

# **3-D Entfernungsdaten-Verarbeitung bei der Umweltvermessung**

am Lehrstuhl für  
Steuerungs- und Regelungstechnik  
der Technischen Universität München  
Prof. Günther Schmidt

eingereichte Diplomarbeit  
des cand. ing. Markus Greim  
Berarbeitungszeitraum 15.06.1987 bis 16.05.1988

Betreuer: Georg Karl

## **Bemerkung rückblickend aus dem Jahre 2022**

Dies ist meine Diplomarbeit aus dem Jahr 1987/1988.

Die Arbeit war Teil der Entwicklung eines mobile autonomen Roboters genannt Makrobe.

Dieser enthielt als Sensorik, unter anderem eine Laser basierte 3D-Entfernungs kamera, heute ein als LIDAR (*Light detection and ranging - analog zum RADAR*) bekanntes System.

Die Kamera und die Makrobe entstanden im Rahmen des Sonderforschungsbereichs 331 der DFG „Informationsbearbeitung in autonomen, mobilen Handhabungssystemen“.

Meiner Diplomarbeit ging eine Studienarbeit am gleichen Lehrstuhl voraus. Hier ging es um den Entwurf der Antriebselektronik für den Drehspiegel der 3D-Kamera.

Diese Kamera, war weltweit einer der ersten, vielleicht auch die allererste zivile und vor allem augensicheren Kameras. Das heißt, die Laserleistung war so gering, dass durch den messenden Laserstrahl Augen nicht geschädigt werden konnten. Solche LIDAR Systeme sind heute, mehr als 35 Jahre später, Standard in Fahrzeugen der automobilen Oberklasse, die autonomes oder teilautonomes Fahren realisieren oder dies zumindest anstreben.

Alle Diagramme in der Arbeit wurden mit Supercalc 3 Version 2, einem der ersten Tabellenkalkulationsprogramme unter MS-DOS erstellt. Der Ausdruck erfolgte auf Nadeldruckern der Firmen Siemens, Typ PT-88 oder Epson, Typ FX-80.

Zeichnungen wurden mit Autocad 2.1x f. MS-DOS erstellt. Diese durfte ich, bei meinem späteren Arbeitgeber, dem Institut f. experimentelle Anästhesie der LMU in der Pettenkoferstraße, auf einem HP-Plotter HP 7475A ausgeben lassen.

Die Software wurde in PASCAL auf einer VAX 11/780 (eigentlich die abgespeckte Version 750, cycle time ca. 3,2 MHz) unter dem Betriebssystem VMS erstellt.

Für diese Diplomarbeit wurde zum Preis von mehreren tausend DM zusätzlich die relationale Datenbanksoftware RDB/VMS (heute als OracleRdb immer noch auf dem Markt) beschafft. Die Lieferung kam aus Irland in Form einer großen sarggroßen Holzkiste, gefüllt mit reichlich Holzwolle. Inhalt der Kiste ein Computer Tape und mehrere Handbücher. Diese Software hatte bereits Ähnlichkeiten mit einer SQL Datenbank. Die erste SQL Norm wurde aber erst 1987 verabschiedet.

Die Entwicklung fand im Alten Elektrotechnischen Gebäude auf dem zentralen Campus der TU in der Arcisstraße statt. Dies war ein Bau aus der Gründerzeit. Wir Diplomanden hatten je einen Arbeitsplatz im Keller des Gebäudes. Da es manchmal von der Kellerdecke tropfte schützten wir die VT220 Computerterminals durch eine Abdeckung aus Pappkarton. Immerhin hatte jeder seinen eigenen Schreibtisch, was aber auch eine Präsenzpflicht implizierte.

Das ein Erdschluss unserer leicht illegalen Kaffeemaschine das Netzwerk des Lehrstuhl zerstörte wurde, uns nicht zum Verhängnis, da sich herausstellte, dass der gesamte Laborbereich im Keller nicht an den Schutzleiter angeschlossen war.

Ebenfalls komfortabel war ein Fernsprech Wandapparat Fe WAp 611 mit Wählscheibe und Amtsberechtigung.

Am besten, war jedoch die Gemeinschaft der Kellerkinder (Diplomanden) u.a. Bernhard Gleinig, Christoph Fröhlich, Anton Voitenleitner, Blacky Schwarz, Peter Klopfer et. al. Kurz nach unserer ruhmreichen Arbeit wurde das Gebäude abgerissen. Wollte man Spuren verwischen? Auf diesem Grund steht heute das Audimax der TU.

Betreuer der Arbeit war Georg Karl - er war nicht immer ganz einfach zu haben, aber ich habe in den Monaten viel gelernt. Die Dauer einer Diplomarbeit war zu Beginn meiner Arbeit noch nicht limitiert. Im Frühjahr 1988 schwenkte die Hochschulleitung aber um, und ich und andere mussten unsere Arbeiten recht ruckartig beenden. Das hatte Vor- und Nachteile.

In Georg Karls Dissertation „Eine 3-D Laserentfernungs kamera zur Bewegungsführung mobiler Roboter“ vom November 1990 sind die Ergebnisse einer der o.a. Diplomanten zum Thema zusammengefasst.

3D-ENTFERNUNGSDATEN-VERARBEITUNG BEI DER UMWELTVERMESSUNG

Am Lehrstuhl für  
STEUERUNGS- UND REGELUNGSTECHNIK  
der Technischen Universität München  
Professor Dr.-Ing. G. Schmidt

eingereichte Diplomarbeit des

cand. ing. Markus Greim  
geb. am 26.03.61  
Guldeinstr. 47  
8000 München 2

Beginn der Arbeit: 15.06.1987  
Einlieferung der Arbeit: 16.05.1988  
Seminarvortrag:  
Betreuer: G. Karl

*Für Liane und Julius*

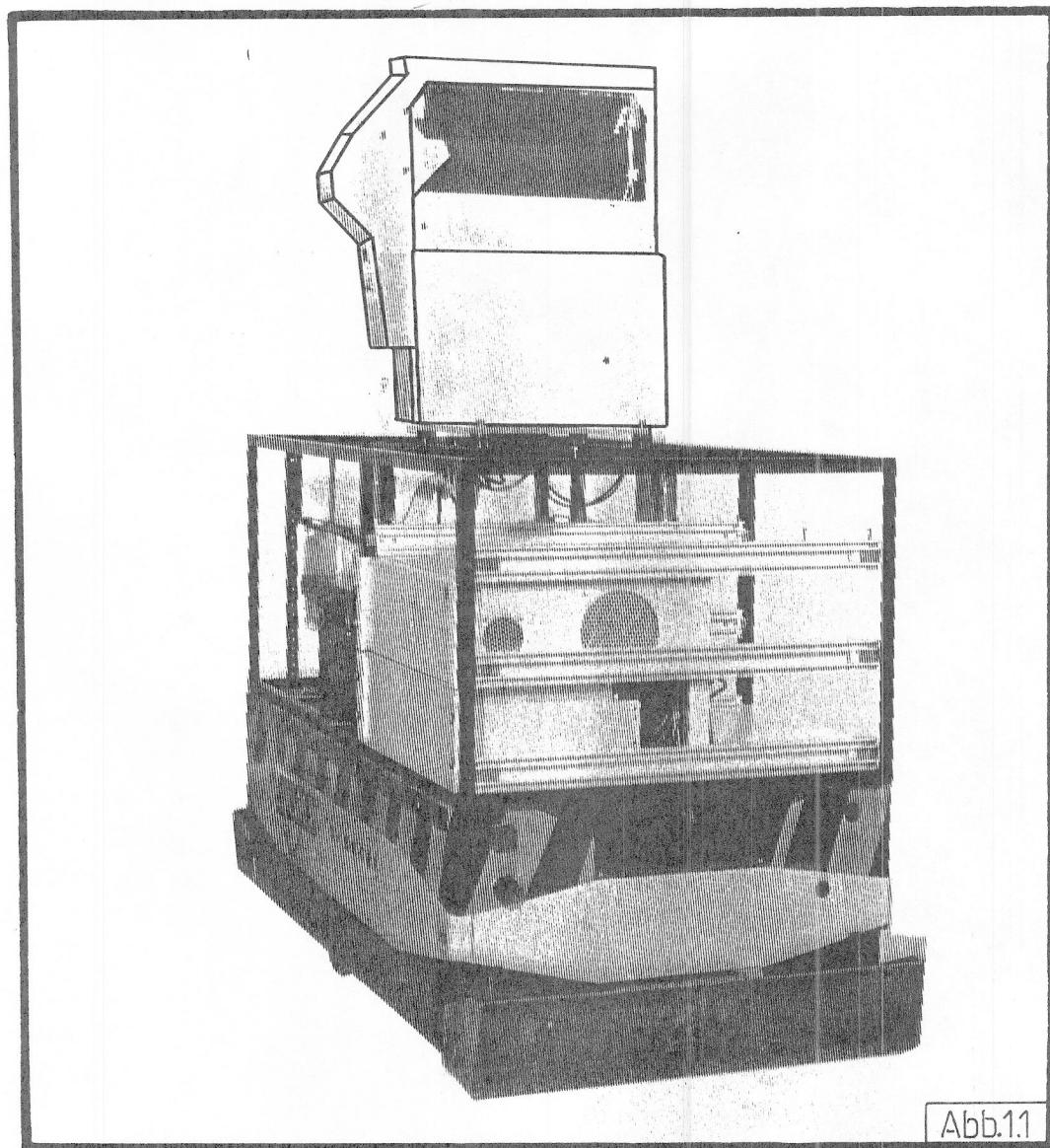
## Inhaltsverzeichnis

1. Einleitung	4.
2. 3-D Entfernungsdatenvorverarbeitung	9
3. Ein Multisensor-Entwicklungsrahmen für das autonom mobile Fahrzeug	
MAKROBE	93
4. Entfernungs bildverarbeitung	133
5. Schnittstellen-, Test- und Transformationsroutinen für den	
Multisensorrahmen ARGUS	197
6. Einsatz des MSS ARGUS	203
7. Zusammenfassung und Ausblick	214
8. Literaturverzeichnis	218
9. Anhang (2.Band)	

## 1. EINLEITUNG

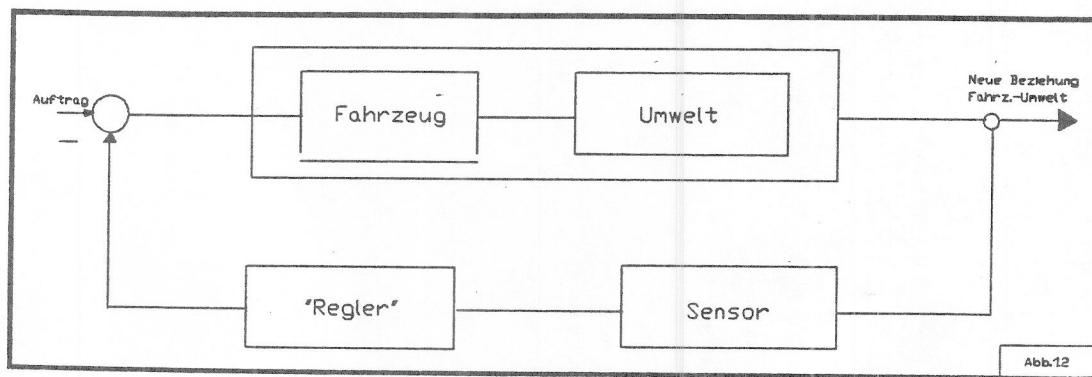
### 1.1. DAS UMFELD: MOBILES AUTONOMES ROBOTERFAHRZEUG MAKROBE

Der Lehrstuhl für Steuerungs- und Regelungstechnik der TU München entwickelt zur Zeit ein mobil autonomes Roboterfahrzeug, genannt MAKROBE (s. Abb. 1.1).



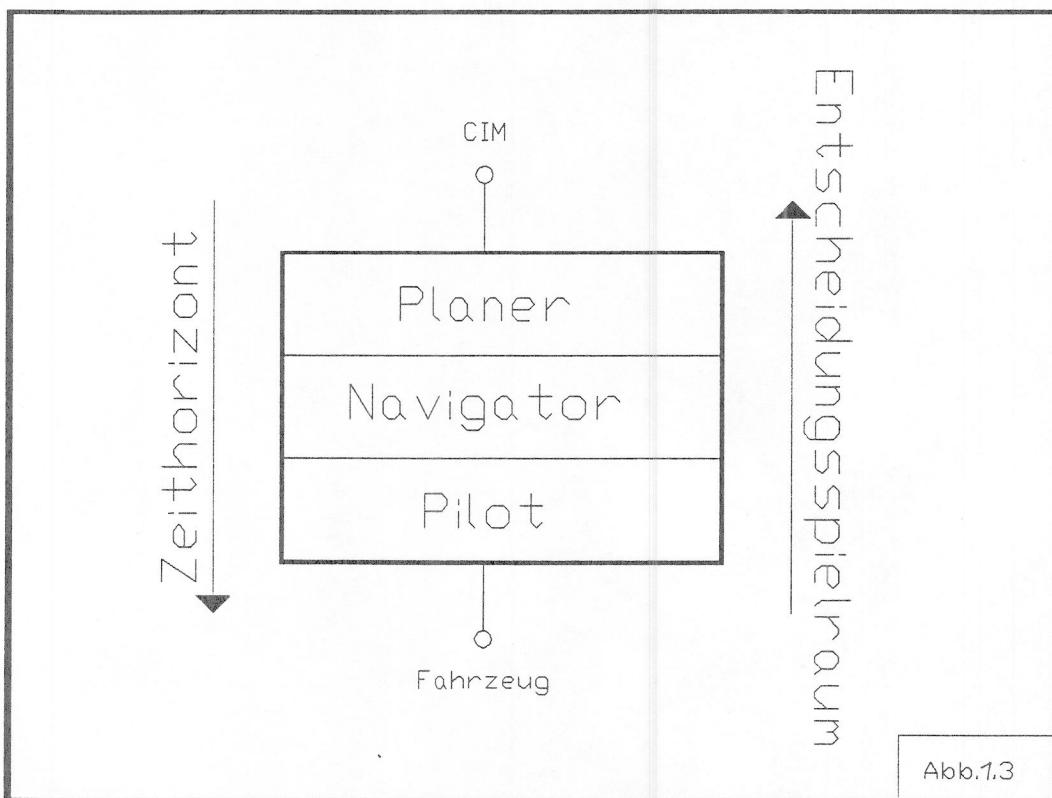
Aufgabe der MAKROBE wird es sein, in einer industriellen Umgebung vom Startpunkt A zum Zielpunkt B zu gelangen, dabei autonom Hindernissen auszuweichen, verstellte Fahrtrassen zu umgehen, und im Hinblick auf neue Aufträge die Umwelt zu erkunden.

Versucht man diese Aufgabenstellung mit den Mitteln des klassischen, regelungstechnischen Blockschaltbildes darzustellen, so ergibt sich ein Regelkreis mit der Strecke Fahrzeug - Umwelt, dem Messglied "Sensor", und einem "Regler" im verallgemeinerten Sinn (s. Abb. 1.2).



1.1.1. "Der Regler": Ein Hierarchisches Planungs-, Entscheidungs- und Steuersystem

Das "Regelungssystem" der MAKROBE ist hierarchisch in 3 Ebenen gegliedert, die Aufgaben mit unterschiedlichem Zeit- aber auch Entscheidungshorizont erfüllen (s. Abb. 1.3).



Die Begriffe Planer, Navigator und Pilot versinnbildlichen Aufgabe und Entscheidungsspielraum der logischen Ebenen.

#### 1.1.2. "Der Sensor": Ein Multisensorsystem

"Der Sensor" der MAKROBE findet seine Ausprägung in einem System vieler Sensoren, die sich in ihren Meßergebnissen teilweise ergänzen, teilweise überlappen, und deren Meßgenauigkeiten, Zeitkonstanten und Auflösungen stark unterschiedlich sind .

- Zu vermessen sind hauptsächlich:
- Die Bewegung des Fahrzeugs
  - Die Umwelt statisch
  - Die Bewegung von Hindernissen

## 1.2. DIE LSSR 3-D ENTFERNUNGSKAMERA ALS HAUPTKOMPONENTE DES MULTISENSORSYSTEMS

Den Kernsensor des Multisensorsystems bildet die LSR. 3-D Entfernungskamera (im weiteren LSR.E.) . Sie liefert ein echt 3-dimensionales Bild der Umwelt in einem Entfernungsreich von 0,3 m bis 10 m.

### 1.2.1. Prinzipielle Eigenschaften der LSR.E.

Die LSR.E. (s. a. Abb.1.1) tastet mit einem Laserstrahl zeilenweise die vor ihr liegende Umwelt ab. An jedem Meßpunkt (81 pro Zeile) ermittelt die LSR.E. die Länge des Laserstrahls von der LSR.E. bis zu dem Umweltobjekt, das den Laserstrahl reflektiert. Die 81 \* 41 Meßpunkte liefern schließlich ein echtes dreidimensionales Bild der Umwelt. Pro Sekunde werden 5 Bilder aufgenommen, die Meßgenauigkeit liegt bei ca. +/- 5 cm. Jede Messung besteht aus Azimut- und Elevationswinkel des Laserstrahls, Entfernung, und als Hilfsgröße die reflektierte Lichtleistung (Qualitätsmaß).

### 1.2.2. Stand der Entwicklung und Anwendung zu Beginn meiner Arbeit

Der Aufbau der LSR.E., sowie deren Anbindung an den datenverarbeitenden Rechner, war zu Beginn meiner Arbeit abgeschlossen. Meßergebnisse konnten als Textdatei auf dem angeschlossenen Rechner abgespeichert werden.

Eine Darstellung der Meßergebnisse einer Zeile war offline, möglich. Die Möglichkeit der quasi 3-D Darstellung, sowie die zumindest nahezu schritthaltende Darstellung einer Meßzeile, war nicht gegeben.

Erste Gehversuche zeigten, daß die Meßgenauigkeit und Ortsauflösung den gestellten Anforderungen im großen und ganzen entspricht.

Erste Anwendungsversuche stützten sich auf die Auswertung einer einzigen Bildzeile.

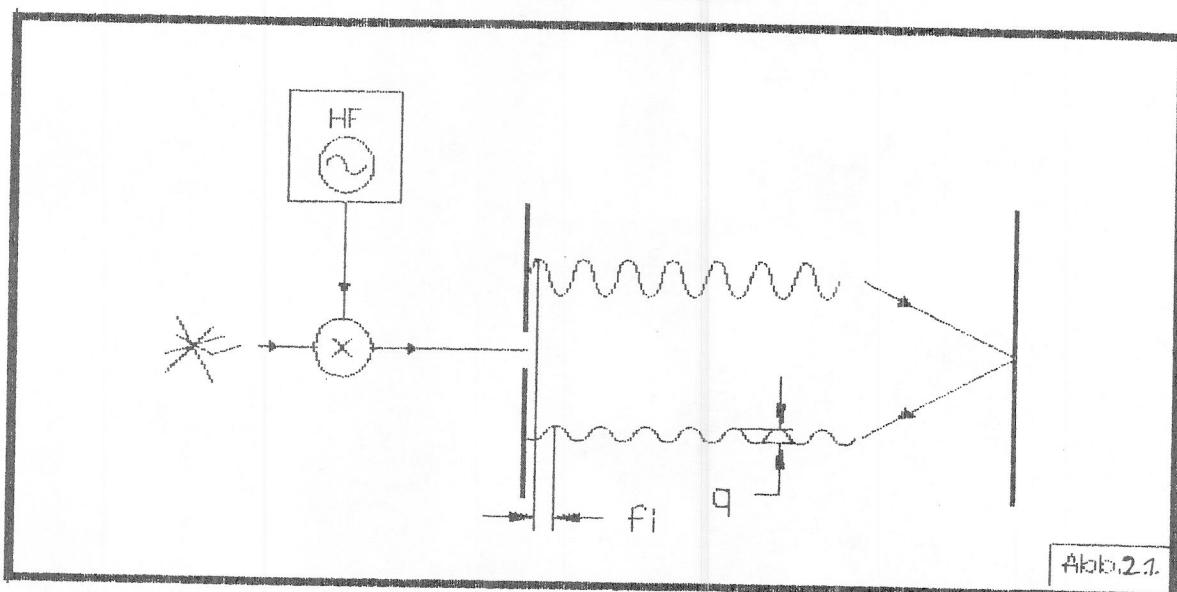
### 1.3. ZIEL DER ARBEIT

Aufbauend auf das oben geschilderte Entwicklungsumfeld und auf den Stand der Entwicklung, ist es Aufgabe meiner Diplomarbeit, Genauigkeit und Meßverhalten der LSR.E. im Sinne einer Kalibrierung zu untersuchen, sowie einer Weiterverarbeitung der Daten mit dem Ziel, eine intelligente Bildauswertung, eingebunden in das Gesamtkonzept der MAKROBE, vorzunehmen. Nach einer Vermessung der LSR.E. zur Bewertung und Validierung der Rohdaten, sind geeignete Filtermethoden auszuwählen und zu realisieren. Zur weiteren Reduktion der Datenflut von 20 KByte pro 3-D Bild werde ich durch geeignete Bildverarbeitungsmaßnahmen die Bildaussage auf das Wesentliche reduzieren, indem ich die Daten als eine wohlgeordnete Menge von geometrischen Primitiven darstelle. Die gesamte LSR.E. Sensordaten Be- und Verarbeitung ist programmtechnisch als offenes System zu konzipieren, d. h. standartisierte Schnittstellen für ein Multisensorsystem, zum Planungssystem und zum Graphik-Monitor System /1/ der Makrobe sind zu implementieren.

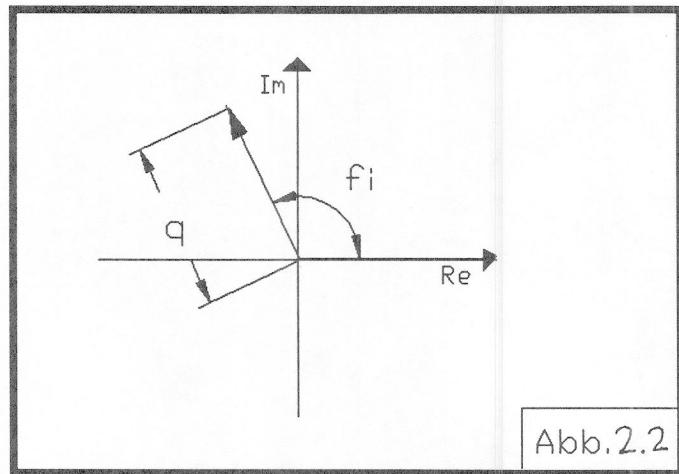
## 2. 3-D ENTFERNUNGSDATEN VORVERARBEITUNG

### 2.1. DAS ENTFERNUNGMESSPRINZIP DER LSR.E.

Das Entfernungsmessprinzip der LSR.E. beruht auf der Laufzeitmessung eines Laserstrahls anhand der Phasenverschiebung eines aufmodulierten sinusförmigen Nutzsignals (s. Abb. 2.1).



Das heißt, jedem Entfernungswert  $r$  kann ein Phasenwinkel  $\phi$  zugeordnet werden. Interpretiert man die empfangene Lichtleistung als Zeigerlänge, so ist die Darstellung des Meßsignals in der komplexen Ebene möglich (s. Abb. 2.2).



Diese Darstellung ist nur innerhalb 0....360 grd entsprechend einem bestimmten Entfernungsreich eindeutig.

Es gilt:

$$\phi = 2\pi \frac{2r}{\lambda} ;$$

$\lambda$  sei die Wellenlänge des aufmodulierten Signals. Um das Problem der Mehrdeutigkeit bei trotzdem hoher Auflösung  $d\phi / dr$  auszuschalten, arbeitet die LSR.E. mit einem Zweifrequenzverfahren, d. h. das Modulationssignal besteht aus zwei verschiedenen Frequenzen. Zu jedem Entfernungswert werden 2 Phasenwinkel detektiert, es existieren 2 Zeiger (s. Abb. 2.3).

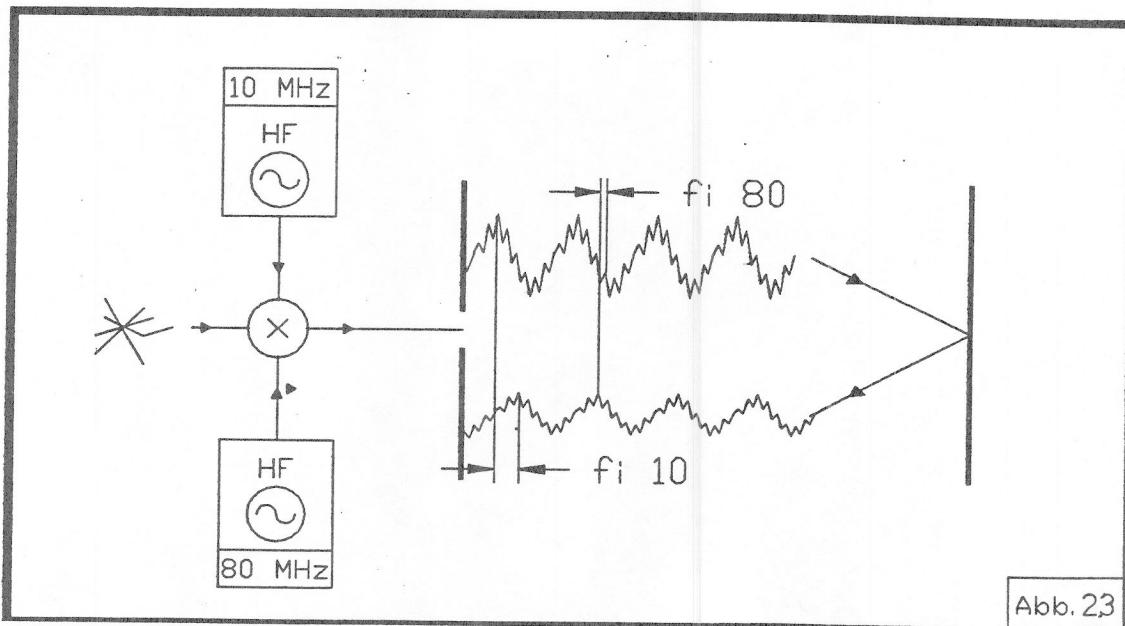


Abb. 23

Zeiger 1 arbeitet durch eine Lasermodulationsfrequenz von 80 MHz eindeutig zwischen 0 m und 1,88 m ( $=\lambda/2$ ), Zeiger 2 (10 MHz) deckt den Bereich von 0 m bis 15,04 m ab (s. Abb. 2.4).

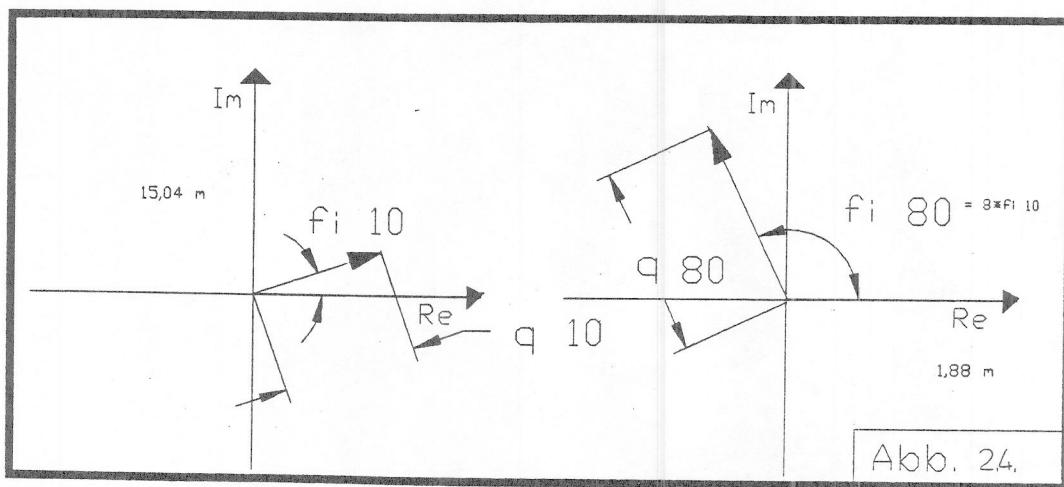


Abb. 24

Da Zeiger 1 die Entfernung zwar mit hoher Auflösung repräsentiert, nur Zeiger 2 aber eindeutig ist im gesamten interessanten Meßbereich, wird das Meßergebnis durch Verknüpfung beider Zeiger ermittelt. Nach getrennter Verstärkung und A/D-Wandlung erfolgt die Weiterverarbeitung der beiden Phasensignale  $\varphi_{10}$  und  $\varphi_{80}$  zum entgültigen Meßergebnis in einer "Noniuslogik" genannten Auswertelektronik (s. Abb. 2.5).

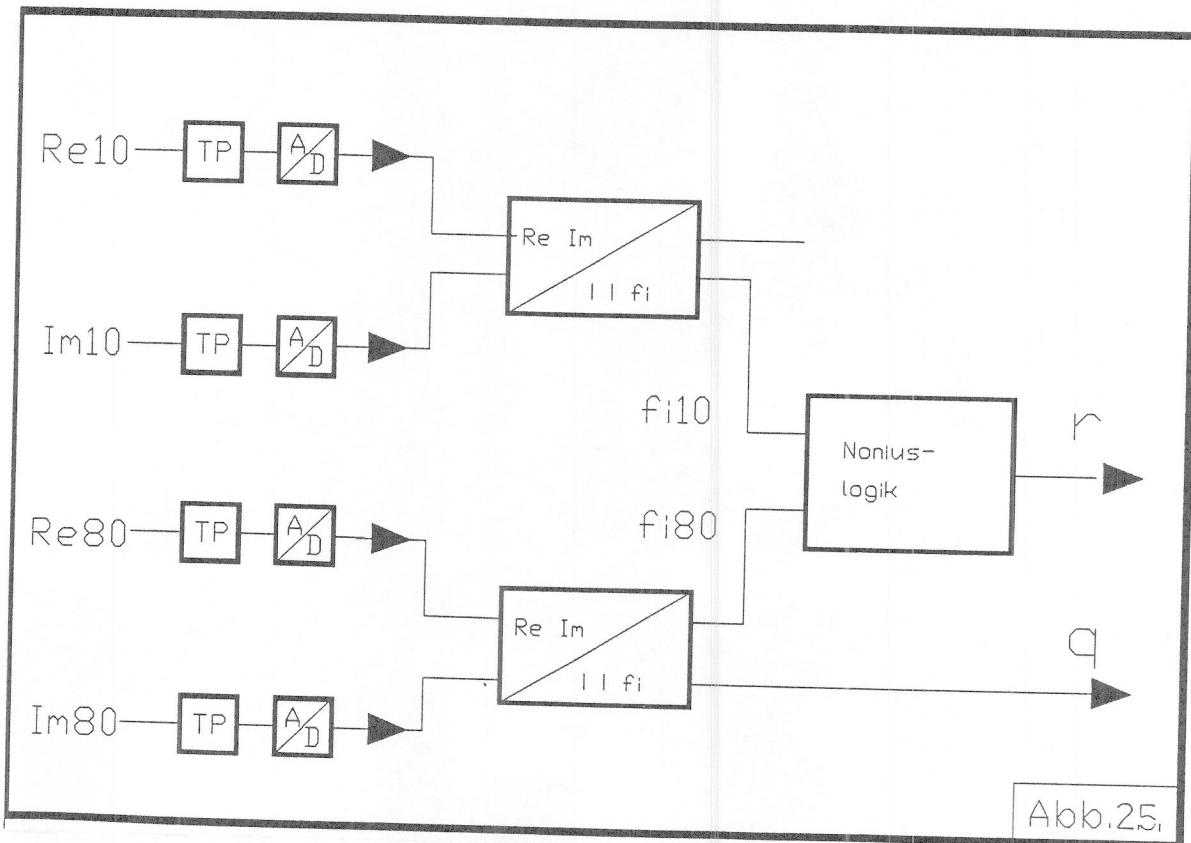
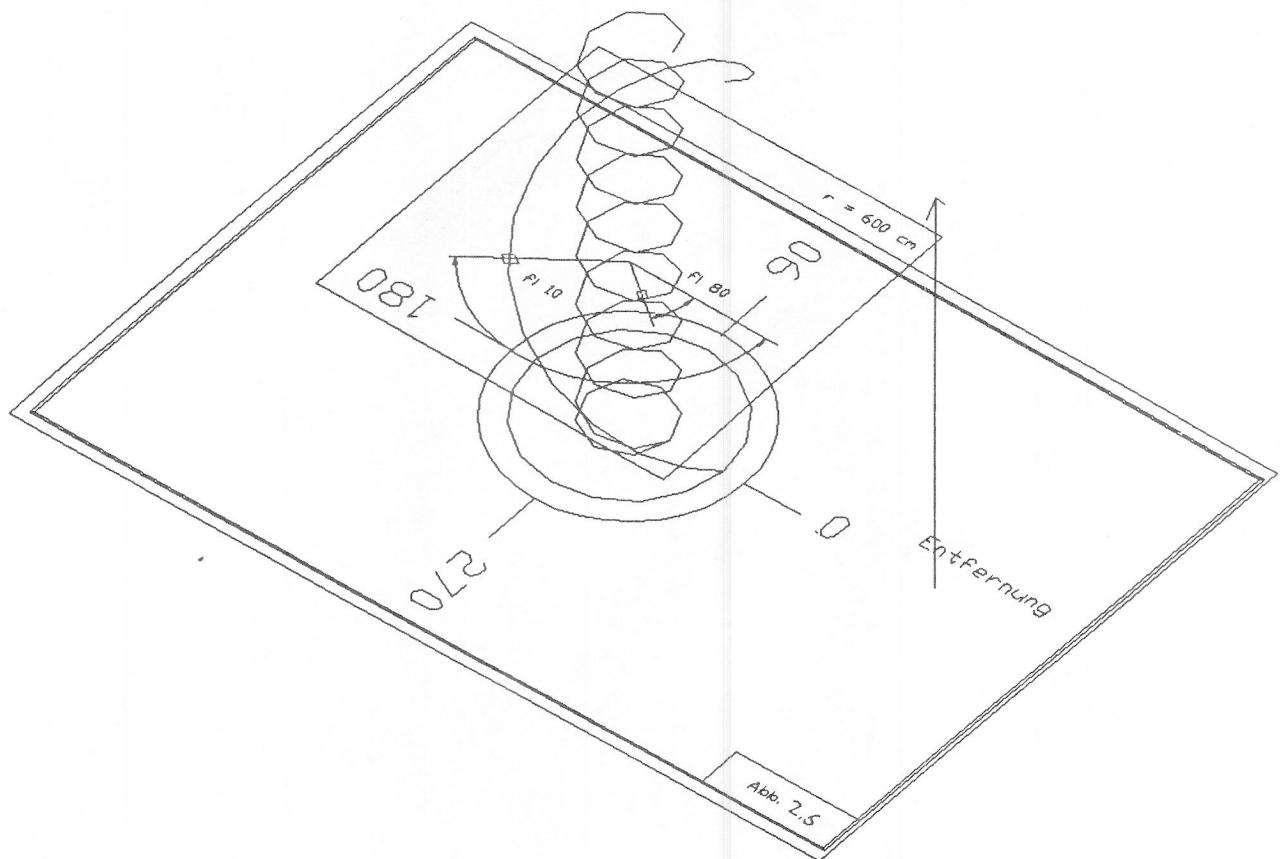


Abb. 2.5

Entsprechend dem Noniusmaßstab in der mechanischen Meßtechnik wird hier ein Referenzmaßstab, nämlich das 10 MHz Phasensignal  $\varphi_{10}$  verwendet, um dem mehrdeutigen Signal  $\varphi_{80}$  eine eindeutige absolute Entfernung zuzuordnen (s. /2/).

Abbildung 2.6 veranschaulicht das Prinzip.



Der Zusammenhang zwischen dem Phasensignal  $\varphi_{10}$  und der Entfernung lässt sich bildlich darstellen. Die Entfernung  $r$  sei Funktion von  $\varphi$ . Der Graph der Funktion ist eine Schraubenlinie auf einem Zylinder. In unserem konkreten Fall hat dieser Zylinder die Höhe 15,04 m, die "10 MHz Schraube" hat in diesem Fall genau eine Windung. Die "80 MHz Schraube" erreicht erst nach 8 Windungen das Ende des Zylinders. Ein beliebiger Entfernungswert wird durch einen Schnitt quer durch den Zylinder dargestellt.

Ich erläutere die Funktionsweise der Noniuslogik schrittweise anhand obigen Modells:

1. Markiere den Entfernungsmeßwert auf der 10MHz Schraube.
2. Lege um diesen Wert einen Fangbereich von  $\pm 94$  cm entsprechend  $\frac{1}{2} \cdot 80$
3. Suche den in diesem Bereich liegenden und hier auch eindeutigen 80 MHz Meßpunkt.
4. Teile den Phasenwinkelbereich von  $\varphi_{80}(0....2\pi)$  in Oktanten ( $k = 0....7$ ).
5. Überprüfe, in welchen Oktanten  $k$  der in Punkt 3 ermittelte Meßpunkt fällt.
6. Addiere zum mehrdeutigen (0....1,88 m) 80 MHz Entfernungswert:  
 $k * 188$

$$r = \frac{1}{2} \cdot 80 \left( \frac{\varphi_{80}}{2\pi} + k \right);$$

(Gleichung 2.1)

## 2.2. VERMESSUNG DER LSR.E.

Um absolute und reproduzierbare Meßergebnisse zu erhalten, ist die LSR.E. zu vermessen und zu testen.

### 2.2.1. Statisches Meßverhalten

Die LSR.E. besteht aus 3 Funktionsblöcken

- a) Die Laser - Entfernungsmeßeinheit
- b) Die Abtasteinheit, bestehend aus 2 bewegten Spiegeln
- c) Die Steuereinheit

Um den Einfluss der Einheiten b) und c) auszuschalten, wird die Vermessung der Entfernungsmesseinheit statisch, d.h. bei stehenden Ablenkspiegeln vorgenommen.

Die physikalischen Gesetzmäßigkeiten und Grenzen des angewandten Meßverfahrens werden durch die folgenden Randbedingungen und Gleichungen zusammenfassend beschrieben (siehe auch /3/).

I. Die reflektierenden Meßobjekte seien Lambert-Reflektoren, d.h. die Rauigkeit der Oberfläche sei groß im Verhältnis zur Wellenlänge des Laserlichts. Daraus folgt für den diffus reflektierten Strahlungsfluß:

$$\Phi_d = \frac{J_T}{\pi} \beta_d \cos \vartheta$$

Gl 2.2.

$\Phi_d$  = Mittlerer diffus reflektierter Strahlungsfluß [W]

$J_T$  = Mittlerer das Meßobjekt erreichende Strahlstärke [W/sr]

$\beta_d$  = Reflexionsfaktor = 1 für weißen Lambert-Reflektor

$\vartheta$  = Einfallswinkel Laserstrahl - Objekt

(s. Abb. 2.7)

## II. Die mittlere empfangene Strahlstärke

Bedingt durch die begrenzte Öffnung und den Blickwinkel der LSR.E. wird nur ein Teil des reflektierten Lichtes wieder empfangen (s. Abb. 2.7).

Für die mittlere empfangene Strahlstärke gilt:

$$J_p = \frac{\alpha A_R J_T}{\pi} \cdot \beta_d \frac{\cos \gamma}{r^2} [W/sr];$$

Es gilt:  $\gamma = \nu$ ; bedingt durch die Geometrie der LSR.E.

$J_p$  = Mittlerer empfangener Strahlstärke [W/sr]

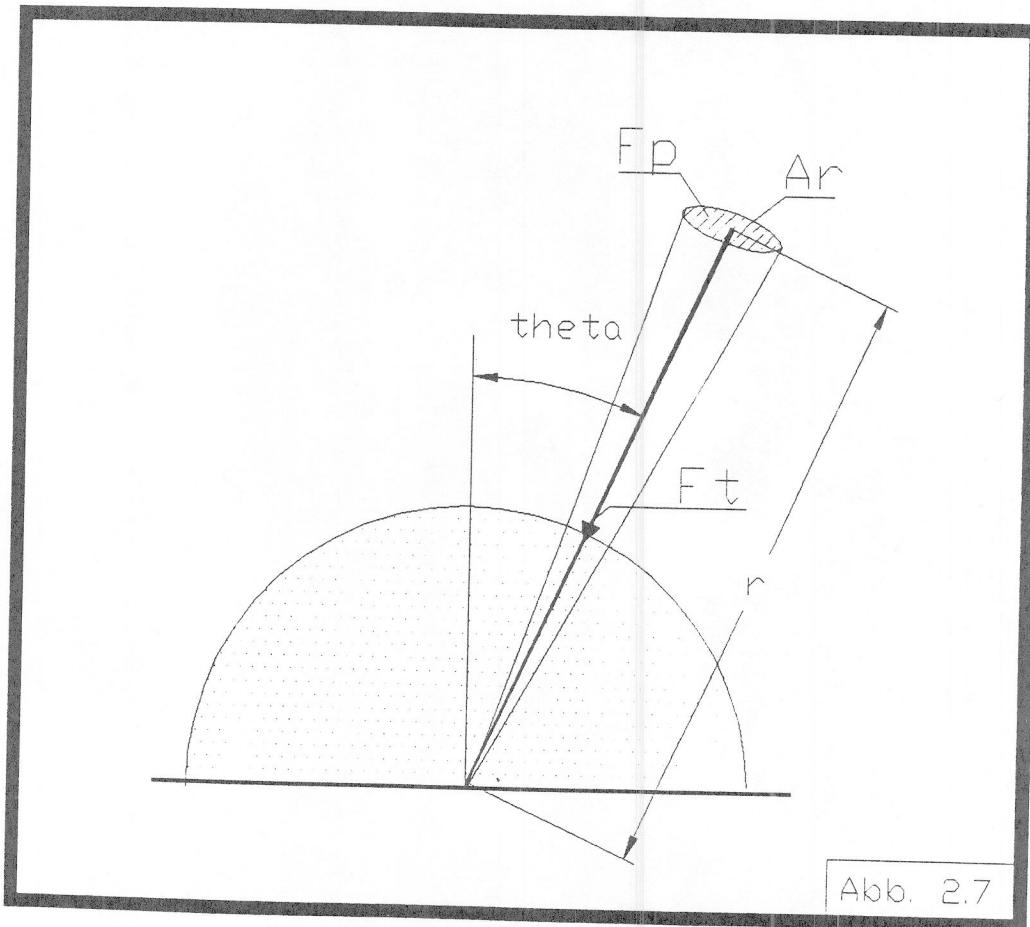
$\alpha$  = Systemkonstante der LSR.E. berücksichtigt Spiegelverluste etc.

$A_R$  = Wirksame Empfängerfläche

$J_t$  = Mittlere ausgesandte Strahlstärke [W/sr]

$\gamma$  = effektiver Blickwinkel der LSR.E. =  $\nu$

r = Entfernung Messobjekt - LSR.E.



### III. Rauschen

Das Rauschen am Ausgang der LSR.E. Meßelektronik ist primär durch infrarotes Umgebungslicht (spektraler Bereich der Empfangsdiode) bedingt. Zur Beschränkung dieses Rauschens wird ein breitbandiger IR-Filter eingesetzt; ein schmalbandiger Interferenzfilter scheidet aus, da dessen Durchlassverhalten, außer vom Lichtspektrum, auch vom Einfallswinkel abhängig ist. Dieser Einfallswinkel ändert sich aber, wenn auch nur leicht, bedingt durch die bewegten Ablenkspiegel während des Scannvorgangs.

Meßtechnische Bestätigung: Schaltet man die Laserdiode aus, lässt die Empfangseinrichtung aber in Betrieb, so bleibt die am Sensorausgang meßbare Rauschleistung in Amplitude und Signaldichte (Standardabweichung) nahezu gleich. Um den maximalen Phasenmeßfehler zu ermitteln der durch Gaussches Phasenrauschen erzeugt wird kann man vereinfacht annehmen das ein Zeiger mit der Länge entsprechend der Rauschleistung vektoriell senkrecht zum Phasenmeßzeiger addiert wird (s.auch Abb. 2.34). Der Phasenmessfehler ergibt sich dann zu:

$$\mathcal{J}_p = \mathcal{J}_T \cdot \tan \Delta_\varphi ;$$

$$\Delta_\varphi = \text{atan} \frac{\mathcal{J}_T}{\mathcal{J}_p} ;$$

$$\text{Gl. 2.3} \Rightarrow \Delta_\varphi = \text{atan} \frac{\pi r^2}{\alpha A_R f_d \cos \vartheta} ;$$

$$\Rightarrow \Delta\varphi \approx \arctan(r^2 c_1) ; \quad c_1 = f(\beta_d, \vartheta) ;$$

$$\Delta\varphi \sim \Delta r \Leftrightarrow \frac{\Delta\varphi}{2\pi} \cdot \frac{1}{2} = \Delta r$$

$$\Rightarrow \underline{\Delta r \approx 30 \arctan(r^2 c_1)}$$

(Gleichung 2.4)

Für die folgenden Messungen wurde ein käufliches Laserentfernungsmeßgerät für Einzelpunktmessung als Referenzmaßstab verwendet. Die Standardabweichung der Messungen dieses Gerätes beträgt:  $\sigma = 10 \text{ mm}$

#### 2.2.1.1. Die Linsencharakteristik

Zur Messung der Linsencharakteristik  $q = f(r)$  wird ein Meßobjekt (konstantes  $\beta_d$ ) senkrecht zum Laserstrahl (konstantes  $\vartheta = 0$ ) von der LSR.E. entfernt. Gemessen wird die Qualität  $q$  in Abhängigkeit von der Entfernung  $r$ . Aus Gleichung 2.3. folgt für die Qualität  $q$  (proportional zu  $\frac{1}{r}$ ) :

$$q = K \frac{\alpha A_R J_T}{\pi} \cdot \frac{S_d \cos \vartheta}{r^2};$$

(Gl. 2.5)

mit  $C_2 = K \cdot \frac{\alpha A_R J_T}{\pi} S_d \cos \vartheta;$

$$\Rightarrow q = C_2 \frac{1}{r^2};$$

(Gl. 2.6)

Abbildung 2.8 zeigt den gemessenen Zusammenhang. Die Kurve gliedert sich in 2 Bereiche:

- a) einen Bereich entsprechend obiger Gleichung  
 Aus der Zeichnung lässt sich ablesen:

$$C_2 = 8 \cdot 10^6 [cm^2]$$

(Gl. 2.7)

für r gemessen in cm

- b) einem Sättigungsbereich für  $r < 100$

Dies beruht auf der nach oben hin liegenden Dynamik von Photosensor, Verstärker und A/D-Wandler. Da es sich bei der Qualität um ein Hilfssignal handelt, wurde die Wortbreite in der digitalen Auswertelektronik auf 8 bit = 0....255 beschränkt.

## Linsencharakteristik

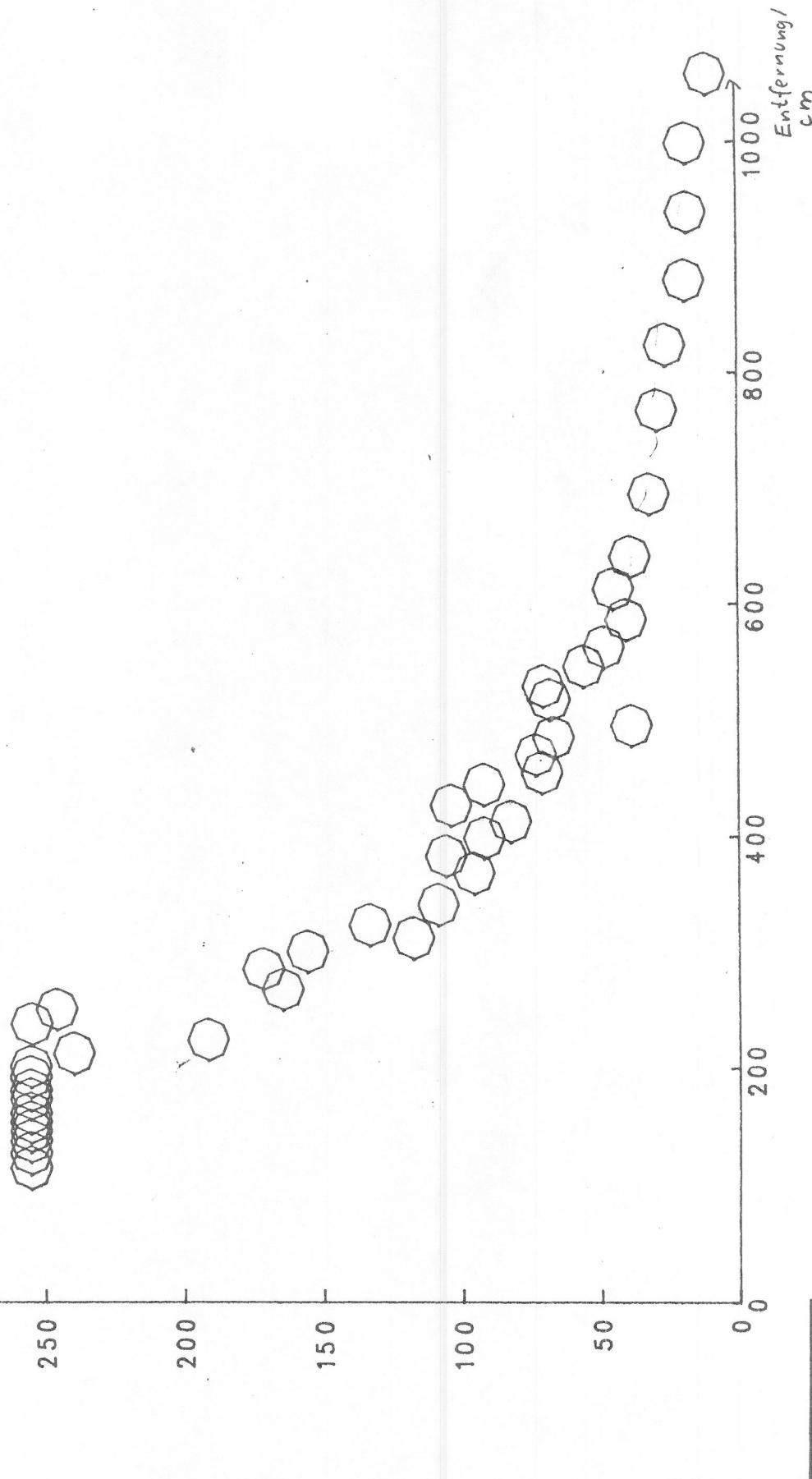


Abb 2.8

### 2.2.1.2. Streuung und Fehler der Entfernungsmeßwerte in Abhängigkeit von der Qualität

Der Entfernungsmessfehler  $r$  wird in Abhängigkeit von der Qualität  $q$  gemessen indem bei konstantem Abstand des Meßobjektes von der LSR.E. d.h. die Farbe des Objektes sowie (= Winkelstellung des Objektes zum Laserstrahl) variiert werden. Abbildung 2.10 zeigt den gemessenen Zusammenhang.

Aus Gleichung 2.4 und 2.5 ergibt sich:

$$\left. \begin{aligned} q &= c_2 \cdot \frac{1}{r^2} \Rightarrow r^2 = \frac{c_2}{q} \\ \Delta_r &= 3\sigma \operatorname{atan}(c_1 r^2) \end{aligned} \right\} \Delta_r = 3\sigma \operatorname{atan}\left(\frac{c_1 c_2}{q}\right);$$

Da Kurve 2.10 nach einem völlig anderen Meßprinzip aufgenommen wird als Kurve 2.9 und 2.8 belegt die Übereinstimmung der aus Gleichung 2.6 errechneten Kurve 2.11 zu Kurve 2.10 die Richtigkeit der errechneten Formeln und physikalischen Voraussetzungen.

Abbildung 2.12 zeigt die Streuung der Entfernungsmeßwerte in Abhängigkeit von der Qualität. Die Streuung liegt im Bereich von  $q > 20$  (= noch akzeptabler Entfernungsmeßfehler) unter 0.6 cm und ist somit vernachlässigbar klein.

# Entfernungs-Messfehler

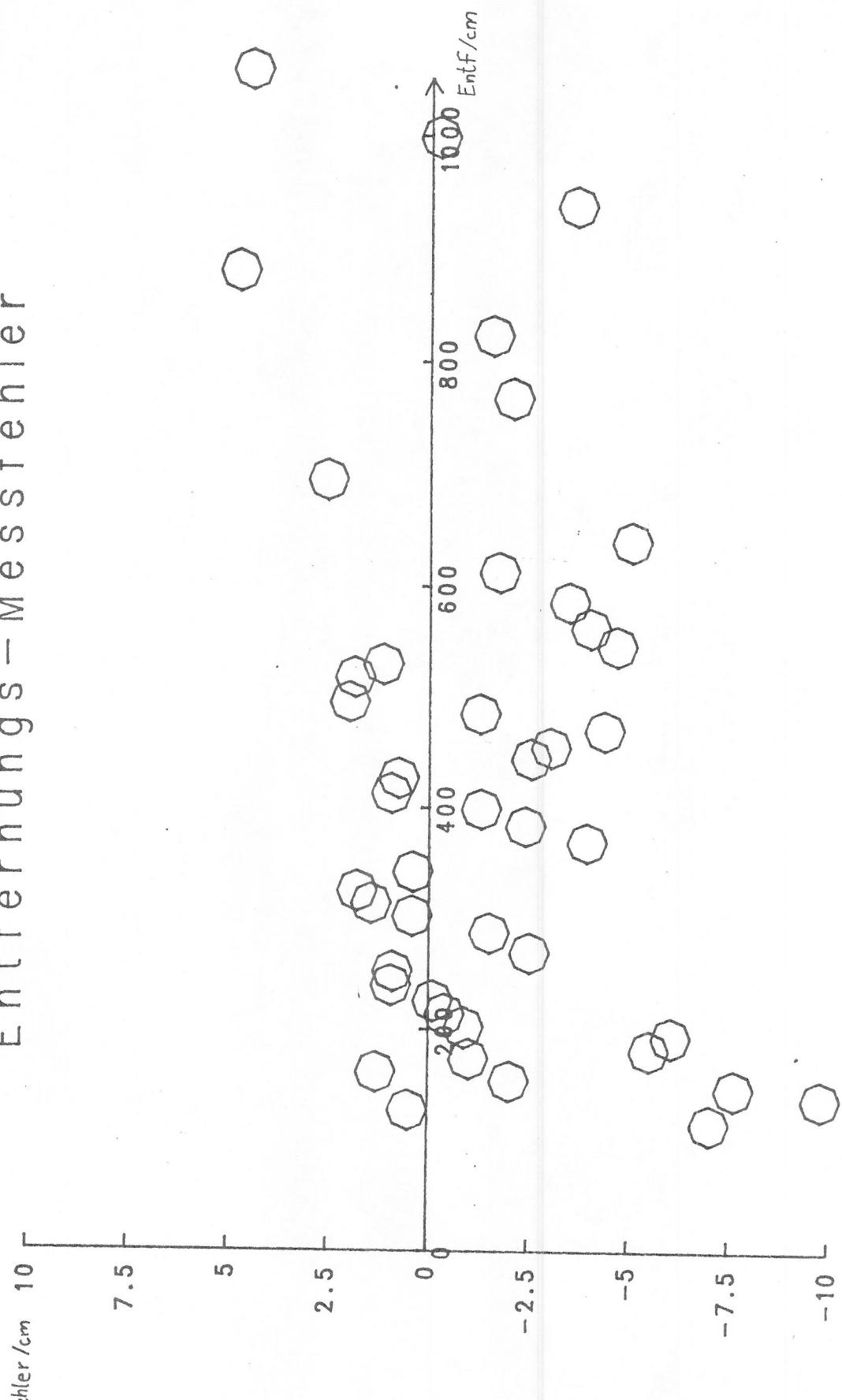


Abb. 2.9

# Messfehler / Qualität

Fehler/cm 10

7.5

5

2.5

0

-2.5

-5

-7.5

-10

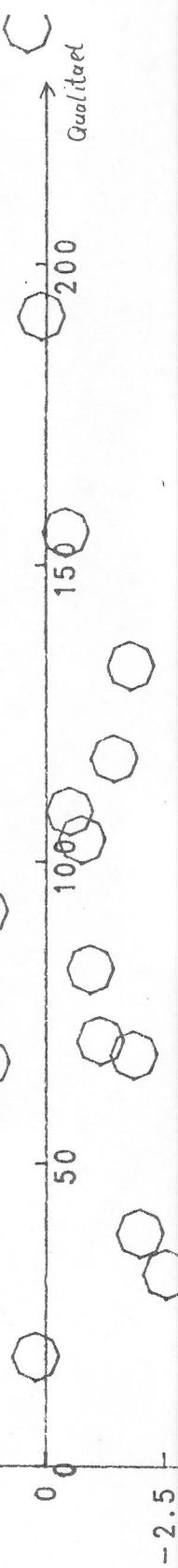


Abb 2.10

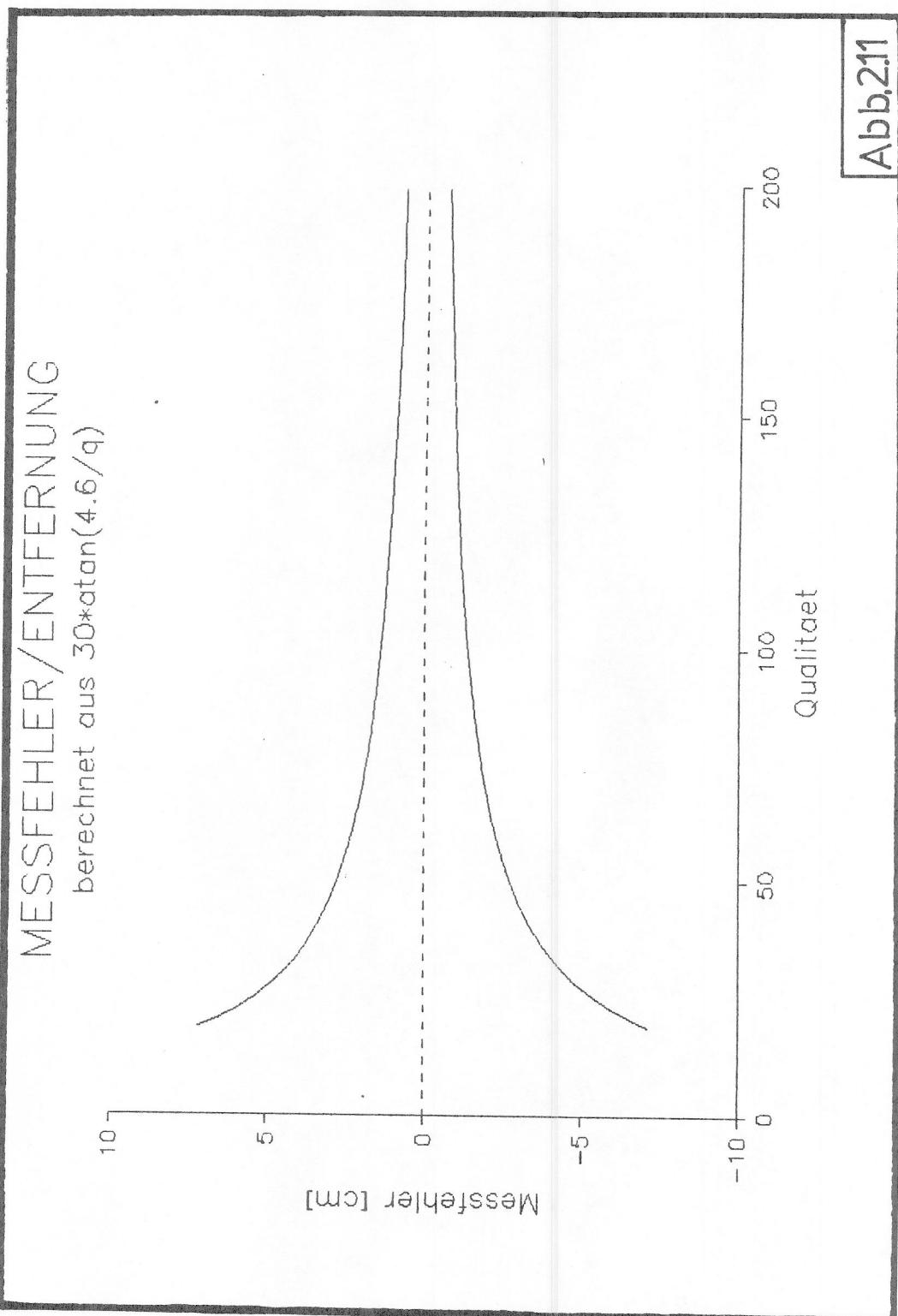


Abb.211

$G/cm$ 

1.4

1.2

1

0.8

0.6

0.4

0.2

0

Streuung / Qualität

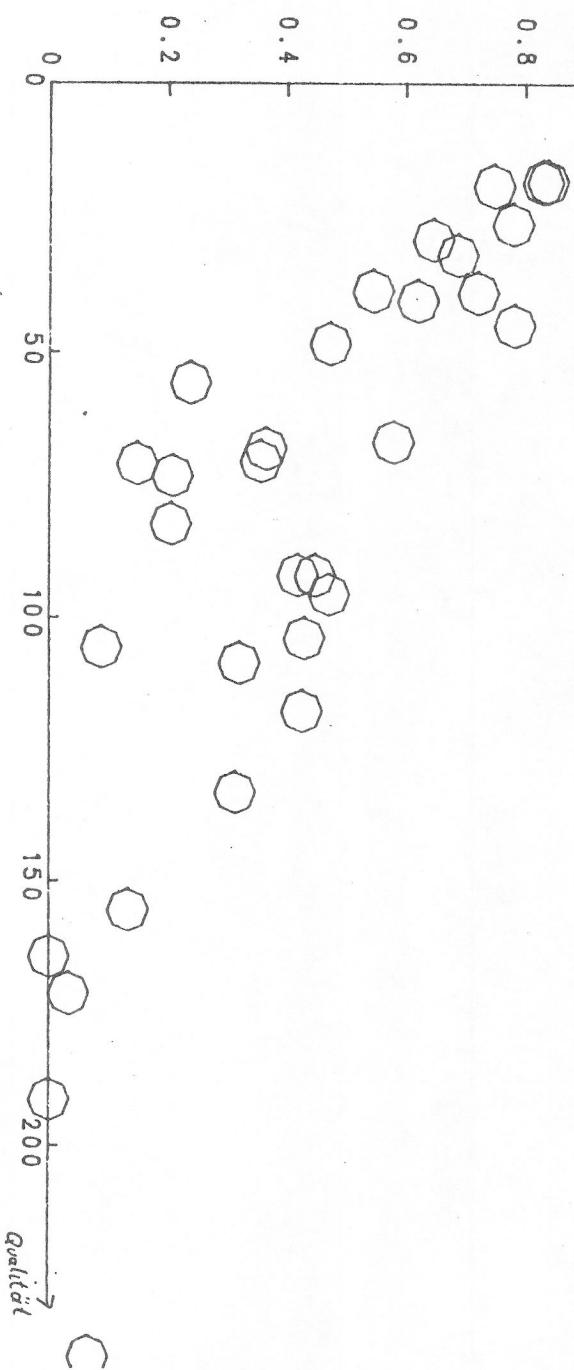


Abb. 212

Entf. Messung mit 10 MHz  
Linearitätsfehler

1000

800

600

400

200

0

Messwert in cm

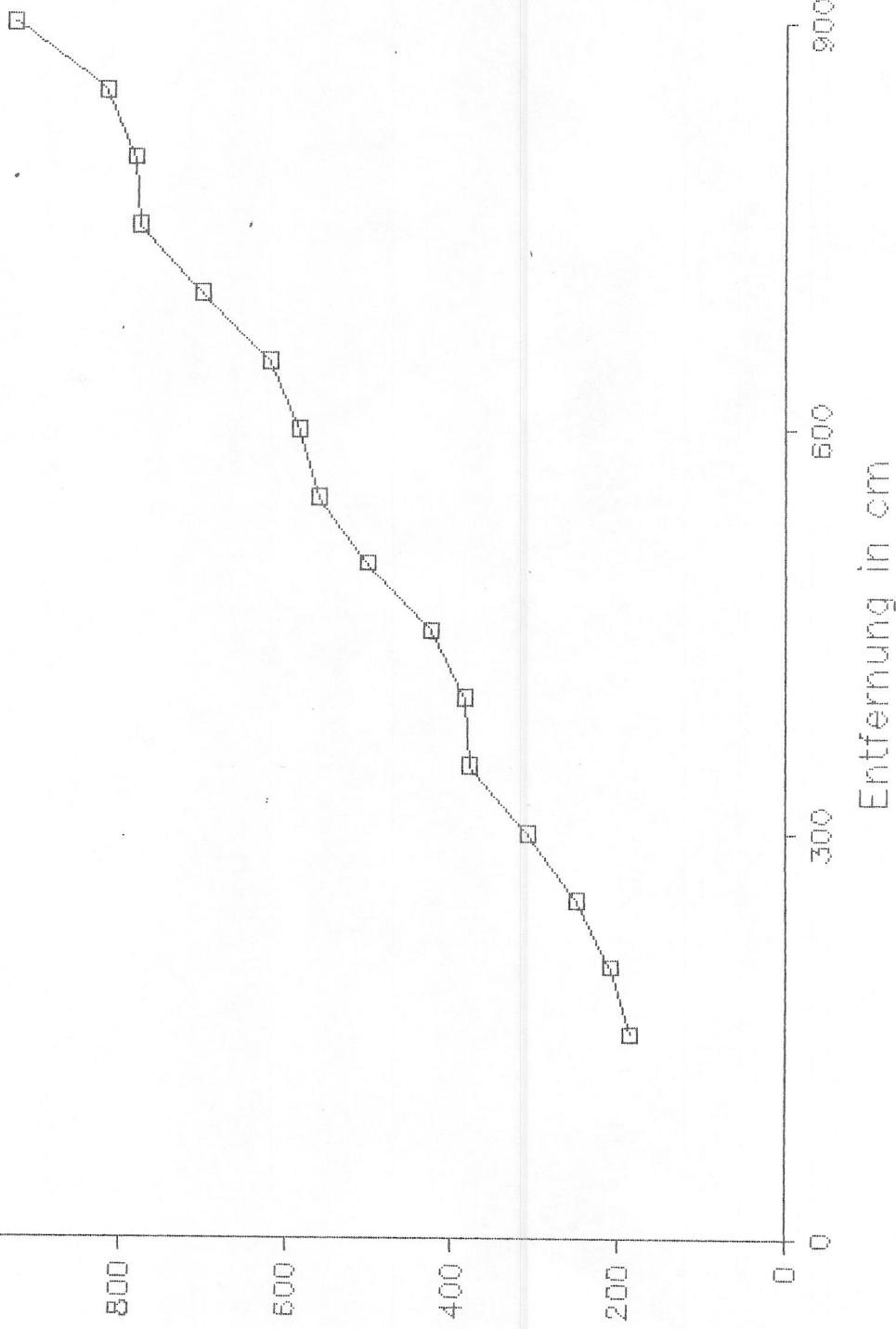


Abb. 213

Entf. LS/cm

1000

800

600

400

200

0

## Entfernungskurve

Entf. tats./cm

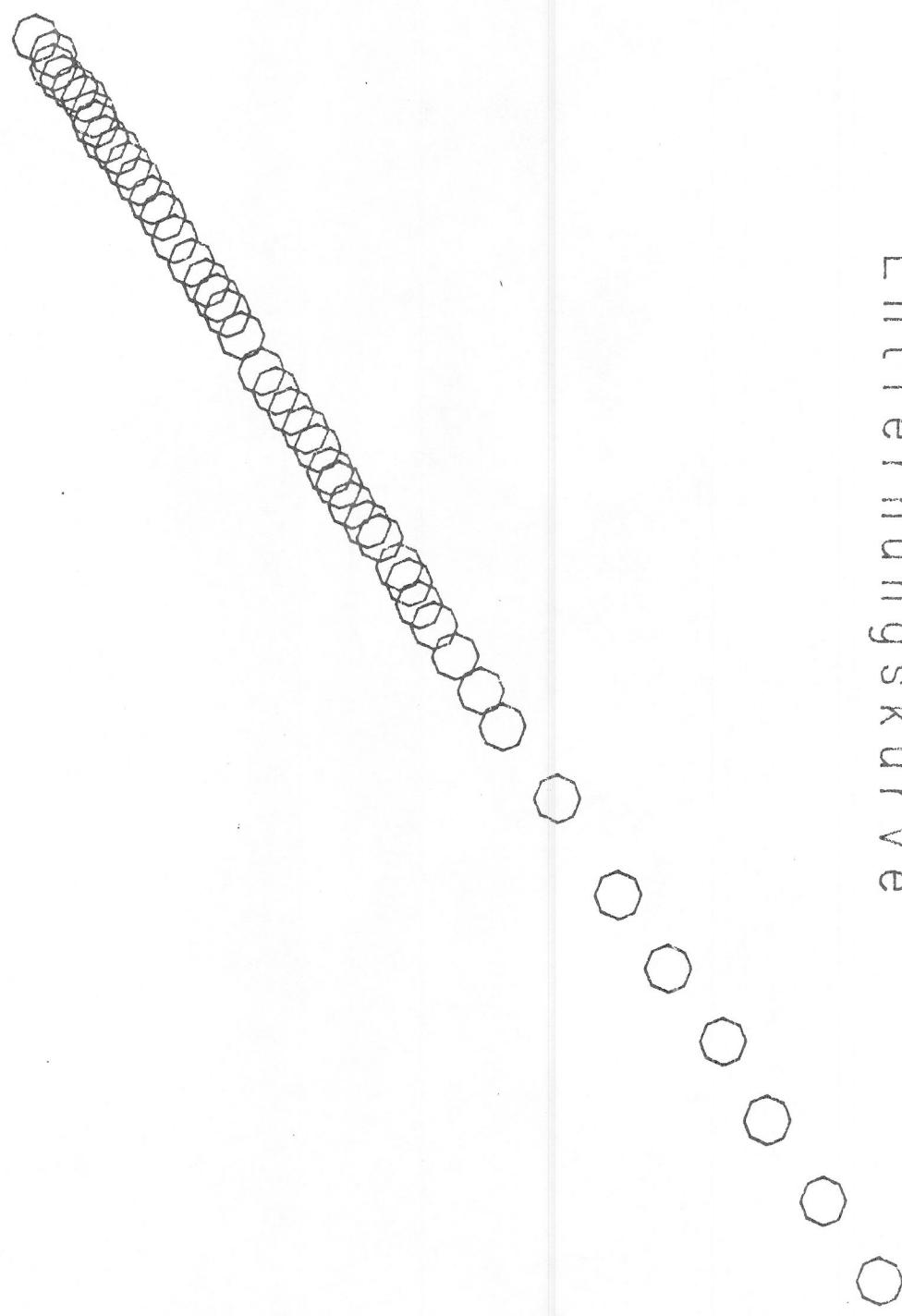


Abb. 2,14

### 2.2.1.3 Entfernungsmeßfehler als Funktion der Entfernung r

Der Entfernungsmeßfehler wird gemessen indem das Meßobjekt bei konstantem und von der LSR.E. entfernt wird. Mit einem Referenzmeßgerät wird gleichzeitig ein  $r_{ref}$  aufgenommen

$$\Delta r = r_{\text{laser}} - r_{\text{ref}}$$

Abb. 2.9 zeigt den Entfernungsmeßfehler, d. h. die Abweichung der gemessenen von der tatsächlichen Entfernung. Zusätzlich ist der Bereich der Streuung eingezeichnet.

Der in Gleichung 2.4. aufgezeigte Zusammenhang zwischen Streuung und Entfernung r lässt sich aus der Zeichnung ablesen:

$C_1$  ergibt sich aus der Zeichnung zu:  $0,6 \cdot 10^{-6} [\text{cm}]$ ;

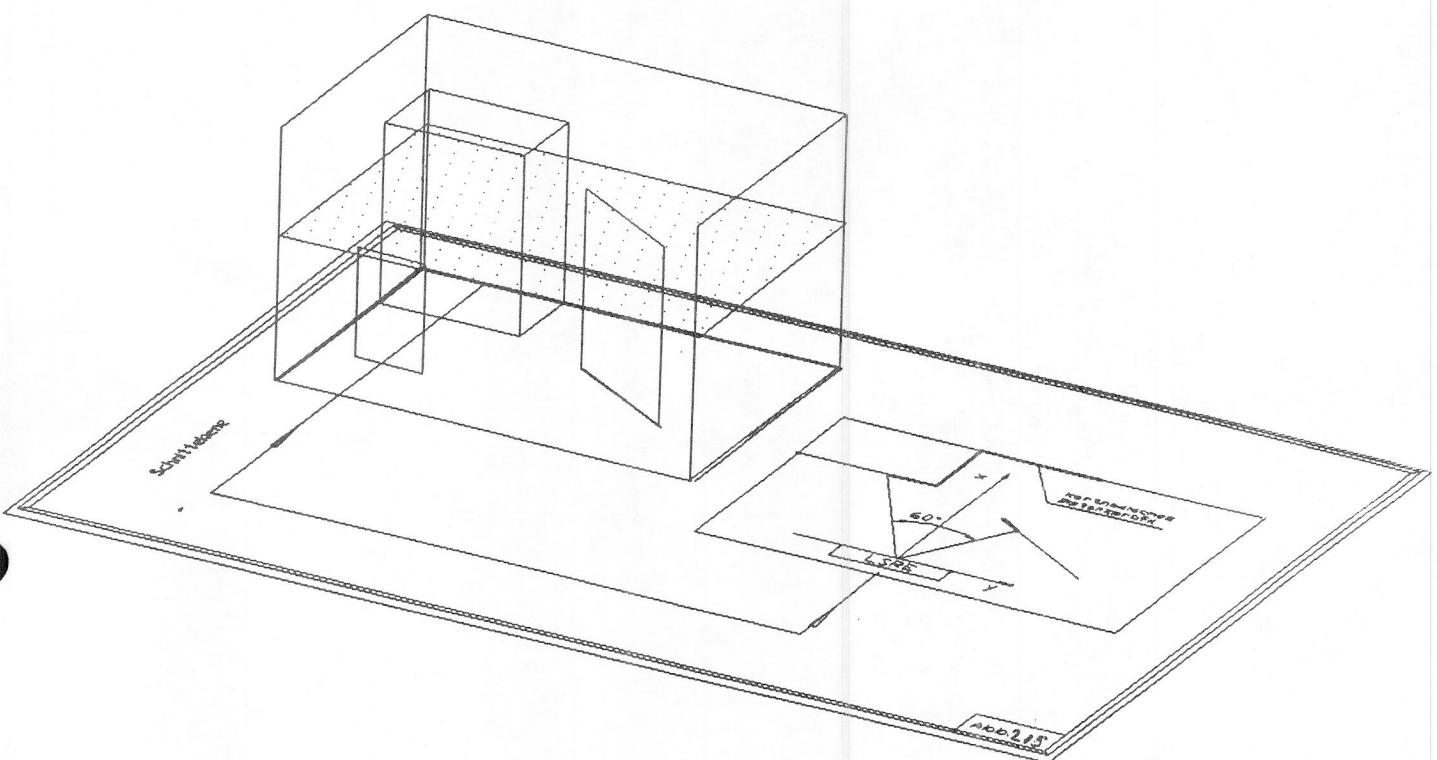
### 2.2.1.4. Entfernungskurve, Linearitätsfehler

Abb. 2.13 zeigt das Meßverhalten des bis jetzt noch nicht explizit betrachteten 10 MHz Kanals. Die Linearität liegt im Bereich von  $\pm 30$  cm. Ein Vergleich mit dem resultierenden Zweifrequenz - Meßergebnis in Abb. 16 zeigt eindrucksvoll den Vorteil dieses Verfahrens.  
(Linearitätsfehler  $\approx 5$  cm)

## 2.2.2. Dynamisches Meßverhalten

Die LSR.E. führt zur Aufnahme eines ganzen Bildes den Laserstrahl zeilenweise über die Meßszene. Pro Sekunde werden 180 Zeilen a 81 Meßpunkte aufgenommen. Zwischen der Messung zweier Bildpunkte liegt somit eine Zeit von  $t_B = 68 \mu s$ , die Aufnahme einer Zeile dauert  $t_Z = 5,5 \text{ ms}$ . Es ist deshalb notwendig, die dynamischen Grenzen der LSR.E. in die Beurteilung ihres Meßverhaltens einzubeziehen.

Im weiteren stütze ich meine Messungen hauptsächlich auf die Aufnahme einzelner Bildzeilen ( $t_B \ll t_Z!$ ) entsprechend einem karthesischen oder polaren Schnitt durch die Meßszene (s. Abb. 2.15).



#### 2.2.2.1. Ortsauflösung der LSR.E.

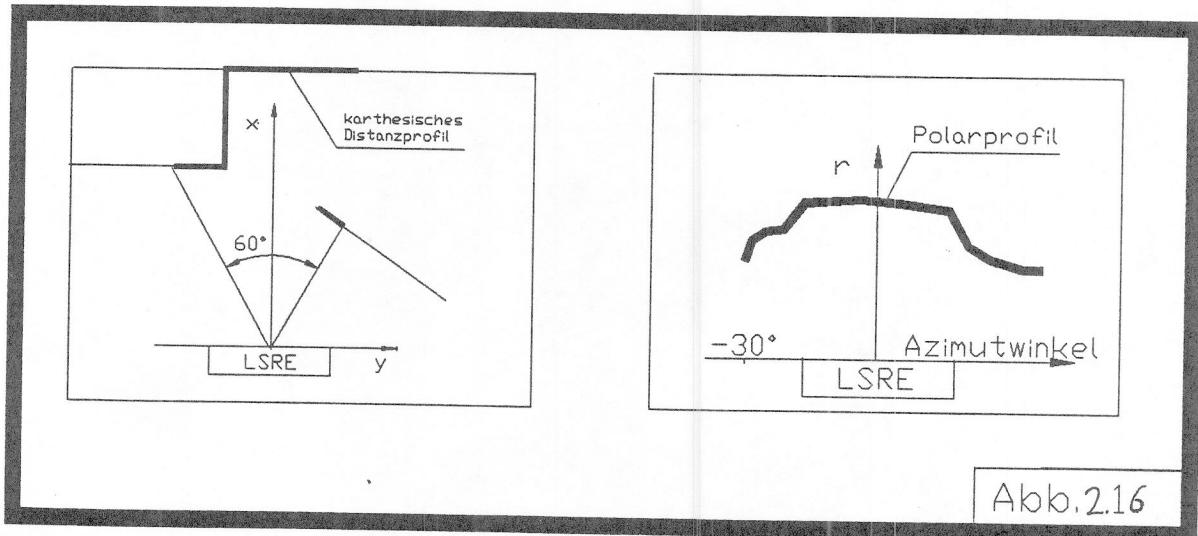
Die Ortsauflösung der LSR.E. wird durch 2 Faktoren begrenzt:

## I. Die Abtastrate

Die LSR.E. tastet eine Szene (s. Abb. 2.16) von links nach rechts ab, der Laserstrahl überstreicht dabei einen Azimutwinkel von  $60^\circ$ .

Bei 81 abgetasteten Bildpunkten haben die Meßpunkte einen Abstand, gemessen in grd Raumwinkel, von:

$$\varphi_{\text{aufl}} = 0,75^\circ;$$



Die Ortsauflösung horizontal von LSR.E., gemessen in Abhängigkeit von  $r$ , beträgt damit:

$$h_{\text{aufl}} = r \cdot \tan(\varphi_{\text{aufl}}) = r \cdot 1,3 \cdot 10^{-2}$$

(Gleichung 2.7)

## II. Die Modulationsübertragungsfunktion

Wie jedes reale nachrichtentechnische System besitzt auch die LSR.E. eine nach oben hin begrenzte Bandbreite (s. Abb. 2.17).

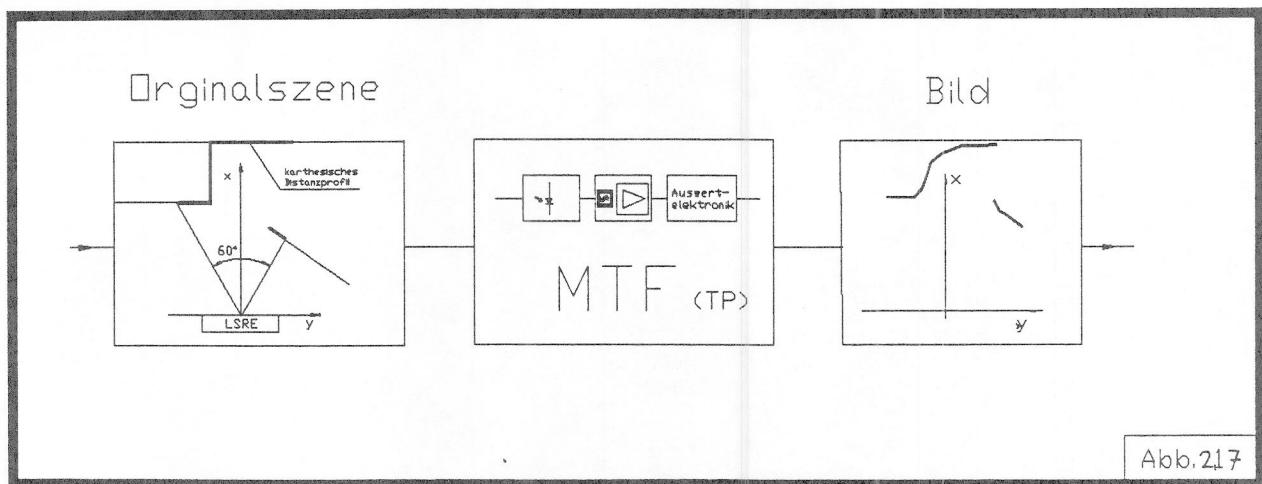


Abb. 2.17

Zum Beispiel besitzen Sprungkanten (Jump Boundaries) in einer Szene sehr hohe Ortsfrequenzanteile (Rechteckspektrum). Die Bandbreite der LSR.E. wird, vor allem durch die notwendigen Tiefpassfilter, vor der A/D-Wandlung der Phasensignale begrenzt.

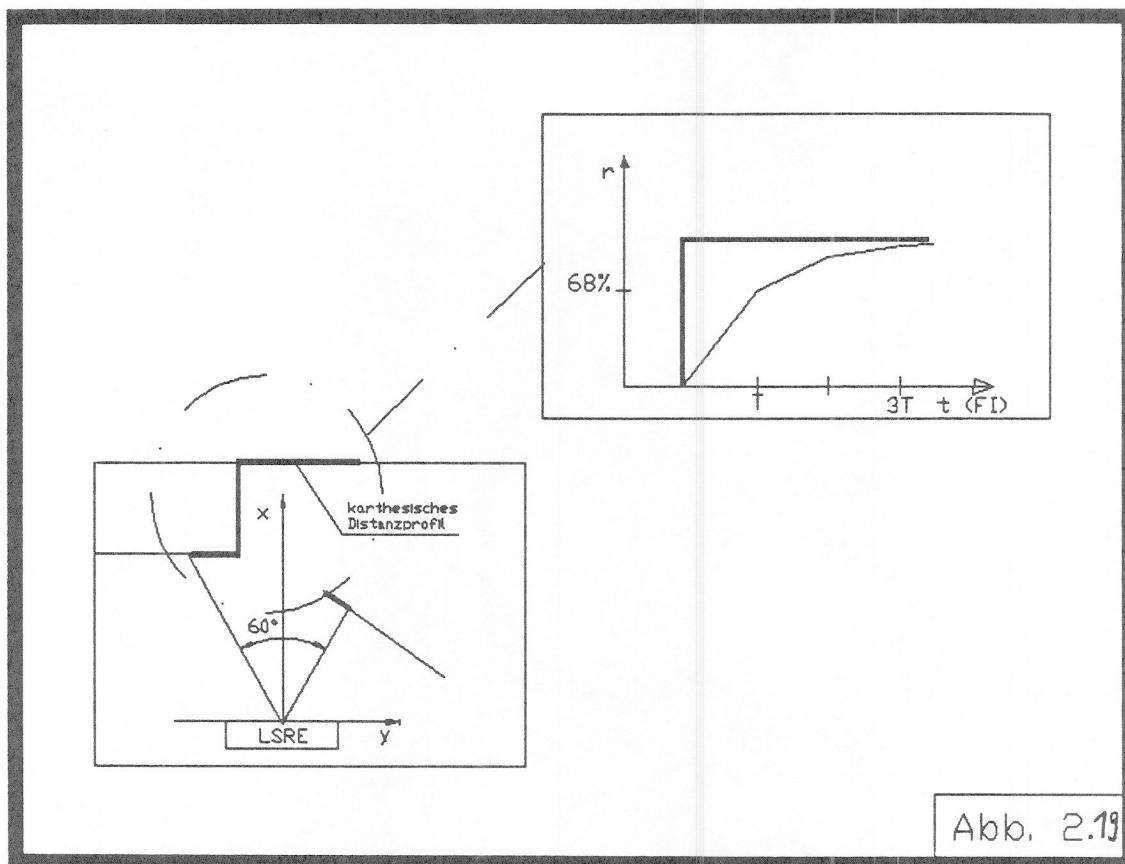
Zum Einsatz kommen hier Bessel-Tiefpässe mit einer Eckfrequenz von 18 KHz. Um die Filtereckfrequenz kann dieser Filtertyp durch ein PT1-Glied angenähert werden. Die Zeitkonstante des angenäherten Filters beträgt

$$T = \frac{1}{2\pi f_e} = 8,8 \mu s;$$

Die Sprungantwort eines PT1-Glieds erreicht bekanntlich nach  $3 * T$ , 98 % ihres Endwertes (s. Abb. 2.19).

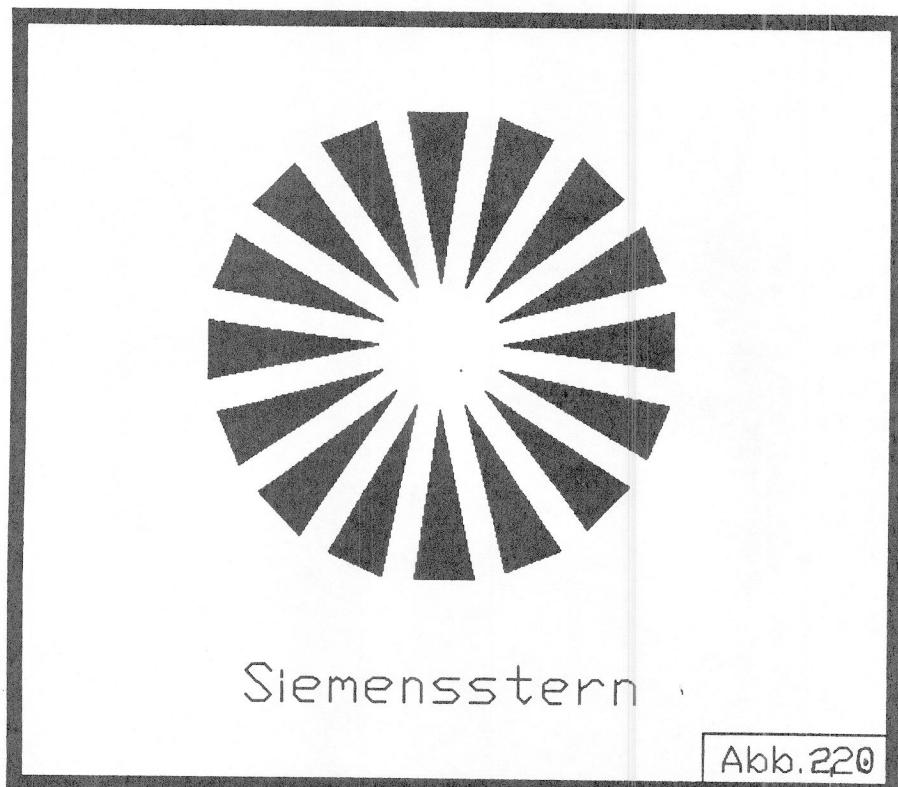
$$3T = 26 \mu s;$$

Dies bedeutet: Sollen alle Ortsfrequenzanteile einer Sprungkante übertragen werden, oder bildlich ausgedrückt, soll die Kante "unverschliffen" abgebildet werden, so muß die Zeit zwischen 2 Abtastpunkten  $T_a > 26 \mu s$  liegen.

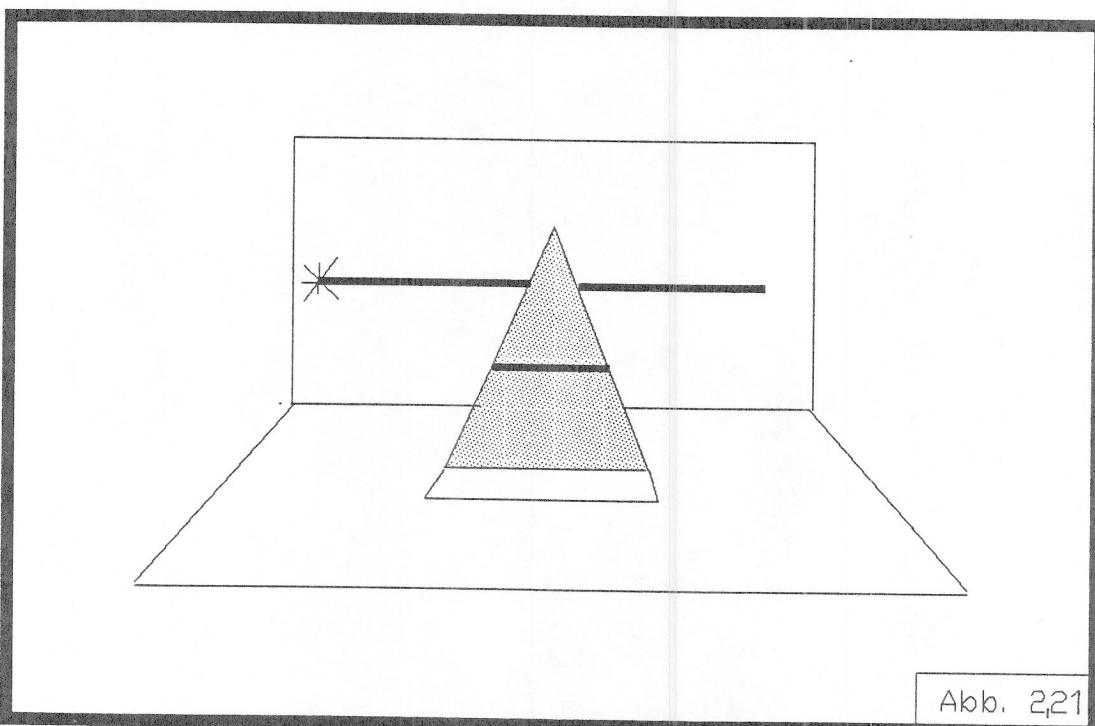


Da die Abtastzeit  $T_a$  der LSRE. größer als 6 mal der die MÜF bandbegrenzenden Zeitkonstante gewählt wurde, wird die Ortsauflösung nur durch die Abtastbreite, nicht aber durch die Dynamik des analogen Übertragungssystems begrenzt.

Zur Messung der MÜF wird in der klassisch optischen Abbildungstechnik ein als Siemensstern bezeichnetes Testmuster verwendet (s. Abb. 2.20).



Mit einem einzelnen, dafür aber räumlichen Segment dieses Sternes, habe ich die Ortsauflösung der LSR.E. überprüft (s. Abb. 2.21).



Wie aus Abb. 2.22 ersichtlich ist der Wert der MÜF = 1 für die gewählte Abtastbreite.

Eine stark tiefpassartige MÜF würde den Punkt A in Abb. 2.22 weiter hinten abbilden. Abbilder als Punkt B, scharfe Kanten, würden abgerundet dargestellt.

#### 2.2.2.2. Kongruenz der Meßzeilen

Zur Gewinnung eines exakten Gesamtbildes ist es notwendig, daß gleiche Meßobjekte in xy-Richtung in verschiedenen Zeilen an der gleichen Stelle abgebildet sind. Die Ablenkeinrichtung muß von Zeile-n zu Zeile n + 1 reproduzierbar arbeiten. Dieser Sachverhalt wird mit der in Abb. 2.23 dargestellten Szene überprüft. Die signifikante Kante A muß in allen Zeilen als Ecke dargestellt werden. Alle Ecken müssen die gleiche Position in xy-Richtung besitzen (Abb. 2.24).

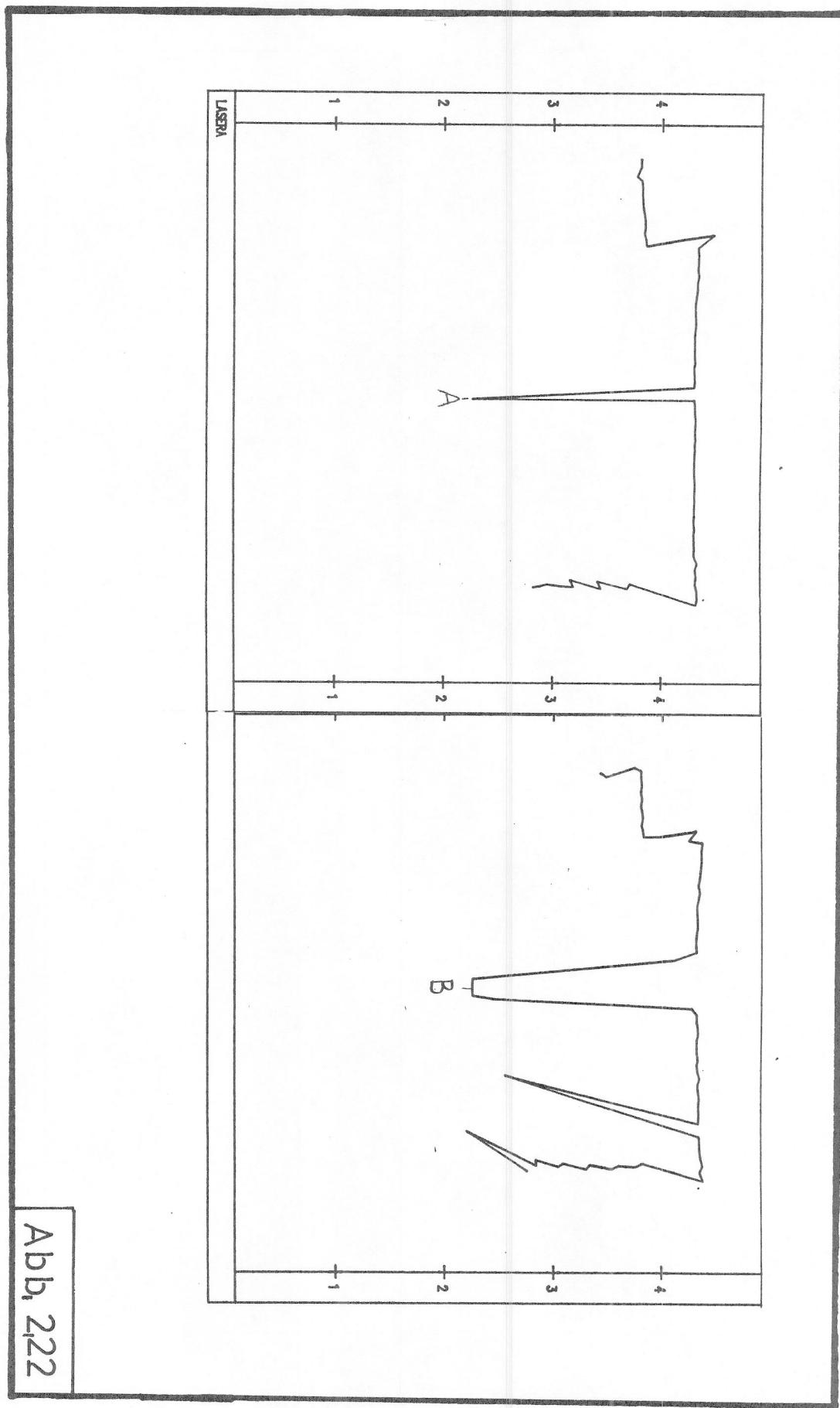
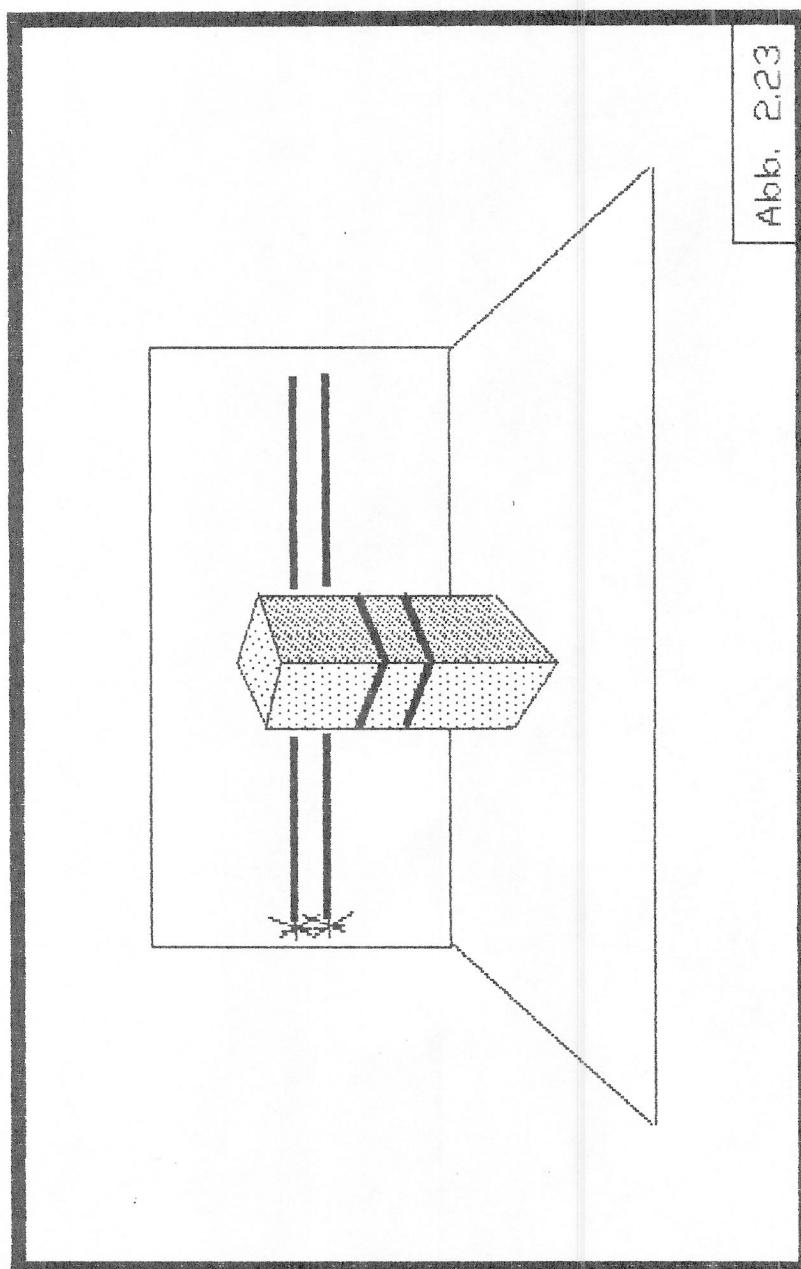
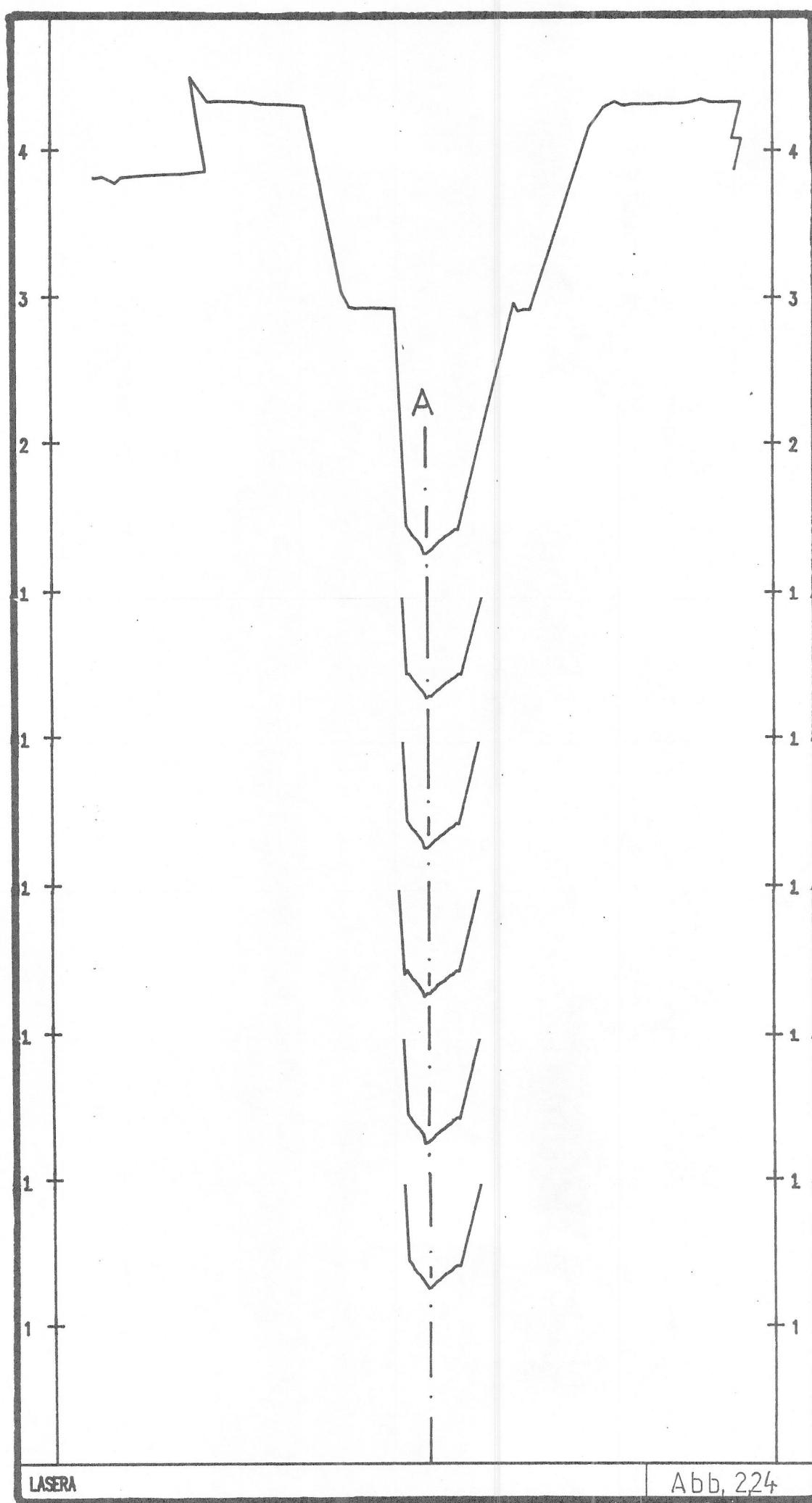


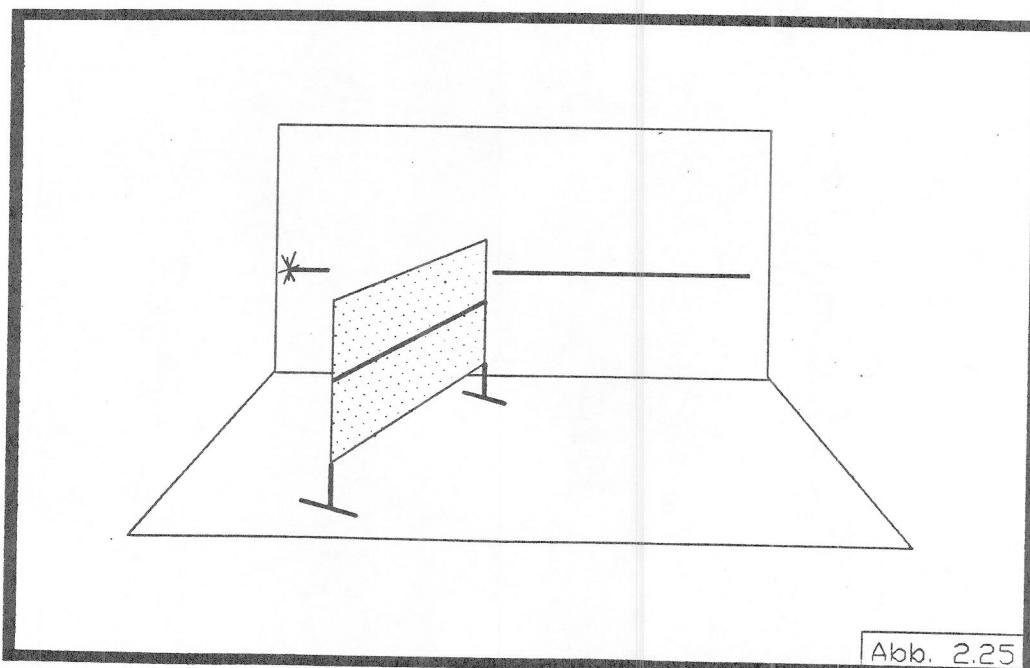
Abb. 222





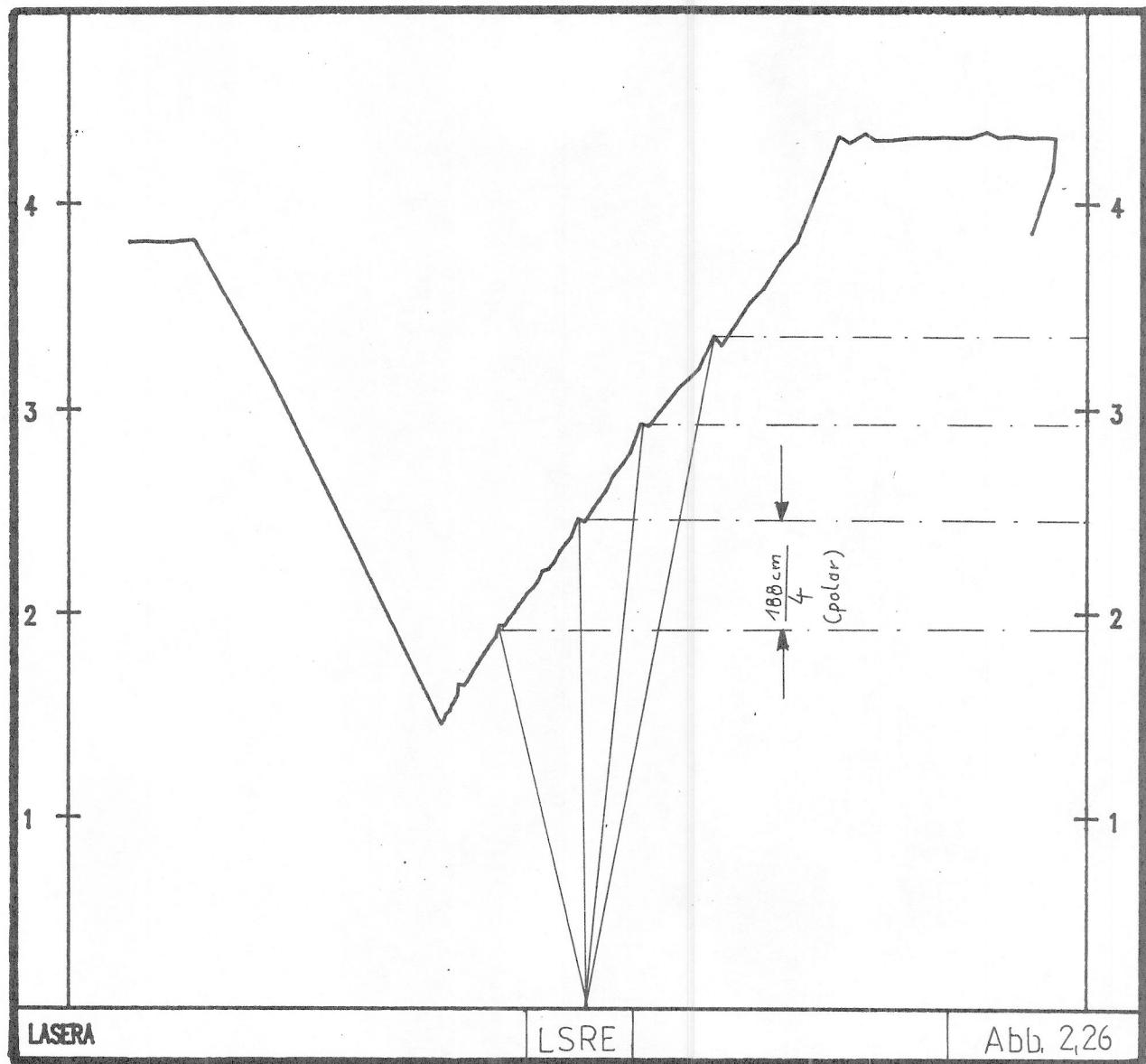
### 2.2.2.3. Linearität

Die in Abb. 2.25 dargestellte Szene wird verwendet, um die Linearität der Abbildung zu überprüfen. Die schrägstehende Wand A muß bei der zeilenweisen Darstellung auf eine Gerade abgebildet werden. Analysiert man das Meßergebnis (s. Abb. 2.251), so zeigt sich, daß die Linearität im Globalen gegeben ist, in zyklischen Abständen von 0,47 m aber Störungen auftreten.

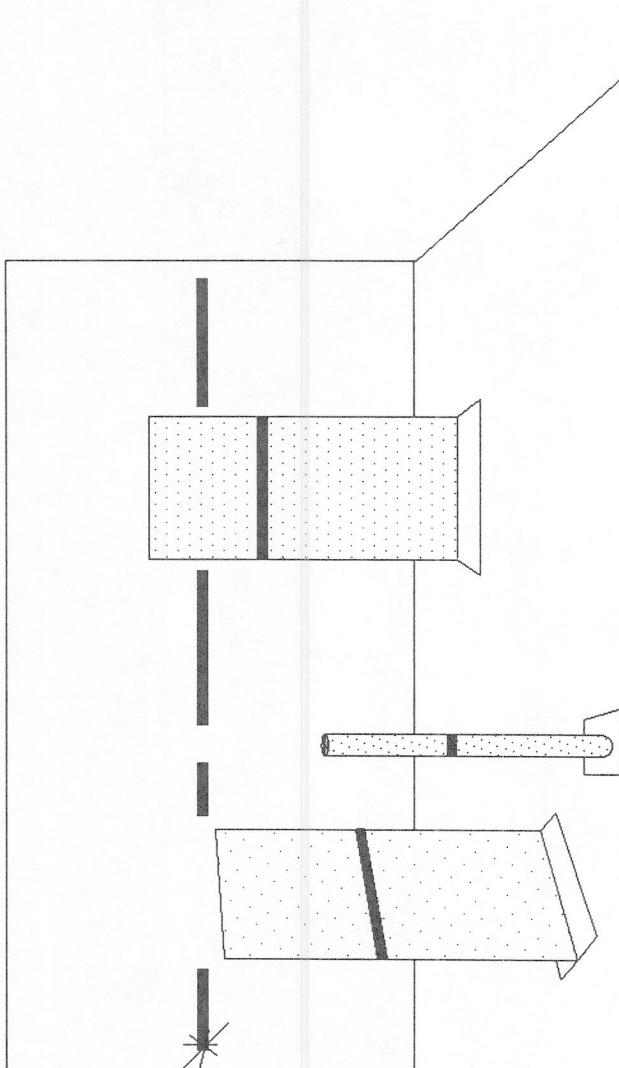


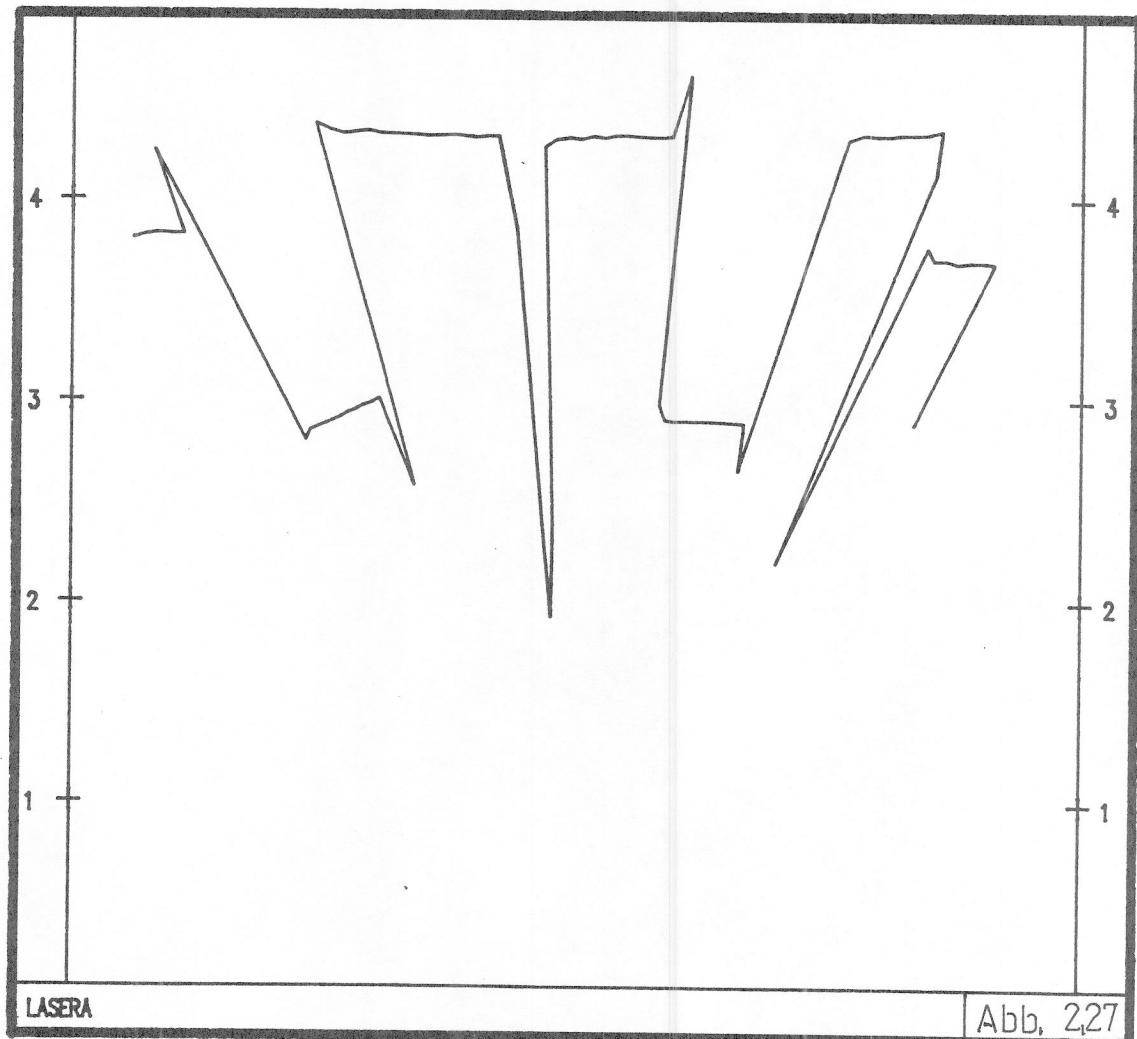
### 2.2.2.4. Kanteneffekte

Bei weiteren Teilszenen, wie z.B. in Abb. 2.26 zu sehen zeigt sich, daß an manchen Sprungkanten ein Überschwingen der Meßwerte auftritt (s. Abb. 2.27). Dieses Überschwingen tritt nur unter bestimmten Sprunghöhen auf und ist durch die analogseitige Übertragungsfunktion, die kein Hochpassverhalten aufweist, nicht zu erklären.



Abl. 2.26





### 2.2.3. Bewertung des Meßverhaltens

Eine zusammenfassende Bewertung des statischen Meßverhaltens der LSR.E. zeigt, daß dort alle charakteristischen Kennlinien anhand einiger grundlegender physikalischer Gleichungen bewertbar sind. Die LSR.E. Eigenschaften sind hier nur von den physikalischen Grenzen bestimmt, weitere "Dreckeffekte" treten nicht auf, die LSR.E. erfüllt hier voll und ganz die Entwicklungsvorgaben.

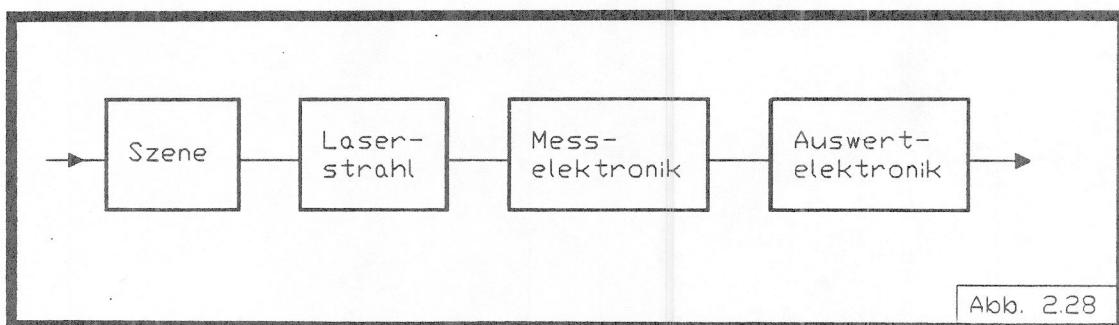
Dies gilt weitgehend auch für das dynamische Verhalten, allerdings bleiben hier 2 Punkte offen. Die Effekte aus Punkt 2.2.2.3. und 2.2.2.4. sind mit naheliegenden Mitteln nicht mehr zu klären.

### 2.3. ENTWICKLUNG EINES SENZORMODELLS

Betrachtet man die Ergebnisse einer LSR.E. Messung z.B. anhand eines Profilschnittes, so ist zwar sofort eine Übereinstimmung mit der vermessenen Orginalszene zu erkennen, eine genaue Validierung ist mit bloßem Auge aber nicht möglich. Insbesondere fällt es schwer, scheinbare Ungenauigkeiten der Messung zu bewerten (s. a. 2.2.2.3. und 2.2.2.4.), da häufig der menschliche Gesichtssinn, der über Details großzügig hinwegsieht, und nicht das Meßgerät LSR.E. falsch arbeitet. Der an dieser Stelle übliche Weg, ein Referenzmeßgerät (= Vergleich auf physikalischer Ebene) einzusetzen, scheidet mangels Verfügbarkeit aus. Stattdessen vergleiche ich die realen LSR.E. Eigenschaften mit einem simulierten Umwelt-Sensor Modell. Hierbei kann ich an verschiedenen Stellen des Modells Störungen aufbringen, um die simulierten an die realen Meßergebnisse anzulegen, und so die Ursachen von Meßfehlern zu ermitteln. Insbesondere erlaubt die Simulation des Umwelt-Sensor Zusammenhänge aufzuzeigen, die durch eine einfache Betrachtung nach dem Schema von "Ursache und Wirkung" nicht mehr zu klären sind.

### 2.3.1. Der Aufbau des Sensormodells

Das gesamte Sensormodell gliedert sich in 4 Teile (s. Abb. 2.28) entsprechend den Komponenten der LSR.E., sowie einer Simulation der Umwelt:

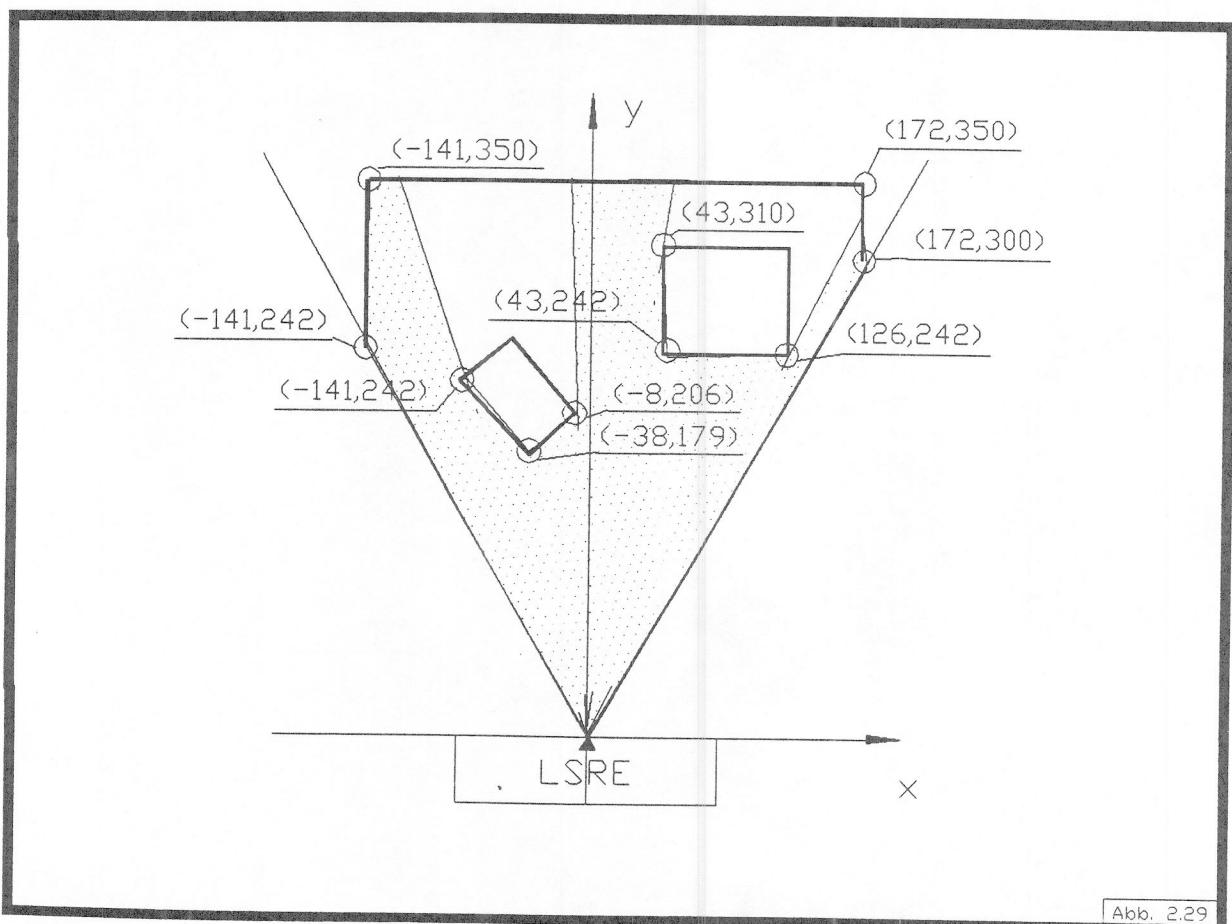


- Umweltmodell zur Simulation der Eingangsdaten
- Modell der Meßelektronik
- Modell der Auswertelektronik

#### 2.3.1.1. Das Umweltmodell

Die LSR.E. setzt ein Gesamtbild aus 41 Zeilen zu je 81 Meßpunkten zusammen. Somit lässt sich die Simulation eines LSR.E. Bildes zeilenweise durchführen. Die weiteren Ausführungen beziehen sich auf jeweils eine Meßzeile.

Jede Zeile beschreibt einen waagrechten Schnitt durch die Szene entsprechend einem Aufriß von oben auf die Umwelt. Dieser Aufriß wird durch Streckenzüge (= Objekte) im karthesischen Koordinatensystem beschrieben (s. Abb. 2.29).



Die LSR.E. blickt vom Koordinatenursprung in Richtung x-Achse. Jede Strecke wird durch Anfangs- und Endpunkt beschrieben. Das Reflektionsverhalten einzelner Objekte, das durch die Winkelstellung ( $\vartheta$ ) zum abtastenden Strahl und deren Oberfläche ( $\vartheta_d$ ) beeinflußt wird, beschreibt man durch eine Zahl  $d$  zwischen 0 und 1 (s. a. 2.2.1.). Es gilt:

$$d = \vartheta_d \cos \vartheta_i$$

Die Zahl 1 steht für den idealen Reflektor senkrecht zum Laserstrahl.

### 2.3.1.2. Modell des Laserstrahls

Die LSR.E. tastet eine Szene der oben beschriebenen Art von links nach rechts ab, der Azimutwinkel liegt  $30^\circ$  links oder rechts der x-Achse.

Der verwendete Laserstrahl besitzt einen Durchmesser von ca. 5 mm, der bei der Simulation nicht vernachlässigt werden darf. Das Laserlicht der LSR.E. wird durch ein Bündel von 100 unendlich dünnen Einzelstrahlen modelliert (s. Abb. 2.30).

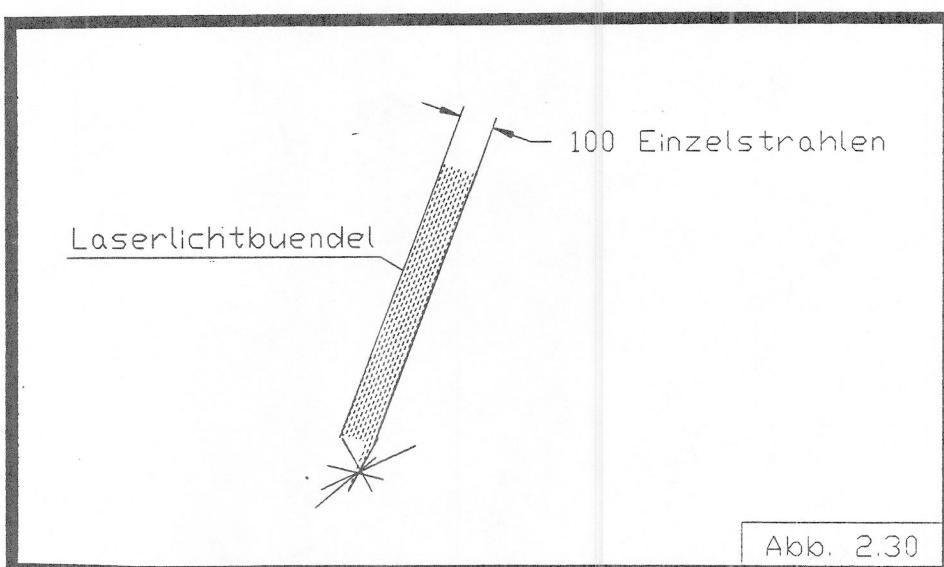


Abb. 2.30

Die Intensität des vom Meßobjekt reflektierten Lichtstrahls hängt, außer vom Objekt, hauptsächlich von der Entfernung zur Laserlichtquelle ab. Wie in 2.2.1. beschrieben (s. a. Abb. 2.8), gliedert sich das Intensitätsverhalten in einen entfernungsabhängigen Teil ( $1/x^2$  Gesetz) und einen Sättigungsbereich. In die Simulation des LSR.E. Empfangspegels gehen beide Komponenten ein.

### 2.3.1.3. Modell der LSR.E. Geometrie und des Abtastverhaltens

Der Meßstrahl wird pro Zeile in 81 Schritten polar über die Szene geführt. Die Entfernung zur LSR.E. sowie die Lichtintensität wird 81 mal für jeden der 100 Einzelstrahlen errechnet (s. Abb. 2.31).

### 2.3.1.4. Modell der Meßelektronik

Die Simulation der Meßelektronik basiert auf dem in 2.1.1. beschriebenen Phasenverschiebungs-Meßprinzip der LSR.E.. Ausgehend von den oben modellierten Entfernungs- und Intensitätsdaten werden jedem Datum 2 Meßzeiger zugeordnet. Für jeden Einzelstrahl des Laserlichtbündels wird für jeden Meßkanal (10 MHz, 80 MHz) ein Zeiger ermittelt. Der resultierende Zeiger pro Abtastpunkt ergibt sich durch die vektorielle Addition aller Einzelzeiger.

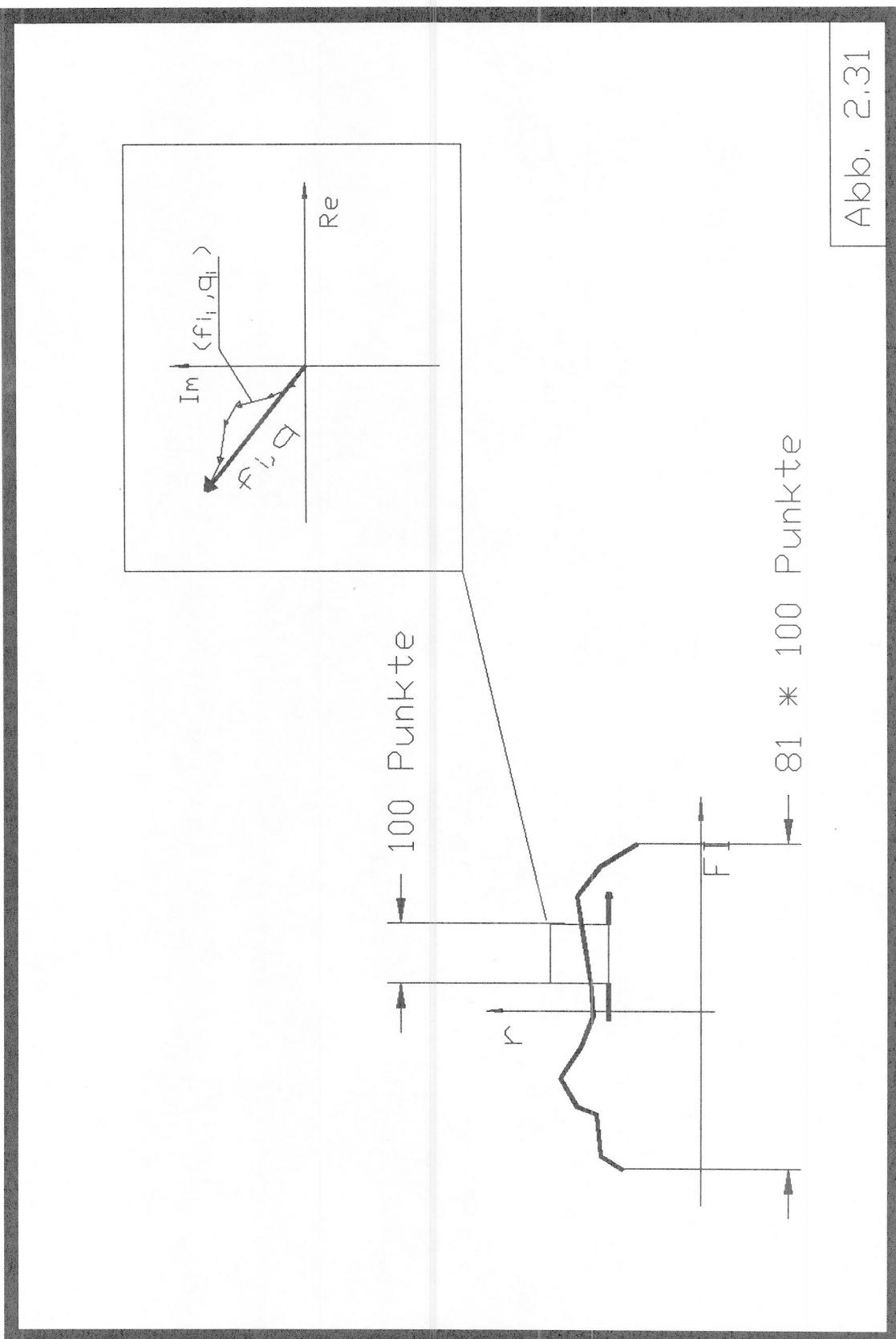
$$Z_{\text{res}} = q_{\text{res}} \begin{pmatrix} \cos \varphi_{\text{res}} \\ \sin \varphi_{\text{res}} \end{pmatrix} = \sum_{n=1}^{100} \begin{pmatrix} q_i \cos \varphi_i \\ q_i \sin \varphi_i \end{pmatrix};$$

(Gleichung 2.8)

$\varphi_i$  ein berechnet sich aus der Entfernung, dem Eindeutigkeitsbereich des Meßkanals und einem absoluten Phasenoffset.

$$\varphi_i = \lambda_{80/10} \cdot 2\pi + \psi_{\text{offset}}$$

(Gleichung 2.9)



Ist die Entfernung bzw. die Qualität gleich für alle Einzelstrahlen, so ergibt sich für den resultierenden Vektor:

$$\begin{pmatrix} \varphi_{res} \\ q_{res} \end{pmatrix} = \begin{pmatrix} \varphi_i \\ 100 \cdot q_i \end{pmatrix};$$

(Gleichung 2.10)

Dies gilt jedoch nicht für schräg zum Lichtstrahl stehende Objekte (Abb. 2.32) oder Sprungkanten (s. Abb. 2.33).

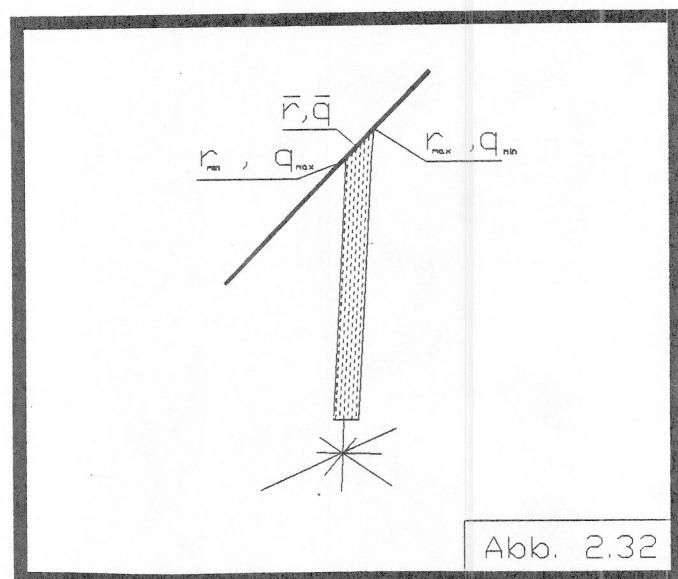


Abb. 2.32

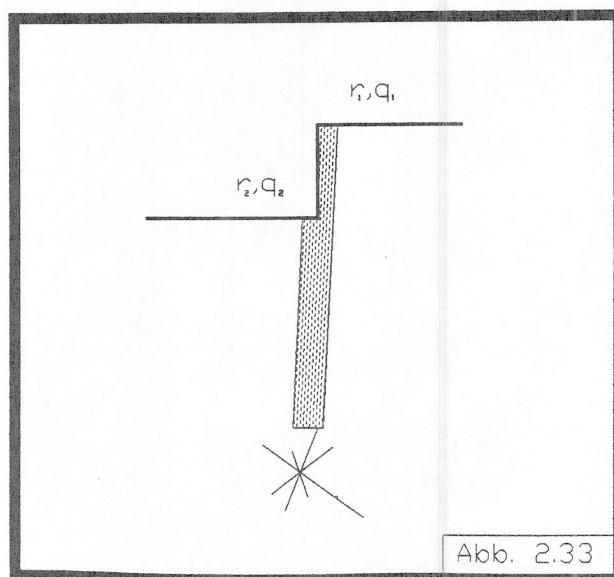
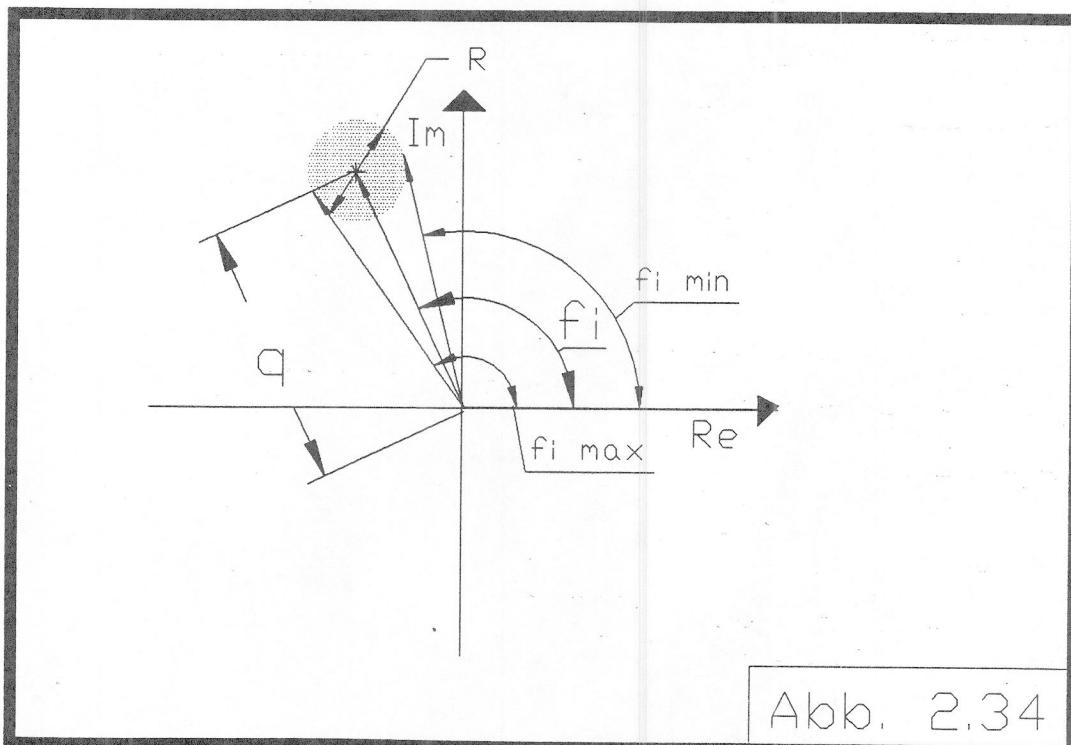


Abb. 2.33

### 2.3.1.5. Modellierung von Störeffekten

#### a) Rauschen

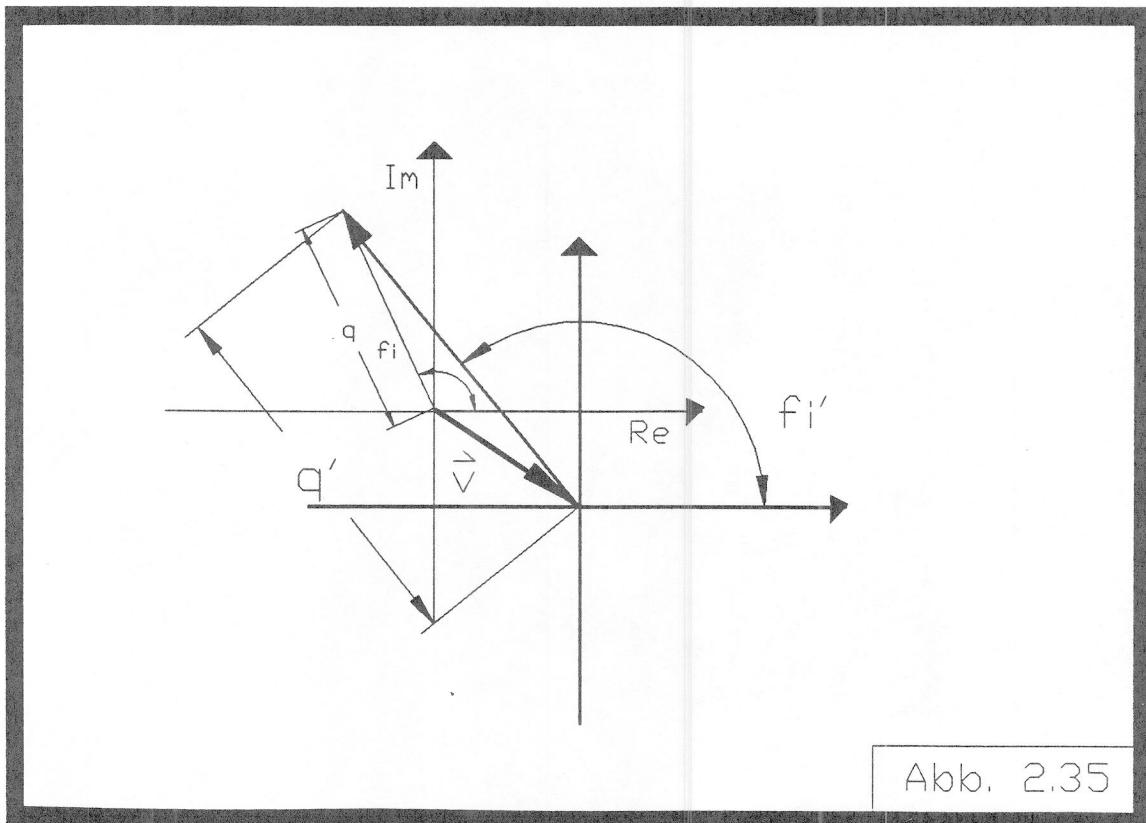
Die ankommenden Meßsignale sind verrauscht (s. 2.1.1.), was durch die Angabe einer Rauschzahl im Modell berücksichtigt wird. Betrachtet man das reale Zeigersignal auf dem Oszilloskop, so zeigt sich das Rauschen als "schneeballförmiges" Gebilde an der Zeigerspitze. Die Rauschzahl  $R$  beschreibt dessen relativen Durchmesser (s. Abb. 2.34).



Da das Rauschen beider Kanäle, bedingt durch die unterschiedlichen Frequenzen, Modulationsgrade, Bauelementestreuungen etc. unterschiedlich ist, wird mit zwei Rauschzahlen  $R_{80}$  und  $R_{10}$  gearbeitet.

#### b) Gleichspannungsoffset

Durch thermischen Drift etc. wird dem Zeigersignal ein geringer, aber messbarer Gleichspannungsoffset überlagert, d. h. der Ursprung der Zeigerkoordinatensysteme wird verschoben (s. Abb. 2.35).



Durch Eingabe des Verschiebervektors  $\vec{v}$  für beide Kanäle wird diese Störung im Modell berücksichtigt.

### c) Phasenoffset

Die beiden Modulationsfrequenzen werden quarzstabil, aber getrennt erzeugt und verarbeitet.

$$\Psi_{\text{offset},80} - \Psi_{\text{offset},10} = \Gamma$$

Alle sechs ( $R10$ ,  $R80$ ,  $\vec{v}$ ,  $\psi_{80}$ ,  $\psi_{10}$ ,  $\Gamma$ ) obigen Werte werden getrennt im Modell berücksichtigt.

### 2.3.1.5. Die Auswertelelektronik

In der Auswertelelektronik werden u. a. die analogen Zeigersignale digitalisiert und zum entgültigen Entfernungsmesswert verknüpft. Zusätzlich wird ein Qualitätssignal ausgegeben.

#### a) Schnittstelle Analog - Digital

Rauschen, Gleichspannungsoffset und Phasenoffset wirken sich hieraus b.z.w. gehen auch in das digitale Signal mit ein.

Das Rauschen beeinflusst hier die Meßgenauigkeit in Abhängigkeit von der Qualität (s. Abb. 2.34).

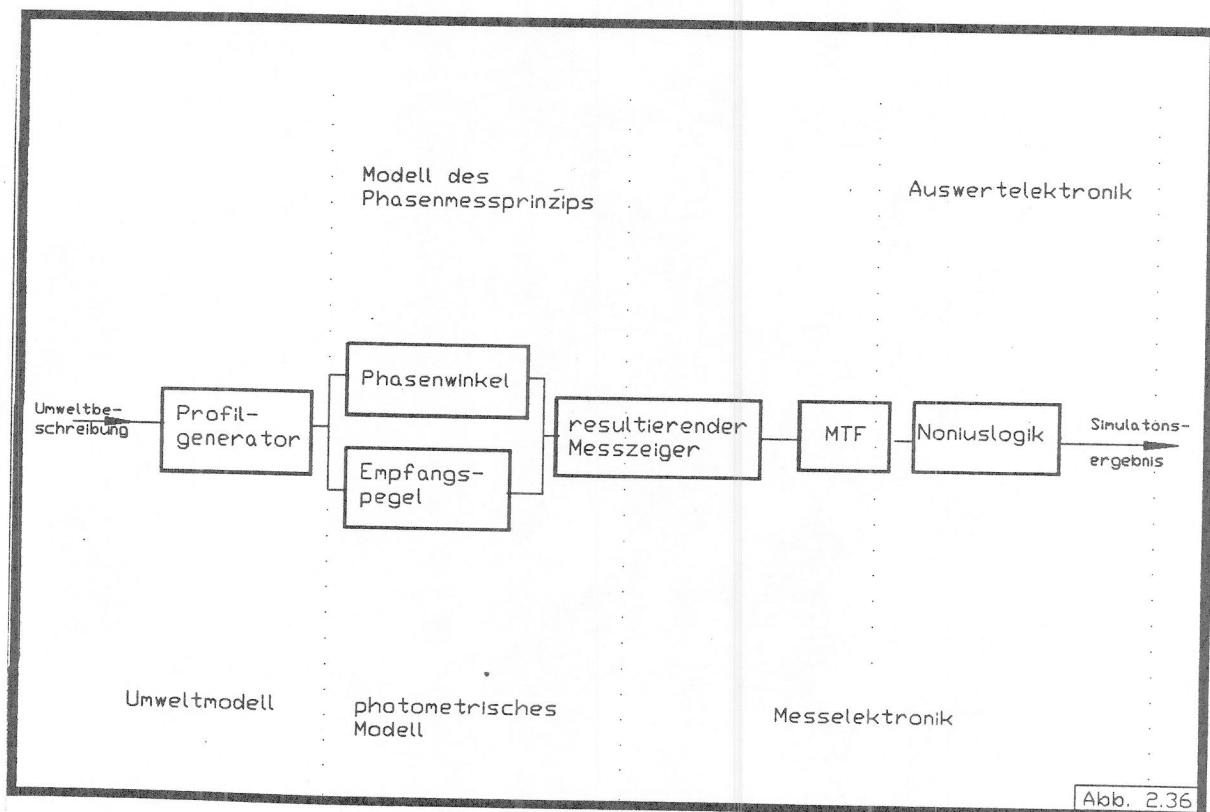
Aus der Abbildung ist ersichtlich, daß es sinnvoll ist, mit drei Alternativen ( $\varphi_{\min}$ ,  $\varphi_{\max}$ ,  $\bar{\varphi}$ ) weiterzuarbeiten.

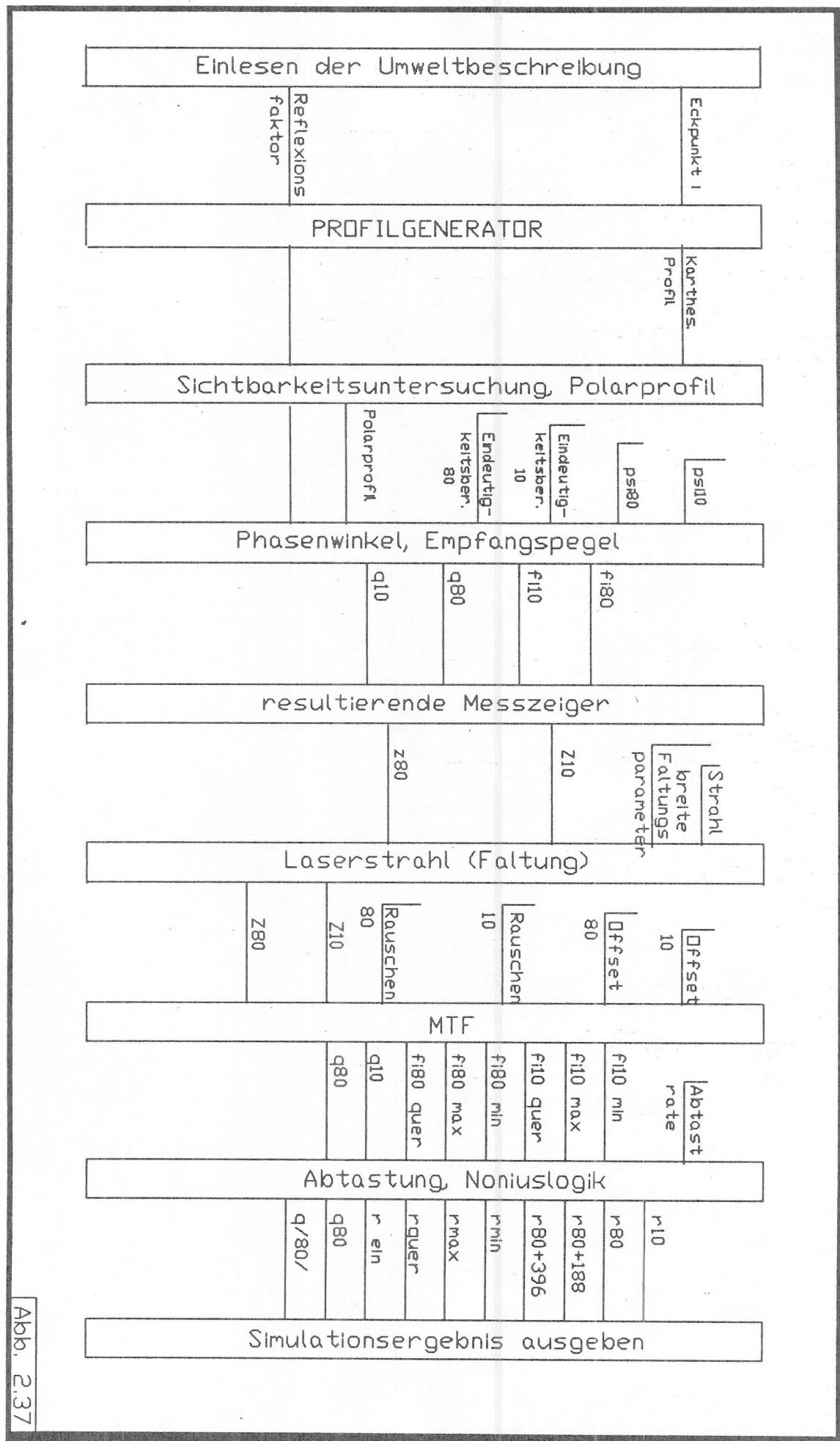
#### b) Qualität

Aus datenverarbeitungstechnischen Gründen wird die Qualität nicht als Länge des Zeigers, sondern als Betrags-Norm weitergeführt  
Echte Zeigerlänge und Betragsnorm werden vom Modell ausgegeben.

#### c) Noniuslogik

Die Verknüpfung der beiden Phasensignale zum entgültigen Meßergebnis erfolgt in der "Noniuslogik". Schliesslich wird hier mit Hilfe von Phasenoffsetwerten (absolut und relativ) der Meßwert auf tatsächliche cm Abstand vom LSR.E. Nullpunkt bezogen. Diese Offsetwerte gehen hier in die Simulation ein. Da wir uns nun auf der digitalen Seite der Elektronik befinden, kann dieser Teil der LSR.E. eindeutig in das programmierte digitale Modell übertragen werden. Einen Gesamtfunktionsplan des Sensormodells zeigt Abb. 2.36. Abbildung 2.37 fasst die Simulationsschritte und die dort benötigten b.z.w.errechneten Parameter nochmals zusammen.





### 2.3.2. Simulationsmöglichkeiten

Abhängig von der Eingangszene, liefert das Modell verschiedene Beschreibungen dieser Szene.

Zum Beispiel:

- Distanz - Qualität
- Verhalten der Messung an schräg stehenden Objekten
- Streuung der Meßergebnisse - Distanz, Qualität
- tatsächliche Qualität - ausgegebene Qualität
- Meßverhalten an Kanten

Die analoge Dynamik des Meßsystems (MÜF) wurde nicht mit berücksichtigt, da die horizontale Ortsauflösung, bedingt durch die Abtastrate, weit unter der Auflösung liegt, die durch die Bandbreite der analogen Kanäle möglich wäre (s. Kap. 2.2.2.). Mit Hilfe dieser Softwaresimulation durch das Programm SIMULI ist es möglich, systematische Ursachen augenscheinlicher Meßfehler nachzuweisen. Da im Sensormodell das gesamte Wissen über die LSR.E.-Umweltstrukturen abgebildet ist, erlaubt die Simulation ins besondere eine Beurteilung des Gesamtverhaltens im Zusammenspiel aller Systemkomponenten. Weiter wird das Sensormodell oder spezielle Teile davon, nicht nur zur Dokumentation des Meßverhaltens (offline), sondern auch prozess-, d.h. hier messbegleitend, eingesetzt. So kann, gestützt auf das Modellwissen vereinzelt im Raum stehender Objekte, von systematisch bedingten Meßfehlern unterschieden werden.

### 2.3.2 Programmtechnische Realisierung des Sensormodells

Das Sensormodell wurde durch das Programm SIMULI (s. Anhang 1) in der Programmiersprache Pascal, unter dem Betriebssystem VMS, auf Rechnern vom Typ VAX 11/750 bzw. MicroVAX II implementiert. Eingangszene und simulierte Ausgangsdaten werden auf Textfiles abgespeichert. Die Eingabe der Simulationsparameter wie Rauschzahl etc. erfolgt interaktiv am Terminal. Einige, im allgemeinen feste Parameter, sind als Konstante im Programm definiert (siehe Programmdokumentation Anhang 1).

### 2.3.2.1. Eingangsdaten und Eingangsdatenformat

Wie in 2.3.1. erklärt wird die Sensorumwelt durch Streckenzüge in karthesischen Koordinaten beschrieben.

Diese Szenenbeschreibung wird vor Programmstart mit Hilfe eines Editors in einer Textdatei abgelegt. Diese Textdatei muß folgendes Datenformat besitzen.

Szenenbeschreibung zu Abb. 2.38:

```
-141 242 -38 179 0.5
-38 179 -8 206 0.6
-141 242 -141 350 0.3
-141 350 172 350
172 350 172 300 0.9
43 310 43 242 0.6
43 242 126 242 0.8
```

Jeder Streckenzug (= Objekt) wird in genau einer Textzeile beschrieben. Die Beschreibung gliedert sich in 2 Koordinatenpaare (Einheit cm!): Axstart, Aystart, Exende, Eyende und dem Qualitätsbeiwert  $q_{bei}$ .

Alle Werte sind durch ein Leerzeichen getrennt. Die Überlappung der Streckenzüge (hintereinanderstehende Objekte) ist zulässig. Die Koordinaten sind vom Typ Integer, der Qualitätsbeiwert vom Typ Real im Bereich von 0,01 ....1.

### 2.3.2.2. Programmbeschreibung

Abbildung 2.39 zeigt das Struktogramm des Programmablaufes.

Nach Einlesen der Eingangsdaten berechnet SIMULI die Raumpunkte zwischen Anfangs- und Endpunkten der Streckenzüge.

Diese Raumpunkte werden in polare Koordinaten transformiert ( $r, \Phi$ ); Punkte, die sich aus der Sicht der LSR.E. verdecken, werden eliminiert. Jedem Entfernungswert ordnet die nächste Routine mit Hilfe des Qualitätsbeiwertes und der Linsencharakteristik eine Qualität zu. Getrennt für den 10 MHz und 80 MHz Kanal erfolgt nun eine Berechnung der Meßzeiger.

Aus den jeweils 100 Abtastwerten wird durch vektorielle Addition ein resultierender Meßzeiger errechnet (100 Einzelstrahlen eines Laserlichtbündels), indem man ein 100 Abtastwerte breites Fenster über die Szene schiebt (Faltung). Das Programm addiert nun zu jedem der resultierenden Zeiger einen Gleichspannungsoffset (vektoriell). Weiterhin wird zum Zeigerwert  $z$  ein Randvektor rechtwinkelig ( $\pm \pi/2$ ) aufaddiert, man erhält so 3 Zeiger:

$z$ ,  $z$  min und  $z$  max.

Um die Abtastung in der LSR.E. nachzubilden, werden die pro Kanal 3 \* 81000 Zeiger = 250000 Entfernungswerte in gleichmäßigen Raumwinkelschritten abgetastet.

Die Zeiger sind bis hier noch im Format Real- und Imaginärteil abgebildet.

Entsprechend der Konvertierung in der LSR.E.-Auswertelektronik werden diese Zeiger nun in die Darstellung Betrag und Phase gewandelt. Dabei findet eine mehrfache Betragsbildung statt.

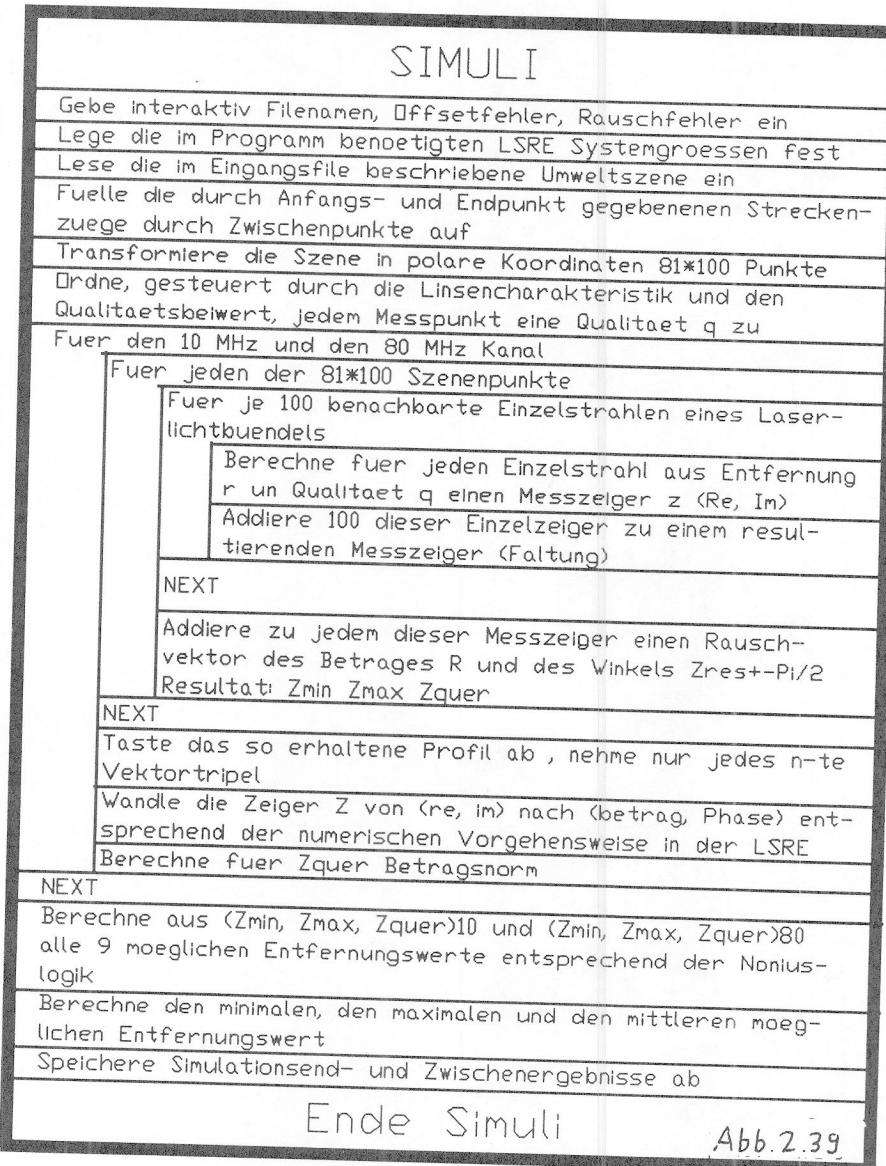
Zum einen wird der echte Betrag (=Qualität) nach dem Euklidischen Abstand (Euklidische Norm) gebildet, zum anderen wird wie in der LSR.E.-Elektronik, die Betragssnorm errechnet. Es stehen nun für jeden der beiden Meßkanäle 3 Zeiger mit  $\varphi$  min,  $\varphi$  max,  $\bar{\varphi}$  und  $q$  min,  $q$  max und  $q$  zur Verfügung.

Die Prozedur "Noniuslogik" verknüpft die 2 \* 3 Zeiger zu 9 möglichen Entfernungswerten.

Von diesen 9 Entfernungswerten werden der Minimale, der Maximale und der Mittelwert errechnet. Die Abspeicherung aller End- und Zwischenergebnisse an einer Textdatei beendet das Programm SIMULI.

### 2.3.2.3. Die Ausgangsdaten und deren Format

Als Ergebnis wird eine Matrix von Integerzahlen, getrennt durch Kommata, ausgegeben. Jede Matrixzeile steht für einen LSRE. Meßpunkt in der Reihenfolge: 1. Meßpunkt (= - 30° Azimut) bis letzter Meßpunkt (= + 30° Azimutwinkel).



Die Matrixspalten haben folgende Überschriften und Bedeutung:

entf:

Tatsächliche Entfernung r von der LSR.E. zum  
Meßobjekt

größtwert:

Aufgrund aller Meßeinflüsse maximal  
ausgegebene Entfernung r

kleinstwert:

Aufgrund aller Meßeinflüsse minimal  
ausgegebene Entfernung r

mittelwert:

Mittlere gemessene Entfernung

zehn:

Entfernungswert, vom 10 MHz Kanal ermittelt

achtzig:

Entfernungswert, vom 80 MHz Kanal ermittelt

achtzig 1:

achtzig + 188 cm

achtzig 2:

achtzig + 2 \* 188 cm

achtzig 3:

achtzig + 3 \* 188 cm

(Mehrdeutigkeitsbereiche)

qualroh:

Qualität, ermittelt nach der Euklidischen Norm (echte  
Zeigerlänge)

pseudoqual:

Qualität, ermittelt nach der Betragssnorm

#### 2.3.2.4. Auswertmöglichkeiten

Mit Hilfe von Standardprogrammen ist es nun möglich, das vom Programm SIMULI errechnete Simulationsergebnis, nämlich die Ergebnismatrix, weiter auszuwerten. Nach dem Übertragen des Ergebnisfiles über das LSSR Rechnernetz auf einen Arbeitsplatzrechner unter MS-DOS verwende ich ein Graphik-Kalkulationsprogramm zur Darstellung und Bewertung der Ergebnisse.

Programme dieser Art erlauben es, Abhängigkeiten zwischen beliebigen Spalten und Zeilen einer Matrix graphisch darzustellen.

Folgende Auswertungen unter vielen Möglichen habe ich durchgeführt:

- Verhalten des 80 MHz Kanal im Verhältnis zum Verhalten des 10 MHz Kanal
- Darstellung Meßfehler - Qualität
- Darstellung Meßfehler - Entfernung
- Darstellung eines Polarprofilschnittes der Rohszene
- Darstellung eines Polarprofilschnittes der simulierten Daten
- Darstellung karthesischer Profile der Eingangs- und Ausgangsdaten

#### 2.4. VALIDIERUNG DES MEßTECHNISCHEN VERHALTENS DER LSR.E.DURCH DAS SENZORMODELL

Durch das oben beschriebene Sensormodell validiere ich das in Kap. 2.2. beschriebene Meßverhalten der LSR.E.. Annahmen über die systematischen Ursachen von Meßfehlern können gestützt oder verworfen werden; auch erlaubt das Sensormodell die Klärung noch unbekannter Ursachen von charakteristischen Meßfehlern der LSR.E.(s. Kap. 2.2.2.2.).

#### 2.4.1. Statische Messungen, simuliert mit dem Sensormodell

Von besonderem Interesse ist die Abhängigkeit der gemessenen Entfernung von Meßfehlern und Qualität. Zur Modellierung dieses Zusammenhangs wird die in Abb. 2.25 gezeigte Szene mit dem Sensormodell simuliert. Die mit  $w$  bezeichnete schräge Wand entspricht kontinuierlich zunehmenden Entfernungswerten bei abnehmender Qualität.

Folgende 4 Fälle wurden untersucht:

##### 2.4.1.1. Meßfehler – Entfernung bei konstantem Rauschmaß $R$

Abbildung 2.40 zeigt den simulierten Zusammenhang, geliefert vom Sensormodell.

Kurve O zeigt die Obergrenze, Kurve U die Untergrenze des möglichen Entfernungsfehlers. Vergleicht man hiermit den real gemessenen Zusammenhang in Abb. 2.9, so zeigt sich eine hervorragende Übereinstimmung der Ergebnisse.

##### 2.4.1.2. Meßfehler – Qualität Bei konstantem Rauschmaß $R$

Abbildung 2.41 zeigt den Zusammenhang von Meßfehlern zu Qualität, wobei Kurve O wieder die obere, und Kurve U die untere Fehlergrenze darstellen. Eine augenscheinliche Übereinstimmung der Kurven zu Abb. 2.10 (gemessene Kurve) ist festzustellen. Allerdings ist die charakteristische Asymmetrie zur Fehlernulllinie in der Orginalmessung nicht wiederzufinden, weiterhin findet auch der tendenzielle Ausreißer zu negativen Fehlern bei sehr kleinen Werten  $q$  keine simulative Bestätigung. Absolute quantitative Unterschiede der Meßfehler sind auf das willkürlich gewählte Rauschmaß zurückzuführen.

##### 2.4.1.3. Meßfehler – Qualität – bei konstantem Rauschmaß $R$ und zusätzlichem Gleichspannungsoffset auf den Phasenmeßkanälen

Messfehler - Entfernung  
 $v_{10}, v_{80} := 0; R = 15$

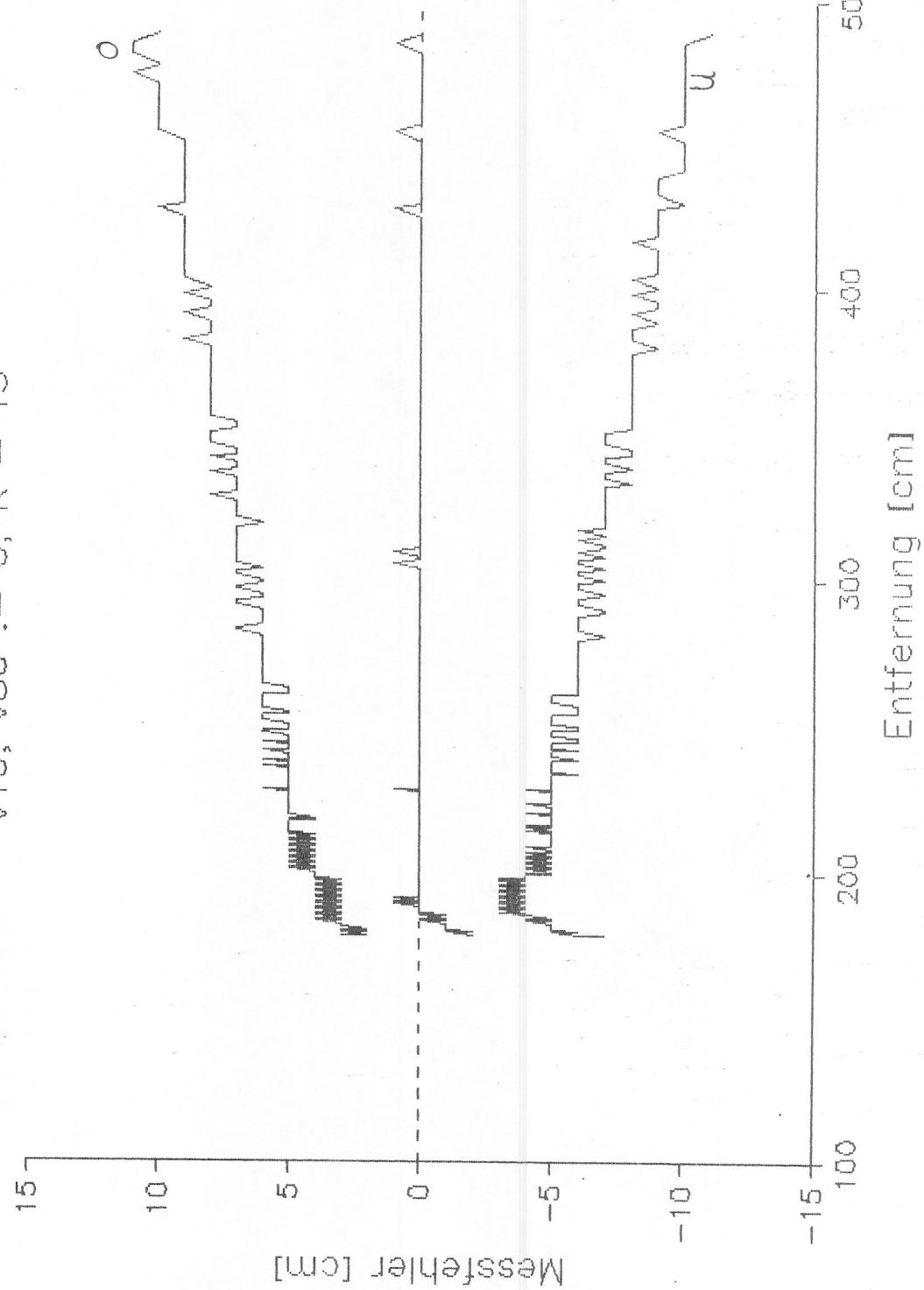


Abb 240

# Messfehler - Qualität

$v_{10}, v_{80} := 0; R = 15$

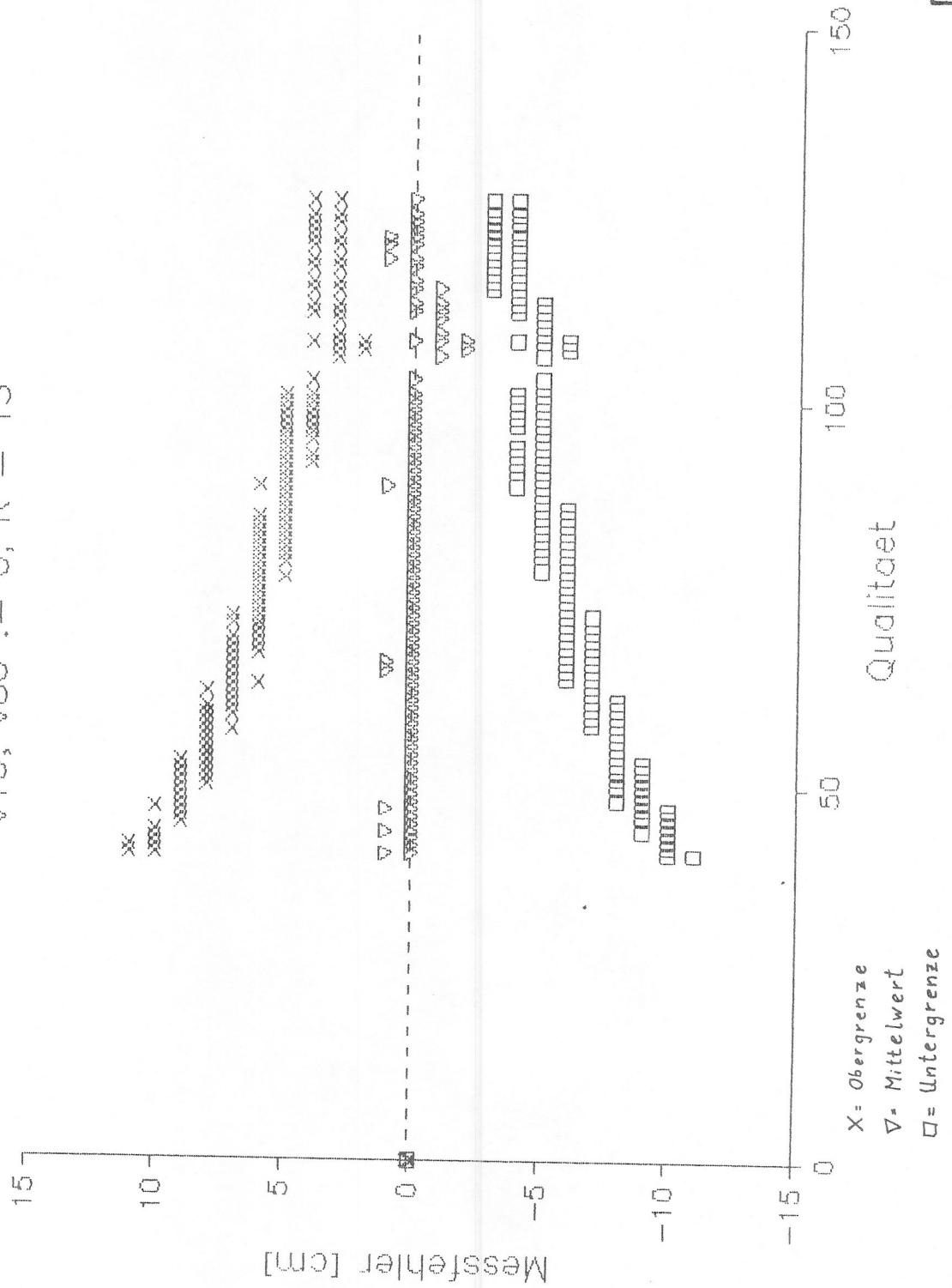


Abb. 241

$v_{10} = (5,5); v_{80} = (-5,5); R = 5$

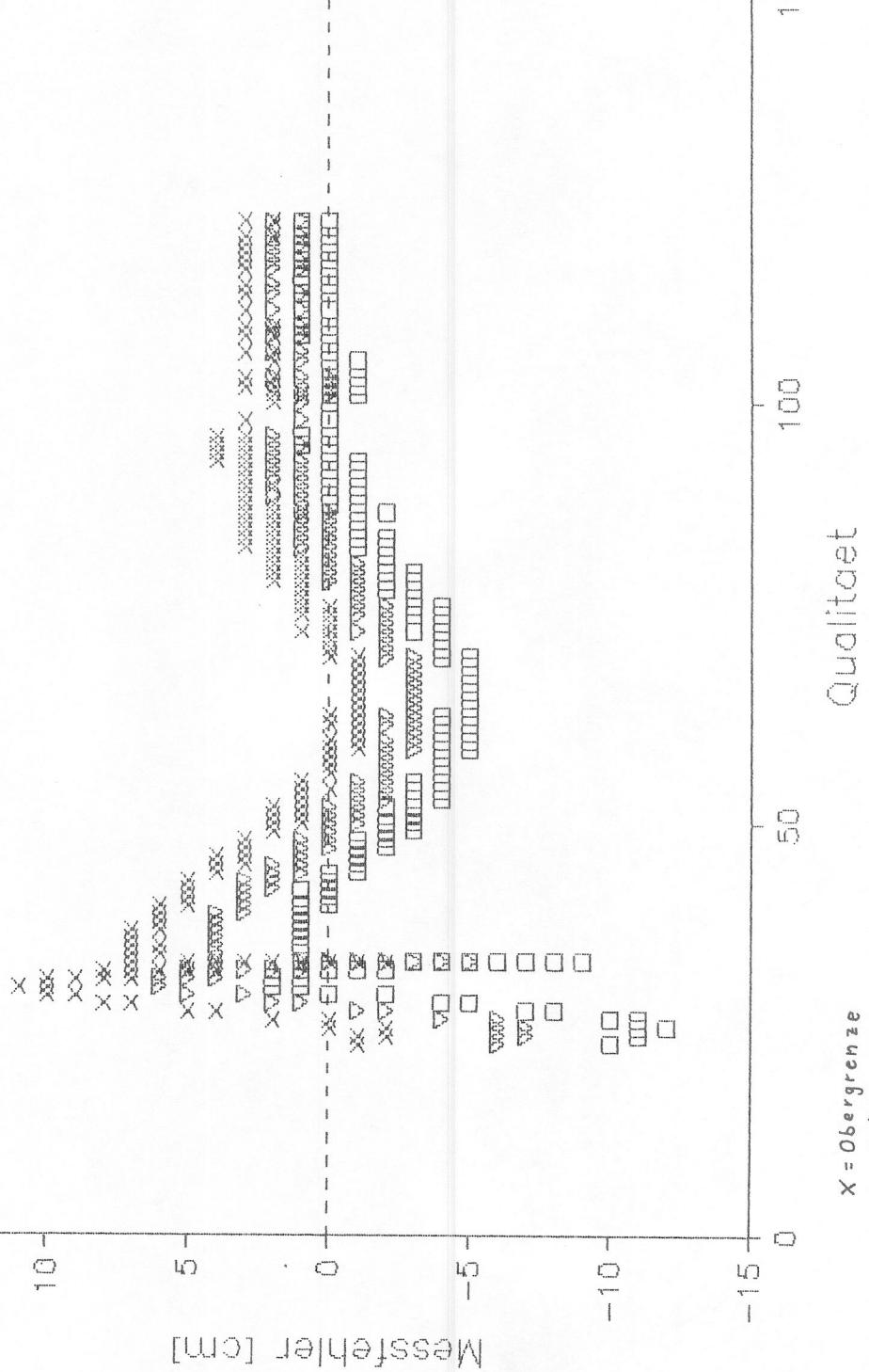


Abb. 24.2

In der in Abb. 2.42 dargestellten Simulation wurde die Rauschzahl R vermindert, gleichzeitig aber ein Gleichspannungsoffset ( $\vec{v}_{10} = +5,5$ ;  $\vec{v}_{80} = -5,5$ ) aufgebracht. Sowohl quantitativ als auch qualitativ konnte so ein besserer Angleich der Simulation an die realen Gegebenheiten errechnet werden. Der Meßfehler nimmt hier ein zyklisches Verhalten an, wobei die Hüllkurve aus Abb. 2.41 erhalten bleibt (s. Abb. 2.43).

#### 2.4.1.4. Meßfehler - Entfernung bei konstantem Rauschmaß R und zusätzlichem Gleichspannungsoffset auf den Phasenmeßkanälen

Mit den gleichen Simulationseckwerten wie in 2.4.1.3. wird nun die Relation Meßfehler - Entfernung dargestellt. Auch hier zeigt sich ein zyklisches Verhalten, bedingt durch den Gleichspannungsoffset. Die Hüllkurve kann aus der nicht offsetbehafteten Simulation bleiben erhalten.

#### 2.4.2. Dynamische Messungen, simuliert mit dem Sensormodell

Besonders das dynamische Meßverhalten (s. Kap. 2.2.2.) der LSR.E. kann mit dem Sensormodell untersucht werden.

##### 2.4.2.1. Modellierung des Linearitätsverhaltens

Wie in 2.2.2.3. näher beschrieben zeigt die LSR.E. zyklische Schwankungen in der Linearität.

Dieser bisher noch nicht begründbare Effekt kann mit Hilfe des Sensormodells erklärt werden.

Betrachtet man in Abb. 2.43 den Meßfehler in Abhängigkeit von der Entfernung, so zeigt sich ein zyklischer Meßfehler mit zunehmender Amplitude. Dieser Meßfehler nimmt alle 188cm/2 ein Maximum an. Wie die Simulation (s. o.) zeigt, tritt dieses periodische Verhalten nur dann auf, wenn die Phasensignale mit einem Offset beaufschlagt werden. Form und Stärke des Linearitätsfehlers ist dabei vom Betrag und der relativen Winkelstellung der beiden Offsetvektoren v 10 und v 80 zueinander, abhängig. Zusätzlich nimmt die Rauschzahl R Einfluß auf das Meßfehler Amplitude.

#### 2.4.2.2. Der Kanteneffekt

Ein weiterer, nur durch das Sensormodell begründbarer Meßeffekt der LSR.E., ist das in Kap. 2.2.2.4. beschriebene Entfernungsmeßverhalten an Kanten.

##### a) Simulation des Effektes

Zum Nachweis des Kanteneffektes wird eine reale Szene ausgewählt, bei der dieses Verhalten besonders deutlich sichtbar ist (s. Abb. 2.44). Diese Szene wird nun durch die Streckenzug - Endpunkte seiner Meßobjekte beschrieben (s. Abb. 2.45).

Diese Szenenbeschreibung dient als Eingangsdatensatz für das Simulationsprogramm SIMULI.

Um störende Nebeneffekte auszuschalten, sind zusätzliche Simulationsparameter, wie Offset und Rauschen, zu Null gesetzt. Das Simulationsergebnis ist in Abb. 2.61 zu sehen. Da die im Realen beobachteten Kantenüberschwinger auch durch das Sensormodell nachgebildet werden, steht fest, daß es sich um ein systematisches reproduzierbares Meßverhalten der LSR.E. handelt. Zur Ursachenforschung werden nun mehrere Ausgangs- und Zwischenergebnisse der Simulation, die in der Ergebnismatrix ausgegeben werden, in Abhängigkeit voneinander ausgewertet. Besonderes Augenmerk gilt dabei natürlich dem Verhalten im Bereich der Kanten.

Messfehler - Entfernung  
 $v_{10} = (5,5); v_{80} = (-5,5); R = 5$

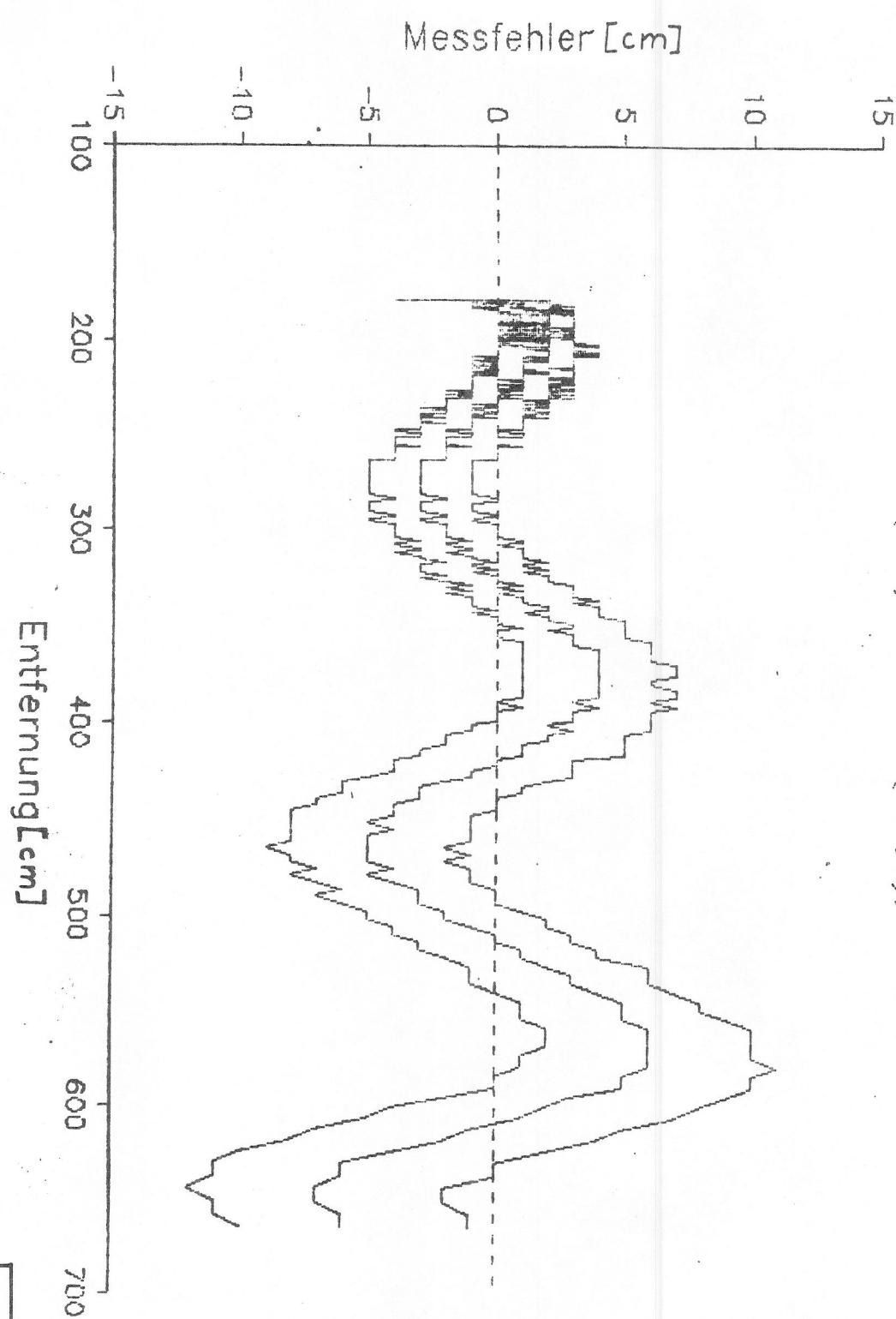


Abb. 243

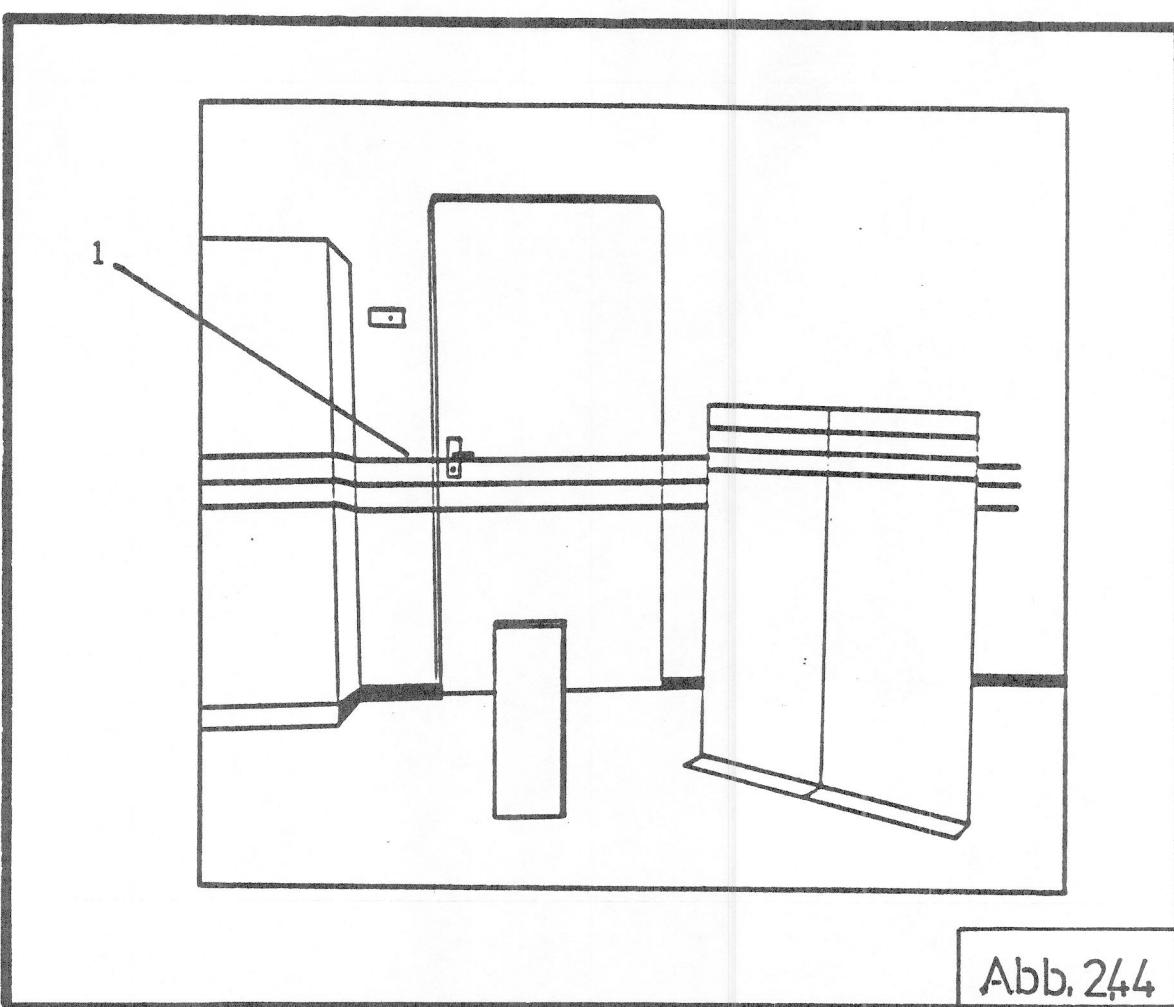
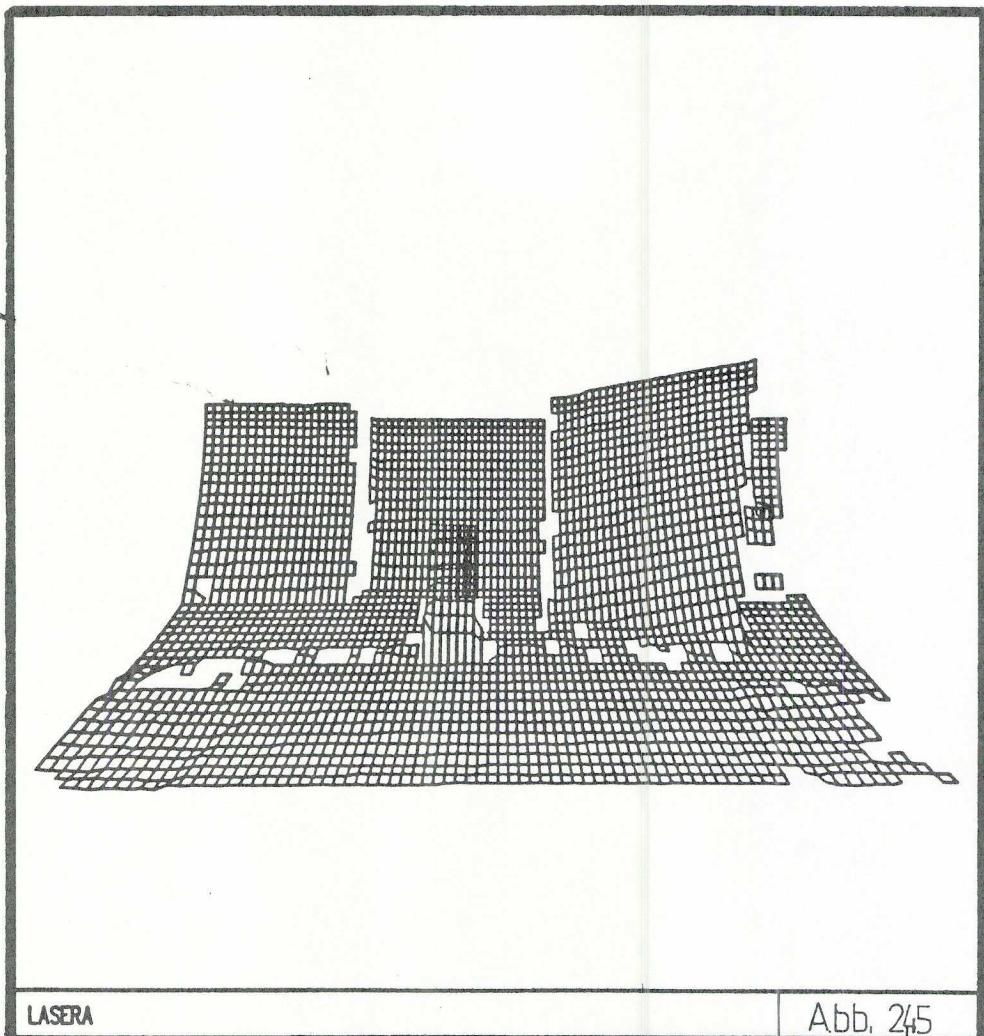


Abb. 244



LASERA

Abb. 245

Die in Abb. 2.46 mit A markierte Kante sei hier Beispiel für die weiteren Überlegungen.

Abb. 2.47 zeigt mehrere interne Zwischenergebnisse der Entfernungsmessung in Abhängigkeit von der Winkelstellung  $\Phi$  des Azimutwinkels. Die interne Winkelstellung  $\varphi_{10}/\varphi_{80}$  wird dabei in Entfernungswerte zurückgerechnet.

Es sind dargestellt:

- grün: Messkurve des 10 MHz Kanal
- gelb: Entscheidungsschwellen der Noniuslogik; sie legen fest, welcher der mehrdeutigen 80 MHz Kanal ausgewählt wird.
- blau: Die 2 in diesem Bereich möglichen, von der Noniuslogik wählbaren, 80 MHz Entfernungsmeßwerte:
- rot: Der von der Noniuslogik ausgewählte Entfernungsmeßwert.

Wie man sofort sieht, entspricht die entgültige Entfernungskurve (rot) genau dem auch real beobachteten Verhalten.

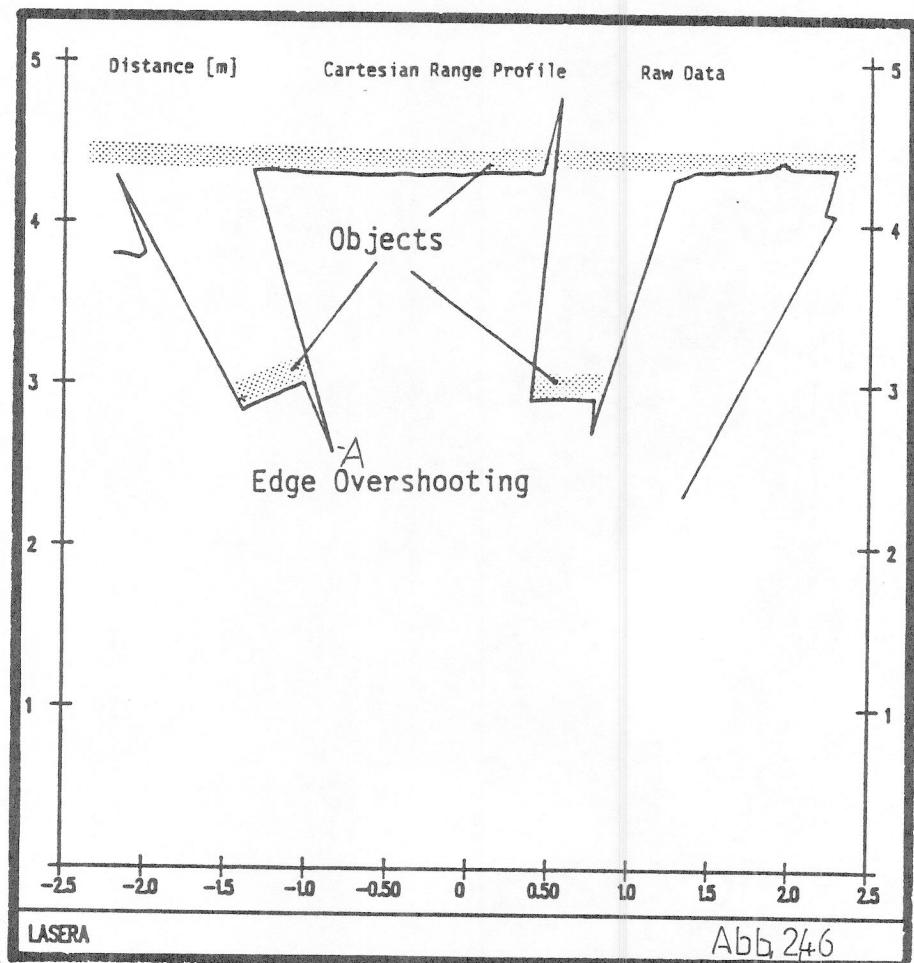
Einem Entfernungssprung von 337 cm nach 489 cm folgen die L.S.R.E. und das Sensormodell, indem zuerst ein Entfernungswert  $< 337$  cm, und dann ein Entfernungswert  $> 489$  cm angenommen werden.

Nach diesem Überschwingen erreicht der Meßwert schließlich den Wert der tatsächlichen Objektentfernung. Die Entfernung von Spitze zu Spitze (s. Abb. 2.41 und Abb. 2.46) beträgt dabei:

$$r_{\text{Sprung}} = n \cdot 188 \text{ cm};$$

#### b) Begründung des Effektes

Wie in Kap. 2.1. erklärt, wählt die Noniuslogik denjenigen 80 MHz Entfernungswert  $r_{80} + n + 188$  cm aus, der im Entscheidungsbereich des 10 MHz Kanals liegt.



Verhalten an Sprungkante  
337 cm nach 489 cm

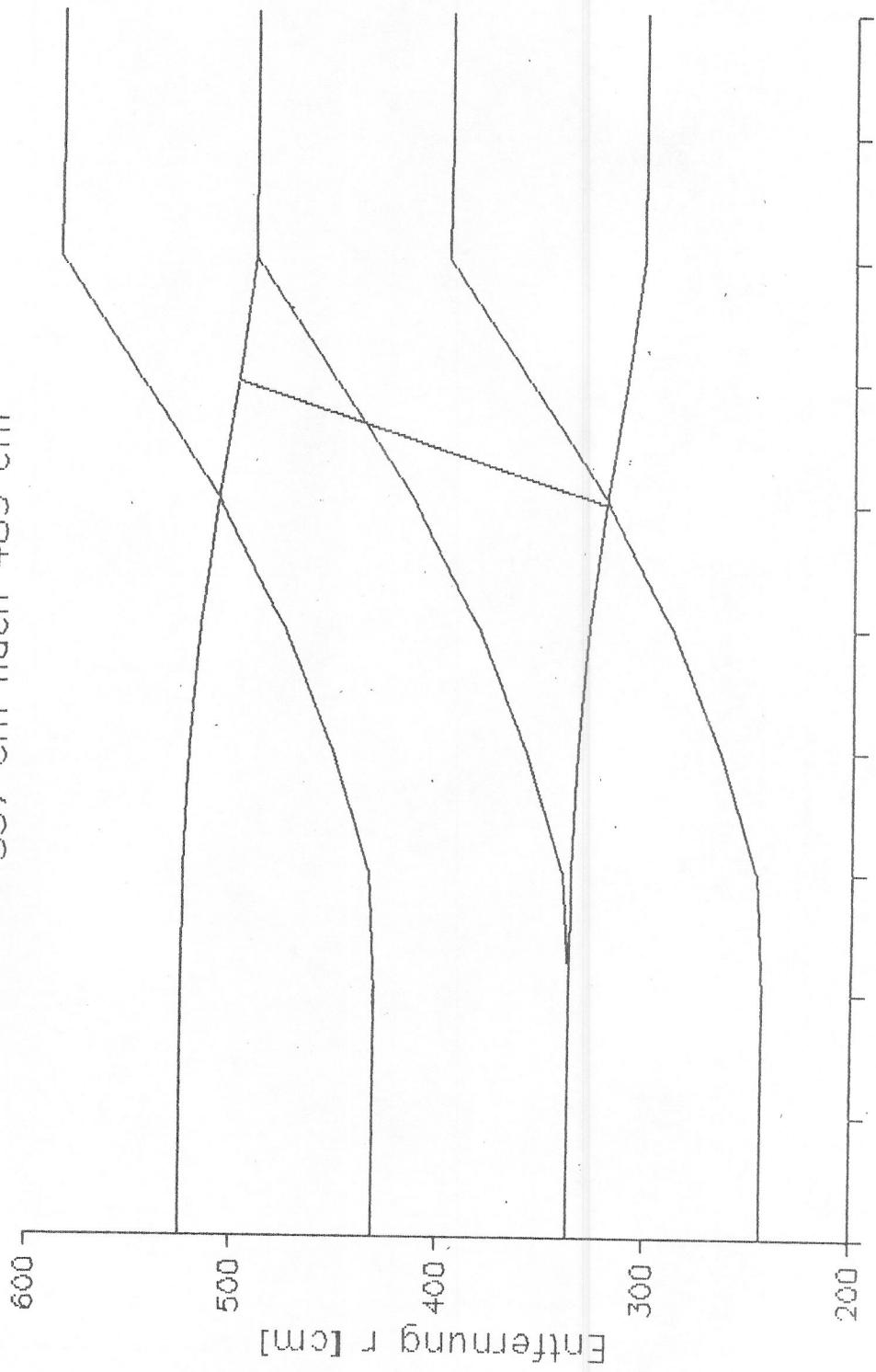
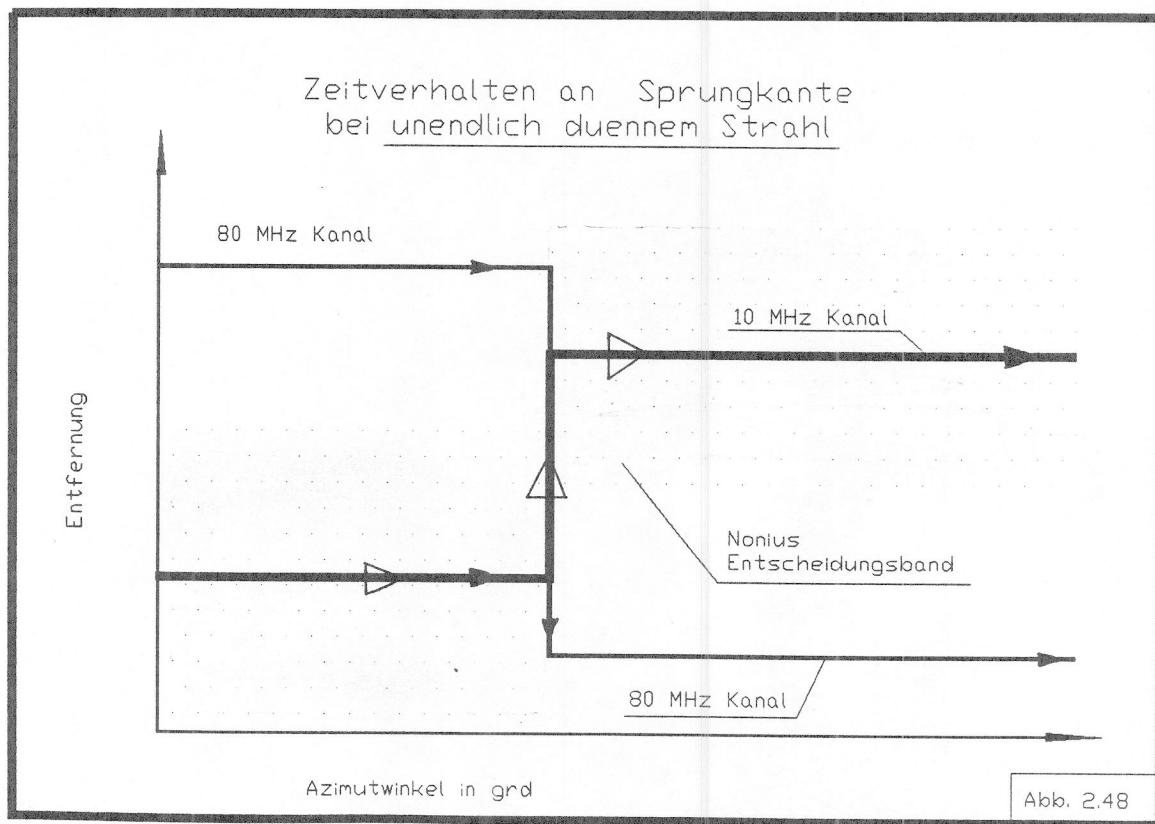


Abb. 24.7

Im Idealfall würde die Auswahl des 80 MHz Kanals nach dem in Abb. 2.48 dargestellten Schema erfolgen.



Allerdings verlaufen real und im Sensormodell Entfernungssprünge nicht rechteckig. Dies wäre nur möglich, wenn der Laserstrahl unendlich dünn wäre. Abb. 2.33 zeigt den realen Sachverhalt. An Kanten empfängt die LSR.E. Mischlicht. Jedes Mischlichtbündel repräsentiert einen anderen Entfernungsmeßwert, und somit einen anderen Meßzeiger.

Der resultierende Meßzeiger ergibt sich durch vektorielle Addition der beiden Einzelzeiger A und B.

Bewegt sich der Lichtstrahl nun über die Sprungkante, so nimmt der Betrag von Zeiger A ab, der Betrag von Zeiger B gleichzeitig zu.  
Dargestellt in der komplexen Ebene (s. Abb. 2.49) gilt für den resultierenden Phasenwinkel

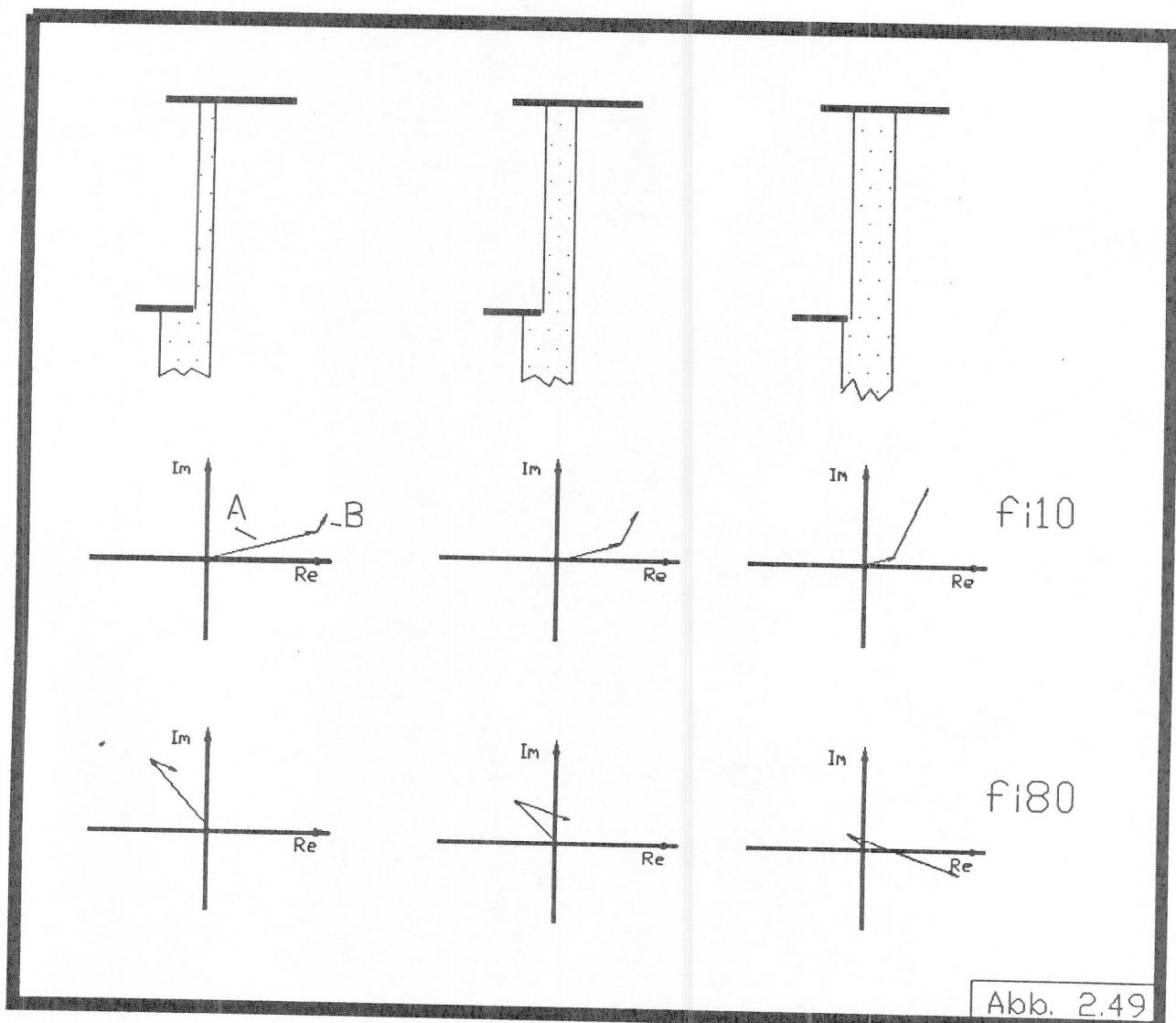


Abb. 2.49

A, B = Zeiger in der komplexen Ebene

A repräsentiert Zeiger zum Entfernungswert links der Sprungkante

B repräsentiert Zeiger zum Entfernungswert rechts der Sprungkante

T = Zeit in der Lichtstrahlbündel Kante überstreicht

Z = Resultierender Zeiger A + B

= Resultierender Winkel

$$A = (a+jb); \quad B = (c+jd);$$

$$Z = (1-T)*A + T*B; \quad T \text{ läuft von } 0..1; \\ = a + T(c - a) + jb + T(d - b);$$

$$\varphi_{res} = \operatorname{atan} \frac{b + T(d-b)}{a + T(c-a)} i$$

$$\varphi_{res}(T=0) = \operatorname{atan} \frac{b}{a};$$

$$\varphi_{res}(T=1) = \operatorname{atan} \frac{d}{c},$$

(Gleichung 2.13)

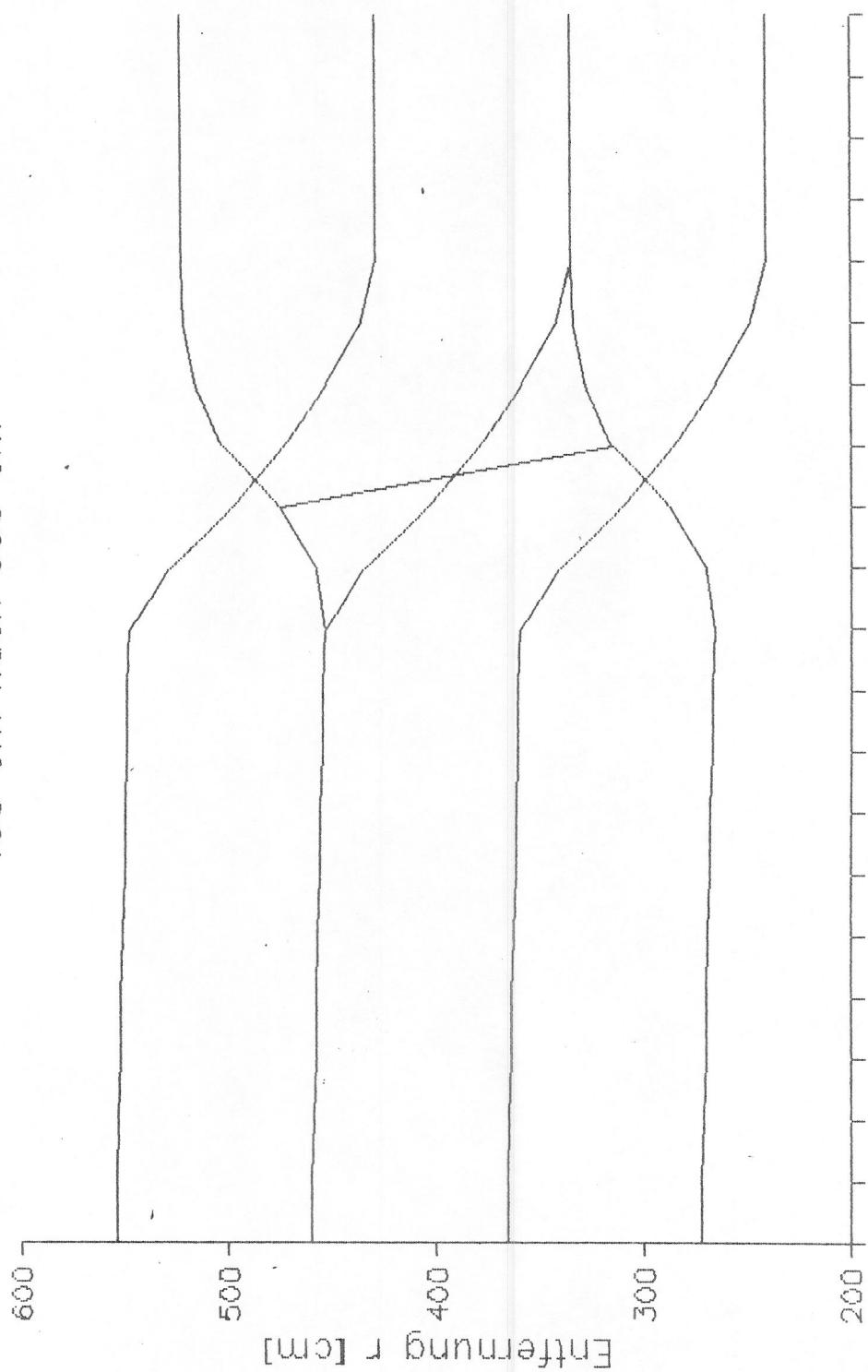
Die resultierenden Winkelverläufe der beiden Meßkanäle während der Zeit, in der der Laserstrahl eine Sprungkante überstreicht, sind also Abschnitte von Arkustangens-Kurven, abhängig vom Zeiger zur Zeit T = 0 und dem Zeiger zur Zeit T = 1.

Da der 10 MHz Zeiger bei gegebener Höhe der Sprungkante nur 1/8 des Winkels überstreicht, den in dieser Zeit der 80 MHz Zeiger zurücklegt, sind die Kurvenverläufe  $\varphi = f(T)$  unterschiedlich.

Es ist so möglich, daß sich beide Meßkanäle in ihrer Entfernungsmessung um mehr als 90 cm unterscheiden. Ein plötzliches Umschalten des Vieldeutigkeitsbereich um 1,88 m tritt auf. Abb. 2.50 zeigt nochmals den modulierten Sprung von 445 auf 335 cm, sowie den Entfernungsverlauf der Einzelkanäle. Bei einer Sprunghöhe von weniger als 94 cm tritt kein Noniussprung auf (s. Abb. 2.51).

c) Meßtechnische Bestätigung

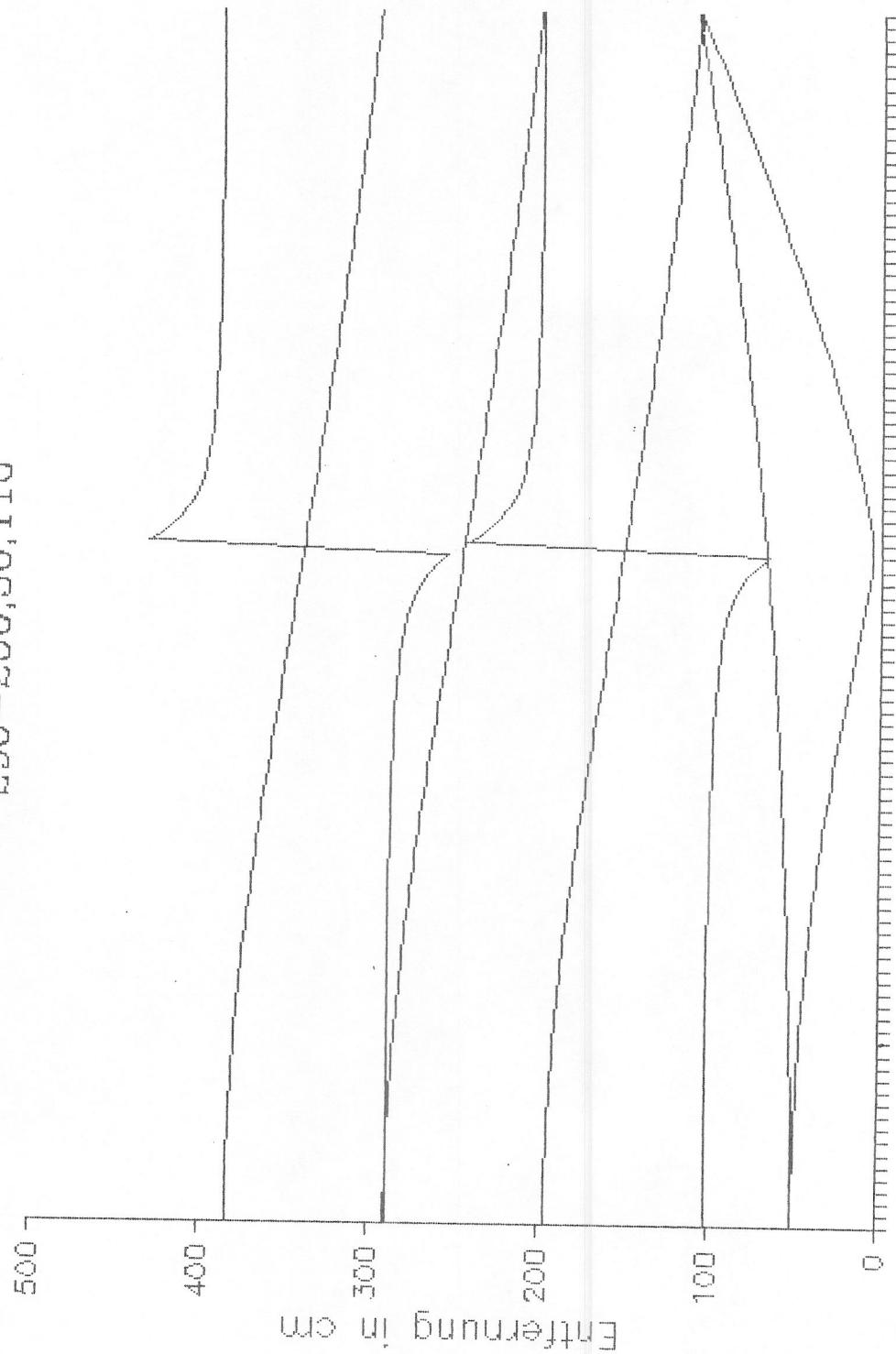
Verhalten an Sprungkante  
455 cm nach 335 cm



Azimutwinkel [grd]

Abb, 250

Zeitverhalten an Kante  
290-200,50,110

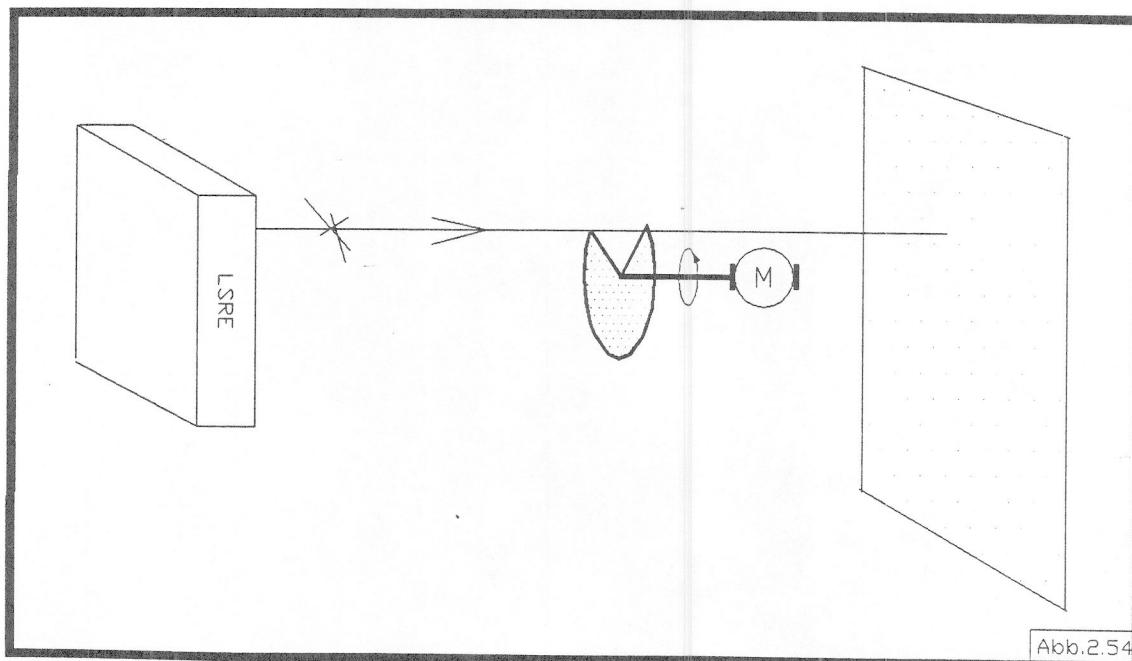


Zeit  $t = \text{Azimutwinkel}$

Abb. 2.51

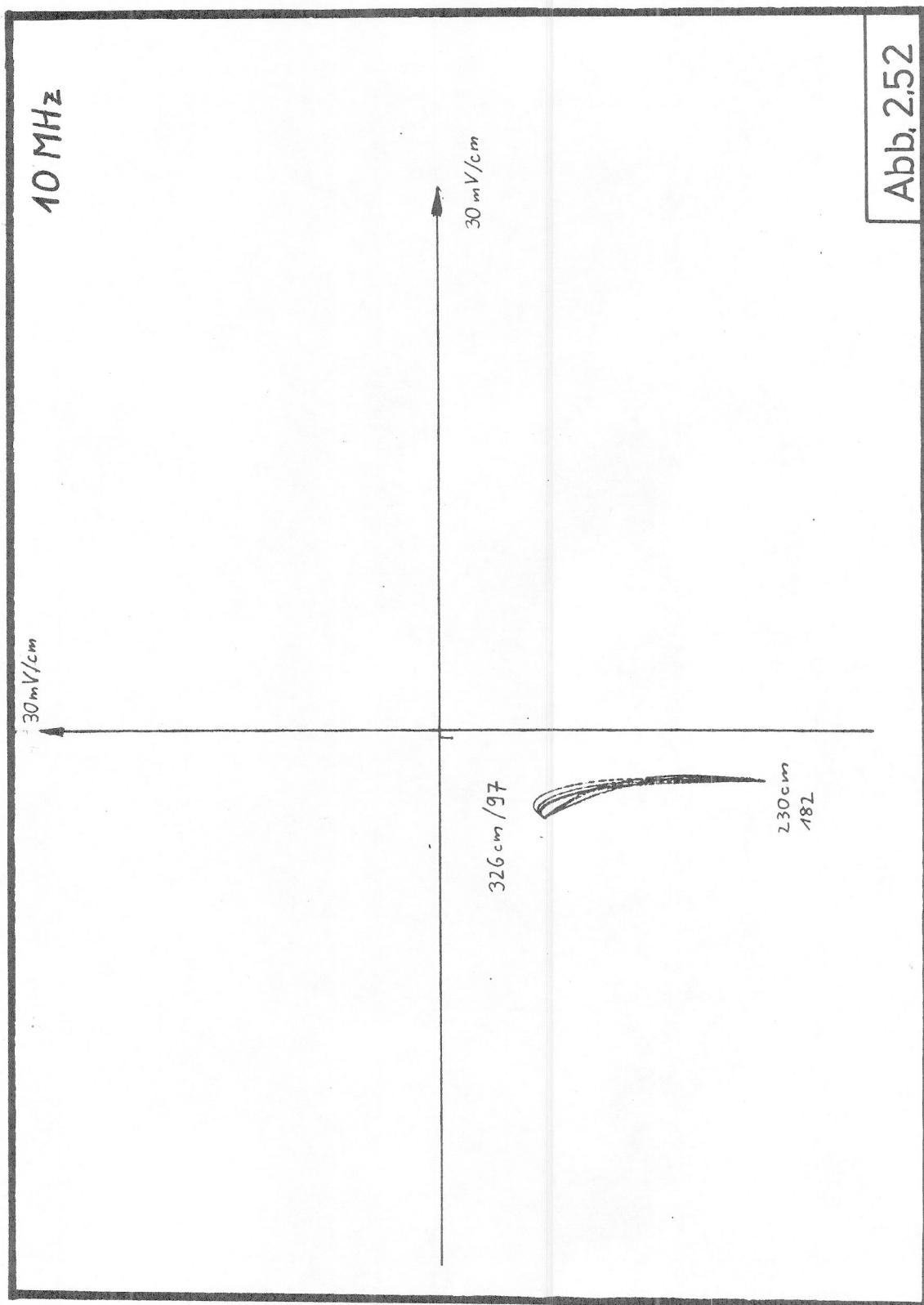
Nachdem es mit Hilfe des Sensormodells möglich ist, den Kanteneffekt und seine Ursachen nachzuweisen, steht eine reale meßtechnische Überprüfung obiger, auf die Simulation gestützte These noch aus. Abb. 2.52 und Abb. 2.53 zeigen die Meßzeiger beider Kanäle und deren Verlauf beim Überstreichen der Sprungkante.

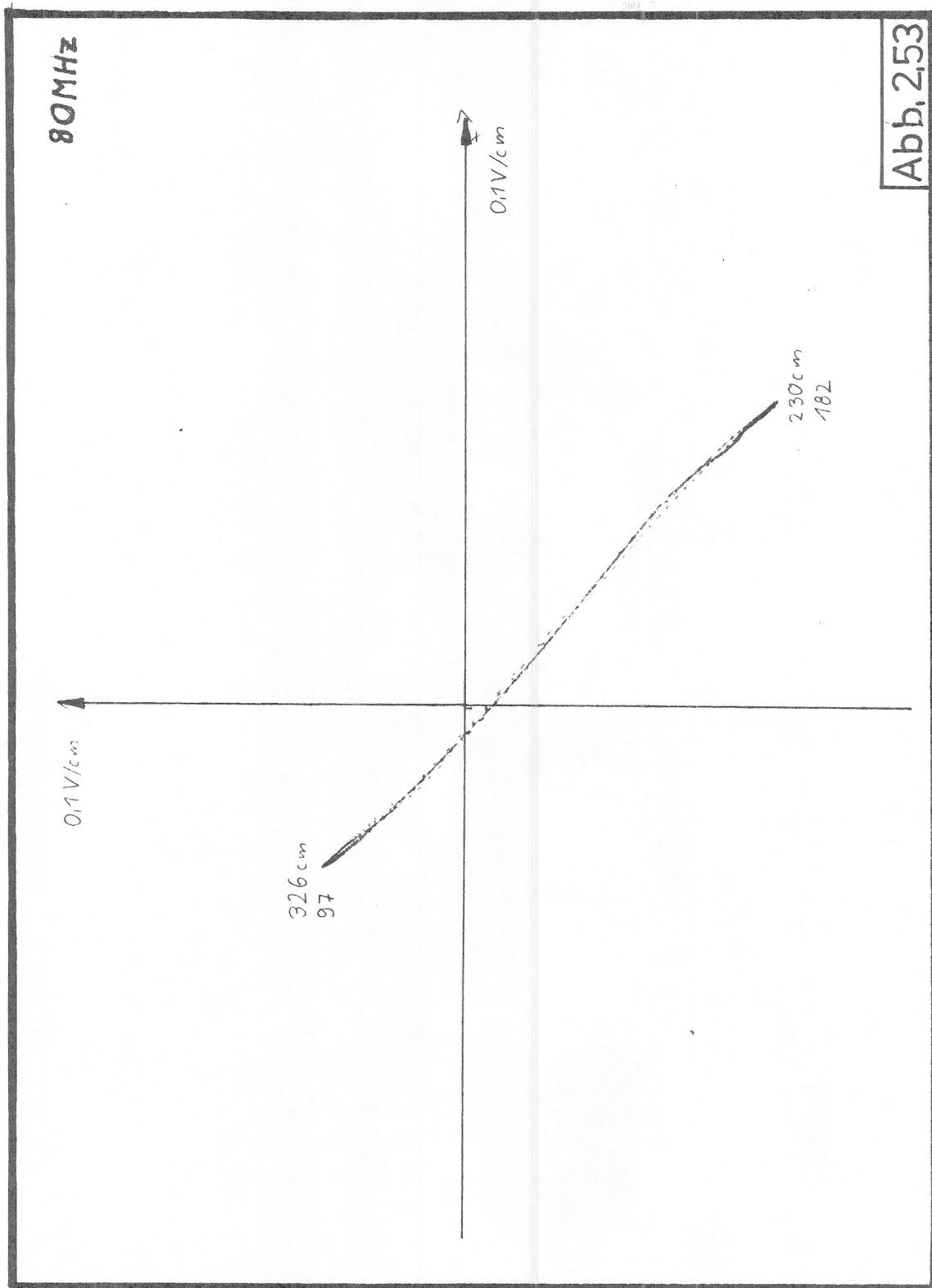
Zur Messung dieser Zeigerverläufe wird folgende Meßanordnung gewählt (s. Abb. 2.54).



Um alle Einflüsse der Ablenkeinheit auszuschalten, wird die LSR.E. im statischen Betrieb eingesetzt.

Zur periodischen Messung und zur Darstellung am Oszilloskop mißt der Laserstrahl auf eine rotierende Schlitzblende, die abwechselnd den Meßstrahl auf eine als Meßobjekt dienende Wand, und auf die Schlitzblende selbst, fallen läßt.





Eingehende Untersuchungen mit simulierten und realen Testszenen zeigen, daß der Kanteneffekt nur dann auftritt, wenn 2 Randbedingungen erfüllt sind:

- a) Die Abtastung des Meßsignals in der Auswertelektronik erfolgt genau zum kritischen Zeitpunkt, d.h. der Laserstrahl wird gerade zum Abtastzeitpunkt durch eine Sprungkante gespalten.
- b) Die Sprunghöhe ist größer als 94 cm.

## 2.5. MODELL UND MEßDATEN GESTEUERTE FILTERUNG

Mit Hilfe des Sensormodells ist es nun einerseits möglich, offline erhaltene, detaillierte Kenntnisse zu nutzen, um durch gezielte Modifikationen an der Hardware der LSR.E. unerwünschte Meßeffekte auszuschalten.

Andererseits kann man das Modell einsetzen, um meßbegleitend Fehler von tatsächlichen Meßwerten zu unterscheiden und diese zu eliminieren.

### 2.5.1. Mögliche Hardwarelösungen

Ich beschränke mich hier auf den signifikant "gefährlichen" Meßfehler der LSR.E., nämlich das Überschwingen der Entfernungsmeßwerte an Kanten (s. Kap 2.4.).

Die oben durchgeführte Analyse anhand des Sensormodells zeigt, daß dieser Fehler nunmehr dann auftritt, wenn gilt:

$$r_{diff} = (|r_n - r_{so}|) \geq 80 \text{ cm};$$

Das heißt, wenn der Wert  $r_{10}$  um mehr als 80 cm vom entsprechenden Wert  $r_{80}$  abweicht.

Eine Analyse von Meßkurve Abb. 2.13, die die Linearität und Meßgenauigkeit des 10 MHz Kanals zeigt, belegt, daß der 10 MHz Entfernungswert nie mehr als 30 cm von der tatsächlichen Entfernung abweicht.

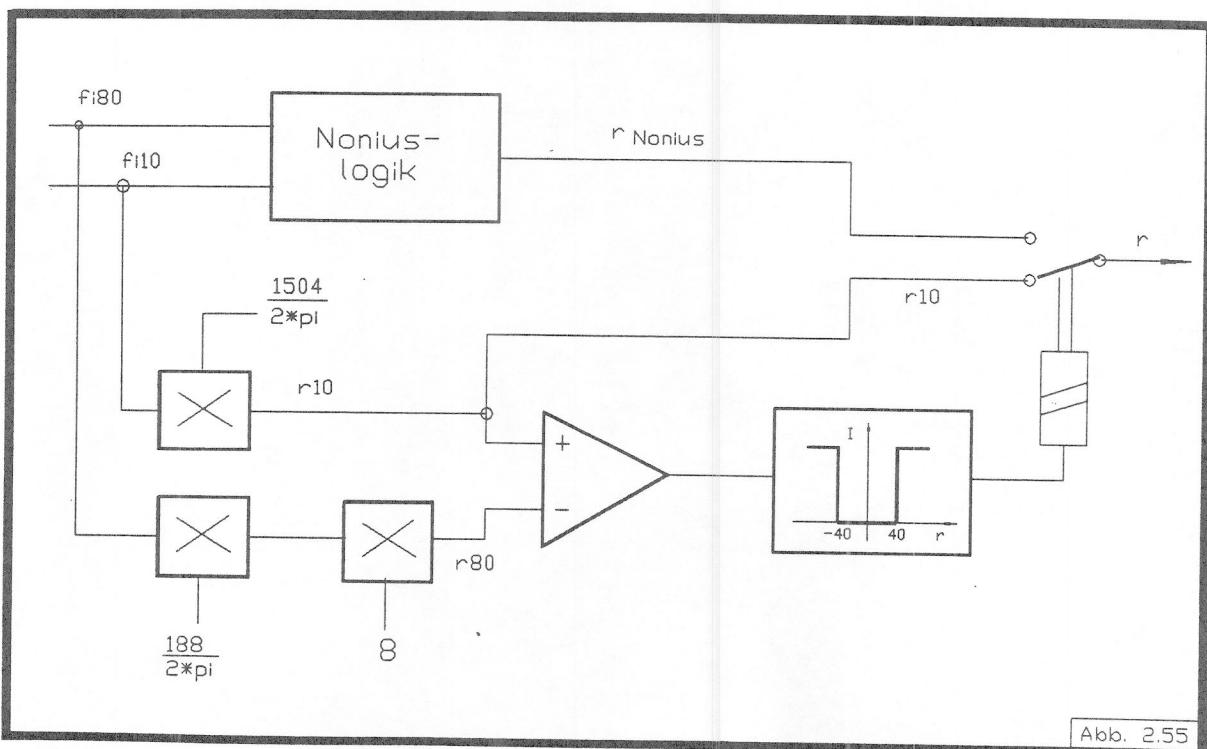
Der 80 MHz Kanal mißt, wenn auch mehrdeutig, genauer als 5 cm.

Es ist nun leicht möglich, in der Auswertelektronik  $r_{\text{diff}}$  aus Gleichung 2.13 zu berechnen.

Die Genauigkeit des 10 MHz Kanals und des 80 MHz Kanal zugrundelegend, darf  $r_{\text{diff}}$  nicht größer als 40 cm werden.

Ist dies doch der Fall, so liegt eine Sprungkante vor, der nächste Meßwert wird ein "Überschwinger" sein und ist zu verwerfen, oder durch den etwas ungenauerer 10 MHz-Messwert zu ersetzen.

Abb. 2.55 zeigt nochmals schematisch das Vorgehen anhand der beiden Meßkanäle.



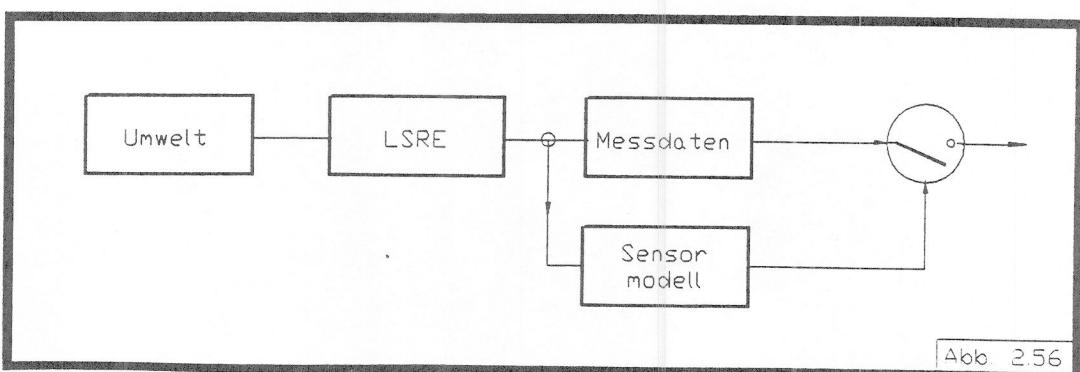
Ausblickend sei angemerkt, daß diese Methode der Sprungkantendetektion (Ausgabe eines zusätzlichen Kantensignals durch die Noniuslogik) ein mächtiges Hilfsmittel zur weiteren Bildverarbeitung darstellt.

## 2.5.2. Softwarelösung

Detektiert man mit Hilfe des Sensormodells fehlerträchtige Umweltszenarien, die Meßfehler hervorrufen, so ist auch eine softwaremäßige Filterung der LSR.E. Daten realisierbar. Als Beispiel dient auch hier wieder die Elimination der an Kanten auftretenden Ausreißer.

### 2.5.2.1. Prinzip des Filters

Abb.2.56 zeigt die Grundstruktur des hier verwendeten Filterkonzeptes. Ähnlich dem in der Systemtheorie üblichen Beobachters, läuft parallel zur LSR.E. Messung eine modifizierte Version des Sensormodells.

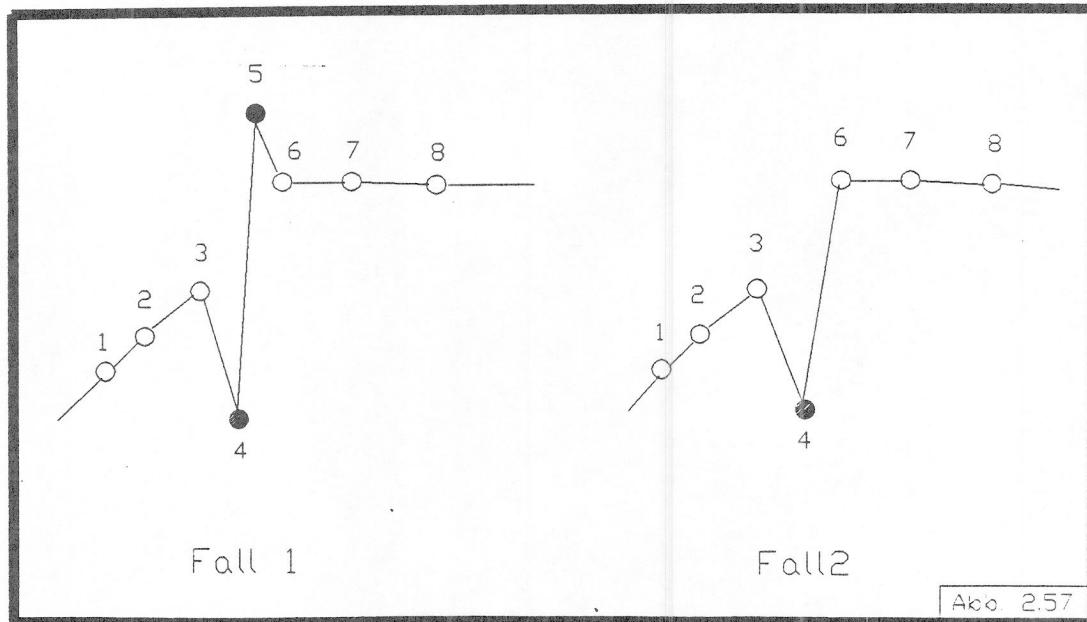


Aufgabe dieses Modells ist es, fehlerproduzierende Kantenkonstellationen zu identifizieren, um im Falle eines Überschwingens (vereinzelter Meßpunkt) den letzten gültigen Meßpunkt nochmals auszugeben.

Im Einzelnen durchläuft die modellgestützte Rohdatenverarbeitung folgende Schritte:

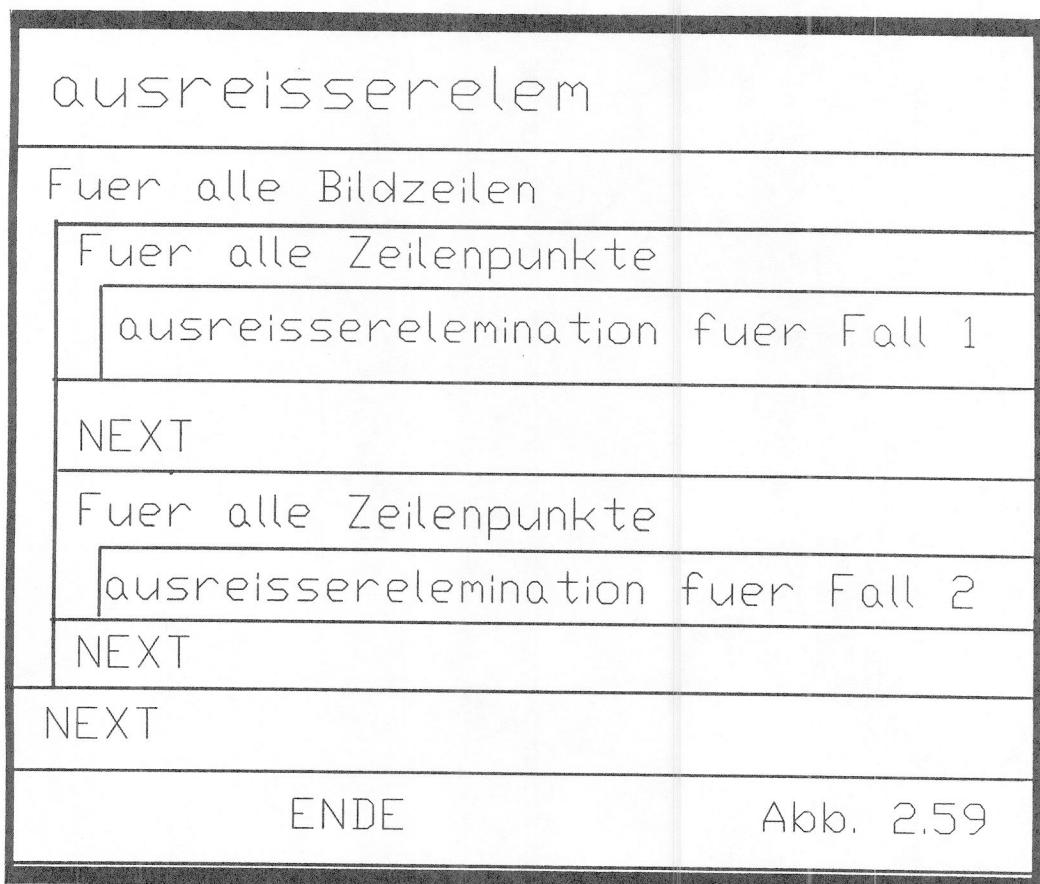
- Entfernungsabhängige Kantendetektion
- Bestimmung der Kantenkonstellationen
- Prädiktion des möglichen Meßfehlers durch Simulation
- Überprüfung, ob Fehler aufgetreten ist
- Ersetzen vereinzelter Ausreißer durch letzten gültigen Meßwert  
(Do Nut Filter)

Durch dieses, im ersten Augenblick aufwendig erscheinende Verfahren, unterscheidet man für das Fahrzeug gefährliche Hindernisse wie Rohre, dünne Betonträger u.ä., von Ausreißern an Kanten; für den Betrieb der LSR.E. auf einem Fahrzeug in industrieller Umgebung eine unabdingbare Randbedingung.



### 2.5.2.2. Programmtechnische Realisierung

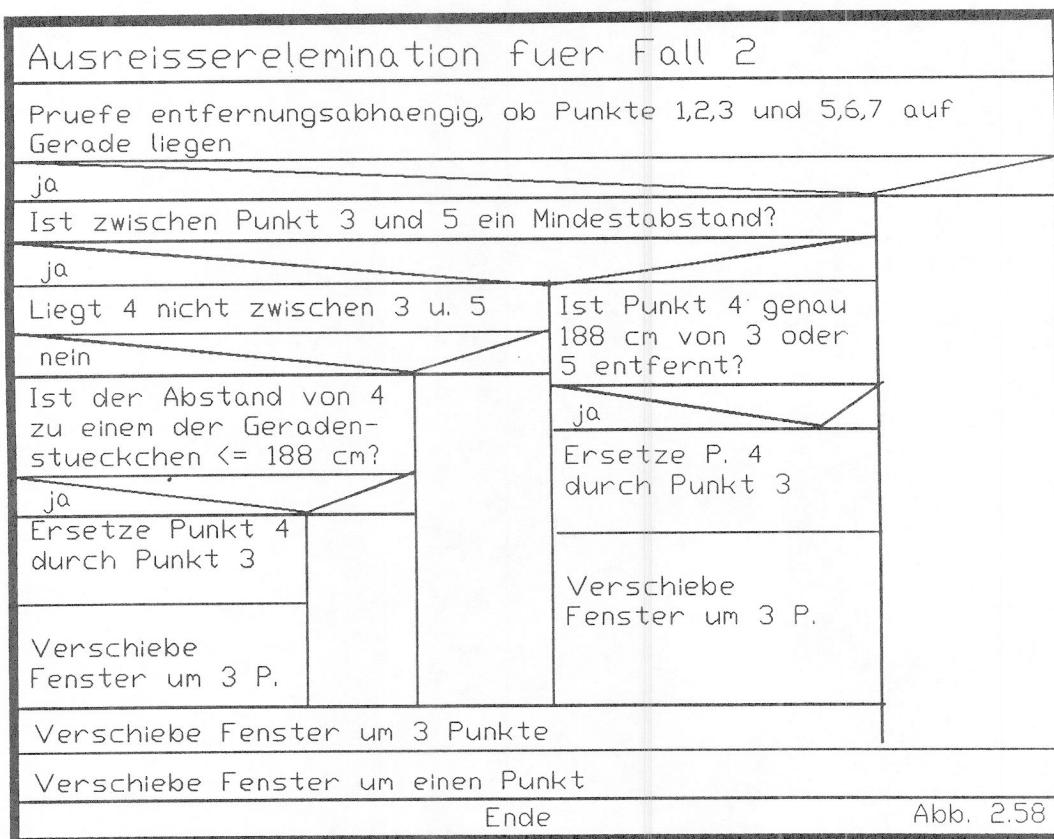
Abb.2.58 zeigt das grobe Struktogramm der realisierten Softwareroutinen. Das Programm gliedert sich in 2 Blöcke entsprechend den beiden möglichen Fehlersituationen (s. Abb.2.57).



Abhängig von der Abtastung, kann ein Überschwingen in zwei oder auch nur in eine Richtung auftreten.

Werte, die auf halber Höhe zwischen den Punkten 3 und 6 liegen, werden, da für das Fahrzeug nicht gefährlich, zwar detektiert, aber keiner Fehlerbehandlung unterworfen.

Im einzelnen zum Programmablauf:



Block 1 reduziert den in Abb.2.57 zu sehenden Fall 1 zu Fall 2, indem nur einer der beiden Ausreißer eliminiert wird. Dies geschieht, indem nur die Punkte 1, 2, 3, 4, 6, 7, 8 in die Bewertung eingehen. Ansonsten ist Block 1 analog zu Block 2 aufgebaut. Im einzelnen werden die in Block 2 (s. Abb. 2.59) gezeigten Routinen ausgeführt.

Abb.2.60 zeigt die Konfiguration an Kanten und Ausreißerpunkte, die auftreten müssen, um eine Elimination dieses Ausreißens durchzuführen. Alle anderen vereinzelten Meßpunkte werden nicht eliminiert.

In Anhang 2 ist ein dokumentierter Programmtext der obigen Routinen zu finden.

#### 2.5.2.3. Implementation

Obiger Algorithmus, genannt "ausreiberelem", wurde in 2 Versionen realisiert.

Die offline Version des Programmes erlaubt es, abgelegte Bilddatenfiles zu filtern. Sie wurde in VAX-Pascal geschrieben und läuft unter VMS auf lehrstuhleigenen VAX-Rechnern.

Eine weitere Version ist direkt auf dem eingangsdatenvorverarbeitenden Rechner vom Typ LSI 11 in Micro-Power-Pascal einer echtzeitfähigen Pascalversion implementiert.

Durch ein von Herrn Fröhlich (siehe /5/) geschaffenes Graphik-MonitorSystem können einzelne Meßzeilen mit und ohne Filterung betrachtet und gleichzeitig weiterverarbeitet werden.

#### 2.5.3. Rohdatenverarbeitung im Einsatz

Abb. 2.46, Abb. 2.61 und Abb. 2.62 zeigen die gleiche Szene. Abb. 2.46 zeigt eine gemessene Rohdatenzeile, der Kanteneffekt ist deutlich zu erkennen.

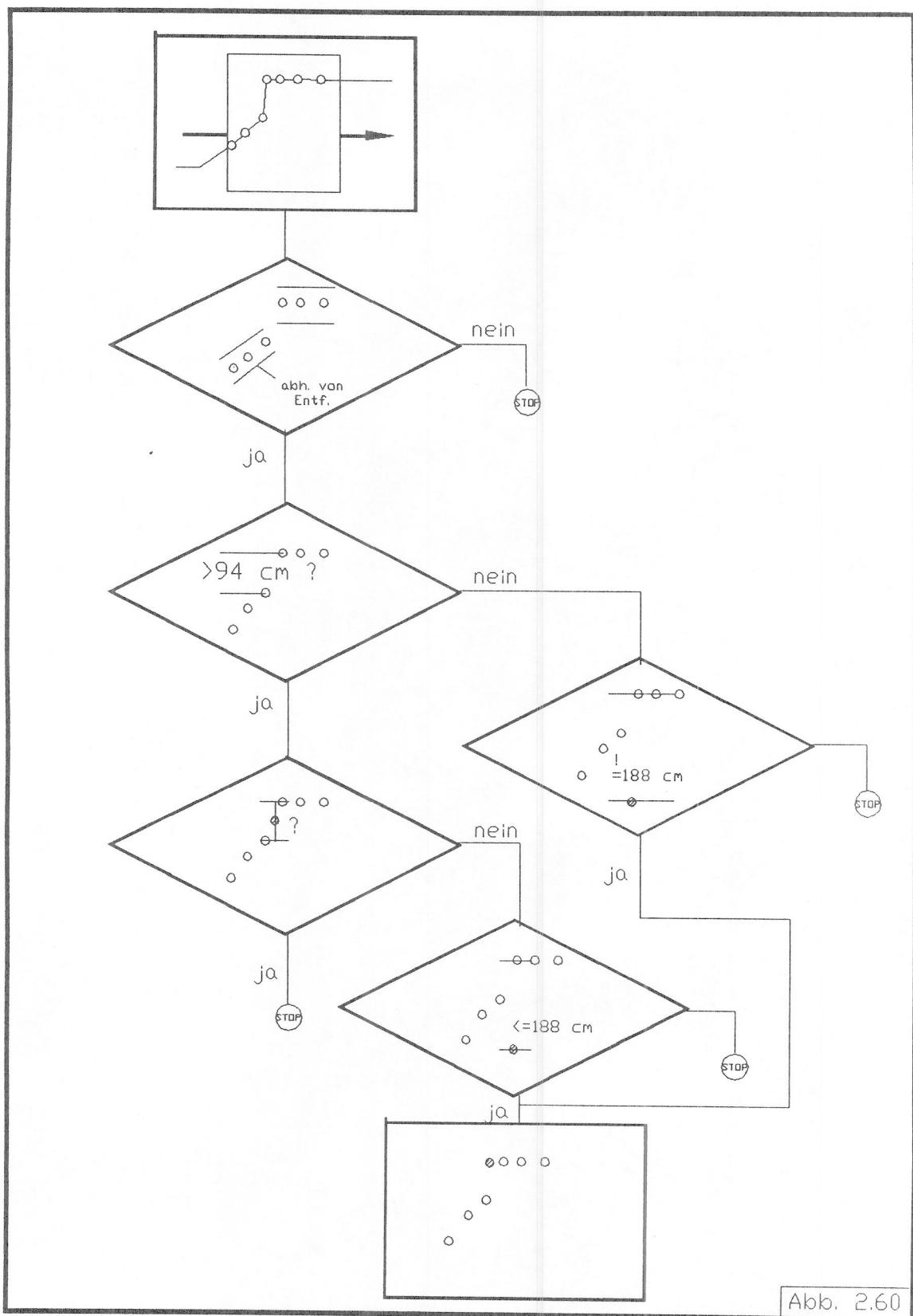


Abb. 2.60

Abb. 2.61 zeigt die gleiche Szene, simuliert durch das Sensormodell; sämtliche Ausreißer finden sich hier wieder; der Kanteneffekt wird als systematischer Fehler bestätigt.

Abb. 2.62 zeigt schließlich die Daten aus Abbildung 2.46, gefiltert mit Hilfe obigen Programms. Sämtliche kritischen Kantenstellungen sind erkannt, die systematischen Fehlermeßwerte werden durch ihre Nachbarmeßwerte ersetzt.

Als weiteres Beispiel dient die Szene in Abb. 2.26; sie enthält nicht nur Sprungkanten, sondern auch ein schmales vereinzeltes Objekt A (Rohr).

Abb. 2.27 zeigt die aufgenommene Rohszene als karthesischer Schnitt;

Abb. 2.64 dieselbe Meßzeile nach Durchlaufen der Filterstufe.

Deutlich ist zu erkennen, daß die Kantenüberschwinger beseitigt sind, das vereinzelte Objekt "Rohr", vom Filteralgorithmus aber eindeutig als ein Objekt und Nicht-Meßfehler erkannt, korrekt abgebildet wird.

Simulierte Szene  
Kai 41.Zeile

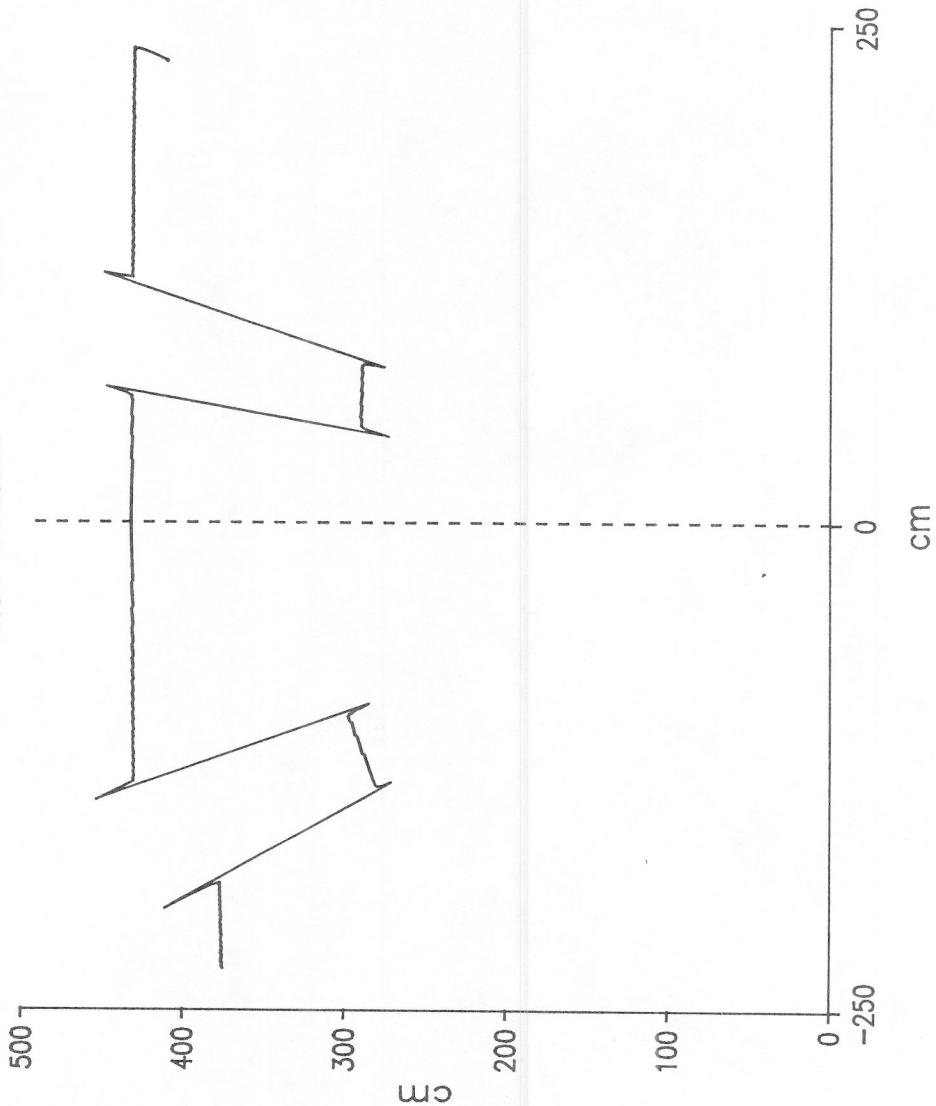
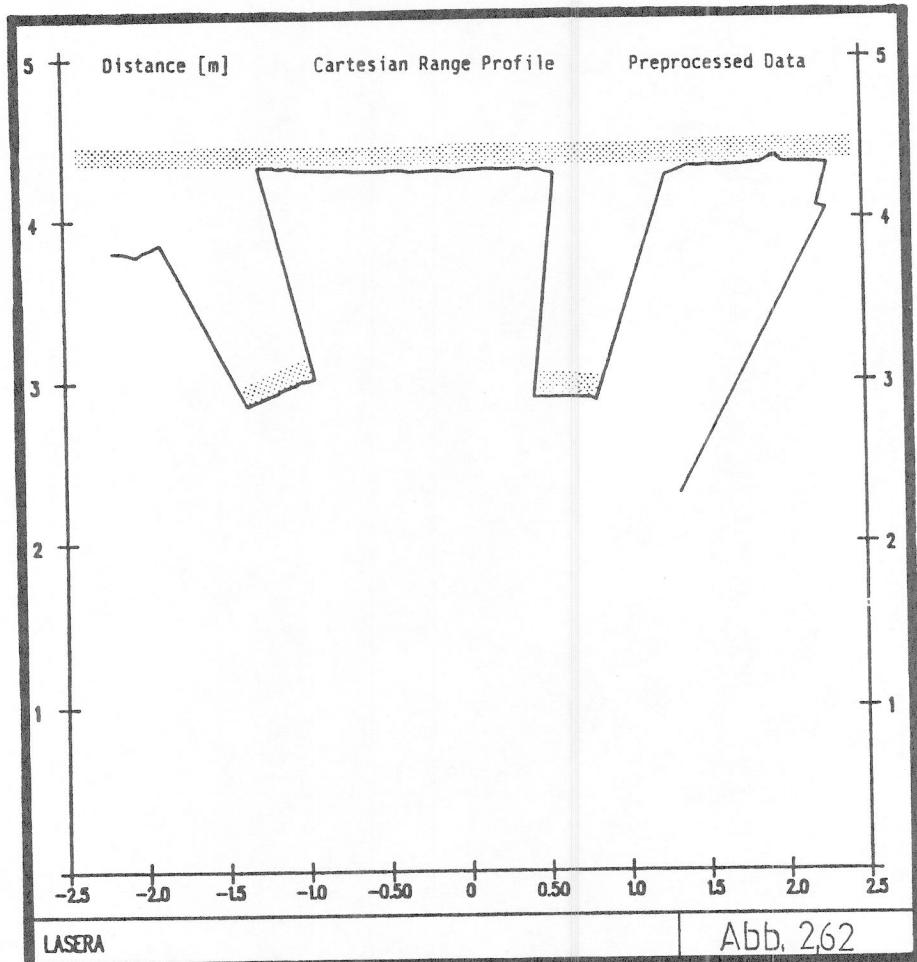
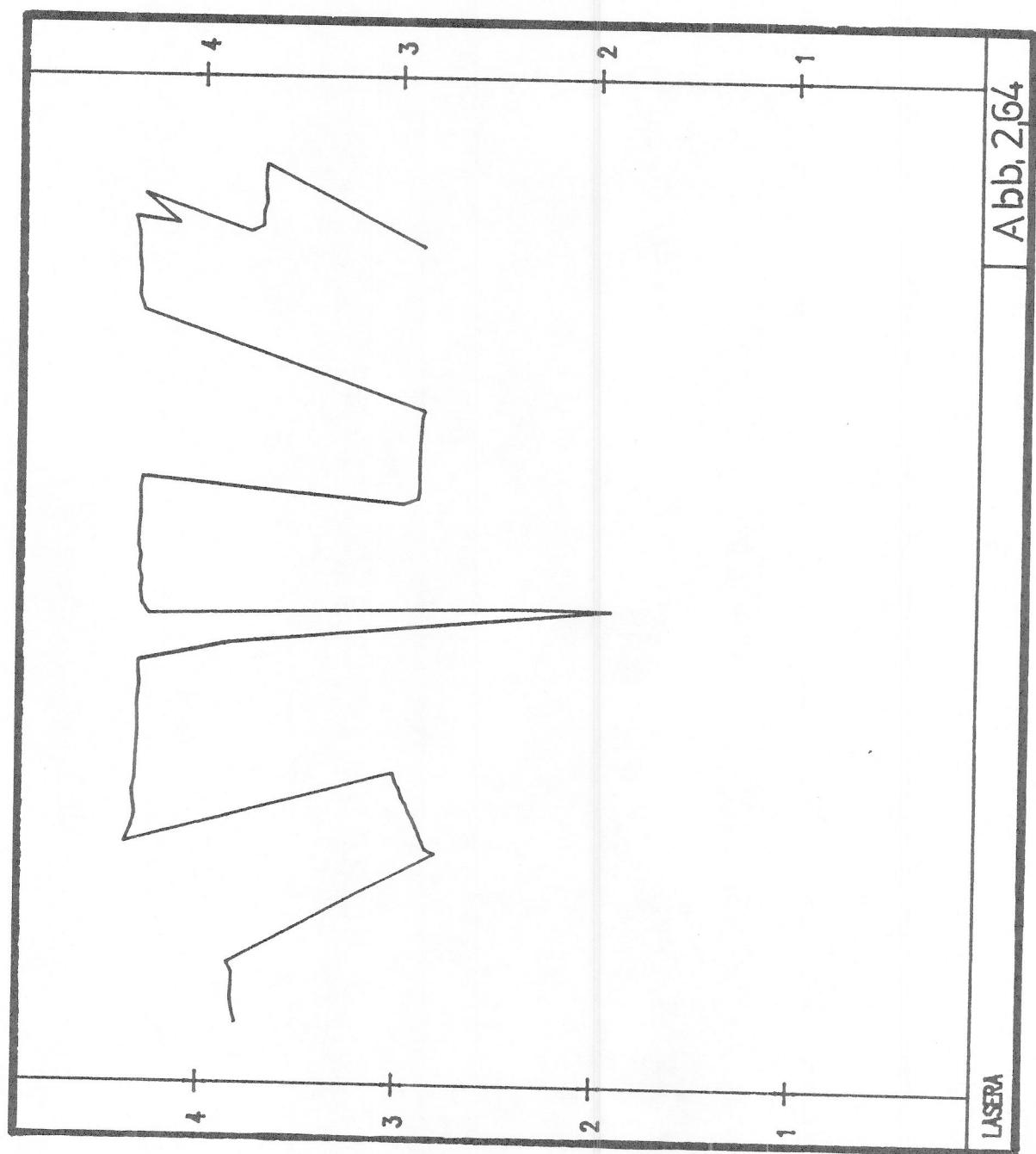


Abb. 2.61





### 3. EIN MULTISENSOR - ENTWICKLUNGSRAHMEN FÜR DAS AUTONOM MOBILE FAHRZEUG MAKROBE

Bevor man beginnt, weitere und fortgeschrittene Bildverarbeitungsverfahren für die LSR.E. anzuwenden, ist es notwendig, die Anforderungen an solche Verfahren und die Möglichkeiten ihrer Anbindung an das Gesamtsystem MAKROBE zu untersuchen.

#### 3.1. DIE MAKROBE UND IHRE SENSOREN

Das autonome Fahrzeug MAKROBE hat den Auftrag, in einer industriellen Umgebung autonom vom Startpunkt A zum Zielpunkt B zu gelangen, sich hierzu aufgrund eines groben Umweltplanes selbst den besten Weg zu suchen, und dabei Hindernissen in der Fahrbahn selbstständig auszuweichen, beziehungsweise verstellte Fahrtrassen zu umfahren.

Hierzu befähigt sie ein hierarchisches Planungs- und Steuerungssystem (siehe 1.1.1.), sowie ein System unterschiedlicher Sensoren zur Umwelterfassung.

Das Fahrzeug und sein Steuersystem benötigt die Sensordaten:

- um sich sicher in seiner Umwelt bewegen zu können
- um für künftige Aktionen seine Umwelt zu erkunden

### 3.1.1. Aufgaben und Fähigkeiten der einzelnen Sensoren

Neben der LSR.E. werden noch folgende Sensoren auf der MAKROBE eingesetzt:

- ein Mikrowellenradar, zur Entfernungsmessung auf größere Entfernung, zur Bildgewinnung, zum Verfolgen bewegter Objekte, sowie zu deren Geschwindigkeitsmessung.
- ein Radsensor, zur Messung von Geschwindigkeit und relativer Bewegungsrichtung.
- ein Kollisionsschutz-Sensor für den Nahbereich: er stoppt die MAKROBE, bevor es zu einer gefährlichen Kollision kommt.
- ein Stereo Bildverarbeitungssystem: hier kommen 2 SW CCD-Kameras zum Einsatz.
- Ultraschall-Sensoren

Diese Sensoren, zum Teil bildgebend, zum Teil nicht, unterscheiden sich zwar stark in ihren meßtechnischen Eigenschaften, bilden aber andererseits das gleiche Meßobjekt ab, nämlich die Umwelt der MAKROBE. Abb. 3.1 zeigt die verschiedenen Eigenschaften, Vorzüge und Nachteile der Sensoren tabellarisch. Aus der Tabelle ist leicht zu erkennen, daß sich die Eigenschaften der Sensoren teilweise ergänzen, teilweise überlappen, sei es der Sichtbereich, die Meßgeschwindigkeit oder die Meßgenauigkeit.

Übersichtstabelle MAKROBE Sensoren (realisiert u. geplant)

	LSR.E.	RADAR	US	Radsensor	Kollisionsschutz	Stereo-BV
Reichweite	1-10m	2-30m	0,3-8m		0,3m	1-20m
Auflösung	3 cm	20 cm	0,3-3m			0,3-5m
Bildpunkte	3200	300	10			512*512
Reaktionszeit Incl. Verarb.	0,3-120s	1-100s	3s	sofort	sofort	5-500 s
Geschw.Messung	indir.	direkt	ungenau	direkt		ungenau

Abb. 31

Betrachtet man in Abb 1.3. nochmals das Entscheidungssystem, so zeigen sich gewisse Analogien. Genau wie im Sensorbereich unterscheiden sich hier der Zeithorizont der Verarbeitungsschritte, aber auch Abstraktions- und Präzisionsgrad der benötigten Eingangsinformationen.

Ziel jeder weiteren Sensordatenverarbeitung muß es also sein, jeder Entscheidungsebene der MAKROBE genau die, und nur die Daten, die sie braucht, so genau und sicher als möglich zur Verfügung zu stellen. (Selektion)

Andererseits sollen redundante Sensoren, sowie Vorwissen, datenbesichernd zusammengefasst werden. (Integration)

### 3.2. DATENVERARBEITUNG IM MULTISENSORSYSTEM DER MAKROBE, RANDBEDINGUNGEN UND AUFGABENSTELLUNG

Ausgehend vom Gesamtsystem MAKROBE, der Fahrzeugumwelt und den messtechnischen Eigenschaften der Sensoren, ergeben sich folgende Forderungen an die Datenverarbeitung in einem Multisensorsystem:

- a) Das System soll Daten verschiedener Sensoren unter Anwendung von Sensorredundanz und Ergänzung verarbeiten können.
- b) Vorwissen über die Umwelt und die darin geltenden Gesetze, das Fahrzeug und die Eigenschaften der Sensoren, sollen Daten abstützen und ergänzen.
- c) Das System soll die Menge der Daten reduzieren.
- d) Das MSS soll ein sicheres, genaues und vollständiges Bild der Umwelt liefern.
- e) Expertenwissen, sei es von der MAKROBE selbst oder vom Bediener, soll geändert und ergänzt werden können, unabhängig vom Programmcode der Verarbeitungsmodule.
- f) Da sich die MAKROBE in einer Umwelt bewegt, in der wiederum bewegliche Hindernisse auftreten, muß es möglich sein, alle Geometrie auf ein ortsfestes Koordinatensystem beziehen zu können.
- g) Für die verschiedenen logischen Verarbeitungsebenen der MAKROBE müssen Umweltdaten in verschiedenem Abstraktionsgrad und verschieden schnell zur Verfügung stehen.
- h) Je nach Fahrauftrag und Umweltsituation müssen die Sensoren konfigurierbar und steuerbar sein.
- i) Die zeitliche Konsistenz der Daten muß gewährleistet sein (Zeitstempel).

- j) Zu allen Sensordaten ist eine offene Schnittstelle zu schaffen, die sich in das Kommunikationsnetz der MAKROBE einfügt.
- k) Eine Möglichkeit zur graphischen Darstellung der Daten bildgebender Sensoren ist zu schaffen.

### 3.3. STAND DER FORSCHUNG AUF DEM GEBIET DER MULTISENSORSYSTEME

Multisensorsysteme (MSS) wurden auf verschiedenen Anwendungsgebieten konzipiert bzw. entwickelt. Einsatzgebiet sind hierbei einerseits autonome Fahrzeuge ähnlich der MAKROBE, andererseits Robotermontagestationen.

#### 3.3.1 Analyse der Literatur

Anhand folgender 8 Autoren lassen sich die weltweit verfolgten Grundkonzepte zum Multisensoraspekt aufzeigen.

##### 1. Henderson

Henderson (siehe /6/, /7/, /8/) arbeitet auf dem Gebiet der Roboter-Montage.

Ausgehend von verschiedenen physikalischen Sensoren (Kamera, Strip-Triangulationslaser) und deren Ausgangsdaten, führt er den Begriff des logischen Sensors ein. Der logische Sensor (im weiteren logS) ist bei Henderson eine Ansammlung von Routinen, die einen speziellen Ausgangsvektor (Kreis (Mittelpunkt, Radius)) errechnen. In /7/ zeigt Henderson die Kopplung der logS über ein sogenanntes Multisensor-Kernel System. Jeder logS besitzt einen Komandointerpreter, der es erlaubt, den Sensor durch Makrobefehle zu aktivieren und anzuweisen, welche Daten weiterzugeben sind. Die Sensoren werden hierarchisch

verschaltet, untergeordnete logS sind so nicht mehr direkt erreichbar. In /8/ wird zusätzlich ein Werkzeug beschrieben, das es erlaubt, aus CAD/CAM Daten die Sensorkonfiguration abzuleiten. Die logS detektieren dabei entweder spezielle CAD-Primitive oder die Konstellation solcher Primitive zueinander.

## 2. Kamat

Auch Kamat arbeitet auf dem Gebiet der Montageplanung. Folgende 3 Probleme sind nach /9/ bei MSS zu lösen:

- Kombination und Integration der physikalischen Sensoren
- Kombination und Integration der Sensoralgorithmen
- Steuerung der Sensoren

Die Verknüpfung der unterschiedlichen Sensorergebnisse geschieht bei Kamat über PROLOG Aussagen und einer Wertefunktion. Jeder Sensorsaussage wird ein Wert (0....1) zugeordnet. Aussagen, die von mehr als einem Sensor gestützt werden, erhalten entsprechend höhere Werte. Alle mögliche Objekte werden vorher in einer Beschreibung als zusammengesetzte Primitive abgespeichert. Diese Beschreibung erfolgt getrennt für alle Sensoren (Kamera, Ultraschall). Das MSS ist in PROLOG gestützt auf eine Datenbank realisiert.

## 3. Almand

Almand /10/, der am Projekt des Autonomous Land Vehicle (ALV) beteiligt ist, geht von einer zweidimensionalen Anordnung der Sensoren, bzw. der Sensordatenverarbeitung aus. Während die eine Dimension zwischen verschiedenen Sensorsaufgaben (Sensorspezialisten mit speziellen Aufgaben) unterscheidet, repräsentiert die andere Achse den Abstraktionsgrad der Aussagen. Zwischen den Spezialisten findet auf jeweils gleicher Abstraktionsstufe eine Verknüpfung der Daten über "Fuzzy"-Mechanismen statt.

#### 4. Meier

Einen anderen Weg beschreitet Meier in /11/ .

Aufgabe ist in seinem Forschungsbereich die Verfolgung einer Rakete mit Kamera, Radar, sowie numerischen Informationen. Das Kamerabild trennt er zu zwei logS:

- Stand der Sonne
- Kantenbild Position der Rakete relativ zur Sonne

Meier konfiguriert vor der Messung ein kausal - Netzwerk, in dem die möglichen semantischen Zusammenhänge aufgeknüpft sind. Die logS füllen das Netz mit Daten auf. Das Netz besitzt Ausgänge zur Sensorkonfiguration und wird nach 2 Gesichtspunkten ausgewertet:

- räumliche Konfiguration
- Information über die Zeit (Spatio-Temporale-Beschreibung)

#### 5. Tujnmann

Hier werden Multisensoraspekte an mobilen Robotern erprobt. Folgende logS hat Tujnmann /12/ implementiert:

- Zeit
- Geschwindigkeit
- Umgebungsplan
- Kollisionsmessung
- relative Position
- Vergleich Umgebungsplan - globale Karte
- Free Way Sensor

Tujnmann fordert die zusätzliche Repräsentation und Verarbeitung eines Genauigkeitsparameters.

Jeder logS ist einzeln implementiert. Ablauffolge und Einsatz werden situationsabhängig gesteuert. Jedes Datum erhält einen Zeitstempel. Die Repräsentation erfolgt über ein sogenanntes "Blackboard"-Konzept, d. h. die Daten werden weitgehend unstrukturiert in einem großen Informationspool (Tafel) dargestellt.

## 6. Elfes

Einen ähnlichen Ansatz wie Tujnmann verfolgt Elfes /13/. Folgende logS sind implementiert:

- Sensor: Horizontales Schnittbild (Polaroid US Sensoren)
- Scanner: Rohdatenverarbeitung für die US-Sensoren
- Mapper: integriert alle USS zu lokalem Plan
- Kartograph: überträgt lokalen Plan in globales Koordinatensystem
- Matcher: vergleicht neuen Plan mit neuem Weltbild
- Objekt Extraktion
- Graph Building: "frame based symbolic description"

Elfes unterscheidet zwischen 3 Sensordimensionen:

- räumliche Auflösung
- Abstraktionsgrad der Sensorergebnisse
- lokaler Blick versus globaler Blick

### 3.3.2. Bewertung der Literatur

Die Untersuchung obiger Veröffentlichungen sowie einiger anderer Aufsätze (/14/, /15/, /16/) zeigt, daß die an ein MSS gestellten Anforderungen nur im Montagebereich ansatzweise gelöst sind. Die Arbeiten im Bereich mobiler Systeme befinden sich noch in der konzeptionellen Phase. Trotz aller Begriffsvielfalt kristallisieren sich doch einige strukturelle Eckwerte für ein MSS heraus:

- a) Die physikalischen Sensoren, deren Ausgangsdaten und die zugehörigen Bildverarbeitungsstrukturen, werden in sogenannte logische Sensoren aufgeteilt, worunter meist Programmstrukturen verstanden werden, die, basierend auf den Daten der physikalischen Sensoren, genau ein Merkmal extrahieren oder eine abgeschlossene Aufgabe erfüllen.
- b) Die logS werden in mehrdimensionalen Strukturen angeordnet, z.B. Abstraktionsgrad, Ortsauflösung, Zeithorizont, lokale-globale Beschreibung des Raumes etc..
- c) Die Zuverlässigkeit und Genauigkeit einer Sensoraussage wird im System meist als Wahrscheinlichkeit oder Fuzzy-Value mitgeführt.
- d) Die Darstellung der Sensorergebnisse, sowie die Kopplung der logS erfolgt über einen der 3 Mechanismen
  - Blockboard
  - Semantisches Netz
  - Kommunikationskanäle
- e) Sind Aufbau und Form auftretender Objekte im voraus bekannt, so können Verknüpfungsregeln für die Ergebnisse verschiedener Sensoren bereits im Voraus definiert werden; dies ist bei Montagezellen, nicht jedoch bei autonomen Fahrzeugen auf Erkundungsfahrt möglich.

Was die Kommunikationsmechanismen in Punkt d) betrifft, so steht beim Blockboarding die Schönheit des gedanklichen Ansatzes der mangelnden Realisierbarkeit gegenüber. Das Semantische Netz sieht demgegenüber zwar Strukturierung der Daten vor, die sich durch die Anlehnung an das Korsett der natürlichen Sprache jedoch als unpraktisch erweisen kann.

Das Prinzip der Kommunikationskanäle ist programmtechnisch sicher leicht zu realisieren, und verarbeitungstechnisch auch schnell, hat jedoch den Nachteil, daß insbesondere in der Entwicklungsphase die Daten nicht sichtbar und lesbar strukturiert sind. Eine Beurteilung des Systemzusammenspiels wird schwierig. Um beliebige logS zu koppeln, ist hier eine genaue und sicher fehleranfällige Definition der Schnittstellen erforderlich.

### 3.4. ENTWURF FÜR EINE MS ENTWICKLUNGSUMGEBUNG

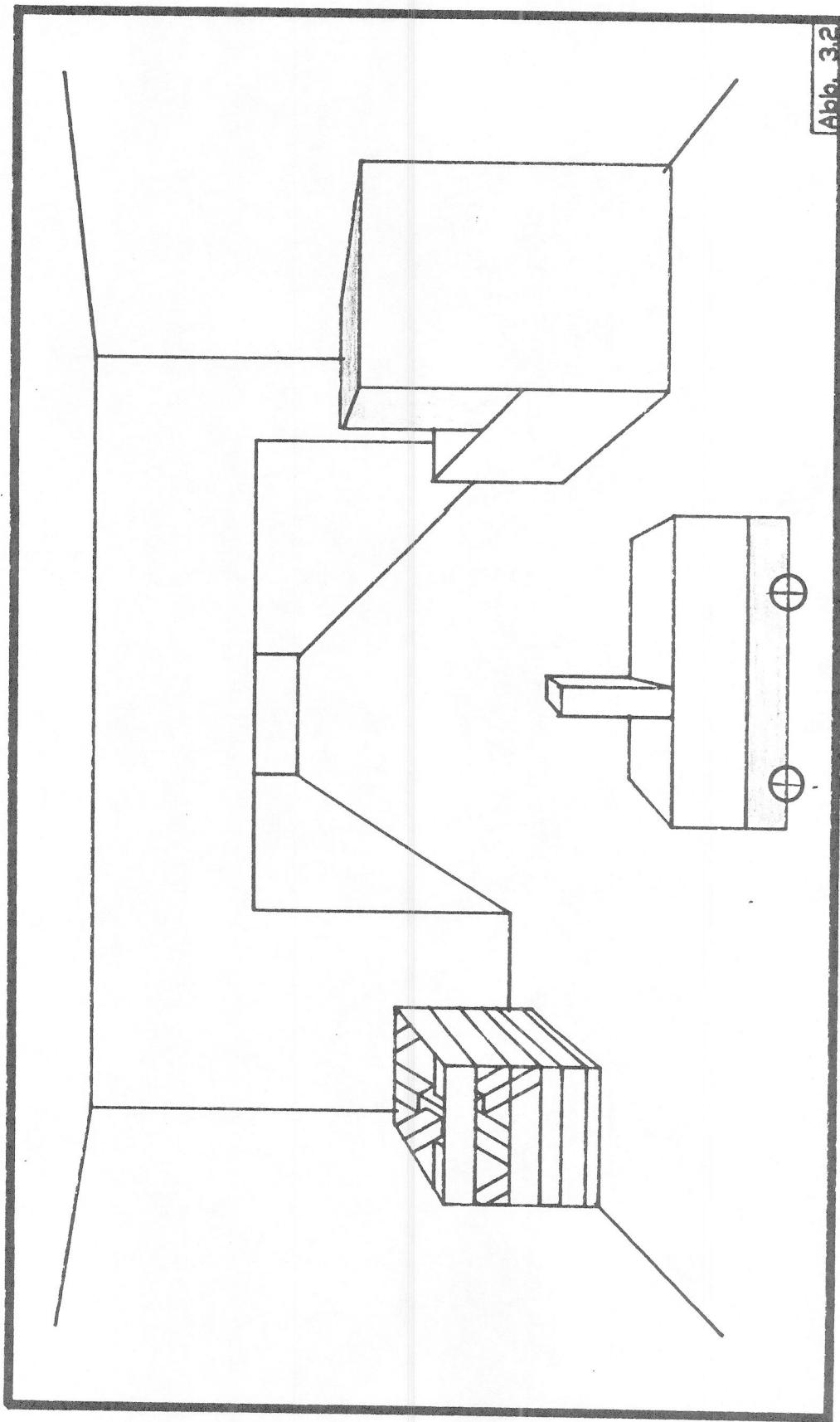
Zur Einbindung fortgeschrittener Bildverarbeitungs- und Segmentierungsalgorithmen für die LSR.E. implementiere ich einen Entwicklungsrahmen, der es erlaubt, ein Multisensorsystem aufzubauen und die verschiedensten, oben zitierten Konzepte, sowie neue Ansätze zu realisieren. Die volle Echtzeitfähigkeit ist nicht Ziel meiner Arbeit. Nach dem hundertäugigen Wächter der Antike, der auch im Schlaf stets 99 Augen offenhielt, nenne ich mein System ARGUS.



#### 3.4.1. Exemplarische Aufgabenstellung an ein MSS

Anhand einer, für die MAKROBE typischen Umweltszene (Industrielle Umgebung), werde ich Aufbau, Mechanismen und Strukturen von ARGUS erläutern (s. Abb.3.2).

Abb. 3.2



Die MAKROBE fährt und arbeitet momentan vom Standpunkt des Bildbetrachters aus. Zu sehen ist eine meist flächige, quaderförmige Welt, Objektflächen sind von Kanten berandet, die sich an Ecken schneiden. Flächen haben unterschiedliche Farben (Reflexionsverhalten). Zusätzlich treten feinstrukturierte Objekte auf (z.B. Materialpalette), die durch die begrenzte Bildpunktanzahl von der LSR.E. nicht mehr detailliert auflösbar sind. Weit entfernte Flächen liegen außerhalb des Sichtbereichs der LSR.E. ( $>12$  m). Durch die Szene bewege sich z.B. ein weiteres Roboterfahrzeug.

Sei das Multisensorsystem der MAKROBE mit folgenden physikalischen Sensoren ausgerüstet:

- LSR.E.
- Mikrowellenradar zur Entferungs- und Geschwindigkeitsmessung
- Stereo-Kamera-Bildverarbeitungssystem
- Rad- und Positionssensor

Um diese Szene vollständig zur Fahrzeugführung zu erfassen und zu erkunden, ist folgende Aufgabenstellung an ein Multisensorsystem und seine Sensoren naheliegend:

#### I. Beginne mit LSR.E.:

a)

- Zerlege das Bild in einzelne räumliche Ebenen
- Zerlege die Ebenen in Flächen
- Detektiere Kanten
- Erkenne Umstrukturen
- Erkenne Bereiche außerhalb des Sichtbereiches

b)

- Zerlege mit Hilfe des Qualitätswertes die Raumflächen in Farbflächen

#### II. Basierend auf Ergebnis I beauftrage die Stereo-Kamera:

- Umstruktur auflösen

#### III. Beauftrage den Radar

- nach bewegten Objekten zu suchen
- den Bereich außerhalb des LSR.E. Sichtbereich zu vermessen

IV. Verknüpfe die Einzelergebnisse aus I, II und III über definierte Zusammenhänge der Einzel-Sensor-Koordinatensysteme zu einem Gesamtbild

V. Transformiere das Gesamtbild in ortsfeste Weltkoordinaten anhand der Eigenposition der MAKROBE

VI. Detektiere anhand vorliegendem Weltbildes die Bewegungsrichtung des fremden Roboterfahrzeuges

Es zeigt sich, daß hier verschiedene Klassen von Aktionen und Daten auftreten. Auf grundlegende Bildverarbeitungsmechanismen setzen Aktionen auf, die sich von Szene zu Szene, abhängig von den Messdaten, ändern können.

Manche Operationen arbeiten direkt auf Sensordaten, andere wiederum verknüpfen verschiedene Sensorergebnisse. Zur Bewegungsdatenverarbeitung ist eine Transformation aller Ergebnisse in Weltkoordinaten, sowie das Mitführen von Zeitmarken unerlässlich.

### 3.4.2. Logische Sensoren

Ein Multisensorsystem besteht aus atomaren Verarbeitungsfunktionen (=logS), die über geeignete Mechanismen Auftrags-, Wissens- und Ergebnisabhängig mit Daten versorgt und konfiguriert werden müssen.

Speziell für die LSR.E. ist also ein Satz von atomaren Bildverarbeitungsschritten zu entwickeln, die eine flächige Umwelt von grober bis mittlerer Struktur, in seine Primitive Ebene, Kante, Farbfläche, Eckpunkte etc. auflöst.

### 3.4.3. Kommunikation zwischen den logS

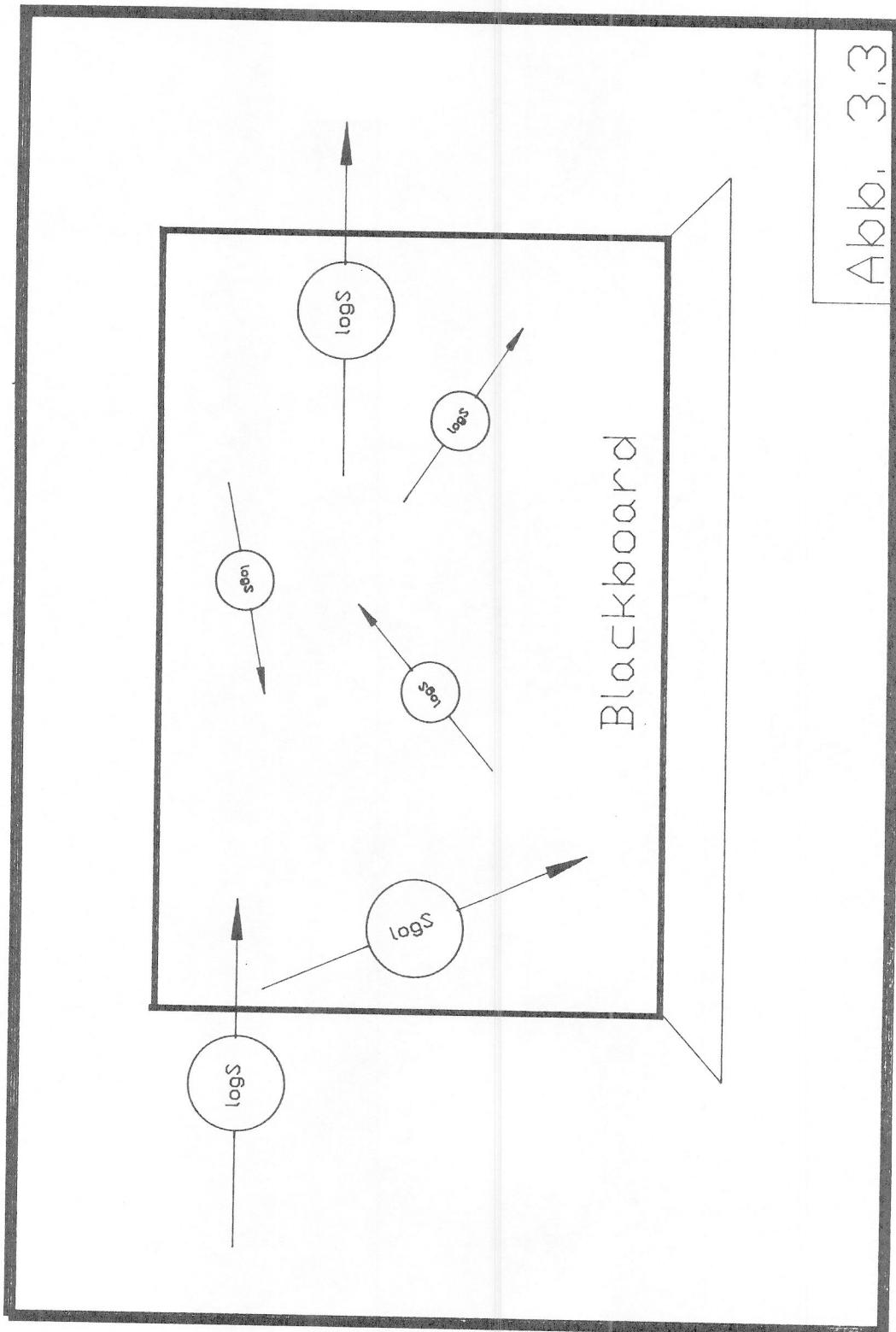
Ein logischer Sensor, wie oben definiert, benötigt Eingangsdaten, sei es bereits existierendes Wissen oder Sensordaten, und gibt wiederum Daten aus. Wie in der Literatur beschrieben sind hier 3 Kommunikationskonzepte denkbar:

- alle Sensoren lesen und schreiben auf ein großes Datenfeld (s. Abb. 3.3)
- das Datenfeld ist durch semantische Zusammenhänge strukturiert (s. Abb. 3.4)
- die logS kommunizieren über feste Kanäle, jeder logS erreicht nur seinen Nachbarn direkt, andere logS nur mittelbar (s. Abb. 3.5).

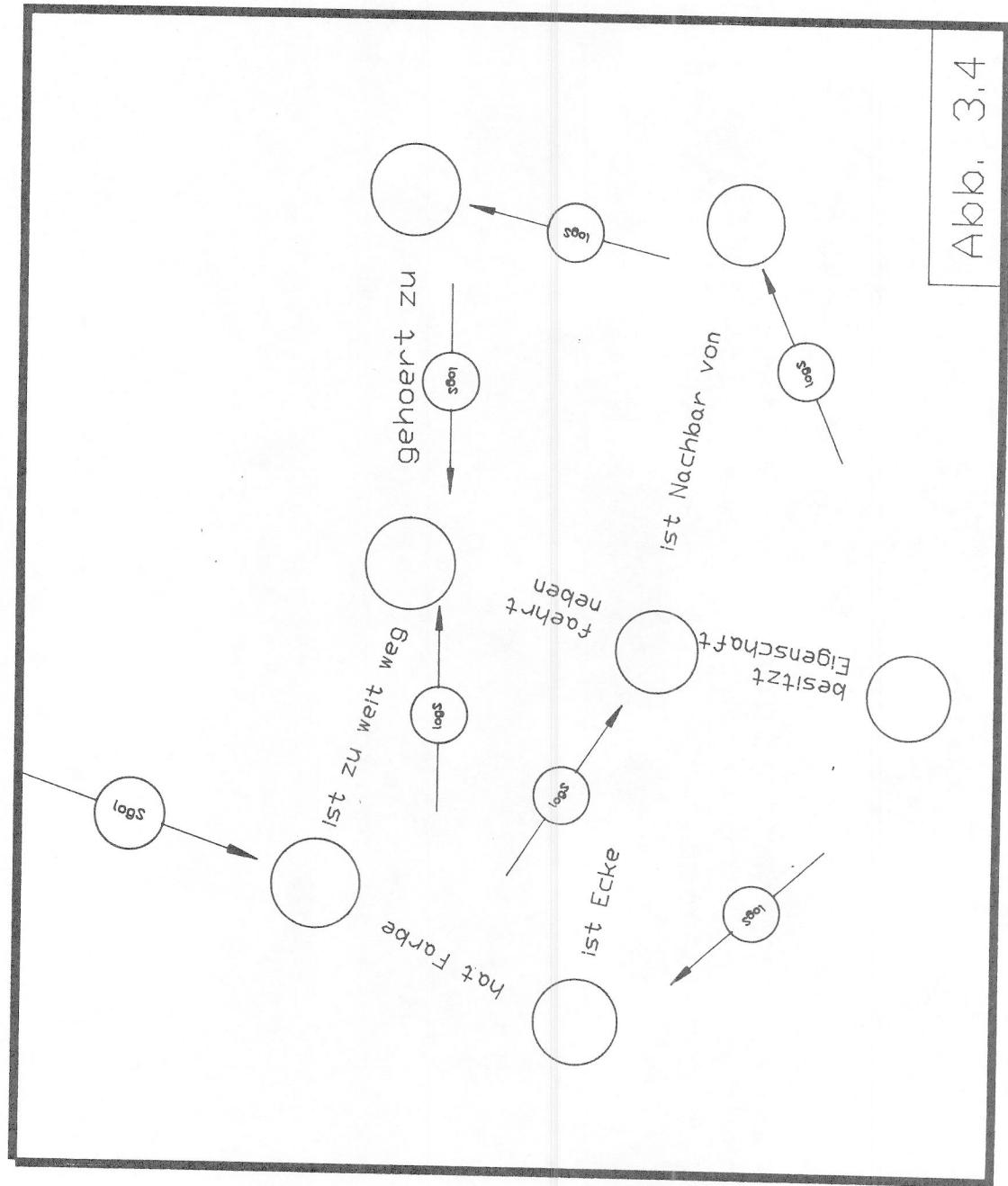
Neben der Verschaltung der bloßen Kommunikationskanäle ist es erforderlich Datenstrukturen, Übertragungsprotokolle und Verriegelungsmechanismen zum strukturierten Datenaustausch zu definieren. Im Falle des MSS für die MAKROBE werden hauptsächlich Geometriedaten verarbeitet. Ein experimenteller Multisensorrahmen für die MAKROBE sollte also in der Lage sein, verschiedenste Kommunikationsmechanismen zur Verfügung zu stellen, Geometriedaten zu verwalten, ein einfaches Aufsetzen elementarer Bildverarbeitungsalgorithmen, sprich logischer Sensoren, zu unterstützen und sich in das LS.R.E. Rechnernetz multitasking und multiuserfähig einbinden lassen.

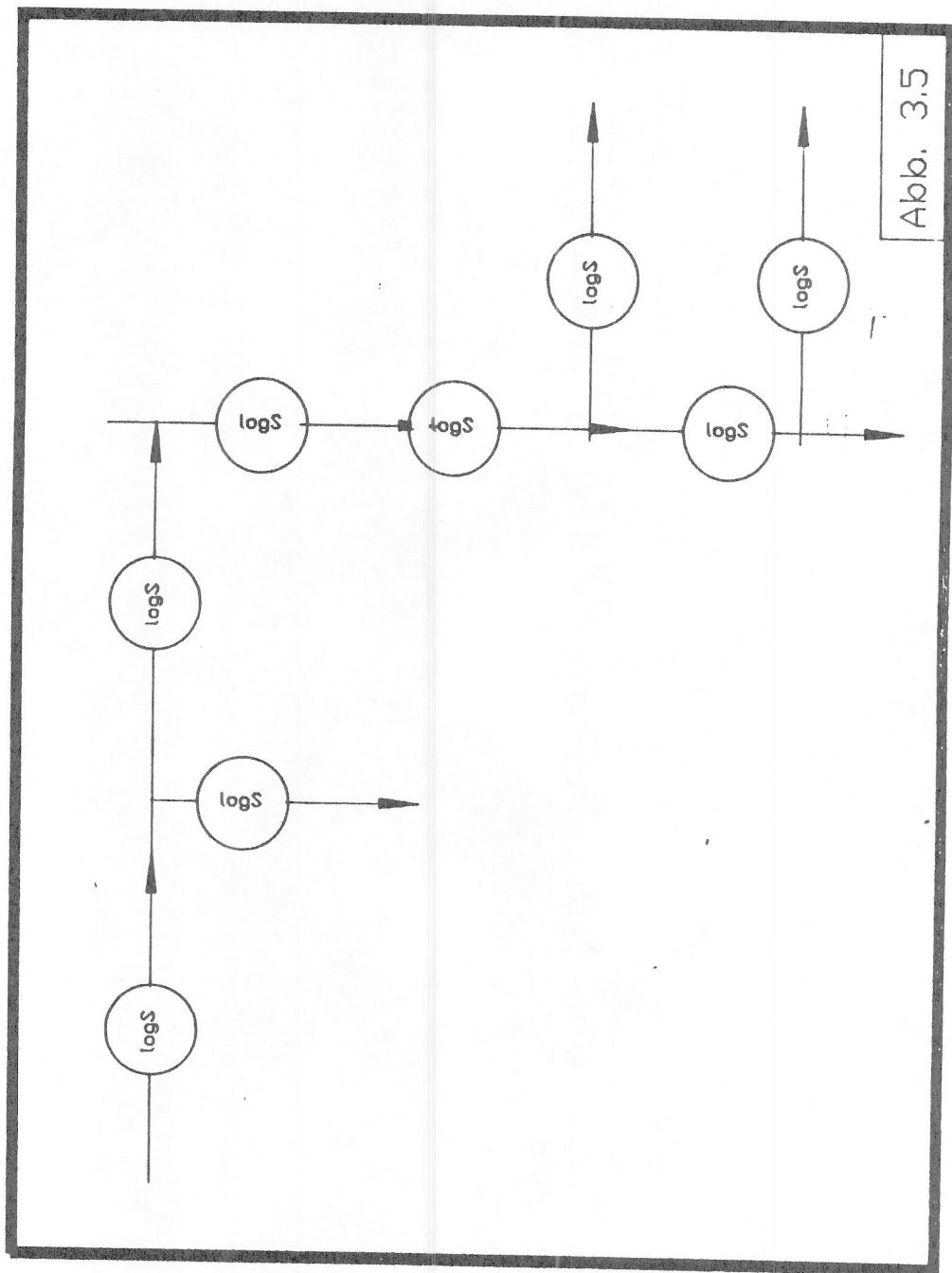
## 3.5. DIE RELATIONALE DATENBANK RDB/VMS UND IHRE WERKZEUGE ALS GRUNDGERÜST ZUR REALISIERUNG EINES MSS

Ausgehend von der Aufgabenstellung und obiger Randbedingungen, habe ich mich entschieden, ein MS Experimentiersystem auf der Basis der relationalen Datenbank (im weiteren RDB) RDB/VMS der Firma DEC zu realisieren.



Akko, 3,3





### 3.5.1. Prinzipieller Aufbau einer RDB

Das Betriebssystem einer Datenverarbeitungsanlage verwaltet seine Daten insbesondere Massendaten, indem es die Steuerung und Organisation der Massenspeicher übernimmt, Inhaltsverzeichnisse verwaltet, Zugriffspfade organisiert etc.. Die logische Organisation der Daten selbst wird kaum unterstützt.

Datenbanksysteme bilden hier ein Bindeglied zwischen dem Benutzer und dem Betriebssystem; mit Hilfe solcher Systeme kann der Anwender Daten, aufgrund logischer Zusammenhänge zwischen den Daten, abfragen, ohne sich um Feinheiten der Fileverwaltung des Betriebssystems zu kümmern. Datenbanksysteme bilden Daten von der physikalischen Filestruktur auf eine logische Benutzeroberfläche ab. Mit Hilfe sogenannter Daten-Definitions-Sprachen definiert der Anwender die Struktur seiner Daten, über Abfragesprachen sucht das Datenbanksystem die gewünschte Information (logische Kombination von Daten) oder aktualisiert die Datenbestände.

Datenbanksysteme (DBS) unterscheiden sich hauptsächlich in der Organisation ihrer Daten. Bekannt sind z.B.

- Hierarchische Modelle
- Netzwerkmodelle
- Relationale Modelle

In der Hierarchischen DB werden Daten durch einen Oberbegriff zusammengefasst, Daten selbst können wieder Oberbegriffe für untergeordnete Daten sein. Ein Zugriff auf untergeordnete Einträge ist nur nach Durchlaufen der ganzen Hierarchie möglich. Wie in jeder strengen Hierarchie ist die Flexibilität solcher Systeme gering.

Relationenname

Student\_der\_Elekrotechnik

Name	Matrikelnr.	Schwerpunkt
Greim	047075	C3
Meier	007003	B1
Huber	418356	A
Schmitt	103854	C3

Atributnamen

3 - Tupel

Schlüselfeld

Attribute

(0...999999) = Domäne der Matrikelnummer

Studienschwerpunkte

Abkürzung	Name_des_Studienschwerpunktes
A	Energietechnik
B1	Nachrichtentechnik
B2	Kybernetik
C3	Automationstechnik

Abb. 3.6

Im Gegensatz hierzu steht die Relationale Datenbank (RDB). Sie besteht aus einer Menge von Tabellen, sogg. Relationen (s. Abb.3.6). Die Zeile einer solchen Tabelle heißt Tupel, die Spalten Felder oder Attribute. Die Menge aller möglichen Werte, die ein Eintrag in eine Spalte annehmen kann, heißt Domäne. Eine Menge von Feldern, deren Eintragungen eine Tupel eindeutig identifizieren kann, heißt Schlüssel. Ein Schlüssel besteht aus 1... bis n Feldern. Ein Schlüssel, der kein leeres Feld enthält, heißt primärer Schlüssel. Es existiert keine Reihenfolge der Tupel. Jedes Attribut oder jede Relation hat im Allgemeinen einen Namen.

Hält man sich an obige Regeln, so sind auch die mathematischen Voraussetzungen für eine Definition der Relation gegeben; heuristisch ausgedrückt: Zwei Feldern A und B einer Relation ist eine Aussage zugeordnet. Die Menge aller geordneten Paare (xy) aus der Grundmenge A x B, die beim Einsetzen in die Aussageform diese erfüllen, definieren eine Lösungsmenge, die man als Relation bezeichnet.

Am Beispiel aus Abb. 3.6 sei dies die Aussage:

Der *Student* der *Elektrotechnik* Greim, *Matrikelnr.* 047075 hat den *Studienschwerpunkt* C3 gewählt

Eine Datenbank, bestehend aus vielen Relationen, ist nur dann effizient, wenn ich mit wenigen effektiven Abfragemechanismen auf jedes Datum in der RDB zugreifen kann. Es stehen nun 5 Operationen zum Zugriff auf eine RDB zur Verfügung.

1. Union
2. Differenz
3. Karthesisches Produkt
4. Projektion
5. Selektion

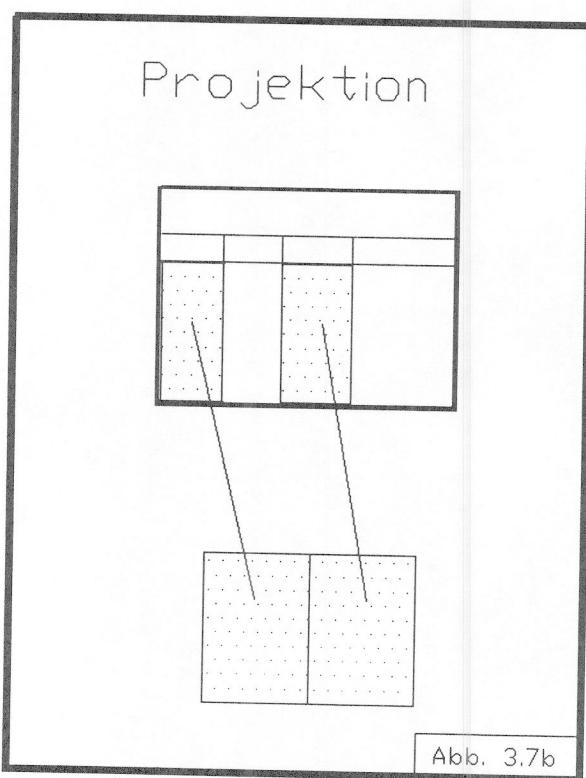
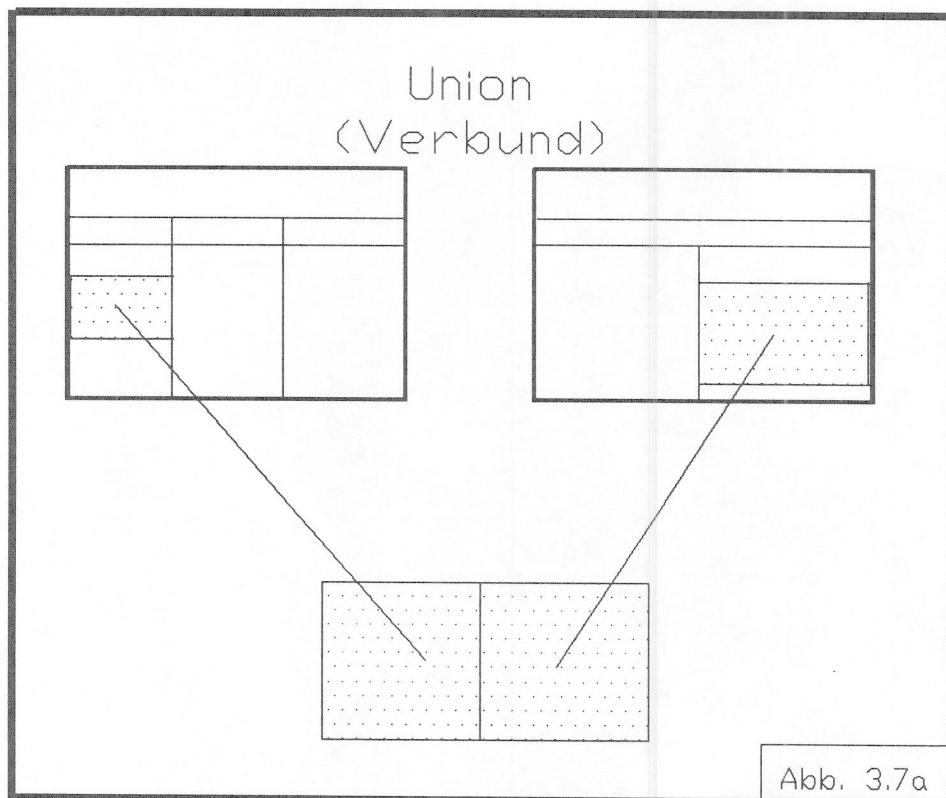


Abb.3.7. zeigt schematisch die Operationen Union und Projektion, (für weitere Beispiele sei auf /17/ verwiesen). Diese Operationen bilden die sogenannte Relationale Algebra, die die hervorragende Eigenschaft der Abgeschlossenheit besitzt, d.h. diese 5 Operationen genügen, um unter allen möglichen Abfrage- und Randbedingungen auf jedes Datum in der RDB zuzugreifen. Die Ableitung weiterer Operationen aus den oben Beschriebenen ist zulässig.

Eine weitere, ebenfalls mathematisch eindeutige und der Relationalen Algebra gleichwertige Klasse von Datenbank-Zugriffsmechanismen, bildet das Relationale Kalkül. Es hat die Form:

< Zielangabe > mit < Bedingung>

In ihr wird über Platzhalter eine logische Beziehung definiert, die die RDB mit all denen Daten ausfüllt, die eine wahre Aussage liefern.

Abb.3.8. zeigt ein Beispiel.

```

FOR X IN STUDENT_DER_ELEKTROTECHNIK
CROSS Y IN STUDIENSCHWERPUNKT
  WITH X.NAME = GREIM
    AND (X.SCHWERPUNKT = Y.ABKUERZUNG)
      PRINT Y.NAME DES STUDIENSCHWERPUNKTES
END_FOR
  
```

Abb. 3.8

Die RDB/VMS verwendet das Relationale - Kalkül. Um jederzeit konsistente Datenbankeinträge zu erhalten, sowie Änderungen der RDB leicht vornehmen zu können, werden bestimmte Regeln angegeben, wie Relationen aufzubauen sind. Es sind 3 Normalformen definiert, die alle Relationen der RDB besitzen sollen:

1. Normalform:

Es existieren keine doppelten Zeilen oder Spalten.

2. Normalform:

Alle Felder sind vom ganzen zusammengesetzten Schlüssel (mehrere Schlüsselfelder) abhängig.

3. Normalform:

Es dürfen keine transitiven Abhängigkeiten der Schlüsselfelder untereinander bestehen.

Alle obigen Regeln sind beim Entwurf einer Relationalen - Datenbank zu berücksichtigen. Der scheinbare Aufwand macht sich aber bezahlt, da nach sorgfältiger Planung die schnelle Realisierung einer RDB (hier RDB/VMS) die Datenabfrage und die Flexibilität überzeugen.

### 3.52 Geometriedaten in einer RDB

Das Einbinden räumlicher Geometriedaten und Objektbeschreibungen in eine Datenbank, meist nicht relationaler Art, ist bisher primär im Bereich von CAD-Systemen untersucht worden (Meier /18/, Pohlmann /19/, Eberlein /20/). Aufgabe eines CAD-Systems ist es, aus wenigen Geometrieprinzipien die vollständige 3-dimensionale Werkstückbeschreibung (alle Raumpunkte) zu synthetisieren. Im Bereich der Umweltsensorik liegt genau der umgekehrte Fall vor; Aufgabe ist die Analyse und Segmentierung eines punktuellen Raumbildes in geometrische Objekte. Trotz dieser Einschränkung erweist es sich als sinnvoll, manche Definitionen und RDB-Layoutregeln aus dem CAD Bereich in den RDB Sensorrahmen ARGUS zu übernehmen.

1. Versuche, die geometrischen Strukturen in den Relationen der DB zu erhalten; sinnvoll sind hier Relationen wie :

- |    |        |               |
|----|--------|---------------|
| a) | Fläche | Eigenschaft   |
|    | Kante  | Eigenschaft   |
|    | Punkt  | Koordinaten   |
|    | Ebene  | Eigenschaft   |
| b) | Ebene  | Fläche        |
|    | Fläche | Kante         |
|    | Ebene  | Punkt         |
|    | Fläche | Punkt         |
|    | Fläche | Nachbarfläche |

2. Verschiedene geometrische Primitive, wie Kanten und Flächen, treten in jeder Szene vielfach auf, sie besitzen außer den Koordinaten ihrer Punkte keinen eindeutigen Identifikationsschlüssel. Um trotzdem eindeutig und schnell auf solche Objekte zugreifen zu können ist es notwendig, daß das System solchen Objekten einen künstlichen Schlüssel, im allgemeinen eine Ordinalzahl, zuordnet (Meier: Surrogat; Pohlmann: Elementkennung).

3. Die Menge der objektbeschreibenden Relationen wird in 2 große Gruppen unterteilt:

Die Relationen unter a) bezeichnet man als metrische /Meier/ oder elementorientierte /Pohlmann/ Relationen.

Die metrischen Relationen beschreiben die Eigenschaften einzelner Objekte meist in Zahlen. In ihnen tritt als einzige Schlüsseldomäne die Elementkennung (= Surrogat) auf.

4. Die topologischen elementorientierten Relationen (b) sind vom Typ - Part of- bzw. -is A-.

-Part of- beschreibt den existentiellen Quantor; zu jedem Teilobjekt existiert genau ein übergeordnetes Objekt, die Umkehrung dieser Aussage -is A- ordnet allen Subteilen die Eigenschaften des übergeordneten Objektes zu. Hier treten paarweise die Surrogate der metrischen Relationen auf.

Die topologischen Relationen sind Translations- und Rotations invariant (allgemein Transformation)! Will man die geometrischen Informationen z. B. in ein Weltkoordinatensystem transformieren, so genügt es, die metrischen Relationen dieser Operation zu unterziehen.

5. Die Datenbank soll geometrische Größen, wie Vektoren und Matrizen, als ein Element behandeln und abspeichern. Da die RDB/VMS ursprünglich für den Verwaltungsbereich konzipiert ist, wird dieses nicht unterstützt.

6. Beim Eintrag sollen die Daten auf geometrische Konsistenz überprüft werden (z. B. 3 Kanten schneiden sich in einem Eckpunkt). Eine solche Überprüfung ist im Falle der LSR.E. Meßdaten, die die Objekte nur in 2,5 Dimensionen in einem begrenzten Sichtbereich beschreiben, nicht sinnvoll.

### 3.5.3 In ARGUS realisierte Geometriedaten Relationen

Abbildung 3.9 zeigt die topologischen Relationen in einer flächigen Umwelt mit den jeweiligen metrischen Eigenschaften der Umweltobjekte. Basierend auf:

- den Regeln für Relationale Datenbanken
- den Normalisierungsbedingungen für RDBs
- den Empfehlungen zur Speicherung von Geometriedaten in Relationen

habe ich folgende Relationen zur Beschreibung von Umwelteigenschaften definiert (s. Abb.3.10):

- LS\_Daten:

enthält alle von der LSR.E. gelieferten Daten eines (metrisch) Bildes, sowie das Surrogat Punktnummer.

- Kartes:

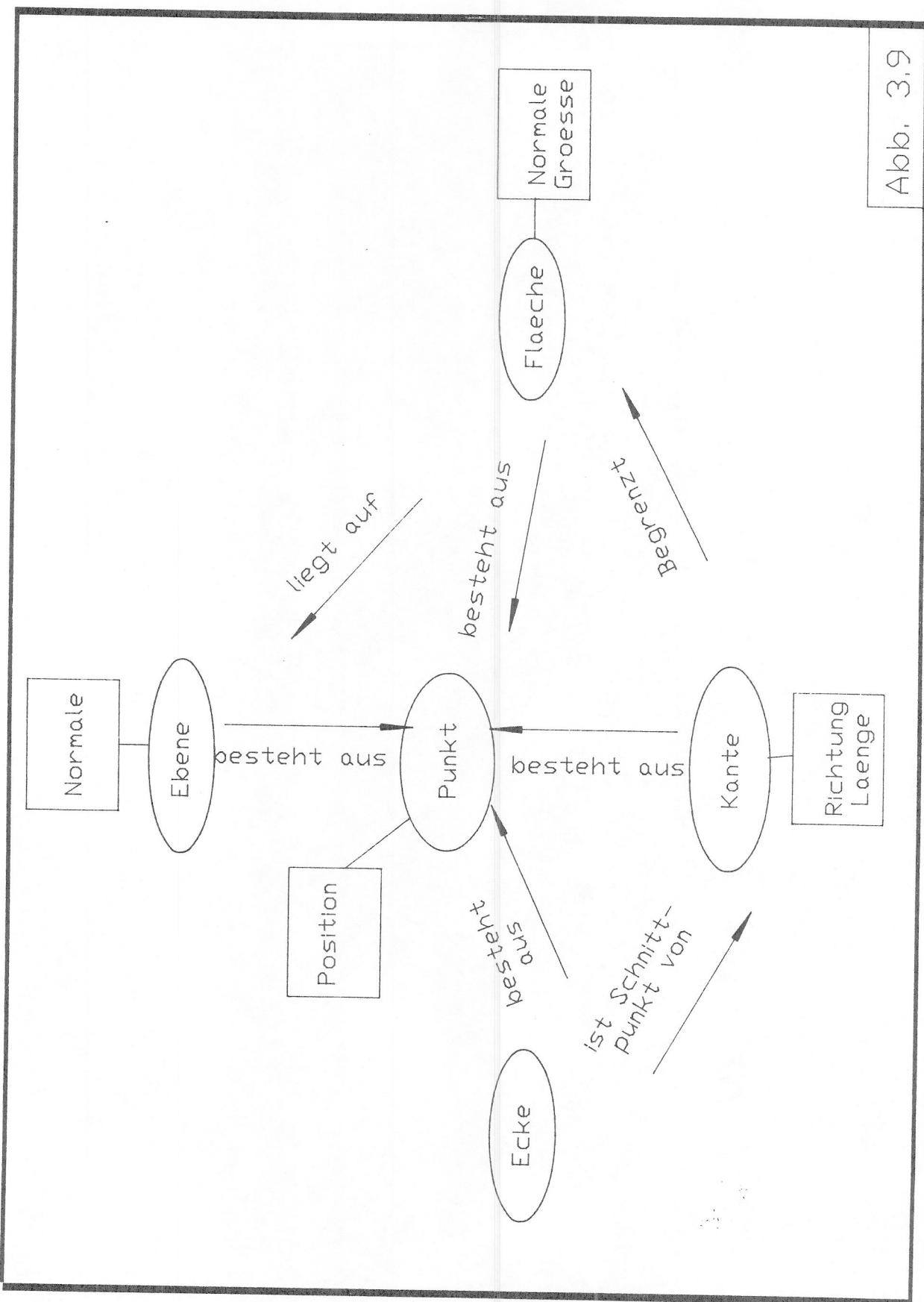
enthält die xyz (karthesischen) Koordinaten aller (metrisch) Meßpunkte; die Meßpunkte werden anhand des Schlüssels Punkt-nummer identifiziert.

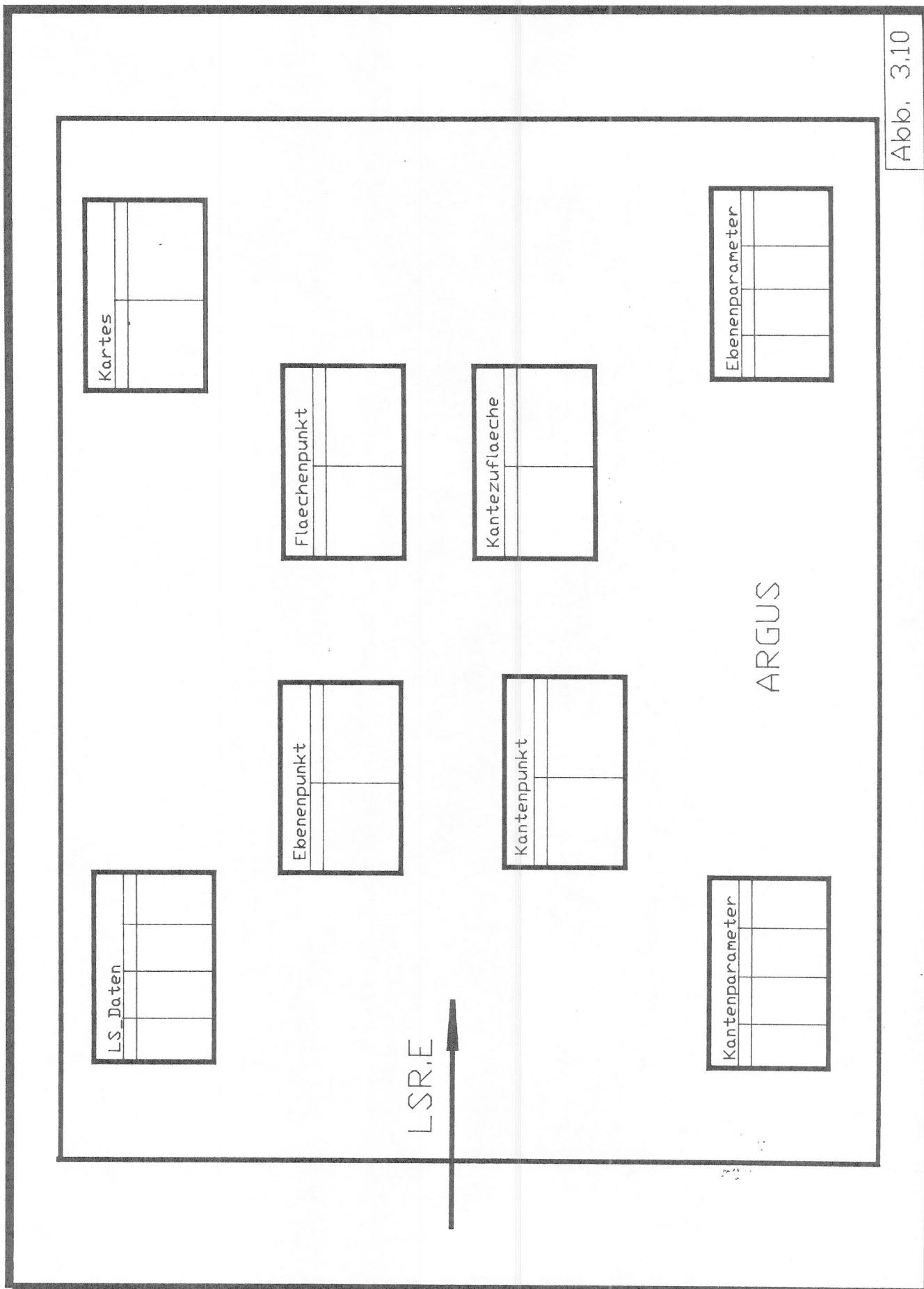
- Ebenenparameter:

Beschreibt jede Ebene, identifiziert durch (metrisch) das Surrogat, Ebenennummer, durch 3 Raumwinkel.

- Kantenparameter:

Beschreibt jede Kante durch die Heßsche Normalform (metrisch).





- Ebenenpunkt:

Zuordnung Ebennummer - Punktnummer (topologisch)

- Flächenpunkt:

Zuordnung Flächennummer - Punktnummer (topologisch)

- Kantenpunkt:

Zuordnung Kantennummer - Punktnummer (topologisch)

- Kantezufläche:

Zuordnung Flächennummer - Kantennummer (topologisch)

Die genauen Definitionen in der Syntax der RDB/VMS finden sich in Anhang.8

### 3.6 ANBINDUNG PHYSIKALISCHER UND LOGISCHER SENSOREN AN DIE RDB

Die in der RDB festgelegten Datenstrukturen bilden nun das Grundgerüst für die logischen Sensoren (hier z. B. elementare Bildsegmentierungsalgorithmen), die lesend und schreibend auf die RDB zugreifen. Hierzu bietet die RDB/VMS hervorragende Werkzeuge. Datenbankabfragen und Schreiboperationen lassen sich fast in alle Programmiersprachen integrieren. Hierbei werden die Befehle in der Abfragesprache der RDB/VMS in den Programmtext integriert, sie stellen somit eine Spracherweiterung der jeweils verwendeten Programmiersprache dar. Unterstützt werden zur Zeit COBOL, BASIC, FORTRAN und PASCAL. Ein Precompiler löst die RDB-Befehle in entsprechende Systemaufrufe auf, das vorübersetzte Programm kann dann vom jeweiligen Sprachcompiler ohne Einschränkung übersetzt werden.

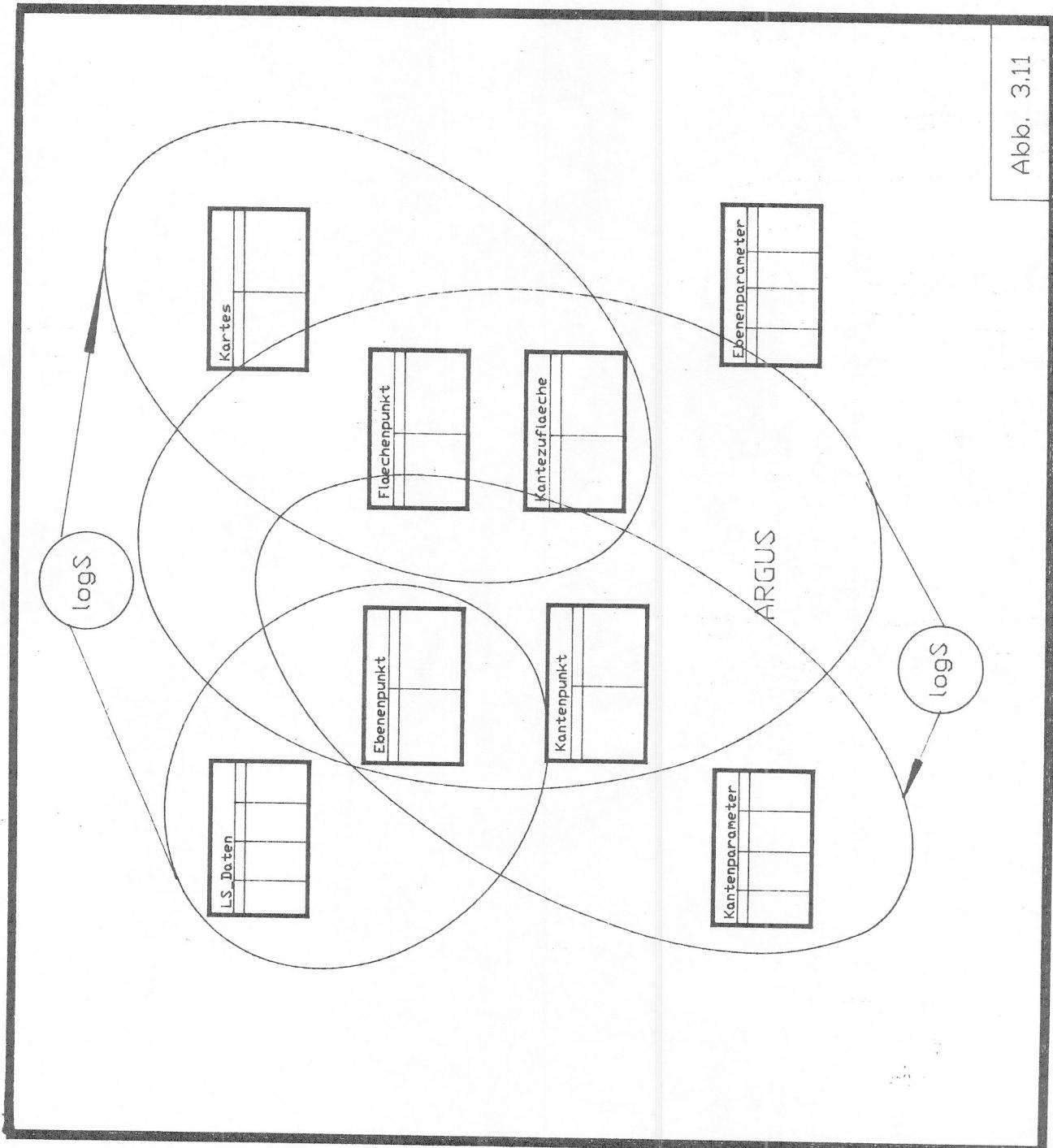
Ein logischer Sensor bzw. dessen programmtechnische Realisierung wird deshalb notwendigerweise aus 3 Blöcken bestehen: Lesender Zugriff auf RDB, Datenverarbeitung, Schreibender Zugriff auf RDB.

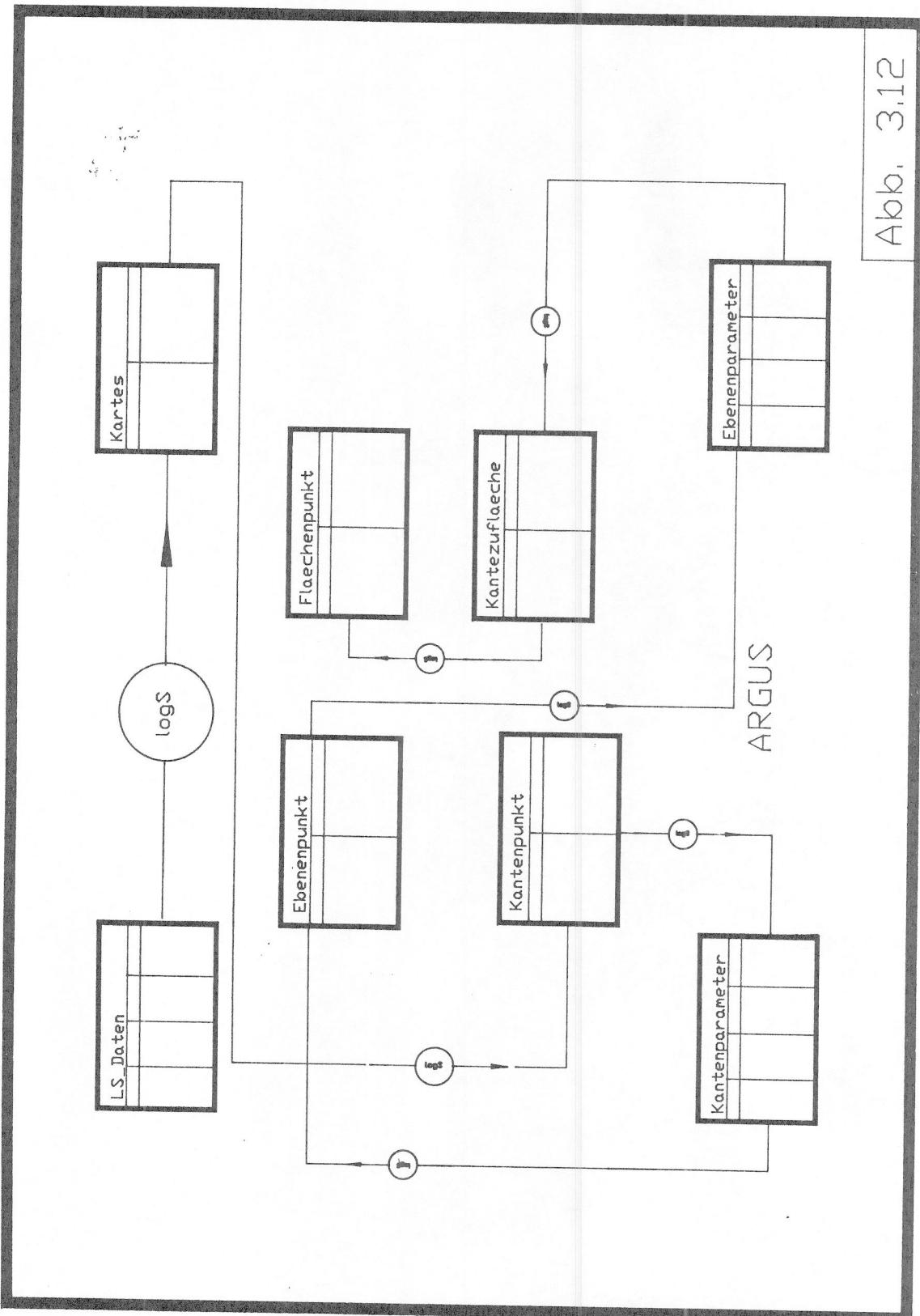
### 3.6.1 Kommunikationsmechanismen logischer Sensoren auf der RDB

Da es mit Hilfe der Abfragesprache möglich ist, auf die gesamte RDB gleichzeitig oder auch selektiv nur auf ein Datum zuzugreifen, lassen sich alle oben erwähnten Kommunikationsstrukturen eines MSS realisieren.

- Blackboarding: Jedem logischen Sensor sei der Zugriff auf alle Daten der RDB gestattet (s. Abb 3.11)
- Hierarchisches oder Kanalkonzept: Jeder logS liest aus genau einer Relation und schreibt in genau einer Relation; auf jede Relation greift nur ein logS lesend, und nur ein logS schreibend zu (s. Abb 3.12).
- Semantisches Netz: Da zwischen den Feldern einer Relation ein inhärenter semantischer Zusammenhang besteht (Name der Relation !), kann der Eintrag in eine Relation als das Ausfüllen eines semantischen Netzes interpretiert werden. So besteht beispielsweise zwischen den Attributen topologischer Relationen der semantische Zusammenhang "is A" oder "is Part of" (s. o. und Abb. 3.13).
- Mischformen: Neben der 3 obigen Deutungen eines RDB Zugriffs logischer Sensoren sind sämtliche Mischformen von Zugriffsrestriktionen möglich.

Ganz automatisch stellt die RDB bei der Kommunikation der logS untereinander, aber auch des MSS, zu anderen Systemen der MAKROBE eine genormte offene Schnittstelle dar. Die Definitionen der Attribute und Relationen können über die interaktive Schnittstelle RDO der RDB/VMS ausgegeben werden.





Alok, 3,12

L.S.	Daten

Punkt hat kartesische Koordinaten

Punkt gehoert zu Ebene

Ebenenpunkt

Ebene besteht aus Flächen

Flächenpunkt

Fläche wird umrahmt von Kante

Karte

Kantenpunkt

Kantewinkel  
Kantewinkel

Kantezufläche

Ebenenparameter

Kante gehoert zu  
Fläche  
ARGUS

Kantenparameter

Akk. 3.13

Neben der mehr logischen Ausformung von Kommunikationsmechanismen auf die RDB unterstützt die RDB/VMS auch auf physikalischer Ebene jede Art von Kommunikation. Die sind:

- Netzanbindung: Die RDB/VMS ist voll netzwerkfähig, d. h. jeder Rechner am Netz des LSSR kann auf die Datenbestände von ARGUS zugreifen, als seien sie in seinem eigenen Arbeitsspeicher abgelegt; sei es der Einplatinenrechner LSI 11 auf dem Fahrzeug MAKROBE, sei es die VAX 11/780, sei es eine Graphikstation.
- Multiuser - Multitasking: Die RDB/VMS unterstützt alle Verriegelungsmechanismen einer Datenbank. Das heißt, alle Konfliktsituationen, die durch gleichzeitigen Zugriff mehrerer User auf den gleichen Datensatz auftreten können, werden abgefangen. So kann zum Beispiel das Multisensorsystem seine Umweltdaten aufbereiten, während gleichzeitig das Planungs- und Entscheidungssystem der MAKROBE auf Zwischen- oder Endergebnisse zugreift.

### 3.6.2 Geometrische Verknüpfung verschiedener physikalischer Sensoren

Zur Gewinnung eines zuverlässigen und vollständigen Umweltwissens setzt die MAKROBE unterschiedliche bildgebende Sensoren ein. Diese unterscheiden sich in ihrer räumlichen Auflösung (LSR.E.-Radar), ihrem Bezugskoordinatensystem, ihrer Reichweite, in ihrer Meßgenauigkeit und ihrer Entfernungsauflösung (LSR.E.-Stereo Kamera). Es ist dem MSS also vorher bekannt zu machen, wie diese Sensoren geometrisch zu verknüpfen sind.

Folgende Möglichkeiten sind hier für eine RDB denkbar:

Eine topologische Relation, die das Surrogat Punktnummer der LSR.E., das für jeweils einen Raumwinkel eines Bildes eindeutig ist, ist mit der zugehörigen Punktnummer Radar verknüpft (s. Abb 3.14).

Verknüpfung_LSR.E_RADAR	
Punktnr._LSRE	Punktnr._RADAR
1	1
2	1
82	1
83	1
3	2
4	2
84	2
85	2
5	3

Abb. 3.14

Eine andere Möglichkeit besteht darin, diskreten Raumwinkeln ( $\phi$ ,  $\Theta$ ) der LSR.E. direkt Raumwinkel des Radars zuzuordnen, wobei ein Toleranzmaß  $d\phi$  und  $d\Theta$  mitgeführt wird (s. Abb 3.15).

Verknüpfung_LSR.E_RADAR					
fi_LSR.E	theta_LSR.E	fi_RADAR	theta_RADAR	delta_fi	delta_theta

Abb. 3.15

Bei der Koppelung von Stereo Kamera und LSR.E. ist eine Verknüpfung über die yz-Koordinaten der LSR.E. Bildpunkte möglich.

Da andere Sensoren zum Zeitpunkt der Arbeit noch nicht verfügbar sind, kann dieser Aspekt nicht experimentell untersucht werden. Der zweite physikalische Sensor der LSR.E., nämlich die Qualität des Signals (entsprechend einem IR-Schwarz-Weiß Bildes) besitzt logischerweise das gleiche Koordinatensystem wie der Entfernungssensor.

### 3.6.3 Steuerung und Konfiguration der Sensoren

Ebenso wie das Umweltwissen über die geometrische Verknüpfung physikalischer Sensoren ist es sinnvoll, Konfigurations- und Kommunikationsregeln für die logS in der RDB abzulegen. Hierzu verschiedene Beispiele:

- a) Trotz aller scheinbaren parallelen Prozessverarbeitung, sind auch die hier eingesetzten VAX-Rechner rein sequentielle Maschinen. Es muß daher festgelegt werden, in welcher Reihenfolge einzelne Verarbeitungsschritte abgearbeitet werden. Denkbar ist eine Auftragsrelation wie in Abb 3.17 gezeigt. Jedem logischen Sensor wird hier ein Nachfolger zugeordnet. Auf die Abarbeitung eines logS folgt die Abarbeitung des Nachfolge-logS.

Auftragsrelation	
logS	Nachfolge logs
Dateneinlesen Datentransformation Kantenpunkte Ebenendetektion Flächendetektion	Datentransformation Datenablegen Ebenendetektion Flächendetektion Flächenkantendet.

Abb. 3.17

b) In den Anfangsüberlegungen zum MSS steht als zentrale Randbedingung das situationsabhängige Abarbeiten dieses Systems. Eine Relation, die verschiedenen Situationen logS zuordnet, enthält das hierzu notwendige Wissen (s. Abb. 3.18). Dabei müssen Mechanismen bereitgestellt werden, die anhand der Daten im MSS eine Situation erkennen und daraufhin, gestützt auf das Wissen in obiger Relation, logS triggern. Da diese Mechanismen über den eigentlichen logS stehen, möchte ich diese als Supersensoren bezeichnen (lat. super = über). Relationen, die nicht direkt von den logS mit Daten versorgt werden, sondern Vorwissen enthalten, seien Metarelationen (griech. meta = über).

Metarelation\_zur\_Steuerung\_der\_Sensoren

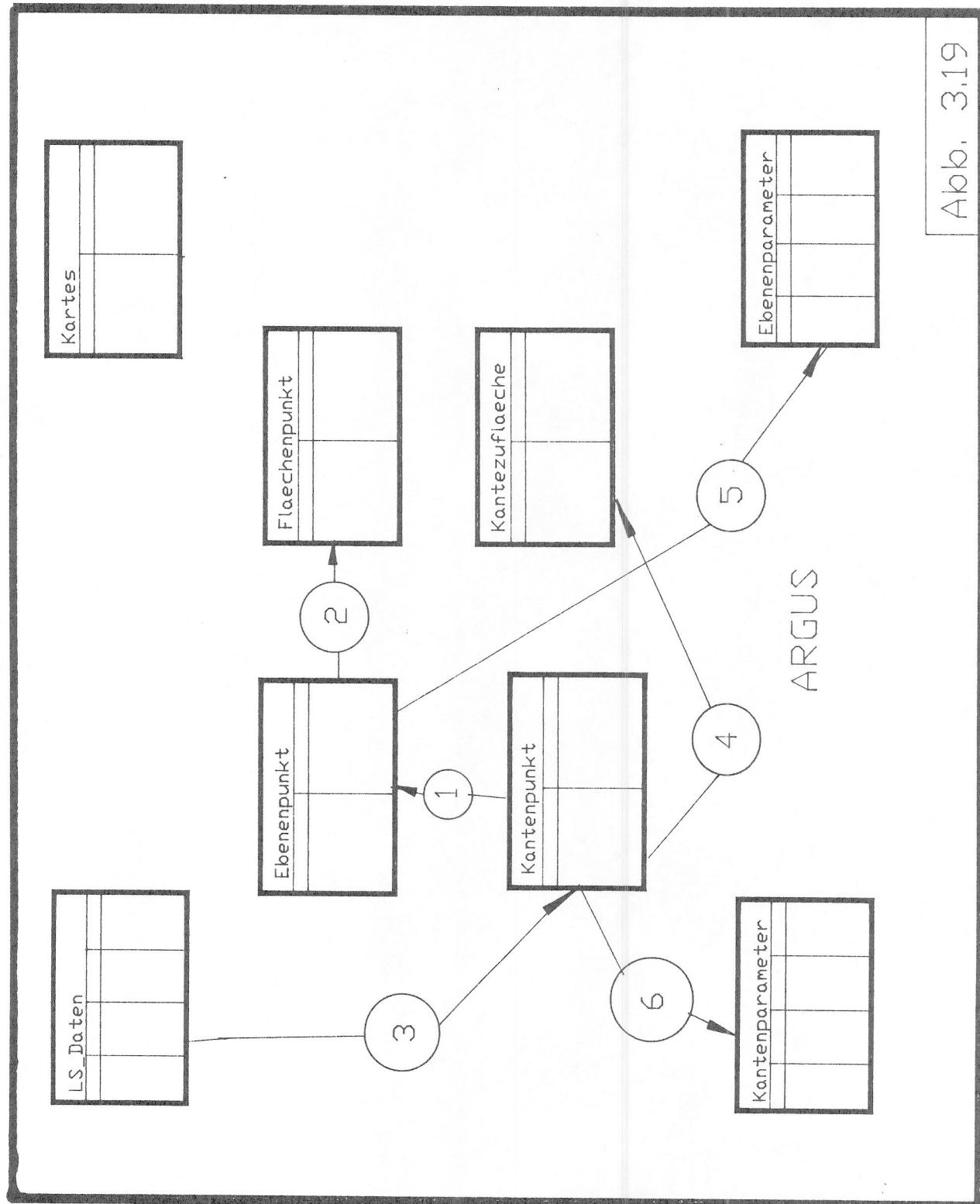
Sensor	Messung falsch	Bewegtes Objekt	Bewegung Fahrzg.	Unstruktur	Objekt zu weit weg	Gefahrsituation	Aktiviere logS
LSRE	0	0	1	1	0	0	RADAR Stereo-BV
LSRE	0	0	0	0	1	0	Entfernungsmeßung RADAR
Rad	0	0	1	0	0	0	Transformation In Weltkoord.
RADAR	0	0	0	1	0	0	Entfernungsmeßung LSRE

Abb. 3.18

Die in Abb.3.10 dargestellten Relationen, die eine flächige Umwelt zu beschreiben vermögen, müssen natürlich über geeignete  $\log S$  = Bildverarbeitungsalgorithmen mit Daten gefüllt sein. Abb.3.19 zeigt die unbedingt notwendigen  $\log S$  und deren Aufgaben:

1. Detektive Ebenen
2. Zerlege die ebenen in Flächen
3. Detektiere Kantenpunkte
4. Segmentiere die Kantenpunkte zu Flächenkanten
5. Berechne die Ebenenparameter
6. Berechne die Parameter der Objektkanten

Es ist also die Grundaufgabe der Entfernungs bildverarbeitung (EBV) zu lösen.



Aklo, 3,19

#### 4. ENTFERNUNGSBILDVERARBEITUNG

##### 4.1. STAND DER FORSCHUNG AUF DEM GEBIET DER ENTFERNUNGSBILD- VERARBEITUNG

Ein Blick auf den Stand der Forschung zeigt, daß es kaum Arbeiten auf diesem Gebiet gibt, die zudem wenig vergleichbar sind. Dies liegt einerseits an der mangelnden Verfügbarkeit echter 3-D Entfernungskameras, andererseits an den verschiedenen Meßverfahren. Während ein Stereo-SW-Kamerapaar ein weitgehend genormtes Teil darstellt, reichen die 3-D Entfernungsmeßverfahren von der Triangulation, mit Hilfe eines Laserzeigers bei Montageobjekterkennung, bis zu aktiven Laserkameras mit 10 W Laserleistung zur Vermessung von Geländeszenen (Einsatz auf dem Gefechtsfeld).

Die Arbeiten lassen sich vom Ansatz her in 2 große Gruppen unterteilen:

I. Detektiere einfache kleine gleichartige Flächenstückchen und füge diese zur Beschreibung von Raumflächen zusammen (bottom up Ansatz); Kanten sind hierbei entartete Flächenstückchen.

II. Zerlege das Bild in einzelne Flächen und bestimme deren Merkmale; Kanten dienen hierbei als charakteristische Grenzen von Segmenten (Top Down Ansatz).

Von den meisten Autoren wird der Weg nach I) beschritten. Hier handelt es sich meist um Arbeiten der Wiedererkennung fester Objekte zur Roboter-Montagesteuerung. Entfernungsbilder werden durch Laser Triangulation gewonnen. Diese besitzen eine geringe Tiefenauflösung bei einer hohen Bildpunktanzahl. Das so erhaltene Bild wird im Fenster zu 3 x 3 bis 9 x 9 Bildpunkten zerlegt, die sich teilweise überlappen. Entweder durch statistische Tests /21/ oder durch Methoden der Optimierung wird untersucht, ob die Punkte dieser Fenster in einer

Ebene liegen, und welche Raumrichtung diese Ebenen einnehmen. Raumkanten werden ebenfalls durch eine Untersuchung innerhalb solcher Fenster detektiert. In einem weiteren Schritt (Merging) werden Flächenstückchen gleicher Normalrichtung zu Flächen zusammengefügt.

Beispielhaft sind hier die Arbeiten von

Cohen und Cayula /22/ :

Hier werden auch zylindrische und sphärische Flächenstückchen berücksichtigt.

Crawley /24/

Crawley definiert zu jedem Flächenstückchen ein Toleranzmaß in Normalen-, Positions- und Entfernungsrichtung.

Vemuri und Aggarwal /26/

Allisto /28/:

Allisto arbeitet ebenfalls nach obigem Verfahren, verwendet aber eine echte 3-D Entfernungskamera auf Laserbasis .

Oshima und Shirai /29/:

Diese Autoren gehen bereits einen Schritt weiter; zusätzlich zur Flächenfindung detektieren sie weitere Merkmale der Flächen wie Schwerpunkt, Innenradius, Außenradius, sowie relative geometrische Beziehungen der Flächen untereinander.

Gil /30/:

Gil wendet sich in seiner Arbeit einem Ausschnitt der Problemstellung zu, indem er Wege aufzeigt, die Kantenpunkte, die er aus einem Entfernungsbild extrahiert, mit einem SW-Bild der gleichen Szene zu besichern.

Bis auf Gils Arbeit befaßt sich keiner der Aufsätze mit der Analyse einer natürlichen oder Laborumgebung. Die Rechenzeiten sind, der stark unterschiedlichen Auflösung der Sensoren wegen, nicht vergleichbar.

Die Arbeiten nach II) sind meist neueren Datums.

Bogaert /31/:

Eine Randstellung nimmt hier Bogaert ein, der die Detektion und Segmentierung von Flächen mit Hilfe der 3-D Hough-Transformation durchführt.

Fan /32/:

Fan beschreitet folgende Strategie:

- Glätte die Bilddaten
- Berechne die Ableitung der Entfernungsdaten in 2 Richtungen
- Suche Nullstellen und Maxima (= Sprung und Dachkanten)
- arbeite mit mehreren Glättungswerten und überlagere die Kantenbilder
- Da Flächen durch Kanten begrenzt werden, lege durch alle, von Kanten umgebenen Bildpunkte, eine biquadratische Fläche und ermittle deren Gleichung

Fan verwendet Daten der ERIM-Entfernungskamera (126 x 126 Punkte) und Daten von Crawley (s.o. Triangulation). Die Auflösung in  $yz$ -Richtung ist mit der LSR.E. vergleichbar. Die Rechenzeiten bewegen sich im 2 bis 7 Minuten Bereich auf einer dedizierten LISP-Maschine.

Boyter u. Aggarwal /33/:

Boyter beschränkt sich in seinem Ansatz auf das Finden von ebenen Flächen, Kantenpunkten und Flächenkanten. Ausgangspunkt ist ein Entfernungsbild, abgespeichert in einer 2-dimensionalen Matrix (Zeilen y, Spalten z) aus Entfernungswerten x.

Strategie:

- Zerlege alle Zeilen in Geradenstücke  
(iterative endpoint curve fitting)
- Zerlege alle Spalten in gleicher Weise. Endpunkte von Geradenstückchen sind Kantenpunkte.
- \*- Greife ein Spalten- und ein Zeilengeradenstück heraus
- Lege die Ebene durch diese Streckenzüge
- Suche alle Streckenzüge, die diese Ebenengleichung erfüllen
- Gehe zu \* bis alle Streckenzüge zugeordnet sind
- Verbinde mit Hilfe der Hough-Transformation Kantenpunkte zu Objektkanten

Anschließend beschreibt Boyter einen Objekt-Matching-Algorithmus.

Eine umfassende Bibliographie zum Thema EBV findet sich bei /34/.

Bewertung der beschriebenen EBV Algorithmen:

Im Hinblick auf den Einsatz von EBV für die Daten der LSR.E. sind die "Bottom up" Algorithmen weniger brauchbar. Sie setzen eine hohe Bildpunktanzahl voraus, lassen aber auch unpräzise Entfernungsmessung zu. Beides trifft für die LSR.E. Daten nicht zu. Gekrümmte Flächen können in der industriellen Umgebung der MAKROBE kaum vor. Ansätze, die sich auf generalisierte Flächen (Sphären) stützen, sind für unsere Aufgaben nicht sinnvoll.

In Frage kommen deshalb primär nur die "Top down" Strategien nach Fan und Boyter, die sich, weniger durch ihr mathematisch geometrisches Vorgehen, als durch ihren pragmatischen Ansatz auszeichnen.

## 4.2. ENTFERNUNGS-BILDVERARBEITUNG IM SENSORRAHMEN ARGUS

In diesem Kapitel werde ich die realisierten EBV Algorithmen, sowie deren Anbindung an die RDB Argus, an die Hard- und Software der LSR.E. und die MAKROBE beschreiben.

Abb 4.1. zeigt einen Gesamtüberblick über die realisierten Funktionen.

Als logS wurden folgende Prozeduren und Programme erstellt:

eingang:

Liest von der LSR.E. kommende Meßdaten in RDB ein; dabei werden die Konturen des Fahrzeugs MAKROBE aus dem Bild ausgeblendet

einlesen:

Liest Daten aus der RDB zur Bildverarbeitung ein.

datenbankeintrag:

Schreibt Ergebnisse der Bildverarbeitung in RDB.

datgriff:

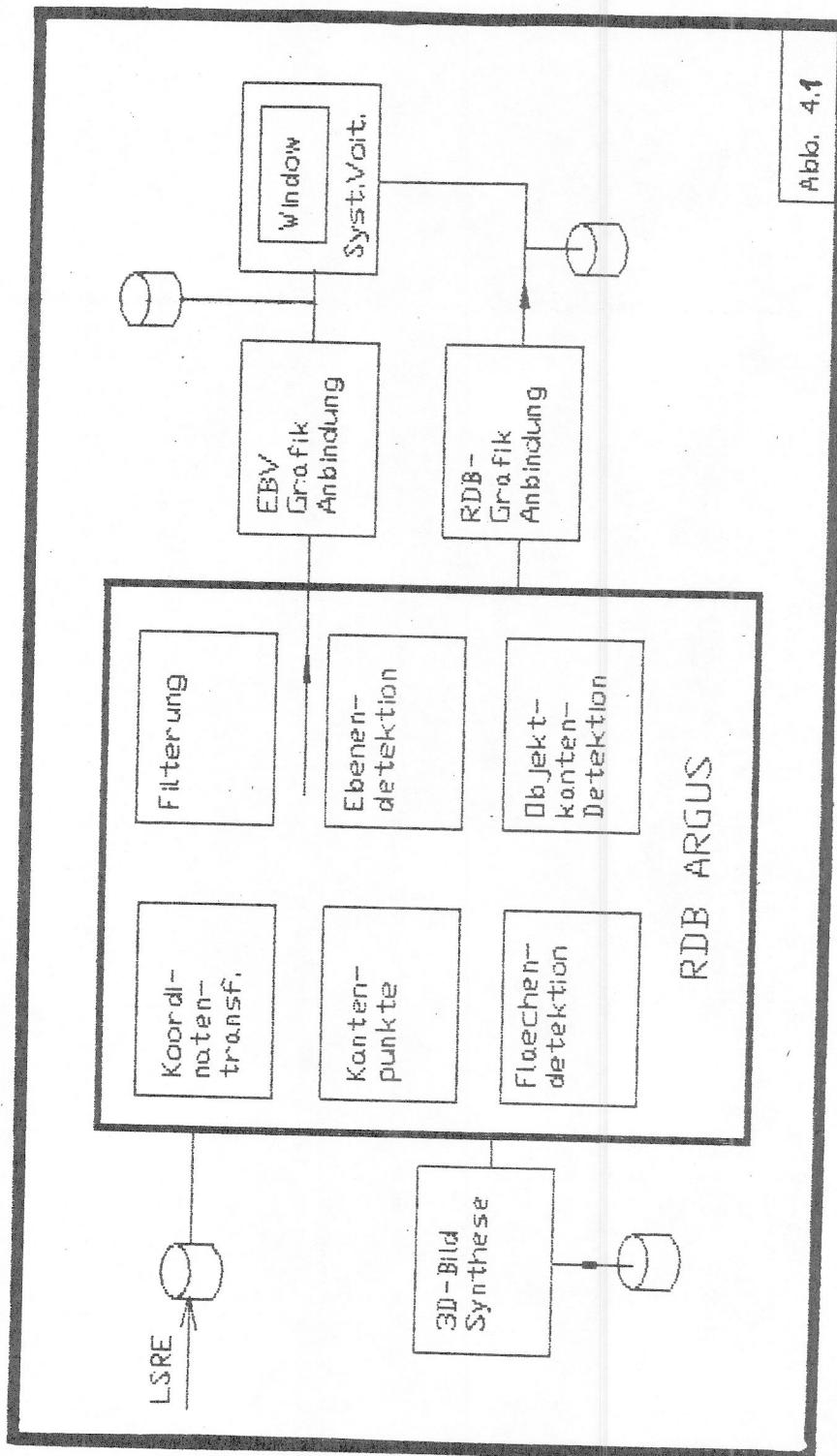
Erzeugt modifizierte Bilddatenfiles, um die Endergebnisse der EBV graphisch darstellen zu können (System Voitenleitner).

xyz :

Liest aus der Datenbank Rohdaten der LSR.E., transformiert sie in karthesische Koordinaten und speichert diese wieder in die RDB;

select:

Programmrahmen, der es erlaubt, mit jeder beliebigen Abfragebedingung auf die RDB ARGUS zuzugreifen und modifizierte Bilddatenfiles zur graphischen Darstellung dieser Daten zu erzeugen.



Abla. 4.1

findunbrokenzeile,

findunbrokenpalze:

Zeilen- und spaltenweise Rohdatenfilterung, sucht stetige Bereiche der Meßprofile

kantenpunkte:

Detektiert Bildpunkte, die auf Sprung- oder Dachkanten liegen.

plan:

Zerlegt das Bild in Ebenen und berechnet geometrischen Parameter dieser Ebenen.

teilflächen:

Zerlegt die Ebenen in Gebiete (Flächen).

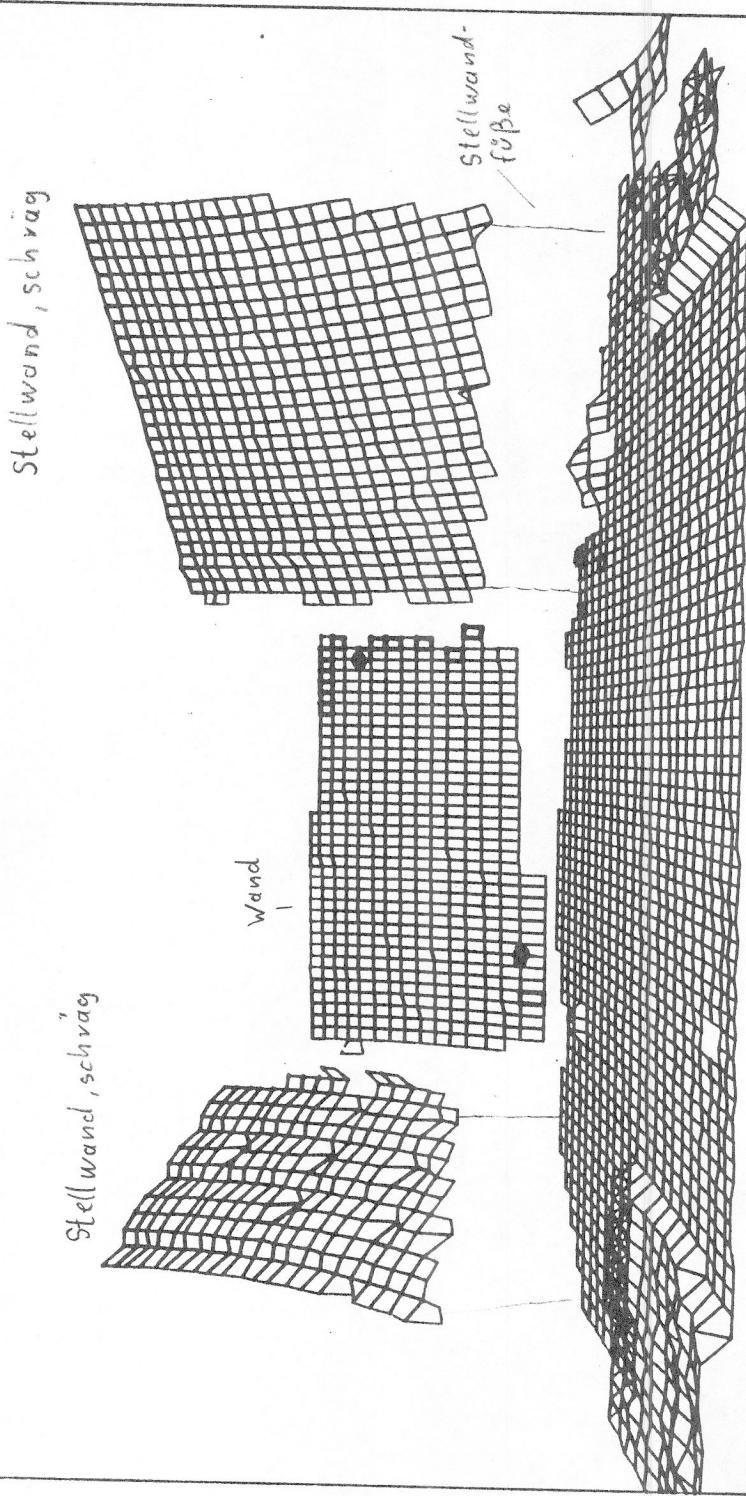
houghtransformation:

Faßt Kantenpunkte zu Objektkanten zusammen und berechnet deren Parameter.

Zusätzlich wurden weitere Komando- und Hilfsprozeduren entwickelt, die den Ablauf obiger Prozeduren steuern, die DB löschen und initialisieren, sowie Zwischen- und Endergebnisse der EBV visuell und textuell dokumentieren.

#### 4.3. ENTFERNUNGS BILDVERARBEITUNGS ALGORITHMEN

Da Bildverarbeitung nur anhand von Bildern zu erklären ist, werde ich den Aufbau und die Funktionsweise meiner EBV Schritte anhand einer Beispielsszene (s. Abb. 4.2), wie sie für einen Innenraum typisch ist, erläutern.



GRAPHISCHE DARSTELLUNG

Abb 4.2

#### 4.3.1. Vorfilterung der Meßdaten

In den Prozeduren `findunbrokenzeile` und `findunbrokenpalte` (s. Anhang 3; S 34 bis 42) werden die Entfernungsmeßwerte der LSR.E. in stückweise stetige Bereiche zerlegt. Die Operation arbeitet dabei auf einem Matrixförmigen-Bildfeld. Jede Matrixzeile aus 81 Punkten entspricht einer Meßzeile der LSR.E.. Die 41 Zeilen der Matrix stehen für die 41 Zeilen des LSR.E. Bildes. Jedes Matrixelement besteht aus einem Record, in dem die Entfernungsdaten, die karthesischen Koordinaten des Meßpunktes, die Qualität des Meßsignals, sowie Hilfsgrößen gespeichert sind. Diese Matrix lässt sich auf einem Textterminal darstellen, indem jedem Matrixelement genau ein ASCII-Zeichen zugeordnet wird.

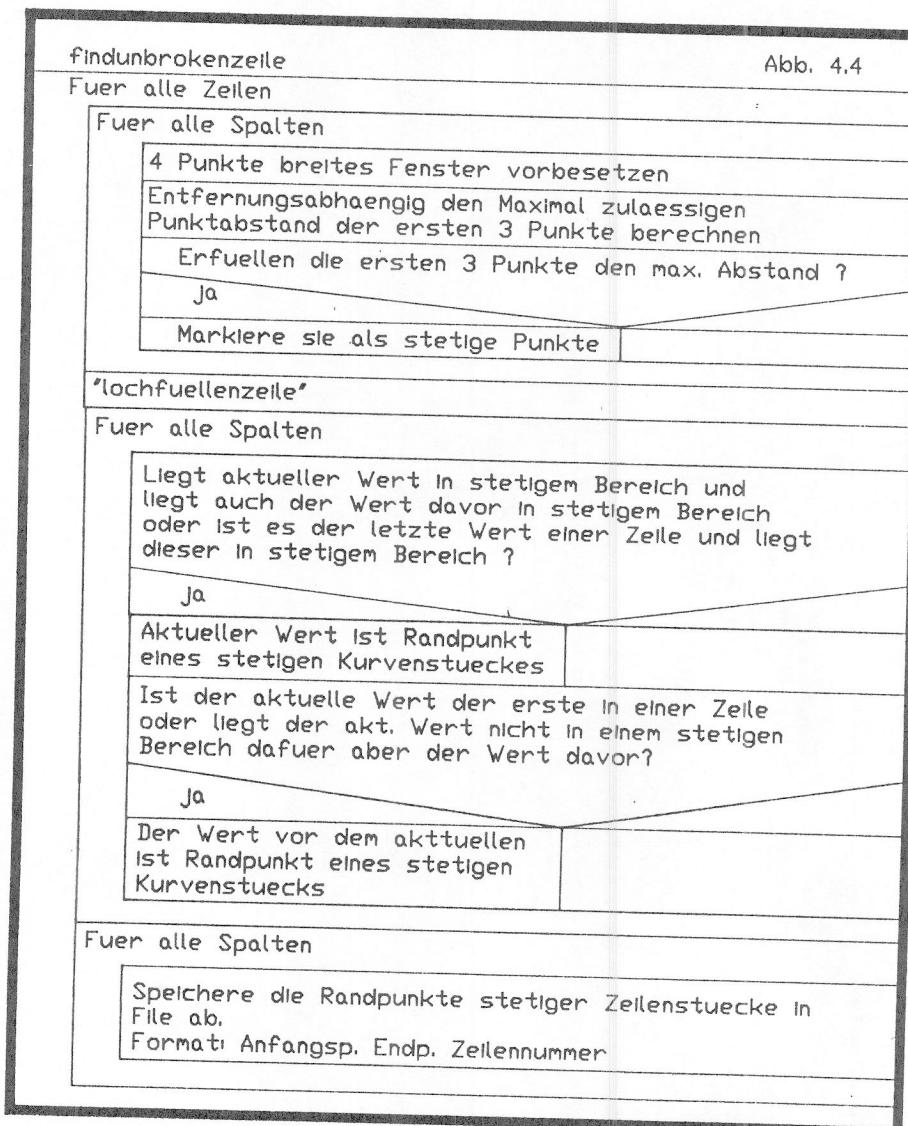
Da im polaren Koordinatensystem der LSR.E. ein funktioneller Zusammenhang zwischen Azimut- bzw. Elevationswinkel und der Entfernung  $r$  gegeben ist, arbeite ich hier mit polaren Werten. Nach der Transformation in karthésische Koordinaten treten Überdeckungen von Originalkanten auf, die das Verfahren nur komplizieren würden. Abb. 4.4 zeigt ein Struktogramm der Prozedur `findunbrokenzeile`; sie arbeitet folgendermaßen:

- Schiebe ein Fenster, das 4 Bildpunkte breit ist, zeilenweise über die Matrix.
- Bewerte entfernungsabhängig (polare Entfernung  $r$ ) den Abstand des 1. zum 2. und des 2. zum 3. Punkt.
- Treffe so eine Aussage, ob diese Punkte annähernd stetig angeordnet sind.
- Ist eine Zeile abgearbeitet, so wird diese Zeile ein zweitesmal durchlaufen.
- Berücksichtige nun zusätzlich den ersten und letzten Punkt der Zeile; prüfe, ob auch der Punkt vorher stetig war oder nicht; markiere so Anfang und Ende stetiger Kurvenzüge.

- In einem 3. Durchlauf werden nun die Anfangs- und Endpunkte stetiger Zeilenstücke in einem File abgespeichert.

Spaltenweise wird absolut analog zu obigem Schema verfahren.

Abb. 4.4



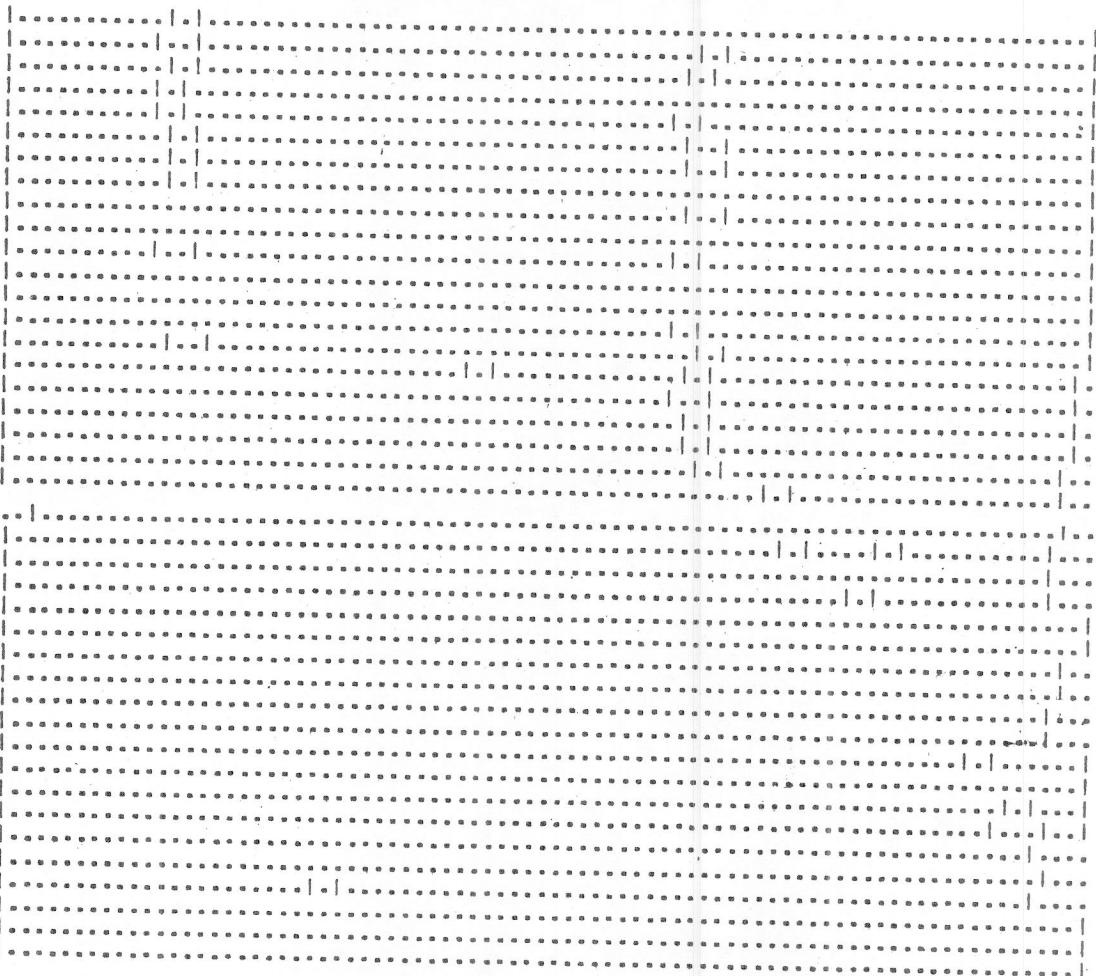


Abb. 4.6

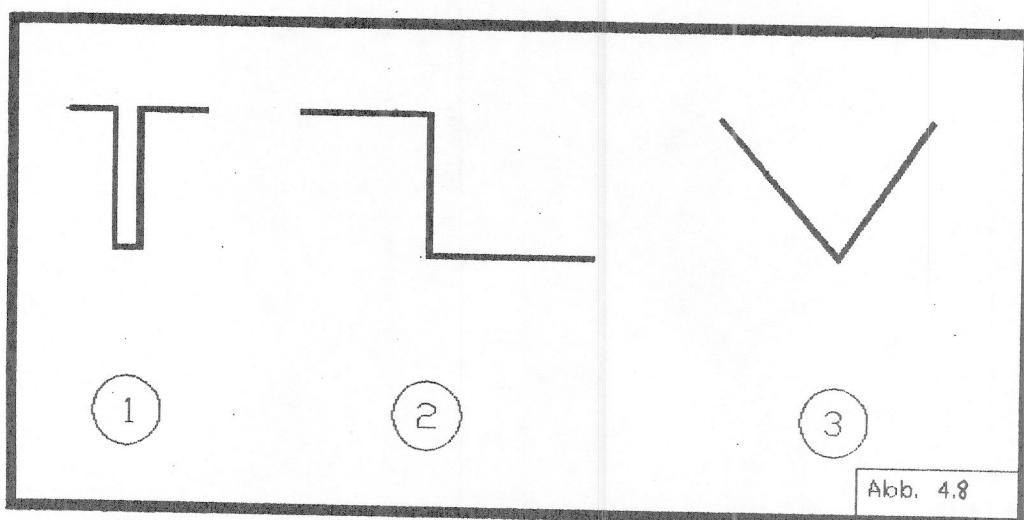


Abb. 4.6 zeigt eine Pseudographik, die das Ergebnis obiger Prozedur anhand den Daten der Beispielsszene aus Abb. 4.2 veranschaulicht. Teil A der Abbildung zeigt die stetigen Meßprofile, markiert durch den Buchstaben Z im Bildfeld, Teil B zeigt die jeweiligen Grenzen dieser Kurvenzüge; Abb. 4.7 zeigt selbiges Verfahren für die Spalten des Bildfeldes.

#### 4.3.2. Detektion von Kanten

Bevor man versucht, in einem Entfernungsbild Kanten zu detektieren, ist es notwendig zu analysieren, welche Kantenkonstellationen in einem Entfernungsbild auftreten können. Während in einem SW-Bild allgemein die sprungartige Veränderung der Leuchtdichte als Kante definiert werden kann, sind beim Raumbild 3 Fälle zu unterscheiden (s. Abb. 4.8).

- 1) Spike-Kanten werden durch dünne Hindernisse (Rohre o. ä.) im Raum erzeugt.
- 2) Jump oder Folding Kanten: sie entstehen dort, wo sich 2 Hindernisse, vom Beobachter aus gesehen, überdecken.
- 3) Roof (Dach-) Kanten: sie treten an vielflächigen Objekten auf.



Der allgemein naheliegende und gängige Weg, Kantenpunkte zu detektieren, liegt in der Verwendung differenzierender Operationen. Kanten vom Typ 1) und 2) werden dadurch gegenüber dem restlichen Bild stark verstärkt. Die einfache Operation bewirkt aber, daß das Meßrauschen im Bild ebenfalls stark zunimmt. Zur Detektion von Kanten vom Typ 3) wäre gar ein 2fach differenzierender Operator notwendig. Beim stark diskreten Bild der LSR.E. (im Vergleich zum SW-Bild sehr wenig Meßpunkte) ist dies nicht mehr sinnvoll. Eine Operation, die sich nur auf die Differenz zweier benachbarter Bildpunkte stützt, ist hier zu störanfällig.

Ein anderer Weg, Kanten zu detektieren, ist die Ermittlung stetiger Bildbereiche; Kanten sind dann solche Bildbereiche, die nicht stetig sind. Da die Anzahl der flächigen Bildbereiche wohl meist größer ist als die Zahl der Kanten, kann sich dieses Verfahren auf mehr Meßpunkte abstützen und ist so wesentlich störsicherer. Es braucht so auch nicht zwischen den einzelnen Kantentypen unterschieden zu werden.

Da ich in meiner Grundannahme von einem Weltmodell ausgehe, das aus Flächen aufgebaut ist, kann ich statt stetiger, direkt gerade bzw. ebene Bildbereiche suchen. Hierfür verwende ich das Verfahren des "iterativ endpoint fit" (s. /35/ ). Das Verfahren kann, wie oben, bildzeilen- und bildspaltenweise abgearbeitet werden.

Ich erläutere den Algorithmus anhand eines karthesischen Entfernungs- schnittes:

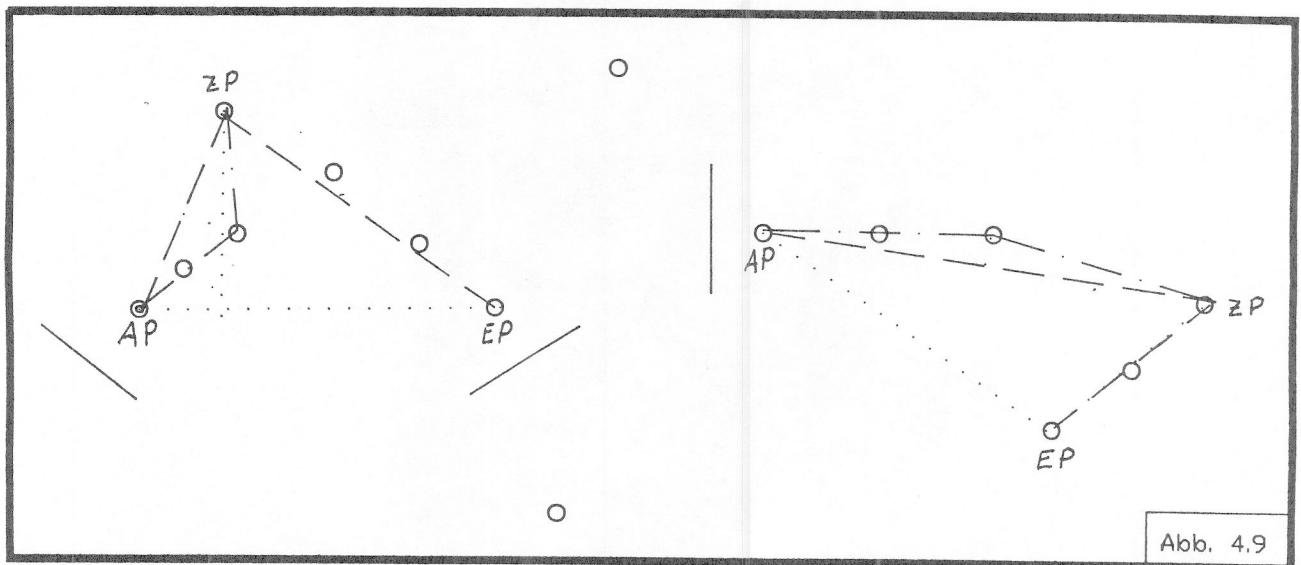


Abb. 4.9 zeigt eine schematisierte Meßzeile; eingegrenzt sind die stückweise stetigen (nicht verrauschten) Meßbereiche (Markierung durch Schritt 4.3.1.).

- \*- Lege nun eine Strecke vom Anfang zum Endpunkt der Kurvenstücke.
- Berechne den Bildpunkt, der von dieser Strecke den minimalen Abstand hat.
- Dieser Punkt wird nun neuer End- bzw. Startpunkt.
- Zerlege also die Kurve an dieser Stelle in 2 neue Strecken (AP - ZP) (ZP - EP) (s. Abb. 4.9.).
- Verfahre mit diesen beiden neuen Streckenzügen genauso wie bei \*.
- Setze das Verfahren solange fort, bis der maximale Abstand eines Meßpunktes von der ideal gedachten Strecke (AP' - EP') unter einer vorher definierten Grenze liegt.

Abb. 4.10 zeigt das Endergebnis. Die Anfangs- und Endpunkte markieren Kanten.

Dieses Verfahren wird nun für alle Zeilen und Spalten des Bildfeldes durchgeführt. Die Struktur des Verfahrens legt eine rekursive Vorgehensweise nahe. Jeder Schritt \* benötigt als Eingangsparameter einen Anfangs- und Endpunkt, und liefert schließlich ein Paar von Anfangs- und Endpunkten (s. Abb. 4.11).

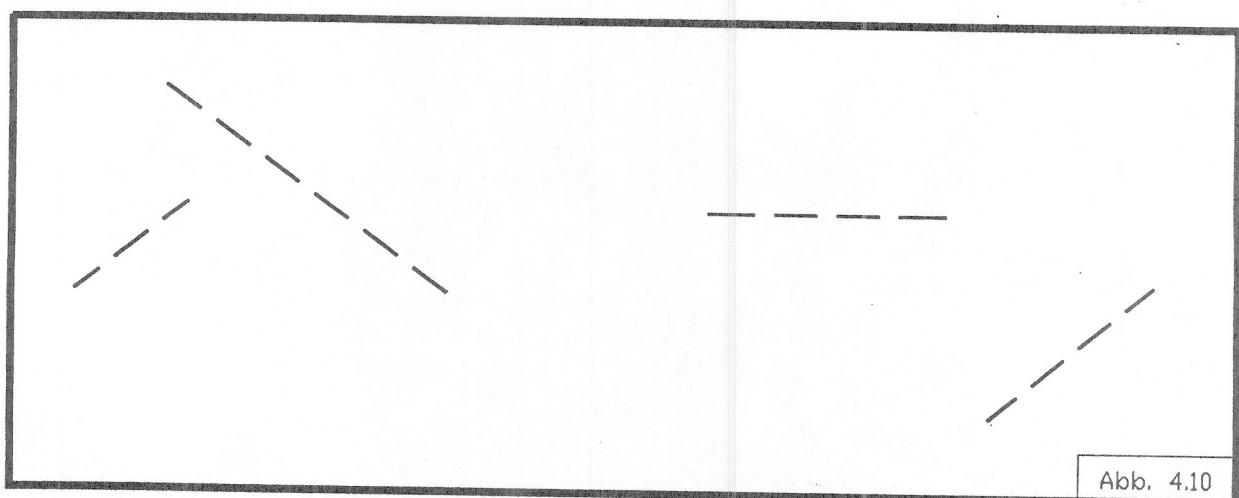


Abb. 4.10

Die adaequate Datenstruktur für die Rekursion ist (aus Abb. 4.11 direkt ersichtlich) ein Binärbaum, der in seinen Knoten die Zwischenergebnisse aufnimmt, die Endergebnisse liegen dann inhärent sortiert in den Blättern des Baumes.

So ist es möglich, je nach erforderlicher Auflösung auf unterschiedlich grobe und robuste, oder feine und störanfällige Zwischenergebnisse zugreifen. Die Aufbau- und Ausleseverfahren für den Binärbaum sind hier, ebenfalls rekursiv programmiert.

Die Länge der Streckenzüge und ihre maximale Abweichung von einer Ideallinie fallen als Nebenprodukt dieses Verfahrens mit an.

Abbildung 4.11

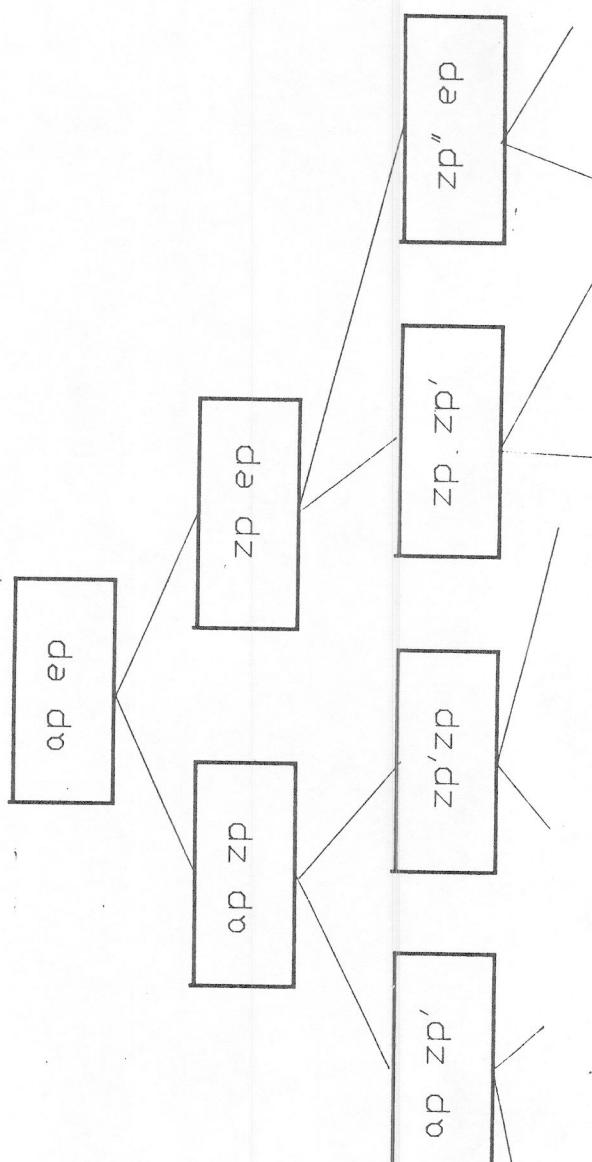


Abb. 4.12

**"kantenpunkte"**

Fuer alle Zeilen und Spalten

Lese die Abbruchbedingungen ein  
Initialisiere Ein- und Ausgabefiles

Fuer alle stetigen Kurvenzuege

Lese Anfangspunkt (ap) und Endp. (ep) eines  
stetigen Kurvenzuges ein

Besetze Arbeitsfeld mit allen Karthesischen Messwerten  
zwischen Anfangs- u. Endpunkt

Ist der stetige Kurvenzug mind. 5 Messpunkte lang?

Ja

Starte Prozedur "zwischenpunkte (ap, ep)"

Berechne die Entfernung von ap nach ep

Fuer alle Zwischenpunkte zp

Berechne Entfernung ap-zp

Berechne den Winkel zwischen  
ap-zp und ap-ep

Berechne den Abstand zp zu  
(ap, ep)

Abstand > als alle bisher er-  
rechneten Abstaende?

Ja

maxabstand = Abstand

Fuege ap, ep, maxabstand, (ap, ep) an Binaer-  
baum an

maxabstand > Abbruchgrenze ?

Ja

Starte zwischenpunkte (ap, zp)

Starte zwischenpunkte (zp, ep)

Speichere  
Spaltennummer  
ap, ep,maxabstand,  
(ap, ep) In File

Kopiere Blaetter des Binaerbaums in File

Abb. 4.12 zeigt ein Struktogramm des durch die Prozedur Kantenpunkte (s. Anhang 3; S. 42 bis S. 50) realisierten Verfahrens; es gliedert sich in folgende Schritte:

- Besetze die Prozedur mit Abbruchbedingung und Anfangs- und Endpunkt eines stetigen Spalten- oder Zeilenprofiles vor.
- rufe die eigentliche "iterative endpoint fit" Prozedur auf
  - \* zerlege die Strecke in 2 neue (AP - EP) Paare
  - speichere diese Punkte im Binärbaum
  - rufe \* mit (AP/ZP) auf
  - rufe \* mit (ZP/EP) auf
  - speichere bei Rekursionsende das letzte Kantenpunktpaar in File
- speichere die Elemente des Binärbaums, die die gewünschte Auflösung besitzen, in File.

Abb. 4.13. zeigt in einer Graphik, die nach dem Durchlauf des Programmes erstellt wird, die auf Abb. 4.2. angewandte Kantenpunkt-suche. Dargestellt sind in a) die zeilenweise gefundenen Anfangs- und Endpunkte. Punkte, die sowohl Anfangs- als auch Endpunkte sind, sind mit \$ markiert.

Abb. 4.14 zeigt dasselbe für die spaltenweise Operation. Alle Kantenpunkte ergeben sich durch Überlagerung von Abb. 4.13 und Abb. 4.14. Abb. 4.29 zeigt die detektierten Kantenpunkte blau markiert als 3D-Grafik.

Abb. 4.14

Abb. 4.15

#### 4.3.3. Ebenendetektion

Im Allgemeinen wird in der EBV zur Flächendetektion folgendes Verfahren angewandt:

- Zerlege das Bild in sich teilweise überlappende rechteckige Flächen (patches) zu  $3 \times 3$  oder  $4 \times 4$  Bildpunkten.
- Versuche, auf diese wenigen Stützstellen eine Ebene zu legen, und ermittle die geometrischen Parameter dieser Ebene.
- Versuche, alle patches mit ähnlicher Raumrichtung zu großen Ebenen zu integrieren.

Ebenso wie der Differentialoperator zur Kantenfindung, stützt sich dieses Verfahren auf sehr wenige Meßpunkte. Bei der Ebenenfindung ist außerdem der Abstand der Ebenenstützstellen möglichst groß zu wählen, und die Auswirkung von Entfernungsmeßfehlern gering zu halten.

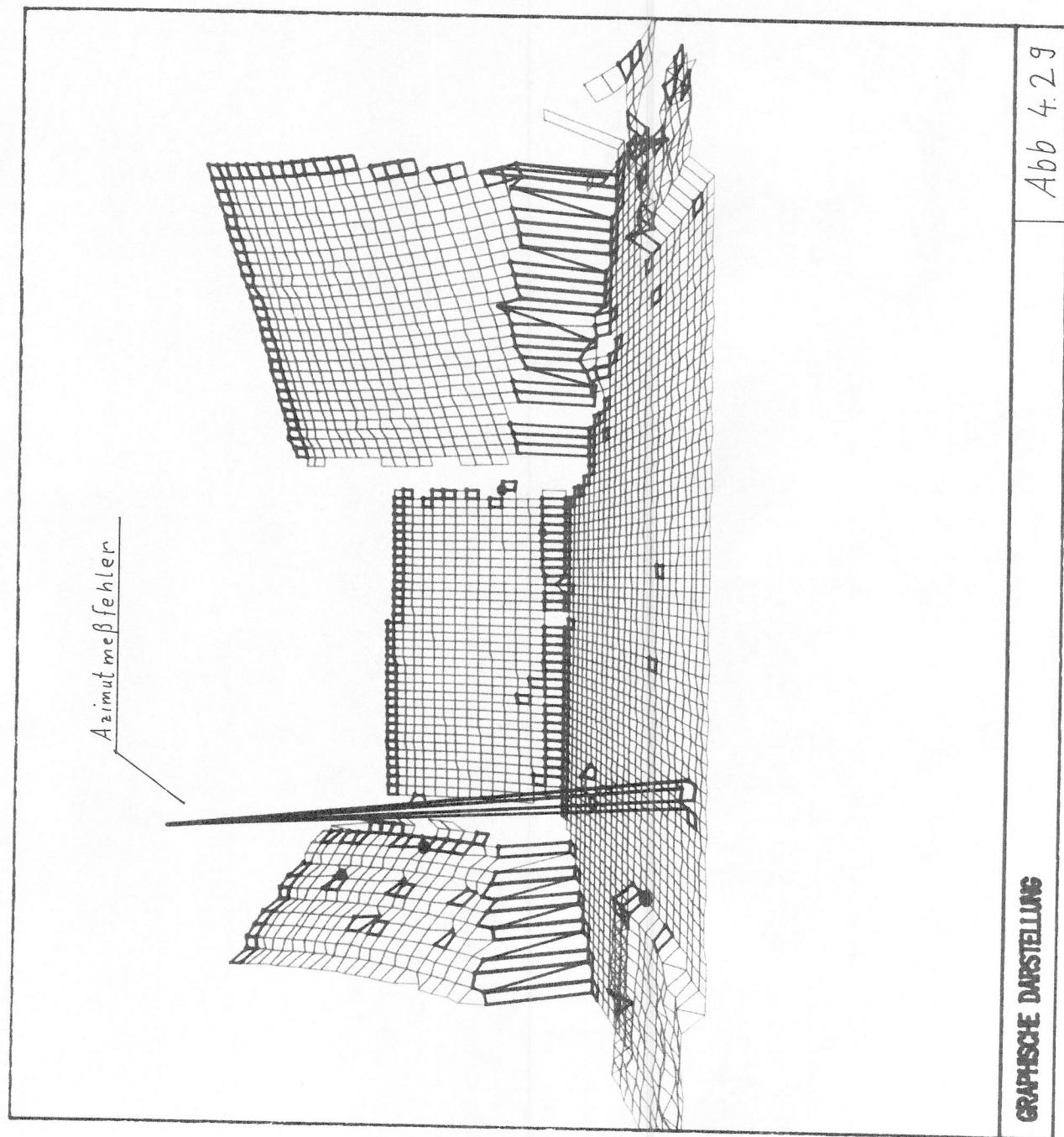
Da ich aus dem Algorithmus der Kantenfindung bereits gerade Streckenzüge erhalten habe, verwende ich dieses Zwischenergebnis und gehe einen globaleren Weg der Ebenenfindung.

Voraussetzungen:

In meiner flächigen Umwelt verlaufen gerade Streckenzüge sicher auf Ebenen. Da ich waagrechte und senkrechte Streckenzüge detektiert habe, schneiden sich mindestens 2 dieser Strecken in einer Ebene. Kantenpunkte, d. h. die Anfangs- und Endpunkte von Strecken, begrenzen die Ebenen.

Ziel:

Finde alle koplanaren Streckenzüge; Fasse alle Meßpunkte, die auf diesen Streckenzügen und somit in einer Ebene liegen zusammen. Berechne die geometrischen Parameter dieser Ebene.



Das Ziel unter obigen Voraussetzungen erreiche ich mit folgender Strategie (s. a. Abb. 4.15):

- \*1. Suche einen waagrechten und einen senkrechten Streckenzug heraus, prüfe ob sich diese kreuzen.
2. Ermittle das Volumen, das die 4 Endpunkte der Strecken aufspannen.
3. Berechne das minimale Volumen, das beide Streckenzüge aufspannen, wenn sie rechtwinklig in einer Ebene liegen aus der Lage der beiden Strecken, sowie der maximalen Abweichung ihrer Streckenpunkte von der geometrischen Idealstrecke (s. 4.3.2.).
4. Ist das (Volumen aus Punkt 2)  $\approx K * (\text{Volumen aus Punkt 3})$ , so sind beide Strecken koplanar.
5. Berechne mit den nun gegebenen 4 Stützstellen 4 Ebenengleichungen, jede stütze sich auf jeweils 3 der 4 Stützstellen (= den 4 Seitenflächen eines Tetraeders).
6. Suche die Ebene, bei der der "Nichtstützpunkt" den minimalen Abstand zur Ebene hat.
7. Suche alle anderen Streckenzüge durch, ob sie die Ebenengleichung (bis auf ein Toleranzmaß) erfüllen.
8. Speichere alle Strecken, die zu einer Ebene gehören, mit den geometrischen Parametern ab.
9. Gehe zu 1 bis alle Streckenzüge abgearbeitet sind.

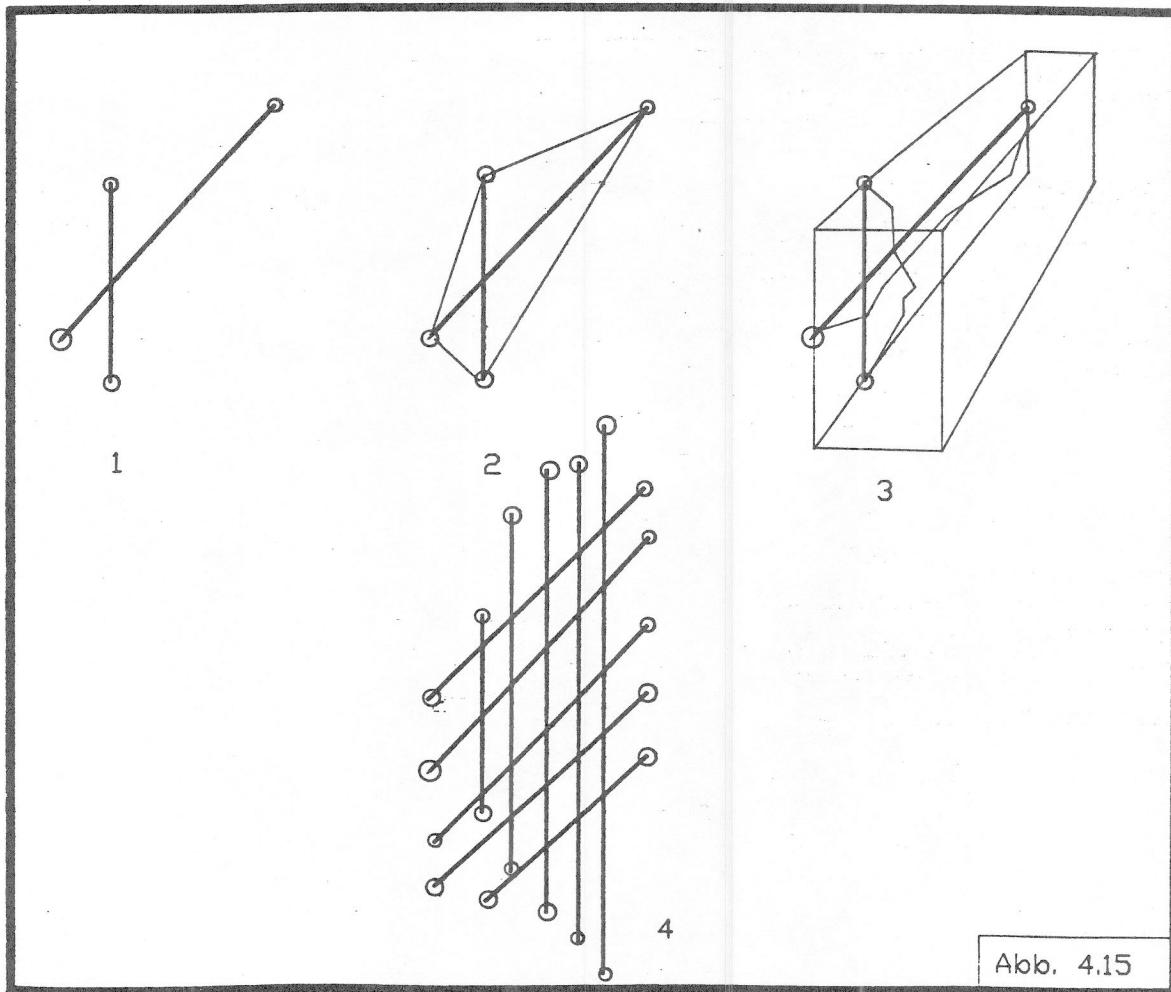


Abb. 4.16 zeigt ein Struktogramm der Prozedur plan (s. Anhang.3; S. 51 bis S 56).

Im weiteren werde ich die wichtigsten Details des Verfahrens erläutern.

Zu 1.:

Die Bedingung, von 2 sich im polaren Koordinatensystem kreuzenden Streckenstücken auszugehen, ist hilfreich und notwendig, wie sich an folgendem Beispiel leicht zeigen lässt (s. Abb. 4.17).

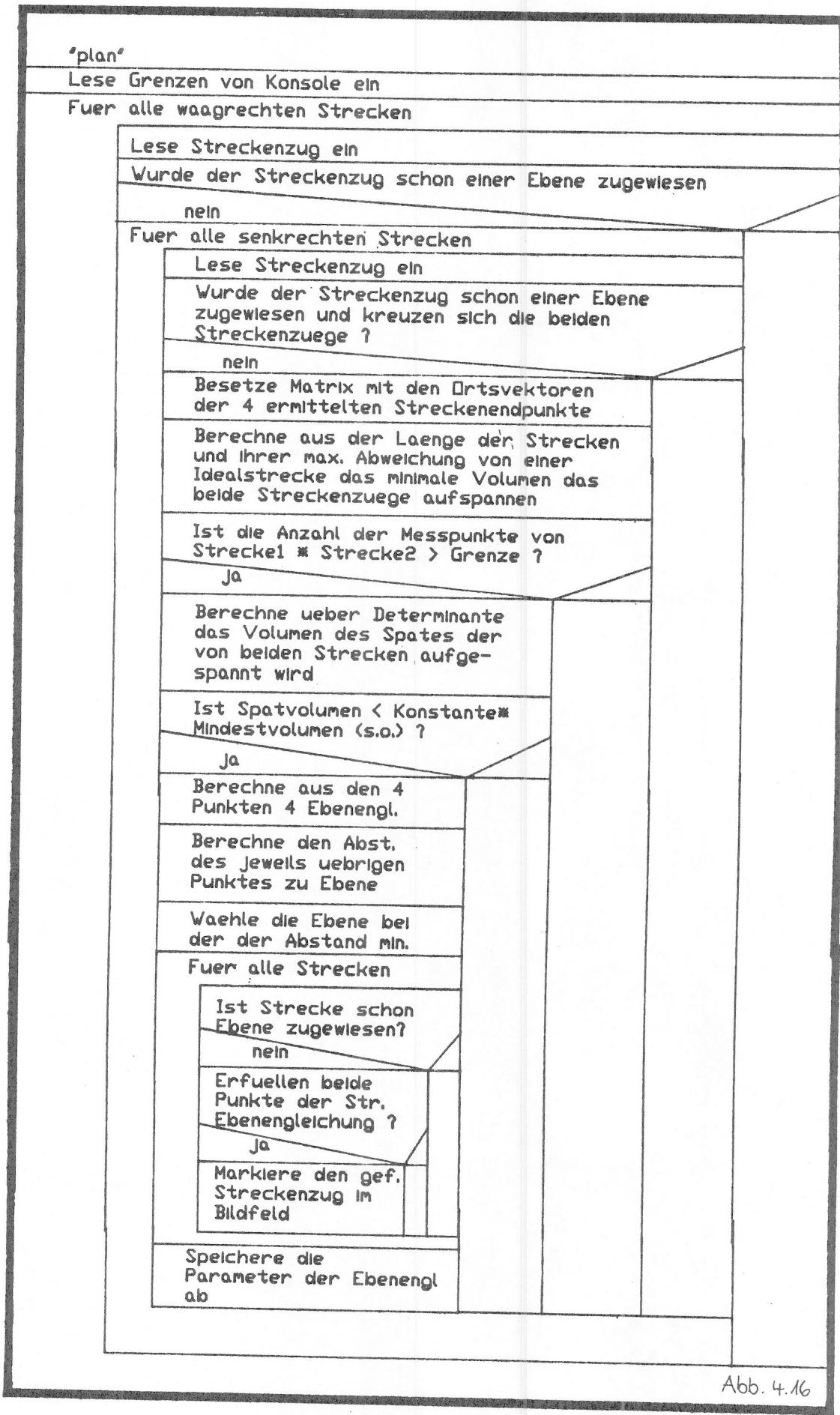
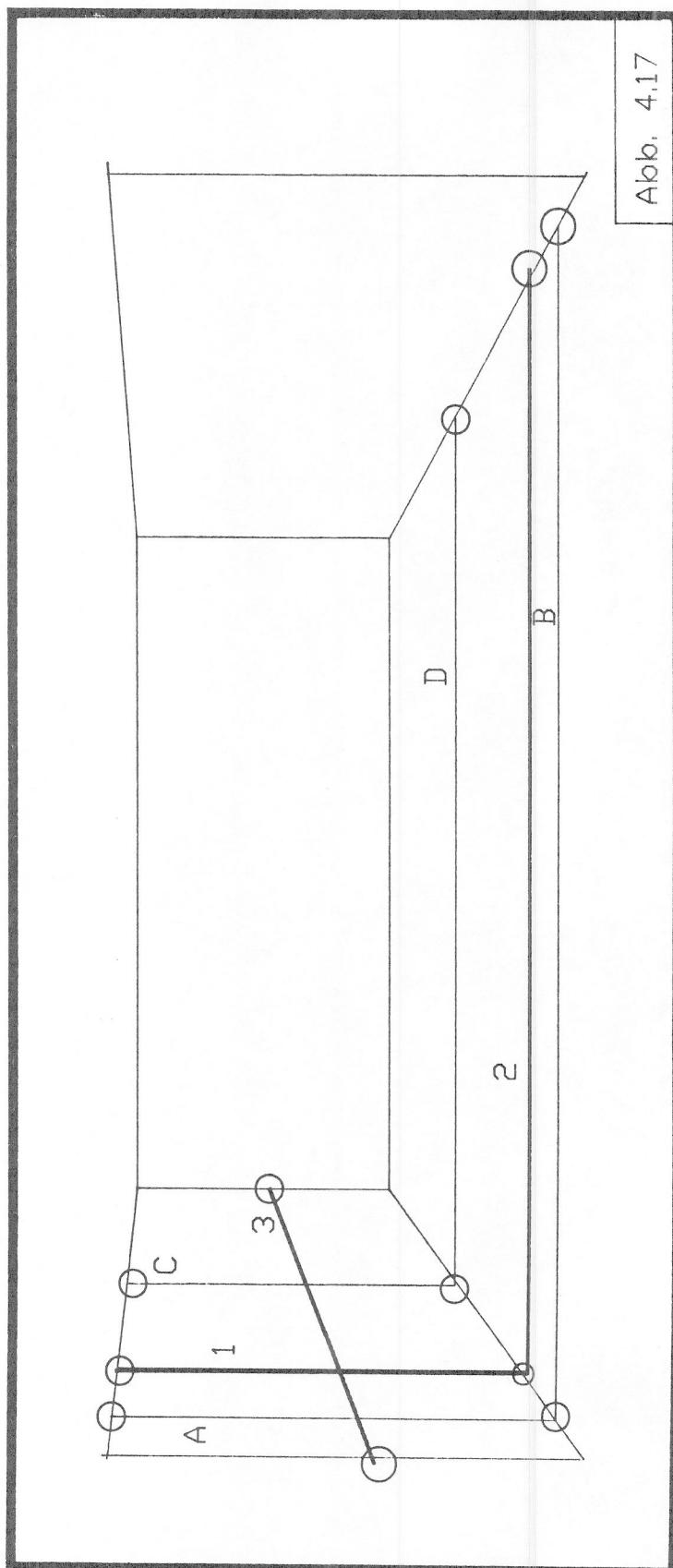


Abb. 4.16

Abb. 4.17



Wie man sieht, liegen die Strecken 1 und 2 in einer Ebene, der aber keine Ebene im Raum entspricht. Da es sich bei den Daten der LSR.E. um Meßdaten handelt, sind diese verrauscht. Scharfe Entscheidungskriterien (gehört zu Ebene: ja/nein) sind deshalb nicht möglich. Jede solche Entscheidung muß Abweichungen im Rahmen der LSR.E. Meßgenauigkeit zulassen. Folge dieser notwendigen Strategie ist, daß auch die Elemente A B und C D einer Ebene zugeordnet werden, die durch die Strecken 1 und 2 aufgespannt, aber im Bild nicht vorhanden ist. Sich kreuzende Strecken werden in der polaren Projektion des Bildfeldes detektiert; die so ausgewählten Strecken sind nicht notwendig echt kreuzend im Sinne, daß sich Strecke A und Strecke B in einem Raumpunkt kreuzen (Verdeckungen). Dies ist jedoch meist der Fall. Da es sich hier nur um die geschickte Auswahl eines Streckentupels als Startwert für das eigentliche Verfahren handelt, sind 100 % hinreichende Randbedingungen hier nicht erforderlich. Im obigen Beispieldfall wird nun das Paar 1/3 ausgewählt, es liegt in einer Ebene, der im weiteren A und C zugeordnet werden, die falsche Konfiguration von oben ist nicht mehr möglich.

zu 2.:

Voraussetzung ist, daß die beiden untersuchten Streckenzüge bzw. deren Eckpunkte genau in einer Ebene liegen; sie spannen dann das Volumen 0 auf. Da dies, bedingt durch die realen Verhältnisse, in der Umwelt und im Meßgerät nie der Fall sein wird, führe ich als Maßzahl für die Koplanarität der beiden Strecken das Volumen ein, das durch die Eckpunkte der Strecken aufgespannt wird. Seien  $\vec{b_1}, \vec{b_2}, \vec{b_3}, \vec{b_4}$  die Ortsvektoren der Streckeneckpunkte. Das von ihnen aufgespannte Spat-Volumen ist dann:

$$V_{\text{Spat}} = \det(\vec{b_1}, \vec{b_2}, \vec{b_3}, \vec{b_4})$$

Der in den 2 Strecken aufgespannte Tetraeder hat das Volumen

$$V_{\text{Tetr.}} = 1/6 V_{\text{Spat}}$$

Zu 3.:

Die in 4.2.3. ermittelten Streckenzüge sind charakterisiert durch:

delta: maximale Abweichung von Idealstrecke  
 l : Länge des Streckenzuges

Unter der Annahme, daß zwei zueinander senkrechte Streckenzüge (A und B) tatsächlich in einer Ebene liegen, ergibt sich das von ihnen maximal aufgespannte Volumen zu (s. Abb. 4.18.)

$$V_E = l_A * l_B * (\delta_A + \delta_B)$$

zu 4.:

Ist das in 2 ermittelte Volumen  $V_{\text{spat}}$  größtenordnungsmäßig im Bereich des Volumens  $V_E$ , so liegen beide Streckenzüge in einer Ebene. Es gilt

$$(S_A, S_B) \text{ sind koplanar} \Leftrightarrow V_{\text{spat}} < k V_E$$

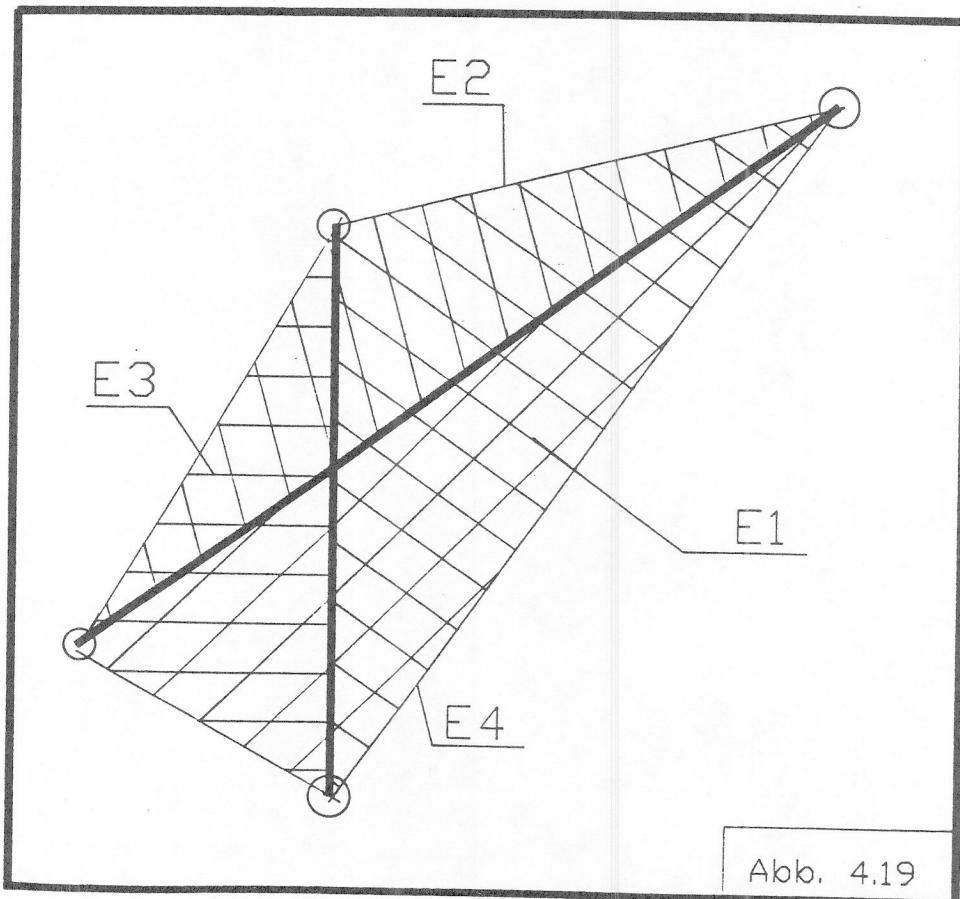
$k$  ist eine Konstante, die während des Programmlaufes eingegeben werden kann. Es hat sich gezeigt, daß eine  $k$  von ca. 10....15 sinnvoll ist (alle Volumina gemessen in Kubikzentimeter).

$$\text{Es gilt: } \frac{V_{\text{spat}}}{V_E} < 10;$$

Die Zahl  $k$  entspricht ungefähr dem kürzesten räumlichen Abstand der beiden Streckenzüge voneinander. Die Berechnung über Determinanten ersetzt trigonometrische Berechnungen durch schnelle Integer Multiplikationen und Additionen.

zu 5. und 6.:

Nachdem zwei Streckenzüge als koplanar detektiert worden sind, ist nun die entsprechende Ebenengleichung zu bestimmen. Durch 4 Raumpunkte lassen sich 4 Ebenen legen; und zwar kommen die Ebenen auf den Seitenflächen des Tetraeders zu liegen, den die 4 Raumpunkte aufspannen (s. Abb. 4.19).



Zu den 4 Punkten werden nun die 4 möglichen Ebenengleichungen, gestützt auf jeweils 3 Punkte, berechnet.

Es gilt:

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = 0 \Rightarrow$$

$$Ax + By + Cz + D = 0$$

Für den jeweils verbleibenden 4. Punkt wird der Abstand zur ermittelten Ebene errechnet.

$$d_a = \frac{|Ax_4 + By_4 + Cz_4 + D|}{\sqrt{A^2 + B^2 + C^2}}$$

Es wird die Ebenengleichung mit ihren Parametern (A, B, C, D) ausgewählt, bei der  $d_a$  ein Minimum annimmt.

Zu 7.:

Es werden nun alle Streckenzüge, die noch keiner Ebene zugeordnet wurden, herausgesucht. Als Kriterium, ob eine Strecke in der gegebenen Ebene liegt, dient die Summe der Abstände der Strecken-Anfangs- und Endpunkte von der Ebene.

Es gilt:

$$d = \left| \frac{Ax_{ap} + By_{ap} + Cz_{ap} + D}{\sqrt{A^2 + B^2 + C^2}} \right| + \left| \frac{Ax_{ep} + By_{ep} + Cz_{ep} + D}{\sqrt{A^2 + B^2 + C^2}} \right|$$

$d \leq d_{max}$

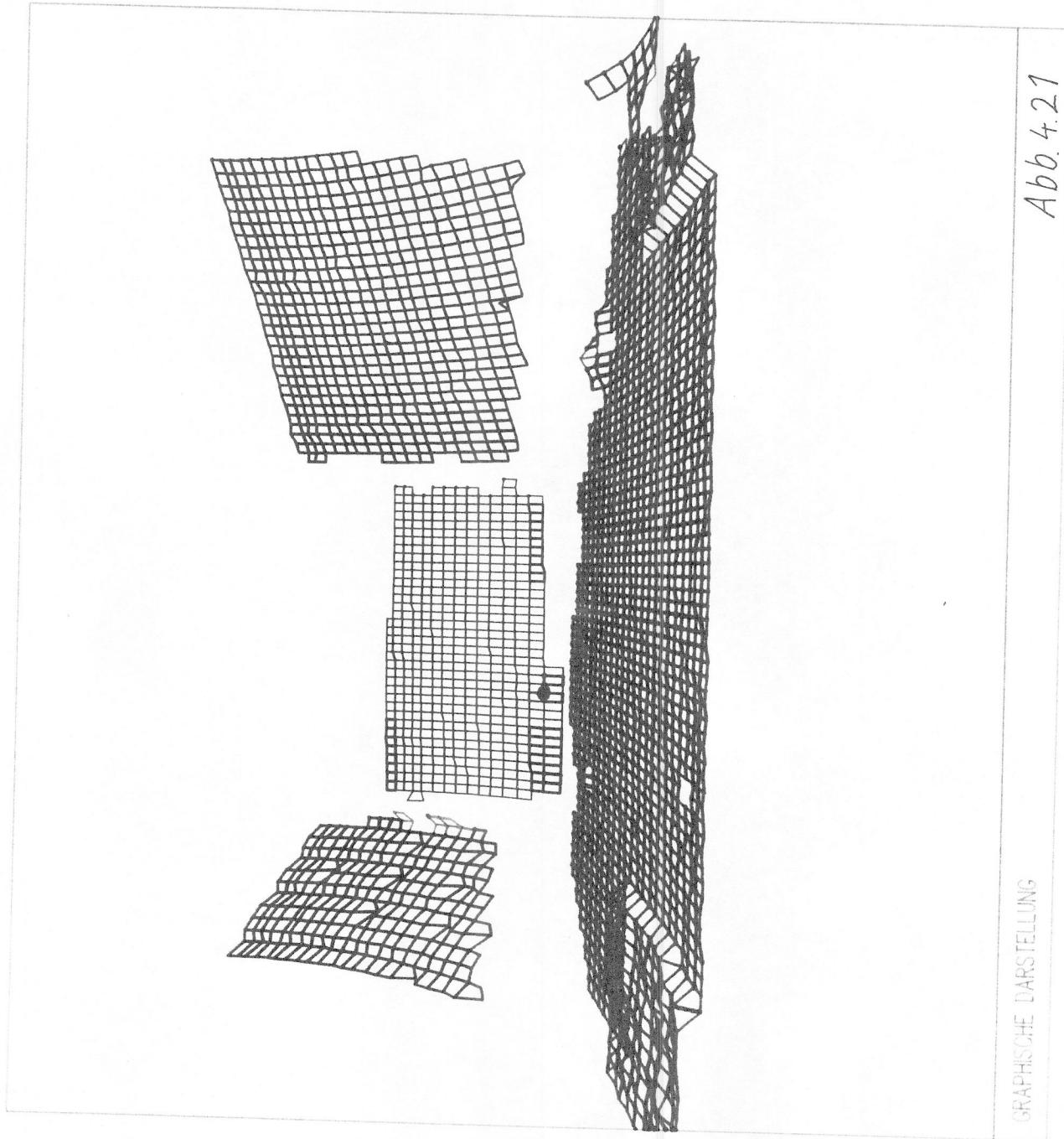


Abb. 4.21

GRAPHISCHE DARSTELLUNG

Die Größe  $d_{\max}$  wird vor Programmablauf eingegeben, ein Wert zwischen  $(2 * 15) - (2 * 30)$  erweist sich hier als brauchbar.

zu 8. und 9.:

Die Arbeitsschritte 1 bis 7 wiederholen sich nun solange, bis alle möglichen Ebenen detektiert und diesen alle Streckenzüge zugewiesen wurden.

Im Bildfeld werden sodann alle Kantenpunkte markiert, die zu Streckenzügen gehören, die einer Ebene zugewiesen wurden. Abbildung 4.20 zeigt das Ergebnis für das schon in 4.3.1. und 4.3.2. verwendete Testbild. Weiterhin wird jeder gefundenen Ebene eine Ebenennummer zugeordnet; alle Meßpunkte, die zu einer Ebene gehören, werden mit dieser Nummer markiert.

Alle Punkte, die zu einer Ebene gehören, werden im Array of file "fläche" mit ihrer Punktnummer (0.....3320) abgespeichert.

Alle Kantenpunkte, die zu einer Ebene gehören, werden im Array of file "ergebnis" abgespeichert.

Die geometrischen Parameter der Ebenen (A, B, C, D) werden im File "ebenenparameter" abgelegt.. Abb. 4.21 zeigt das Ergebnis als 3-D Graphik.

#### 4.3.4. Flächendetektion

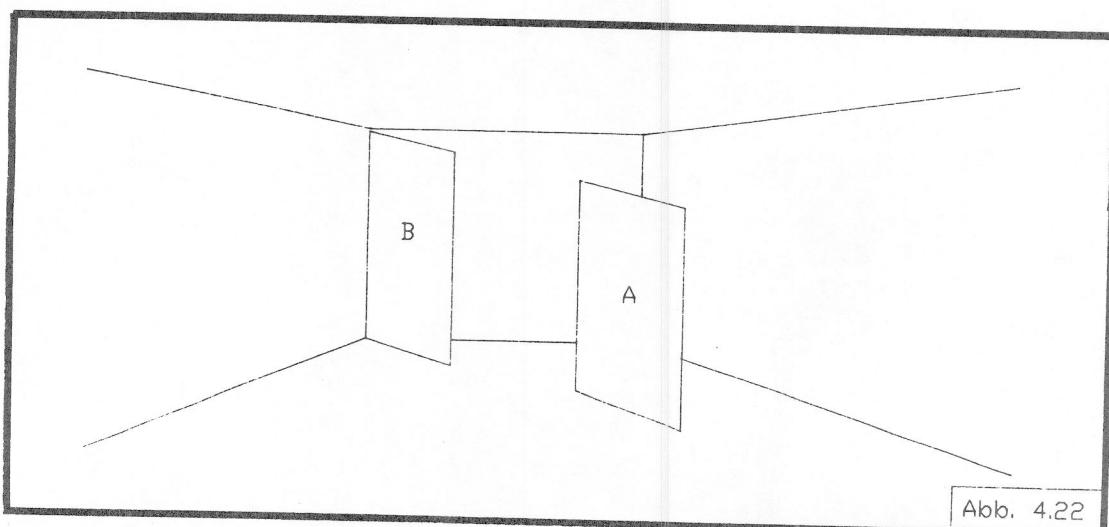


Abb. 4.22 zeigt eine Szene, bei der 2 verschiedene Hindernisse (A und B) in einer Ebene auftreten. Eine EBV, die nur Ebenen detektiert, versagt hier; erst durch eine Flächendetektion können die Hindernisse als getrennte Objekte erkannt werden.

Da aus 4.2.1. bekannt ist, welche Bildpunkte in einer Ebene liegen und auch die geometrischen Parameter dieser Ebene vorliegen, können weitere EBV-Algorithmen auf 2dimensionale Binärbildverfahren aufbauen. Die oben detektierte Ebene bildet dabei den 2-D Arbeitsbereich, die Entfernungswerte der Bildpunkte stecken zum einen in den Geometriedaten der Ebene, zum anderen in den 2-D Koordinaten der Bildpunkte in der Ebene.

Einen solchen 2-D Binäralgorithmus zur Zerlegung und Segmentierung zusammenhängender Gebiete in einer Ebene, beschreiben Lumia, Shapiro und Zuniga in /36/. Deren Verfahren wende ich in leicht abgewandelter Form bei der Flächensegmentierung an.

Ich gehe folgendermaßen vor:

1. Markiere alle Bildpunkte, die zu genau einer Ebene gehören, in der Bildmatrix ( $41 * 81$  Punkte) mit einer 1; alle übrigen Punkte seien 0 (s. Abb. 4.23 ).

	0.....80
0	000000000000.....
.	000000000000.....
.	00111000000.....
.	01111100001.....
.	01111000011.....
40	01100001111.....
	.....
	.....

2. Schiebe das in Abb 4.24 gezeigte Fenster zeilen- und spaltenweise über die Matrix. Die 3 Fensterfelder seien mit A, B und C markiert.

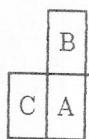


Abb. 4.24

3. Ordne nach folgender Strategie dem Bildpunkt A, auf dem das Fensterfeld gerade sitzt (A-Pixel), eine Ordinalzahl (Label) zu:

$(A = 1) \wedge (B = 0) \wedge (C = 0)$       A-Pixel bekommt das nächste noch freie Pixel zugeordnet.

$(A = 1) \wedge (B = 0) \wedge (C = 1)$       A-Pixel bekommt das Label von C-Pixel

$(A = 1) \wedge (B = 1) \wedge (C = 0)$       A-Pixel bekommt das Label von B-Pixel

$(A = 1) \wedge (B = 1) \wedge (C = 1)$       A-Pixel bekommt Label min (B-Label C-Label), gleichzeitig wird abgespeichert, daß die Label an der Stelle B und C gleichwertig sind.

4. In einem 2. Durchgang durch das Bildfeld werden gleichwertige Label durch das niedrigerwertige ersetzt.

Abb. 4.25 zeigt das Ergebnis der Clusterung von Abb. 4.23 nach dem ersten, Abb. 4.26 das Ergebnis nach dem 2. Durchgang.

#### Äquivalenzbedingungen

```

00000000000000000000000000
000000000000000000000000...
001110000000000000000000...
02111100003.....(2 = 1) (1 = 2)
02111000043.....(2 = 1) (4 = 3) (1 = 2) (3 = 4)
02100005543.....(2 = 1) (4 = 5) (3 = 4) (5 = 4)
.

```

Abb. 4.25

```

000000000000.....
000000000000.....
001110000000.....
01111100003......
01111000033......
01100003333......
.

```

Abb. 4.26

5. Nun werden die Label durch lückenlos aufsteigende Nummern ersetzt  
(s. Abb. 4.27.)

```

000000000000.....
000000000000.....
001110000000.....
01111100002......
01111000022......
01100002222......
.

```

Abb. 4.27

6. Speichere das Ergebnis ab; besetze das Bildfeld mit den Punkten der nächsten Ebene vor und gehe zu 1).

Abb. 4.28. zeigt ein Struktogramm des Clusteralgorithmus. Der Programmtext findet sich in Anhang 3; S 65 bis S 68.

Näher zu erläutern ist die Prozedur des Zusammenfassen und Sortieren gleichwertiger Label (im weiteren L.).

Bei obigem Verfahren treten 1...n verschiedene, für ein Cluster mehrere gleichwertige Label auf.

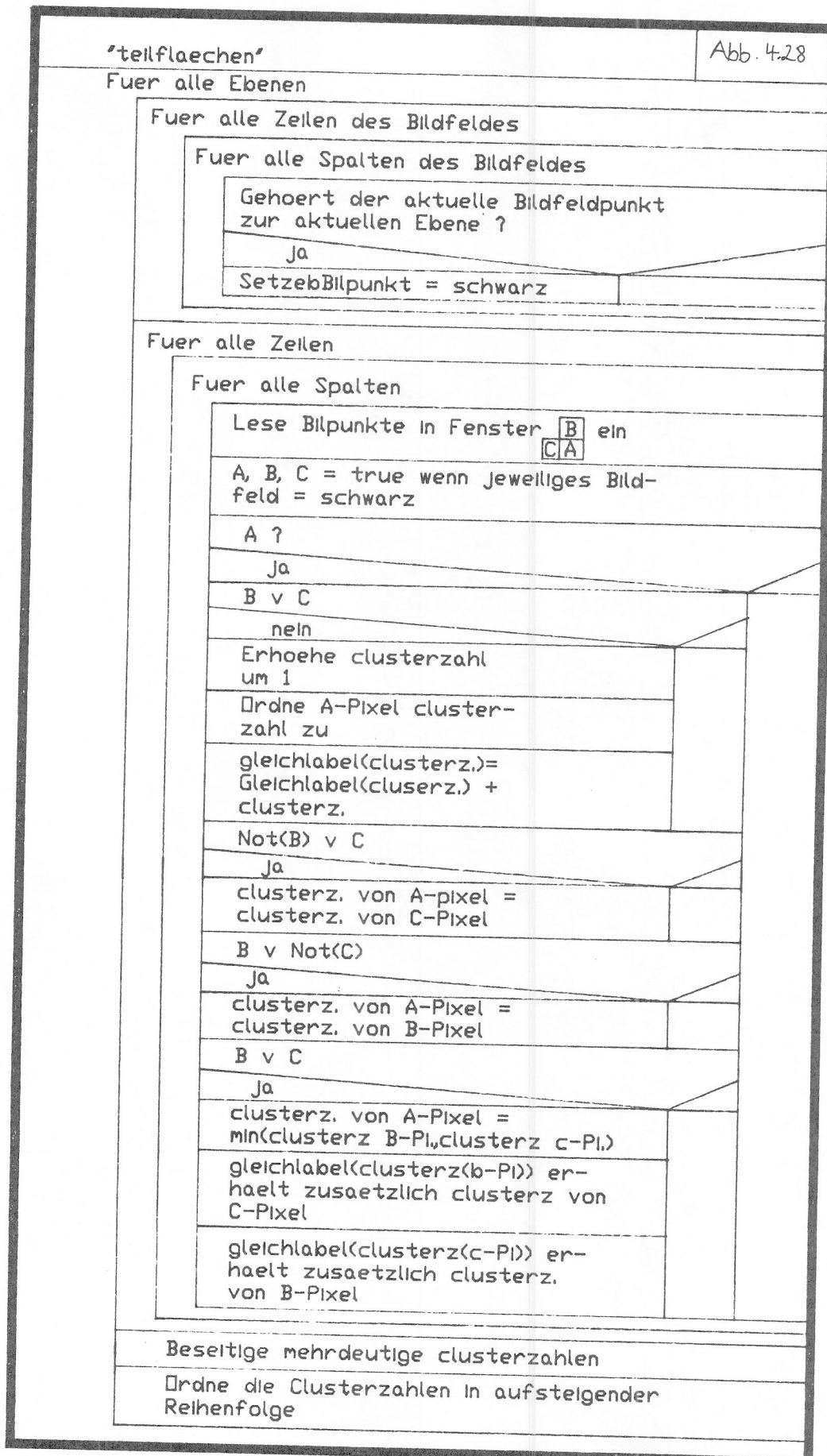
Es gilt:

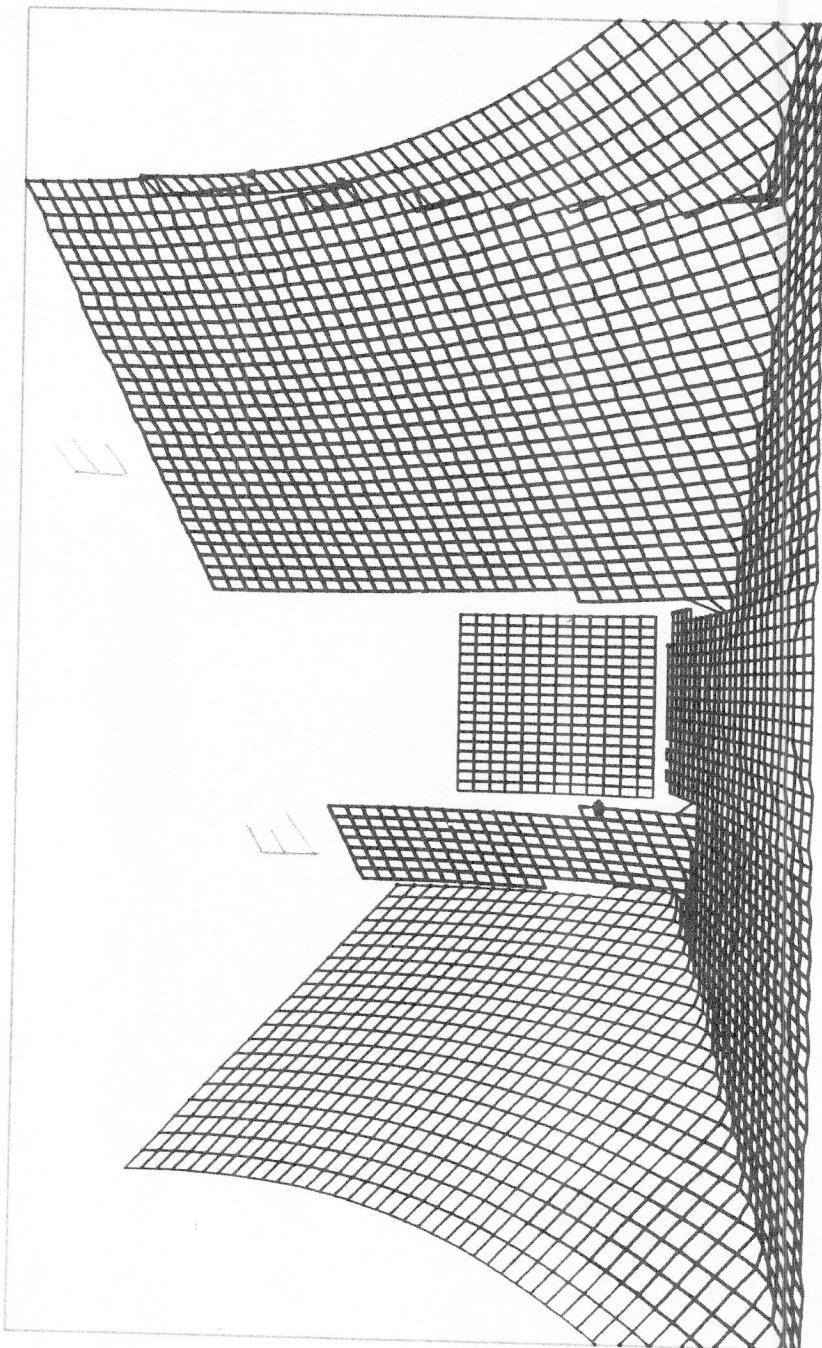
1	{	$l_1, l_2, \dots, l_n$	}
:	{	$l_2, \dots$	}
.	{	...	}
n	{	...	, $l_n$ }

Abgespeichert werden diese Label in einem eindimensionalen Feld von Mengen der Dimension k. Jedes Label erhält hierin eine Menge zugeordnet, in der es selbst, sowie alle gleichwertigen L. zusammengefasst sind. Die 1. Menge des Feldes enthält dabei alle L., die dem 1. L. gleichwertig sind, die 2. Menge alle L., die dem 2. L. gleichwertig sind u.s.w...

Nach folgendem Verfahren werden mehrdeutige L. eliminiert, sowie eine aufsteigende Reihenfolge von Flächennummern gewährleistet.

1. Greife die k-te L-Menge des Mengenfeldes heraus.
2. Gehe alle Elemente des Befehlfeldes durch, in dem das Label von jedem Pixel gespeichert ist und überprüfe, ob das aktuelle L. in der L-Menge gespeichert ist.
3. Wenn ja, weise dem Pixel das neue Label k zu.
4. Sperre das aktuelle Pixel für weitere Zugriffe.
5. Erhöhe k um 1 und gehe zu 1).





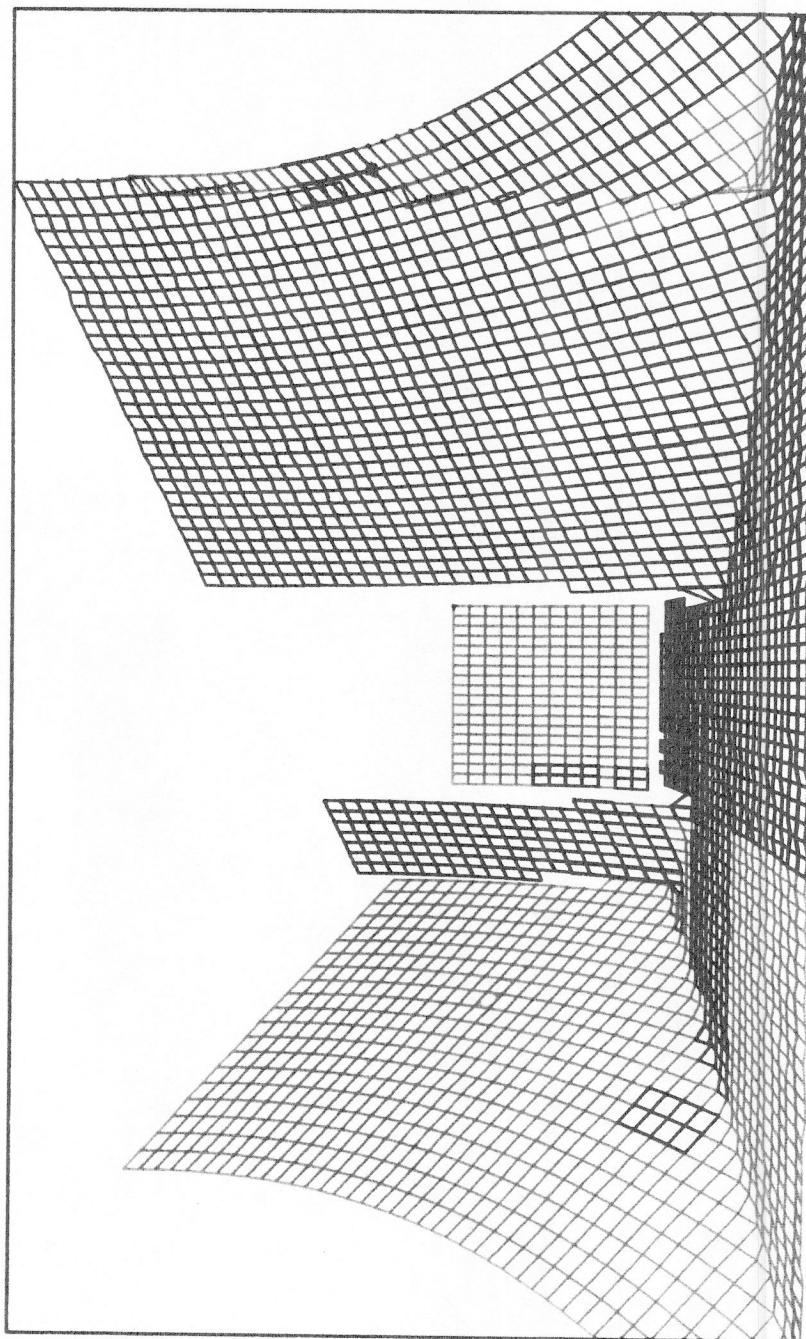
GRAPHISCHE DARSTELLUNG

Abb 4.30

Abb 4.31

Teilflaechenzahl :

Abb 4.32



GRAPHISCHE DARSTELLUNG

Abb 4.33

Abb. 4.30 zeigt eine Originalszene; die mit E bezeichneten Flächen liegen in einer Ebene.

Abb. 4.31 zeigt das Ergebnis der Ebenendetektion als Pseudographik,  
 Abb. 4.32 zeigt auf gleiche Art die Flächenzerlegung von Ebene E.  
 Abb. 4.33 zeigt farbig markiert die Flächen F 1 und F 2 in E.

#### 4.3.5. Zusammenfassung von Kantenpunkten zu Objektkanten

Bei der Zusammenfassung von Kantenpunkten zu Objektkanten sind wiederum globale und lokale Verfahren bekannt.

Ein Beispiel für lokales Vorgehen sind die Suchverfahren. Hier sucht man die lokale Umgebung eines Kantenpunktes A ab, bis der nächste Punkt B gefunden wird; man berechnet die Richtung AB und sucht in dieser Richtung weiter.

Dieses Verfahren reagiert empfindlich auf verrauschte Signale und fehlende Punkte in einer Kantenlinie.

##### 4.3.5.1. Kantendetektion mit Hilfe der Houghtransformation

Entsprechend meiner Strategie, global arbeitenden Verfahren den Vorzug zu geben, habe ich ein Verfahren realisiert, das auf der Houghtransformation fußt.

Die Houghtransformation (zum ersten Mal 1962 von P. V. C. Hough beschrieben) bildet (wie jede mathematische Transformation) Originalpunkte in einem Bildbereich ab.

Seien  $P(x, y)$  Punkte im Originalbereich

$$P(x, y) \xrightarrow{H} x \cos \vartheta + y \sin \vartheta - p = 0;$$

$\vartheta$  läufe von  $0..2\pi$

Das heißt, ein Originalpunkt, aus  $P$ , beschrieben durch seine kartesischen Koordinaten, wird im Bildbereich auf einen sinus-Kurvenzug abgebildet, wenn  $(p, \vartheta)$  die orthogonalen Koordinaten des Bildbereiches sind.  $p$  und  $\vartheta$  entsprechen den beiden Parametern der Heßschen Geradenbeschreibung.

Abb. 4.34 zeigt ein Beispiel nach (35/).

Läuft  $\vartheta$  von  $0....$  bis  $2\pi$ , so kann dies gedeutet werden, als drehe man eine Gerade, die durch den Punkt  $P$  läuft, und stelle ihre Heßschen Parameter in einem rechtwinkeligen Koordinatensystem dar. (Andere Koordinatensysteme und Beschreibungsformen sind möglich, z. B.:

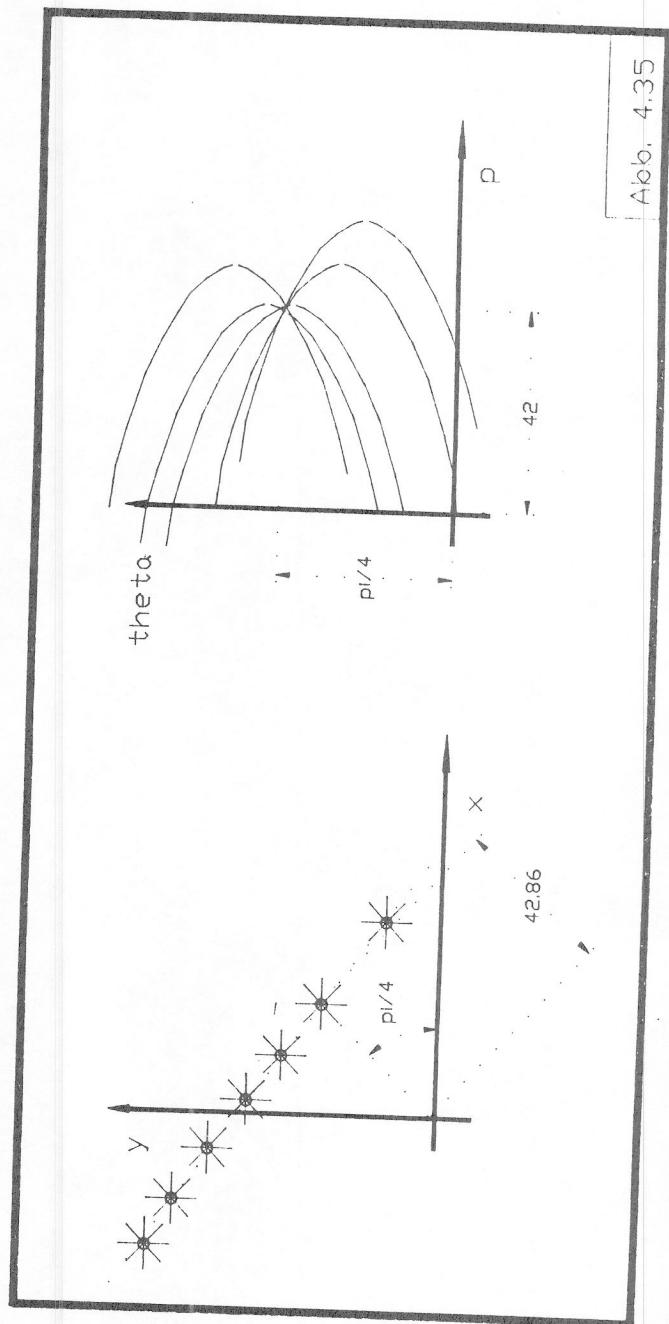
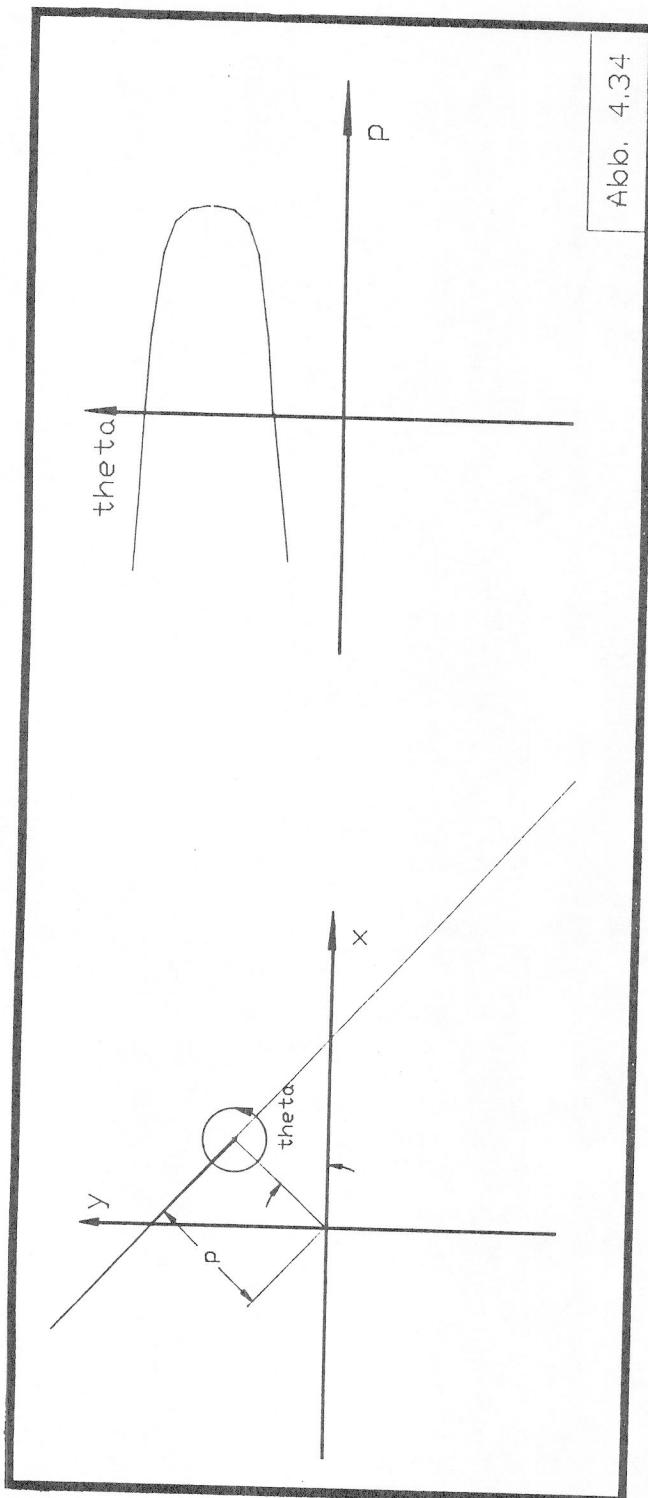
$$P(x, y) \xrightarrow{H} ax + b - y = 0; \quad m \in [0, \infty]$$

Bildet man nun mehrere Punkte aus dem xy Originalbereich, die auf einer Geraden liegen, in die H-Ebene ab, so stellt man fest, daß sich die entstehenden Sinuskurven genau in einem Bereich verdichten (s. Abb. 4.35).

Vergleicht man die  $p, \vartheta$ -Parameter im Hough Bereich mit den Parametern der Gerade, auf der die Originalpunkte liegen, so gilt:

$P_o = P_H$
$\vartheta_o = \vartheta_H$

Diskretisiert man dieses Verfahren zu endlich vielen  $\vartheta_i$ , woraus endlich viele  $p_i$  resultieren, so entsteht ein diskreter H-Bereich (s. Abb. 4.36).



Läßt man  $\nu_i$  nun von  $\nu_0 \nu_1 \dots \nu_j$  laufen und errechnet dazu nach der Transformationsgleichung  $\xrightarrow{H}$  die gerundeten  $p_0 p_1 \dots p_n$  so wird jeder Originalpunkt auf n "H-Zellen" abgebildet. Ordnet man jeder dieser H-Zellen einen Zähler zu und inkrementiert diesen, sobald einer von insgesamt j Originalpunkten aus  $P(xy)$  dorthin abgebildet wird, so ist nach der Houghtransformation jede Akkuzelle mit einer Zahl zwischen 0 und  $j * n$  belegt. Die zum Zählermaximum gehörenden Koordinaten  $\nu_i$ ,  $p_i$  beschreiben eine Gerade, auf der alle Originalpunkte, abgesehen von Diskretisierungsfehlern, liegen. Der Abstand der Originalpunkte voneinander ist hierbei unerheblich. Gegen Abweichungen der Originalpunkte von einer Idealgerade ist dieses Verfahren sehr unempfindlich (siehe /37/). Ein Nachteil ist der relative große Aufwand. Als Aufwandmaß gilt:

$$\text{Aufwand} = n * l * j$$

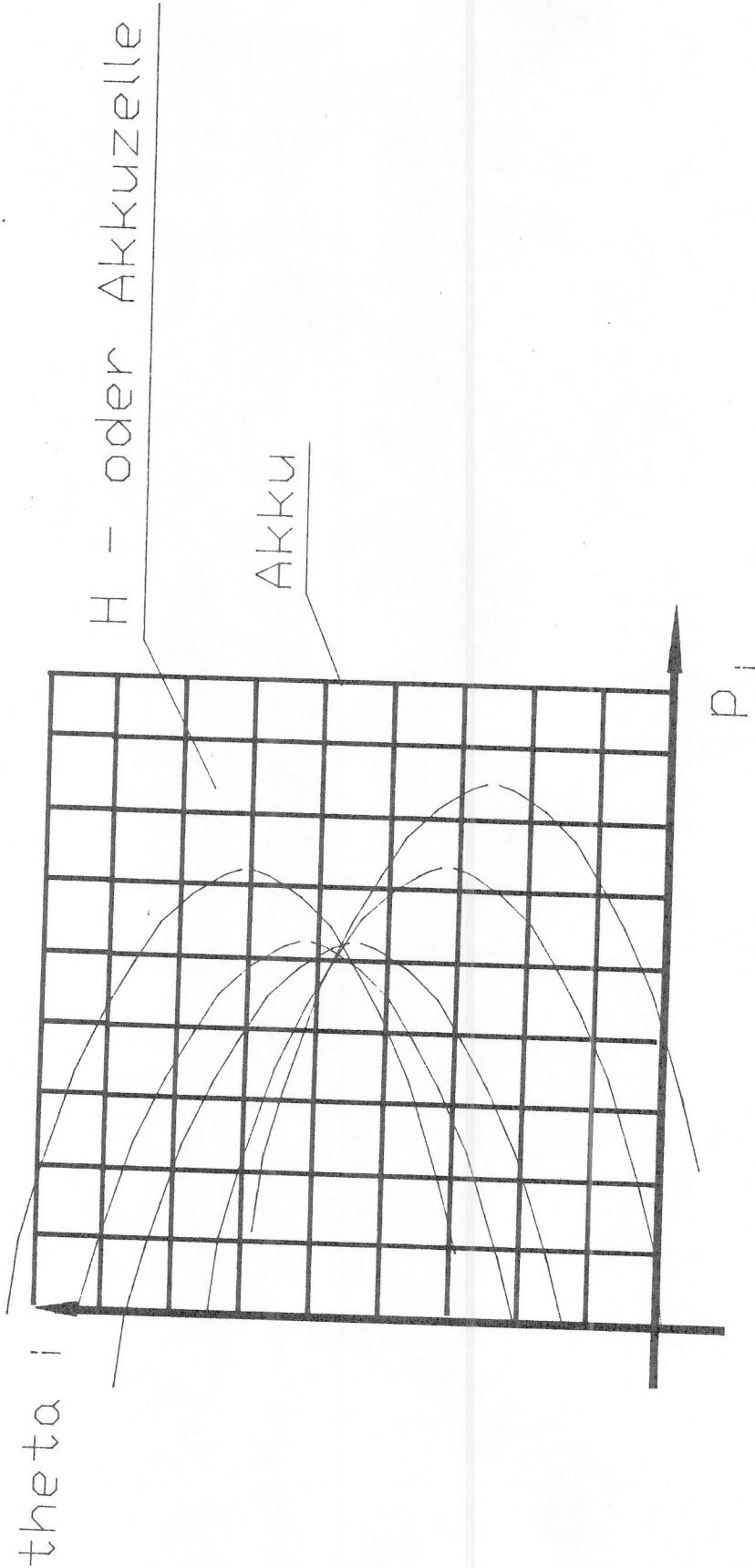
Das Verfahren ist für beliebig  $n^*$ -dimensionale Originalbereiche, sowie verschiedenste Parameterstellungen (für Kreise = Mittelpunkt/Radius, Ebenen etc..) erweiterbar (siehe /38/).

#### 4.3.5.2. Randbedingungen zur Realisierung der Houghtransformation (HT)

Vom mathematischen Verfahren zur realisierbaren Programmstruktur sind, um mit endlicher Rechenzeit und endlichem Speicher auszukommen, wichtige Randbedingungen zu berücksichtigen.

1. Da sich der Rechenaufwand bei der HT mit jeder zusätzlichen O-Bereich-Dimension vervielfacht, sollte man, wenn möglich, im 2-D Bereich arbeiten.

Abl. 4,36



2. Die Auflösung in kartesischen Koordinaten der LSR.E. beträgt 1 cm, der Meßbereich umfaßt  $\pm 1000$  cm (y-Richtung). Die Winkelauflösung des Verfahrens sollte  $5^\circ$  betragen. Daraus ergäbe sich eine Größe des diskreten H-Raumes von:

$$N_H = 2 \times 1000 \times \frac{360}{5} = 144 \cdot 10^3 \text{ Elementen}$$

Eine Kante im LSR.E.-Bild besteht aus maximal

$$\sqrt{81^2 + 41^2} = 90 \text{ Bildpunkten}$$

Wird mit einer  $5^\circ$  Schrittweite transformiert, so ergibt dies

$$T = \frac{360}{5} \times 90 = 6480 \text{ Treffer im H-Bereich}$$

Dies bedeutet:

$$t_{\text{relativ}} = \frac{6480}{144 \cdot 10^3} = 0,045$$

Bestenfalls erhalten 4,5% der H-Ebene einen Treffer, d. h. 95,5% der Maximumsuche im Akku verlaufen in leeren Feldern. Werden zusätzlich noch Störpunkte d. h. Kantenpunkte, die auf keiner Objektkante liegen, abgebildet, bzw. liegen die Kantenpunkte nicht genau auf einer Geraden, so verschwindet jedes Akku Maximum in einem Hintergrundrauschen, d. h. es treten kaum ausgeprägte Maxima auf.

3. Da nicht bekannt ist, wieviele Linien sich in einer Ebene befinden, muß das Verfahren mehrere Linien im H-Bereich auflösen können. Es sind also auch relative Maxima in der H-Ebene zu detektieren.

4. Nicht nur die Detektion von Kanten mit bestimmten geometrischen Parametern im LSR.E.-Bild ist notwendig, fast wichtiger ist die topologische Zuordnung, welcher Meßpunkt zu welcher Kante gehört. Dies ist von der HT nicht gewährleistet, wenn mehrere Linien erlaubt sind.

5. Ist der Akku nicht unendlich groß, und sind mehrere parallele Kanten endlicher Länge im O-Bereich vorhanden, so treten unerwünschte Nebenmaxima auf, da sich durch 2 parallele Strecken aus je n Punkten mindestens n Geraden legen lassen, die genau senkrecht zu den Strecken sind.

#### 4.3.5.3. Realisiertes HT - Verfahren

Das Verfahren zur Objektkantenfindung wurde in der Prozedur houghtransformation (s. Anhang 3; S 68 bis S 100 ) realisiert. Kernstück der Houghtransformation ist die Prozedur polirath. (Poli-Line-Recursive-Adaptive-Hough-Transformation)

Im einzelnen gehe ich wie folgt vor:

I. Wähle alle Kantenpunkte einer Ebene aus (s. Randbedingung 1 in 4.3.5.2.) und projizierte diese Punkte in die Grundebene des kartesischen Koordinatensystems (xy, yz, xz), die zur aktuellen Ebene die geringste Winkelabweichung besitzt (möglichst geringe Verzerrung!!).

II. Greife aus den Punkten von I nur diejenigen heraus, die zu einer Fläche gehören; dies hält die Anzahl der Linien gering ( Rb. 3) und erlaubt eine eindeutige Zuordnung von Kanten zu Objekten (Flächen).

III. Wähle nur die Kantenpunkte aus, die Anfangs- oder Endpunkte der in 4.3.2. detektierten Streckenzüge sind. Dies reduziert das Auftreten paralleler Kanten (s.Rb. 5.).

IV. Setze Startfile, belegt mit den ausgewählten Originaldaten.

V. Transformiere die Punkte aus IV in die H-Ebene; verwende hierzu einen Akku mit lediglich  $9 \times 9$  Zellen (s. Rb 2. und /39/).

VI. Ermittle die Maxima des Akkus, indem alle Akkuzellen, deren Zählerstand  $< 80\%$  (Größe wählbar) des Maximalzählerstandes sind, zu 0 gesetzt werden. Fasse die verbliebenen besetzten Akkufelder, die benachbart sind, zu n-Clustern zusammen.

VII. Bestimme die Ausdehnung aller Cluster in der  $p$ -Ebene ( $p_{\min}, p_{\max}, \mathcal{V}_{\min}, \mathcal{V}_{\max}$ ). Für jedes Cluster ergibt sich genau ein solches Grenz-Quadrupel.

VIII. Starte für jedes Cluster wieder bei V, wobei der Laufparameter der HT nun aus dem Bereich  $(\mathcal{V}_{\min}, \mathcal{V}_{\max})$  in 9 Schritten (Akkugröße) gewählt wird (REKURSION!). Die Originalpunkte werden von Rekursionsschritt zu Rekursionsschritt mitgeführt.

IX. Breche die Rekursion ab, wenn die gewünschte Auflösung  $p_{\text{grenz}}, \mathcal{V}_{\text{grenz}}$  erreicht ist oder, bei Verkleinerung der H-Ebene, wenn die Summe aller Akkuzählerstücke stärker abnimmt als dies die Verkleinerung der H-Ebene erwarten lässt.

X. Speichere alle Punkte einer gefundenen Kante sowie deren  $p$  Parameter ab (Parameter des Maximums).

XI. Lösche die gefundenen Kantenpunkte aus Startfile und gehe zu IV, solange die Anzahl der Originalpunkte nicht zu klein ist.

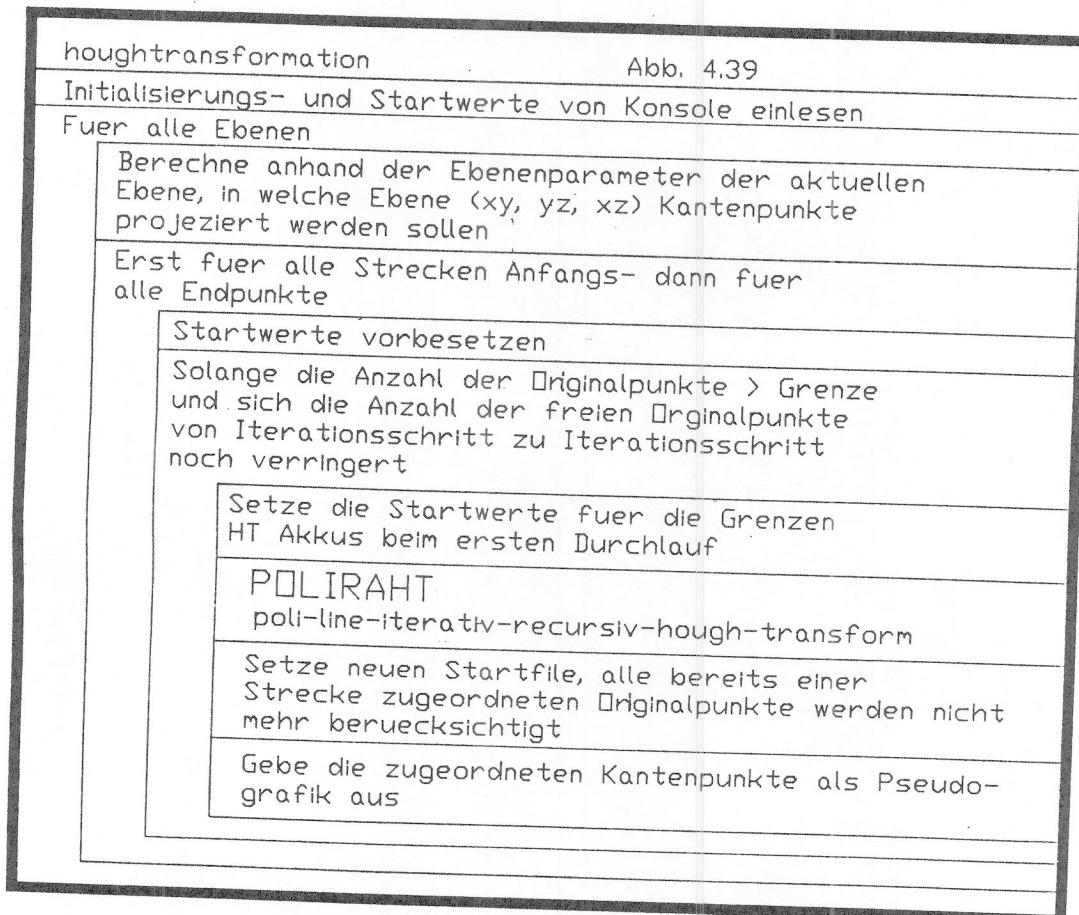
XII. Bearbeite nun alle Endpunkte der Streckenzüge, die zu einer Fläche gehören (gehe zu III.).

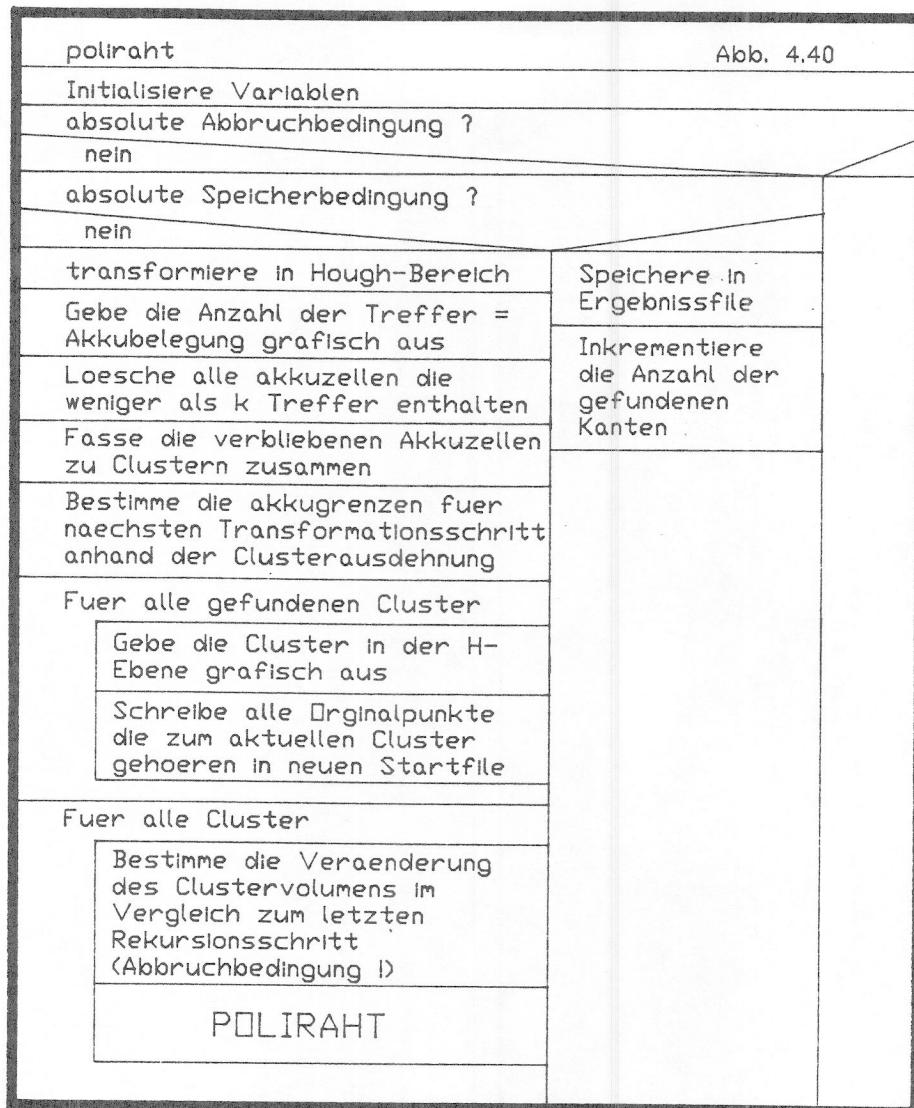
XIII. Bearbeite nun die Punkte der nächsten Fläche einer Ebene und gehe zu II.

XIV. Bearbeite nun die nächste Ebene; gehe zu I.

Abb. 4.39 zeigt ein Struktogramm der Rahmenprozedur houghtransformation. Sie enthält die Schritte I....IV. und XI...XIV. obiger Beschreibung.

Die Arbeitsschritte V....X., nämlich die eigentliche HT und Kantenfindung, sind in der rekursiven Prozedur polirath implementiert, deren Ablauf im Struktogramm Abb. 4.40 veranschaulicht ist.





Die Transformation bildet die Originalpunkte in den Houghbereich ab und hängt alle Originaldaten, die zu einem Bildpunkt gehören als Zeigerkette an das entsprechende Akkufeld an (s. Abb. 4.41). Die Grenzen der H-Ebene und damit des Akkus sind variabel.

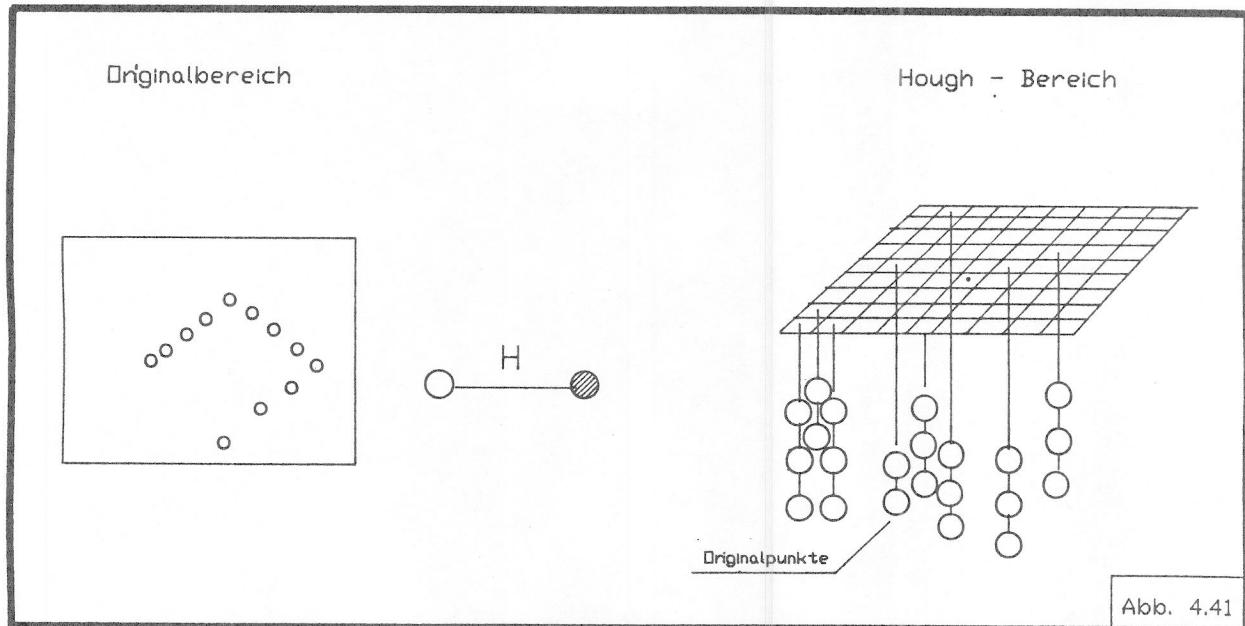


Abb. 4.41

Die Akkugröße ist fest und beträgt  $9 \times 9 = 81$  Punkte.

Abb. 4.42 zeigt einen während des Programmablaufs protokollierten Graphik-Ausdruck des Akkus. Die Akkubefüllung wird als ASCII-Zeichen ausgegeben. Jeweils 10 Treffer erhöhen die Wertigkeit des ASCII-Zeichens um 1. Abb. 4.43 zeigt ein Struktogramm der Transformationsroutine.

S/N 1.35000E+01  
 Grundflaeche : 4.74169E+02  
 d Huellvolumen / d flaeche : 1.10000E+01  
 d Volumen / d flaeche : 1.44462E+00  
 Projektionsebene YZ  
 pschrittw. / thetaschrittw. / Proportionalf. : 3.78125E+01 1.95937E-01 2.61818E-02  
 1.66667E+00 9.50550E-02  
 Treffer knapp daneben : 0  
 Treffer daneben : 0  
 maximales p : 7  
 minimales p : 4  
 Mittelwert : -1.51250E+02  
  
 Flaechenzahl : 5  
 pmin, pmax, thetamin, thetamax :  
 -3.02500E+02 0.00000E+00-7.85000E-01 7.82500E-01  
 Maechtigkeit : 10  
 Gipfelhoehe : 11  
 Trefferanzahl neu : 99  
  
 ....65....  
 ....1:....  
 ....47....  
 ....50....  
 ....1:....  
 ....1:....  
 ....:1....  
 ....56....  
 ....83....

Abb 4.42

Peak-Hoehe : 11  
Clusternummer : 1  
Peakkoordinaten (px, theta) : -1.512505+02 1.94687E-01  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
0  
X  
0  
\*\*\*\*\*  
  
Peak-Hoehe : 10  
Clusternummer : 2  
Peakkoordinaten (px, theta) : -1.512505+02 -5.89063E-01  
\*\*\*\*\*  
X  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Abb. 4.4.4

## transformation

Abb. 4.43

Lege die fuer die Transformation benötigten sin und cos Werte in Tabelle ab

Bestimme Normierungsfaktor um p-Werte in den Zahlenbereich 1..9 zu bringen (akku !)

Fuer alle Originalpunkte

Fuer alle 9° Winkelschritte

Wurde der aktuelle Originalpunkt schon einer Kante zugewiesen ?

nein

Projizierte Originalpunkte in Arbeitsebene

Berechne  $p = f(f_i)$

Inkrementiere akku( $p, f_i$ )

Zaehle die Anzahl der Gesamttreffer

Schreibe Originaldaten in Zeigerfile das jedem akkufeld zugeordnet ist

Loesche den File der die Originalpunkte enthaelt

## neuegrenzen setzen

Abb. 4.45

Fuer alle detektierten Cluster

Fuer alle akkufelder eines Clusters

Bestimme linke, rechte obere und untere Grenze des Clusters in akkukoordinaten

Bestimme die Hoehe und die Breite des Clusters in akkukoord.

Hat Cluster die Form eines senkrechten Striches ?

ja

dekrementiere p.min Koord.

inkrementiere p.max Koord

Liegt peak des Clusters an akku-Rand ?

Ja

Bestimme an welchem Rand

Breite oder Hoehe des Clusters  $\leq 2$  und keine Strichform ?

ja

Vergroessere p.min, p.max bzw. theta.min, theta.max so das Cluster Breite und Hoehe mind 3 akku Felder

Liegt peak des Clusters an akku-Rand?

ja

Bestimme abhaengig von der oben ermittelten peak-Lage in welche Richtung die akku-Grenzen zu verschieben sind

Setze neue akku-Grenzen Behalte die akku-Groesse bei verschiebe die Grenzen so, das der akku-Peak in der Mitte des akkus zu liegen kommt

Setzte neue akku-Grenzen, und zwar so, das die neuen Grenzen an p.min/max und theta.min/max zu liegen kommen

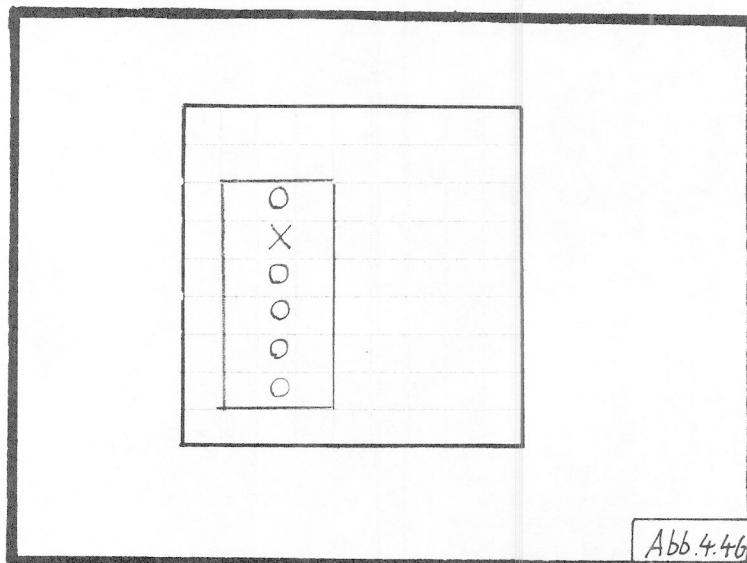
Speichere die neuen Grenzen ab

In der Prozedur clusterung werden alle Akkuwerte, die keine signifikanten relativen Maxima sind, gelöscht. Verbleibende Maxima werden zu Clustern zusammengefasst. Dies geschieht nach demselben Verfahren wie die Flächenclusterung in 4.3.4.

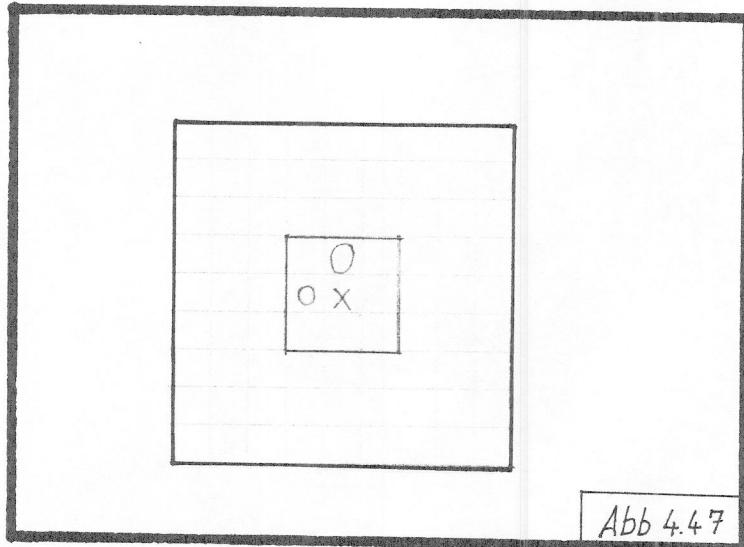
Abb. 4.44 zeigt die aus der Akkubellegung in Abb. 4.42 resultierenden Cluster. Der "Gipfel", d. h. der maximale Akkuzählerstand eines jeden Clusters wird durch X markiert, alle übrigen Punkte durch O.

In der Prozedur neuegrenzensetzen (Struktogramm Abb. 4.45) werden nun für den nächsten Rekursionsschritt die Akkugrenzen an die Ausdehnung der einzelnen Cluster adaptiert. Hier wird folgende Strategie verfolgt:

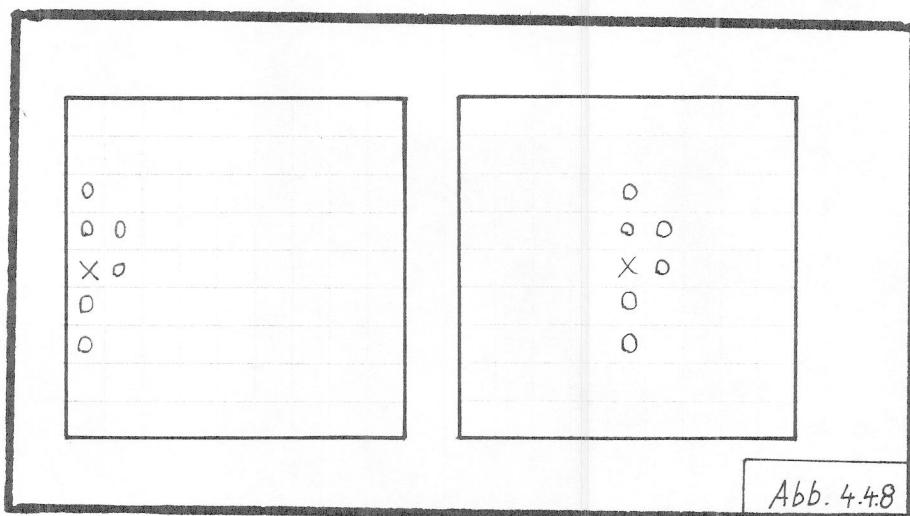
- a) Senkrechte Strichformen des Clusters (s. Abb. 4.46) (tritt vor allem beim ersten Rekursionsschritt auf).



- b) Cluster ist weniger als 2 Felder breit oder hoch (s. Abb. 4.47).

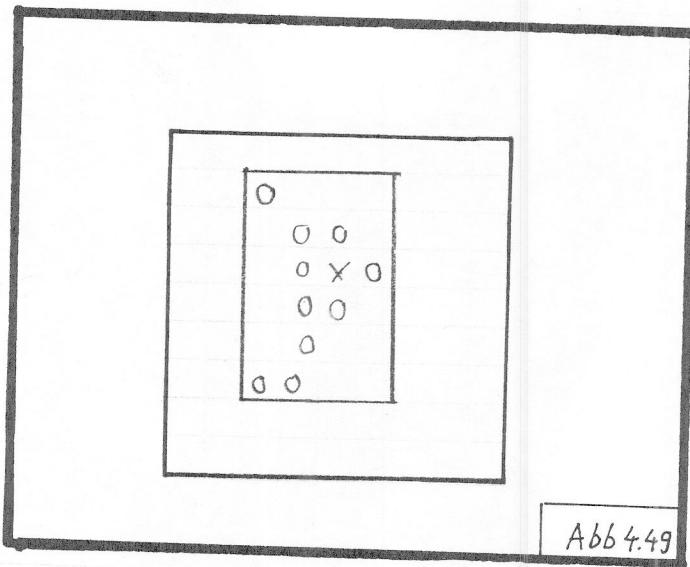


c) Clustermaximum liegt am Rand des Akkufeldes (s. Abb. 4.48).



Behalte hier die absolute Ausdehnung des Akkus bei Verschiebung zu dessen Position in der H-Ebene.

d) Keiner der obigen Fälle (s. Abb. 4.49):



Passe die Grenzen direkt an die Clusterausdehnung an.

Von der Prozedur clusterdatenausgangsfeld werden nun alle Originaldaten, die zu einem Cluster gehören, zusammengefasst, indem alle an den entsprechenden Akkufeldern "hängenden" Zeigerketten zu einer Zeigerkette verbunden werden. Mehrfach auftretende Originalpunkte werden aus der neuen Kette eliminiert.

Zu  $k$  Clustern werden also  $k$  neue Originaldatensätze generiert, im Kopf eines solchen Datensatzes sind die neuen Akkugrenzen, sowie die Anzahl der Originalpunkte abgespeichert. Mit diesen  $k$  neuen Originaldaten und Grenzwerten steuert nun die Prozedur polirath  $k$  mal von neuem. Abb. 4.50 zeigt schematisch den Rekursionslauf.

2) Flächenzahl : 1  
 Distanz, die zu betrachten ist max : 2.35000E+00  
 $-1.0000E+01 \cdot 1.0000E+01 = -5.0000E+01$   
 $-5.0000E+01 \cdot 1.2200E+02 = 1.05312E+00$   
 Wuchtigkeit : 150  
 Gisfalleanzahl : 131  
 Trefferanzahl neu : 1089

3) Flächenzahl : 1  
 Distanz, die zu betrachten ist max : 1.44423E+00  
 $-1.0000E+01 \cdot 1.0000E+01 = -1.2720E+01$   
 $-1.2720E+01 \cdot 1.2200E+02 = 80$   
 Wuchtigkeit : 150  
 Gisfalleanzahl : 131  
 Trefferanzahl neu : 729

4) Flächenzahl : 1  
 Distanz, die zu betrachten ist max : 1.35525E+00  
 $-1.0000E+01 \cdot 1.0000E+01 = -3.5500E+00$   
 $-3.5500E+00 \cdot 1.2200E+02 = 45$   
 Wuchtigkeit : 150  
 Gisfalleanzahl : 131  
 Trefferanzahl neu : 397

5) Flächenzahl : 1  
 Distanz, die zu betrachten ist max : 1.35525E+00  
 $-1.0000E+01 \cdot 1.0000E+01 = -3.5500E+00$   
 $-3.5500E+00 \cdot 1.2200E+02 = 45$   
 Wuchtigkeit : 150  
 Gisfalleanzahl : 131  
 Trefferanzahl neu : 397

6) Flächenzahl : 1  
 Distanz, die zu betrachten ist max : 1.35525E+00  
 $-1.0000E+01 \cdot 1.0000E+01 = -3.5500E+00$   
 $-3.5500E+00 \cdot 1.2200E+02 = 45$   
 Wuchtigkeit : 150  
 Gisfalleanzahl : 131  
 Trefferanzahl neu : 153

7) 2.1) Flächenzahl : 1  
 Distanz, die zu betrachten ist max : 1.35525E+00  
 $-1.0000E+01 \cdot 1.0000E+01 = -3.5500E+00$   
 $-3.5500E+00 \cdot 1.2200E+02 = 45$   
 Wuchtigkeit : 150  
 Gisfalleanzahl : 131  
 Trefferanzahl neu : 185

7) 2.2) Flächenzahl : 1  
 Distanz, die zu betrachten ist max : 1.35525E+00  
 $-1.0000E+01 \cdot 1.0000E+01 = -3.5500E+00$   
 $-3.5500E+00 \cdot 1.2200E+02 = 45$   
 Wuchtigkeit : 150  
 Gisfalleanzahl : 131  
 Trefferanzahl neu : 185

7) 2.3) Flächenzahl : 1  
 Distanz, die zu betrachten ist max : 1.35525E+00  
 $-1.0000E+01 \cdot 1.0000E+01 = -3.5500E+00$   
 $-3.5500E+00 \cdot 1.2200E+02 = 45$   
 Wuchtigkeit : 150  
 Gisfalleanzahl : 131  
 Trefferanzahl neu : 185

Abb 4.51

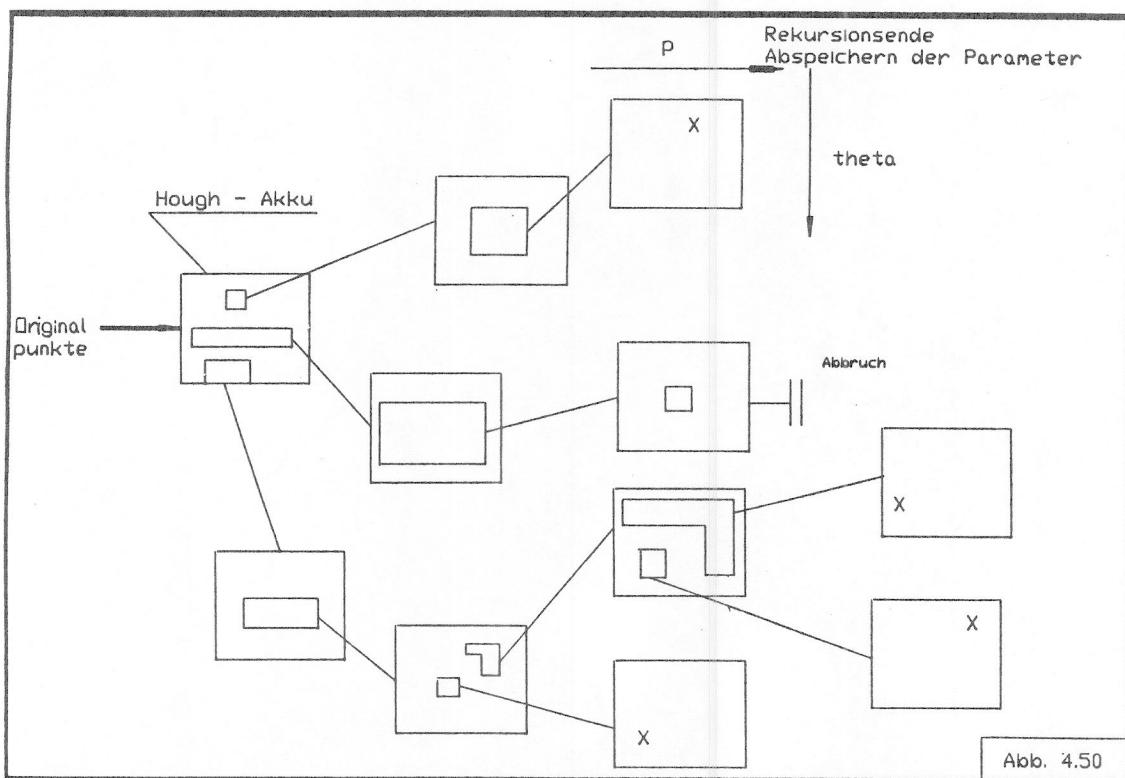


Abb. 4.51 zeigt dies an Originalprotokollen der Prozedur houghtransformation. Informationen, wie neue detektive Grenzen, Trefferanzahl, Maximafläche, aktuelle Projektionsrichtung etc., werden mit ausgegeben.

#### 4.3.5.4. Abbruchbedingungen für die rekursive HT

Wichtig bei jedem rekursiven bzw. deduktiven Verfahren ist es, geeignete Abbruchbedingungen zu definieren, die die Rekursion sobald als möglich beenden, sowie das bis dahin erreichte Ergebnis anerkennen und abspeichern bzw. verwerfen.

Es werden verschiedenste Maßzahlen als Abbruchkriterien ausgewertet:

1. treffer: Gesamtzahl TM der Treffer im Akku = Summe aller Akkuzählerstände.
2. gipfelhöhe: Höchster Akkuzählerstand GH
3. rauschzahl: Gipfelhöhe/Treffer
4. volumengradient:  $dh/df$

Die Akkugrenzen  $p_{\min}$ ,  $p_{\max}$ ,  $v_{\min}$ ,  $v_{\max}$  definieren ein Flächenmaß:

$$F = |p_{\min} - p_{\max}| * |v_{\min} - v_{\max}| \text{ [cm}^2\text{rad]}$$

Die Grundfläche einer Akkuzelle beträgt somit:

$$F_{\text{akku}} = F / 81$$

Erweitert man den Akku in die 3. Dimension und interpretiert jeden Treffer als Einheitswürfel der Grundfläche  $F_{\text{akku}}$  und der Höhe 1, so lässt sich der Akku als "Gebirge" darstellen (s. Abb. 4.52).

Das Gebirgsvolumen beträgt dann:

$$V_{\text{akku}} = T_H * F_{\text{akku}}$$

Das Hüllvolumen des Würfels, der dieses Gebirge vollständig einschließt, errechnet sich zu

$$H_{\text{akku}} = G_H * F$$

$F_{\text{alt}}$  und  $G_H$  werden an den nächsten Rekursionsschritt übergeben, der Volumengradient berechnet sich dann zu:

$$V_g = \frac{dh}{df} = \frac{G_{\text{neu}} F_{\text{neu}} - G_{\text{alt}} F_{\text{alt}}}{F_{\text{neu}} - F_{\text{alt}}} ;$$

$V_g$  ist ein Maß dafür, wieviele Treffer dem Akku noch verbleiben, auch wenn die Akku-Grundfläche verkleinert wird.

5. Ähnlich wie der Hüllvolumengradient ist der Volumengradient definiert, der anstelle des Hüllvolumens das "tatsächliche Gebirgsvolumen" heranzieht. Es gilt:

$$V_H = \frac{dv}{df} = \frac{T_{\text{neu}} F_{\text{akku neu}} - T_{\text{alt}} \cdot F_{\text{akku alt}}}{F_{\text{akku neu}} - F_{\text{akku alt}}} ;$$

6. p-schrittweite: Maß, wie genau die HT eine Gerade in P-Richtung bereits annähert = Maß für die Rasterung des Akkus.

7. theta-schrittweite: Maß für die Genauigkeit der Winkelauflösung

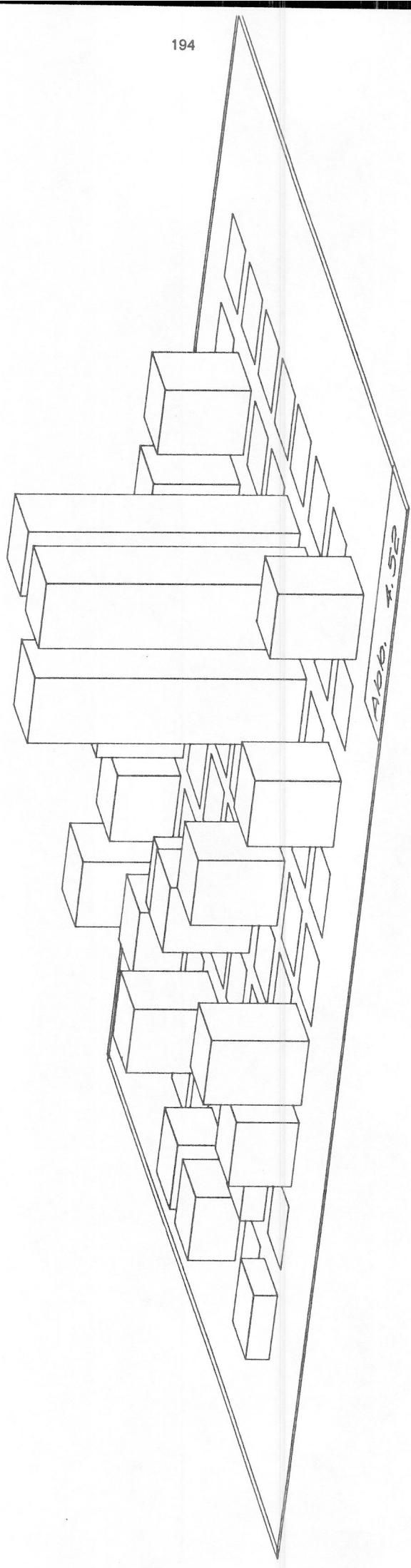
Folgende Bedingungen determinieren die Rekursion:

Absolute Abbruchbedingung ohne Ergebnis abzuspeichern:

- (treffer < grenze 1) v
- (gipfelhöhe < grenze 2) v
- (rauschzahl < grenze 3)
- ( $V_g$  < grenze 4) v
- ( $V_H$  < grenze 5)

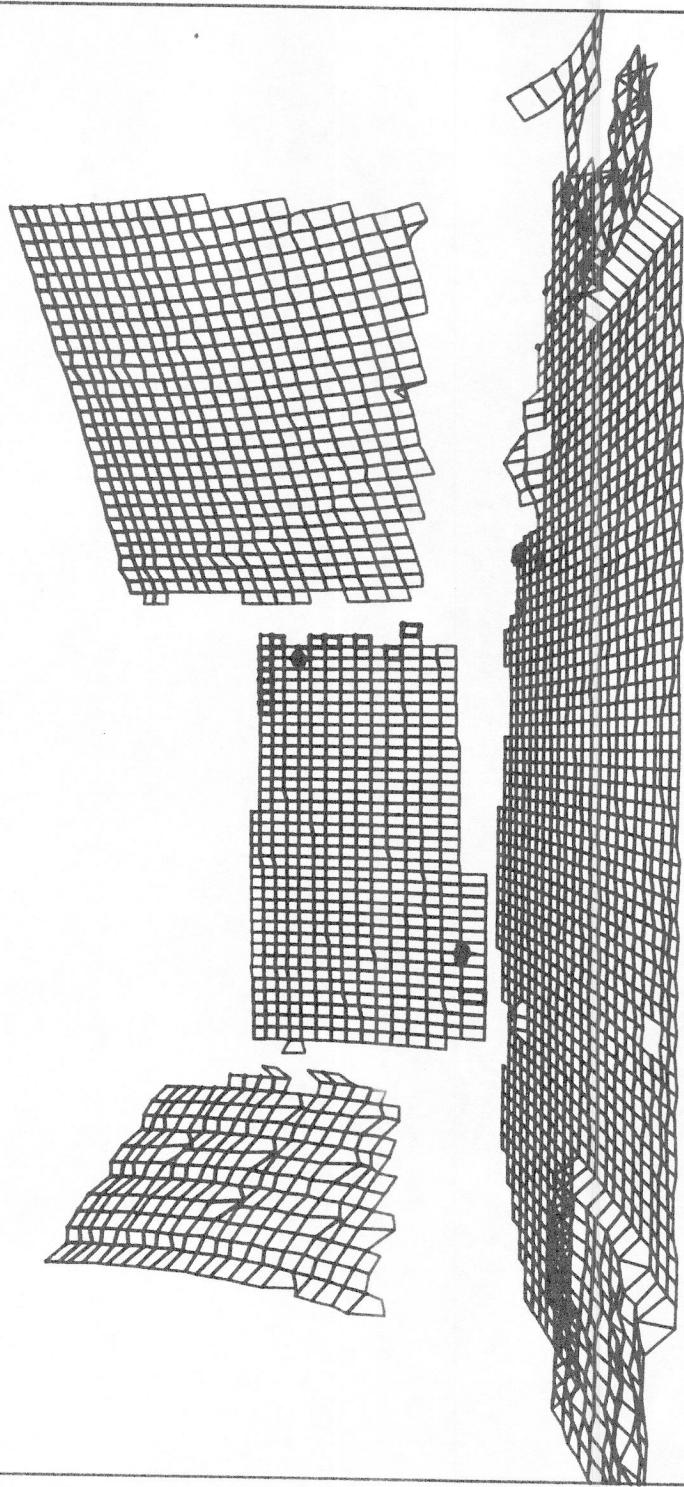
Absolute Abbruchbedingungen mit Abspeicherung des Ergebnisses:

- ((p schrittweite < grenze 6) thetaschrittweite < grenze 7)) v
- (( $V_g$  < grenze 8) v ( $V_H$  < grenze 9))
- (pschrittweite < grenze 6))



In allen übrigen Fällen fährt die Rekursion fort. Grenzen 4, 5, 6, 8, 9 können während des Programmlaufes eingegeben werden.

Abb. 4.53 zeigt das Endergebnis der Houghtransformation zur Szene aus Abb. 4.2 als 2D - Grafik; Abb. 4.54 zeigt dieselbe Szene als 3D Grafik.



GRAPHISCHE DARSTELLUNG

Abb 4.54

## 5. SCHNITTSTELLEN- TEST- UND TRANSFORMATIONS PROZEDUREN FÜR DEN MULTISENSORRAHMEN ARGUS

Für den Einsatz des MSS ARGUS sind Schnittstellen zu den LSR.E. Rohdaten und zum Graphiksystem zu schaffen. Weiterhin ist ein logS zu implementieren, der die polaren LSR.E. Koordinaten in das karthesische Koordinatensystem wandelt. Zu Testzwecken wird ein Programm entwickelt, das synthetische LSR.E. Bilder generiert.

### 5.1. SCHNITTSTELLE VON DEN LSR.E. ROHDATEN ZUM MSS ARGUS

Die Meßdaten eines Bildes der LSR.E. werden in einem Standardformat in Dateien abgespeichert. Das Programm "eingang" liest diese Daten ein, ruft die RDB auf und speichert die Daten dort in der Relation LS\_Daten ab. Die Fahrzeugkonturen der MAKROBE werden von der LSR.E. miterfaßt. Da diese aber für die Fahrzeugführung oder Umwelterkundung keinerlei Information enthalten, wird die MAKROBE aus dem Bild ausgeblendet, indem an dieser Stelle die Meßwerte auf den virtuellen Fahrzeuguntergrund (Boden) gelegt werden. Ist die Qualität des Meßsignals < 20 (s. Kap. 2), wird ebenso verfahren.

Ein Ausdruck des Programms "eingang" findet sich in Anhang. 4. S.113 bis S.118.

## 5.2. SCHNITTSTELLE VON MSS ARGUS ZUM GRAPHIKSYSTEM

Da die Beurteilung einer EBV nur mit den Augen, und somit durch eine graphische Darstellung möglich ist, wurden zwei Schnittstellen vom MSS zum Graphiksystem nach /1/ geschaffen. Die Direkten Ergebnisse der EBV, errechnet im Programm "ratio", werden durch die dort implementierte Prozedur "datgriff" als Bilddatenfiles ausgegeben (s. Anhang 3, S.29 bis S.111).

Um die Möglichkeiten von ARGUS voll ausschöpfen zu können, wird mit dem Programm "select" ein Rahmen zur Verfügung gestellt, mit dem situations- und bildabhängig die gezielte RDB-Abfrage auf ARGUS möglich ist, selektierte Bilddaten werden im Originalbild farbig markiert auf der Graphikstation ausgegeben.

Hierzu muß lediglich eine RDB-Abfrage formuliert werden und in die Prozedur "datgriff" des Programmes "select" eingegeben werden. Diese Abfrage kann online durch den RDB Kommandointerpreter RDO ausgetestet werden.

Ein Beispiel zur Selektion aller Flächen, die parallel zur üBlickrichtung der MAKROBE stehen ("Gangwände"), zeigt Abb. 5.1.. Erfüllt ein Blickpunkt das Selektionsmerkmal, so wird ein Schalter im Bildfeld (page [] kp.) auf true gesetzt; der entsprechende Bildpunkt wird farbig markiert.

```

FOR L IN EBENENPARAMETER WITH
(L. ALPHA > 1.3) AND (L. ALPHA < 1.7) AND (L. BETA < 0.3)
AND (L. GAMMA > 1.3) AND (L. GAMMA < 1.7)

hilfnummer:= L. EBENENNR;

FOR M IN EBENENPUNKT WITH M. EBENENNR = hilfnummer
verti:= (M. EBENENPNR DIV 81);
horiz:= (M. EBENENPNR MOD 81);
page [horiz, verti].kp:= true;

END_FOR;

```

Abb. 5.1

Anhang.5. zeigt ein Listing des vollständigen Programmes.

### 5.3. EIN TESTPROGRAMM FÜR DIE EBV

Wie alle Meßdaten, so sind auch die LSR.E. Daten mit Meßfehlern behaftet. Bei der Entwicklung von EBV-Algorithmen erweist sich dies als sehr störend, da meist nicht mehr zu rekonstruieren ist, ob Fehler in der EBV systematischer Art sind oder auf verrauschten Eingangsdaten beruhen. Aus diesem Grunde wurde das Programm "synth" (s. Anhang 6 S.; 129 bis S.135) entwickelt, das erlaubt, mit wenigen Raumpunkten eine Szene zu beschreiben und daraus ein synthetisches LSR.E. Bild zu generieren. Synthetisiert werden alle LSR.E. Daten, nämlich:

Entfernung:  $r$  in cm

Elevation:  $\vartheta$

Azimut:  $\varphi$

Qualität:  $q$  gestützt auf das Sensormodell in Kap. 2.

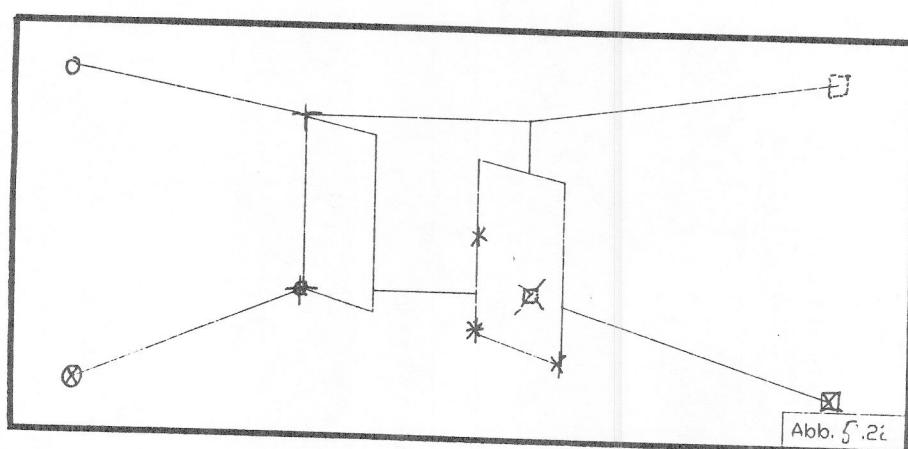


Abbildung 5.2. zeigt eine einfache Szene, markiert sind die Stützpunkte zur Beschreibung der Raumflächen. R ist ein Rahmen, beschrieben durch seinen unteren linken und oberen rechten Eckpunkt.

Durch die Stützpunkte wird jeweils eine Ebene definiert; der Rahmen legt fest, in welchen Raumausschnitt sich diese Ebene ausdehnt, eine einfache Eingabe von Flächen ist so möglich.

```
0 -100 -64 500 -100 -64 0 100 -64 -500 -65 500 300
0 -100 -64 500 -100 -64 0 -100 200 -500 -65 500 300
0 100 -64 500 100 -64 0 100 200 -500 -65 500 300
500 -100 -64 500 100 -64 500 -100 200 -500 -65 500 300
200 0 -64 120 80 -64 200 0 100 20 -65 80 70
200 0 -64 120 80 -64 200 0 100 -80 -65 -40 70
```

Abb. 5.3

Abbildung 5.3. zeigt den zu 5.2. gehörenden Eingangsdatensatz für "synth".

Im Einzelnen arbeitet das Programm folgendermaßen:

\* Lese aus Textfile die Raumbeschreibung ein (maximal 15 Flächen)

Berechne zu jedem der Stützpunkttripel die Ebenengleichung

Lasse Azimut und Elevation entsprechend dem LS.R.E. Abtastverhalten durchlaufen

Berechne aus Azimut, Elevation und der Ebenengleichung die Entfernung r

Liegt der berechnete Wert in karthesischen Koordinaten außerhalb des Rahmens R, so setze die Entfernung r auf 10000

Gehe zu \* bis alle Ebenen eingelesen und bearbeitet sind

Taste nun das Bild nochmals ab und suche für jeden Raumwinkel den minimalen r-Wert aller möglichen Ebenen heraus

Berechne entfernungsabhängig einen Qualitätswert für alle r

Schreibe den synthetisierten Bilddatensatz in die Relation LS\_Daten der MSS ARGUS

Abbildung 5.4. zeigt das zum Datensatz in Abb. 5.3. gehörende synthetische Bild graphisch dargestellt, indem das Programm "select" aus den in ARGUS vorhandenen Daten einen Bilddatenfile generiert hat.

#### 5.4. TRANSFORMATIONSROUTINE

Die EBV arbeitet zum Teil in den polaren Koordinaten der LS.R.E., zum Teil in karthesischen Koordinaten. Das Programm "xyz" (s. Anhang 7; S 137 bis S 140 ) liest aus der ARGUS-Relation LS\_Daten die polaren Koordinaten, wandelt diese in karthesische Koordinaten und legt sie in der Relation Kartes ab. Der Koordinatenursprung wird 64 cm über den Boden auf die Grundfläche der Sensorplattform (s. 1.1.) gelegt.-

## 6. EINSATZ DES MSS ARGUS

### 6.1. BEDIENUNG UND RECHENGESCHWINDIGKEIT

Die Handhabung des MSS ARGUS gliedert sich in folgende Schritte:

1. Rufe den RDB Kommandointerpreter RDO auf.
2. Lösche mit Hilfe des Kommandofiles "wegdamit. com" alle Eintragungen in der RDB.
3. Verlasse den Komandointerpreter.
4. Starte das Programm "eingang"; dieses Programm frägt ab, welcher Bilddatenfile in die RDB geladen werden soll.
5. Starte das Programm "xyz"; es transformiert die polaren in karthesische Bildkoordinaten.
6. Starte das Comand-File "runrat. com"; hierdurch wird die gesamte EBV, sowie die Ausgabe aller Ergebnisse in die RDB und in Bilddatenfiles zur graphischen Darstellung der Daten veranlaßt.

Nach Durchlaufen der obigen Schritte sind alle Meßdaten EBV-Ergebnisse in ARGUS abgelegt. Ein Protokoll des Programmablaufes liegt in der Datei Temp.dat vor. Zur graphischen Darstellung der Ergebnisse werden folgende 4 Dateien erzeugt:

- |              |   |
|--------------|---|
| Ebene. pt:   | Alle detektierten Ebenen                                  |
| Fläche. pt:  | Alle in Flächen zerlegten Ebenen                          |
| Kante. pt:   | Alle gefundenen und einer Ebene zugeordneten Kantenpunkte |
| Strecke. pt: | Alle detektierten Objektkanten                            |

Zur Berechnung obiger Ergebnisse benötigt ein Rechner vom Typ VAX-Station 2000 folgende CPU-Zeiten für die Szene in Abb. 4.30.

- Sekunde 0: Programmstart (Punkt 6)
- Sekunde 33: Die RDB ist initialisiert, alle benötigten Daten sind eingelesen.
- Sekunde 35: Die Daten sind gefiltert.
- Sekunde 47: Spalten- und zeilenweise sind alle geraden Streckenzüge im Bild errechnet.
- Sekunde 58: Alle Ebenen, ihre Parameter und alle Punkte, die zu dieser Ebene gehören, sind errechnet.
- Sekunde 122: Die Houghtransformation ist beendet; alle Objektkanten sind errechnet.

Es ist ersichtlich, daß die wichtigsten EBV-Schritte (Filterung, Kantenpunkte, Ebenen) in 25 CPU-Sekunden abgearbeitet werden.

Ein Vergleichsmaßstab sei das Ergebnis von /32/, wo als annähernd vergleichbare Szene das Entfernungsbild einer Kaffeetasse (3 Flächen) in 97 CPU-Sekunden auf einer speziellen LISP-Machine segmentiert wird.

Zusätzlich zu den allgemeinen Resultaten der EBV ist es nach dem Programmdurchlauf möglich, gezielt spezielle geometrische Informationen über die Szene aus dem MSS ARGUS heraus abzufragen.

## 6.2. BEWEGTE OBJEKTE IM BLICKFELD DER LSR.E

Als erstes Anwendungsbeispiel für das MSS ARGUS habe ich die in Abb. 6.1. gezeigte Szene gewählt. Vor dem Fahrzeug MAKROBE, aufgestellt im Roboterlabor des LSSR, ist ein quaderförmiges Objekt aufgehängt, das sich längs seiner senkrechten Achse dreht. In Abb. 6.1 sind die detektierten Kantenpunkte rot markiert. Da aus Gründen der graphischen Darstellung nicht alle Kantenpunkte abgebildet sind, zeigt Abb. 6.2 zusätzlich eine Pseudographik der Kantenpunkte in polaren Koordinaten; in gleicher Darstellung sind in Abb. 6.3 die segmentierten Ebenen zu sehen.

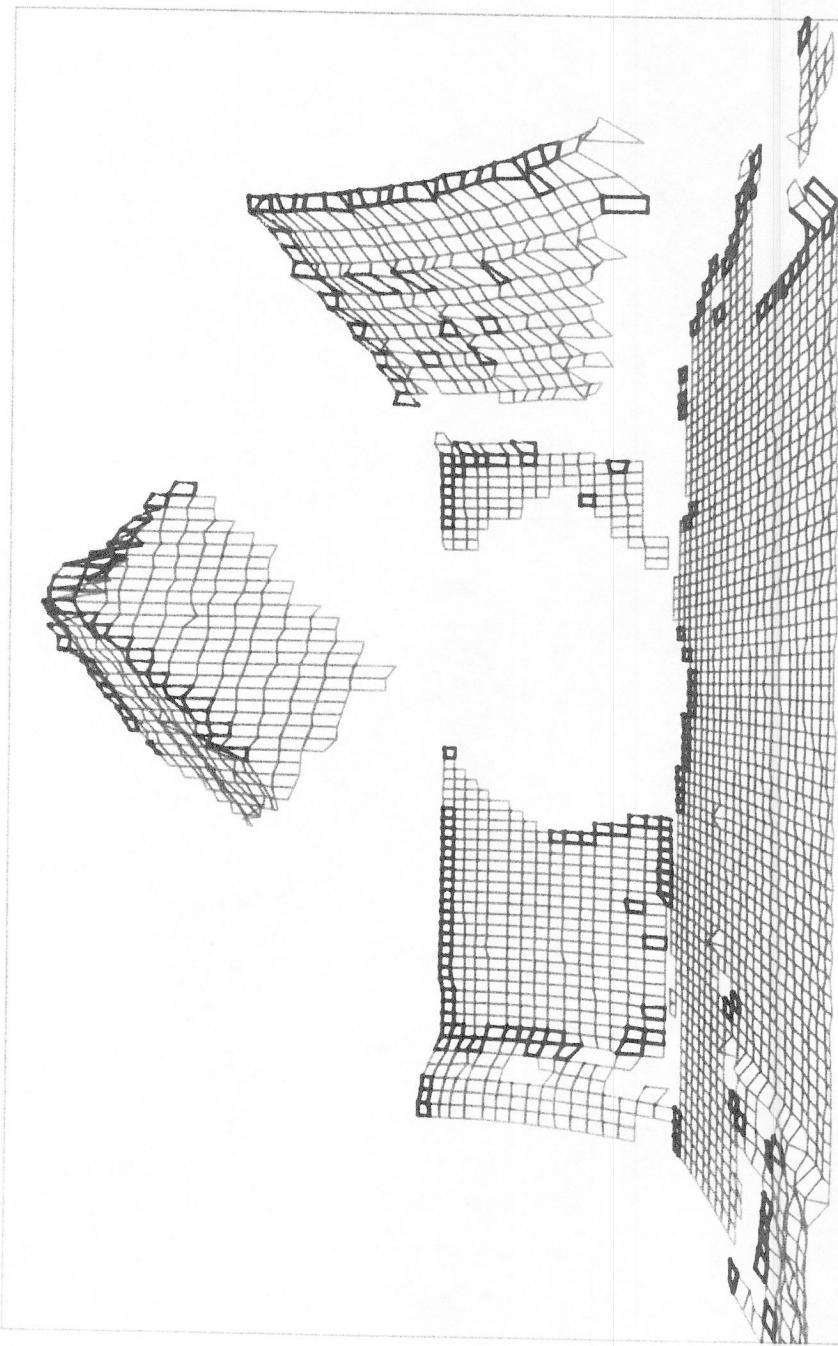
Die Abbildungen 6.4, 6.5 und 6.6 zeigen nun eine Ausschnittsvergrößerung der gleichen Szene, dargestellt ist das bewegte Objekt. Die Bilder sind in ihrer zeitlichen Ablauffolge angeordnet. Der Quader dreht sich im Uhrzeigersinn. Segmentierte Ebenen sind farbig markiert. Abb. 6.7 zeigt die segmentierten Ebenen von Bild 6.4.

Es ist nun Aufgabenstellung an das MSS, ein bewegtes Objekt wie diesen Quader, wiederzuerkennen. Interessant sind hierzu die geometrischen Beziehungen der Objektflächen zueinander. Durch eine Abfrage an ARGUS werden nun die geometrischen Parameter ausgegeben (s. Abb. 6.8). Der Winkel zweier Ebenen zueinander berechnet sich zu:

$$\cos \varphi = \frac{\cos \alpha_1 \cos \alpha_2 + \cos \beta_1 \cos \beta_2 + \cos \gamma_1 \cos \gamma_2}{\sqrt{\cos^2 \alpha_1 + \cos^2 \beta_1 + \cos^2 \gamma_1} \sqrt{\cos^2 \alpha_2 + \cos^2 \beta_2 + \cos^2 \gamma_2}}$$

Zwischen den beiden Seitenflächen des Quaders errechnet sich so ein Winkel von

Schon an den Winkelwerten selbst lässt sich bei gleichem Beta und Gamma eine Winkeldifferenz im Raumwinkel Alpha (Winkel zwischen Boden und Ebene) von  $\pi/2$  ablesen.



GRAPHISCHE DARSTELLUNG

Abb 6.1

Abb. 6.3

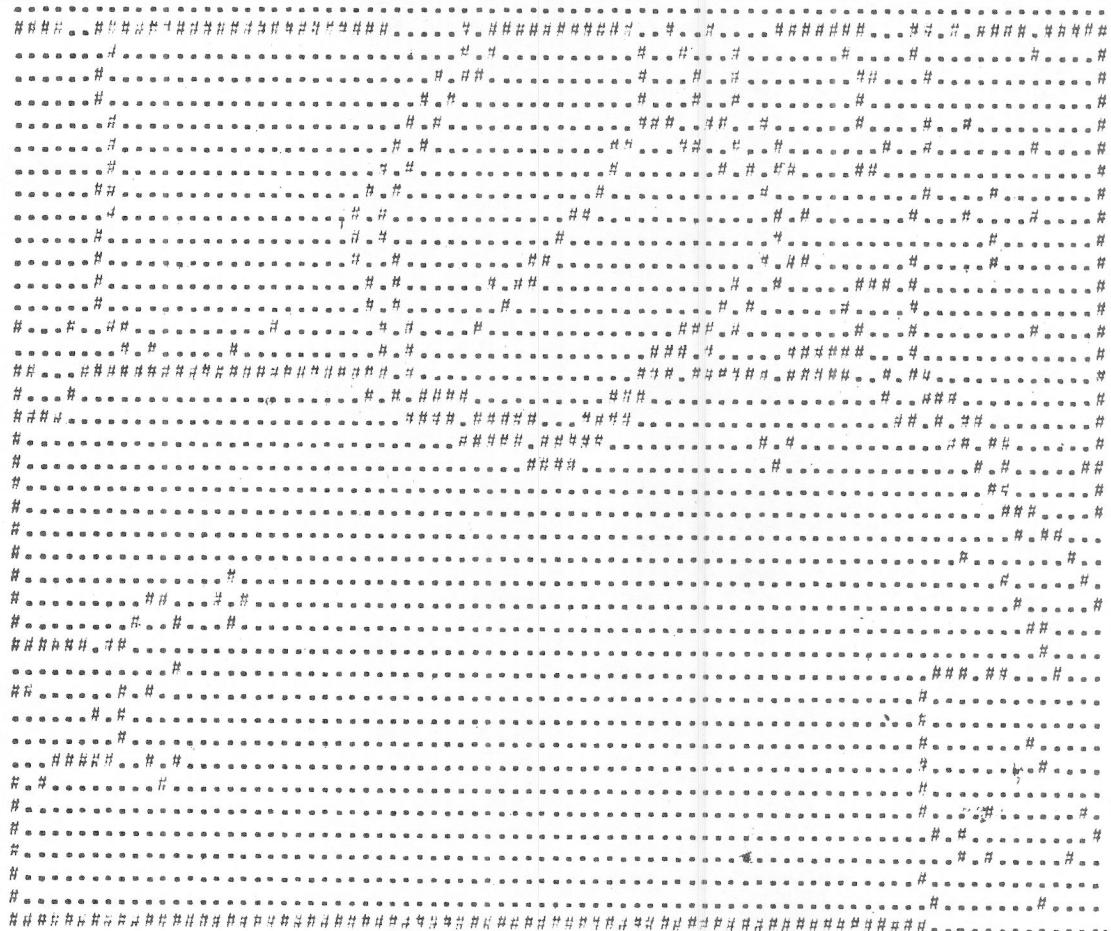


Abb 6.2

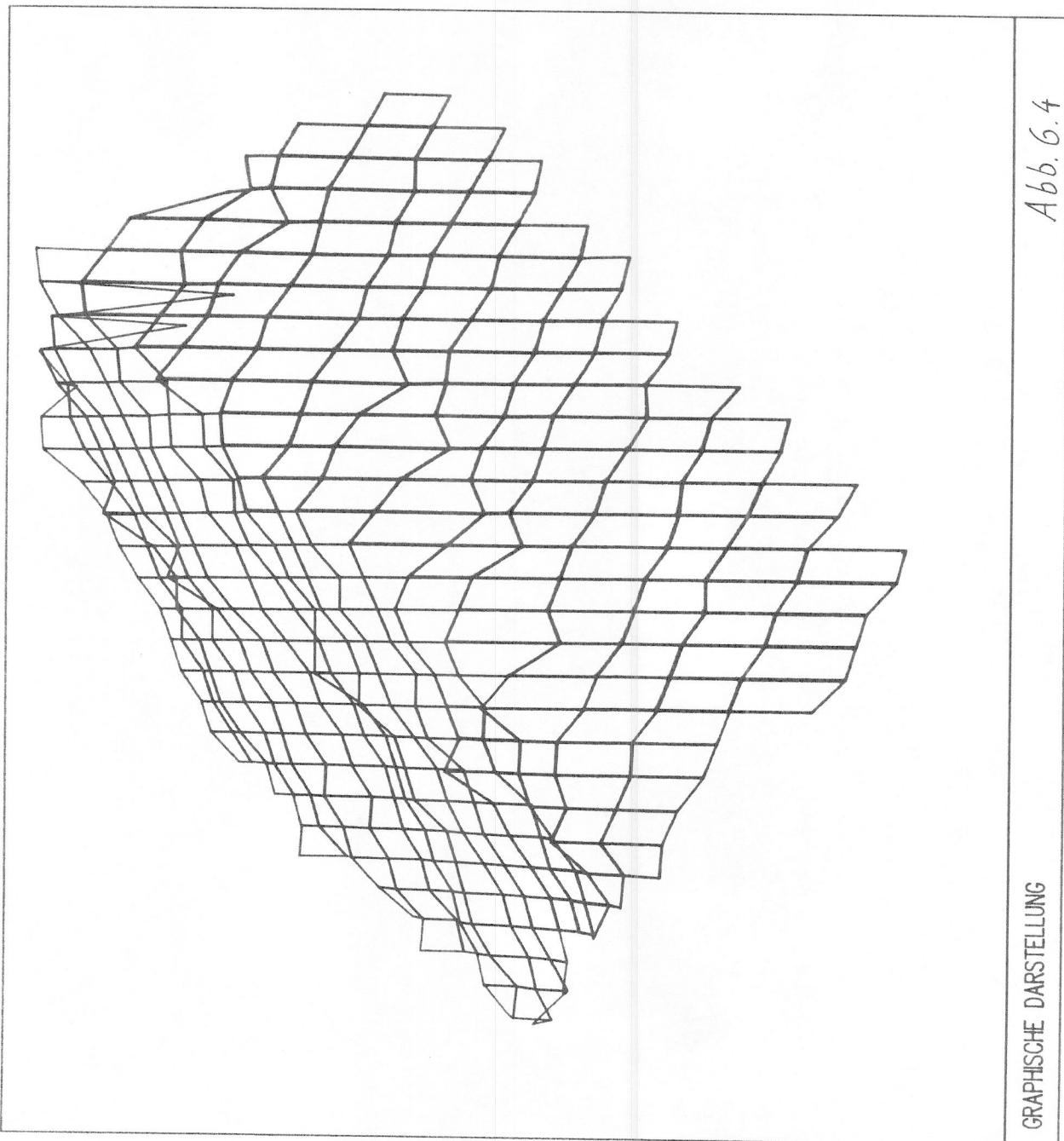
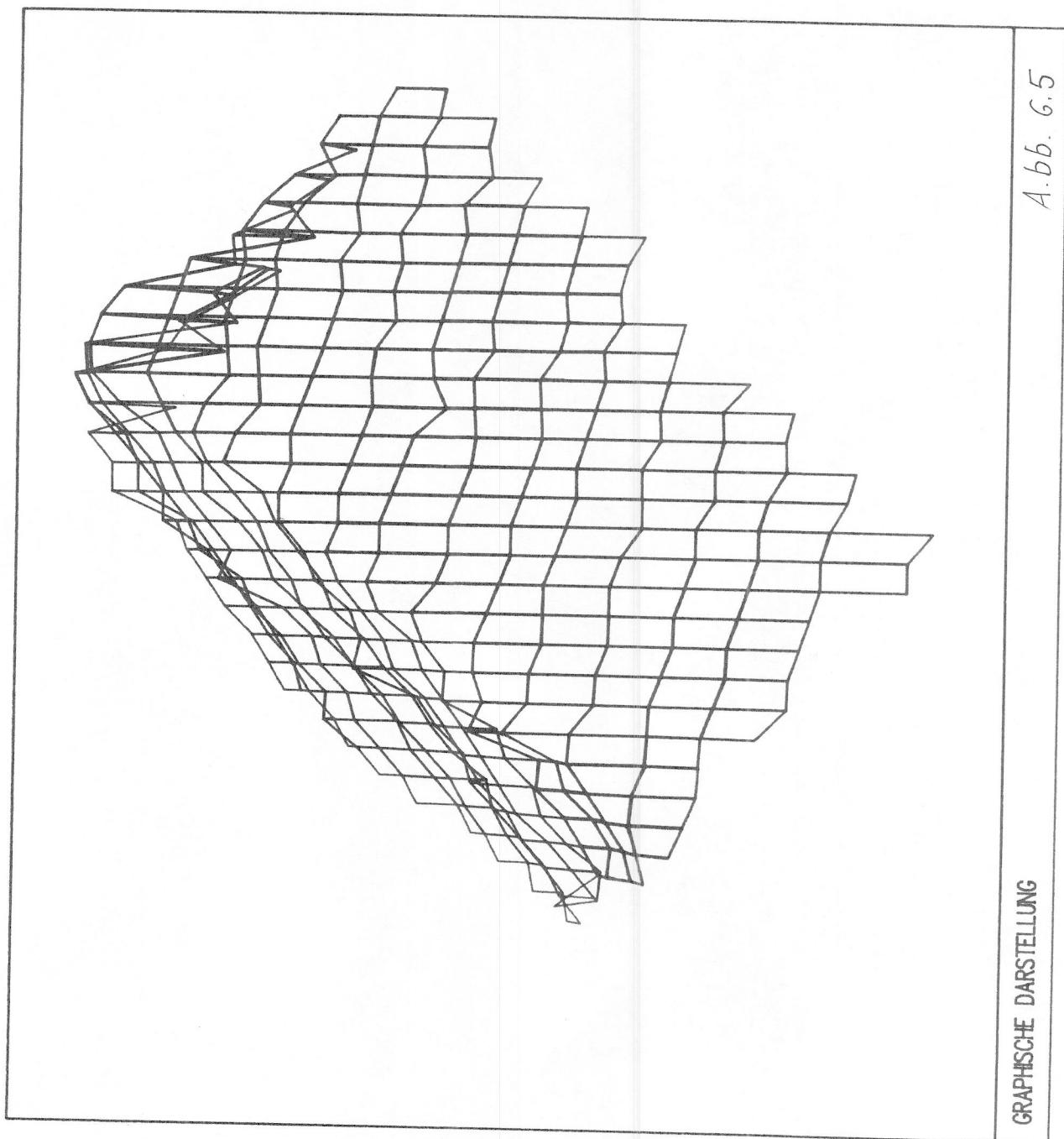


Abb. 6.4

GRAPHISCHE DARSTELLUNG



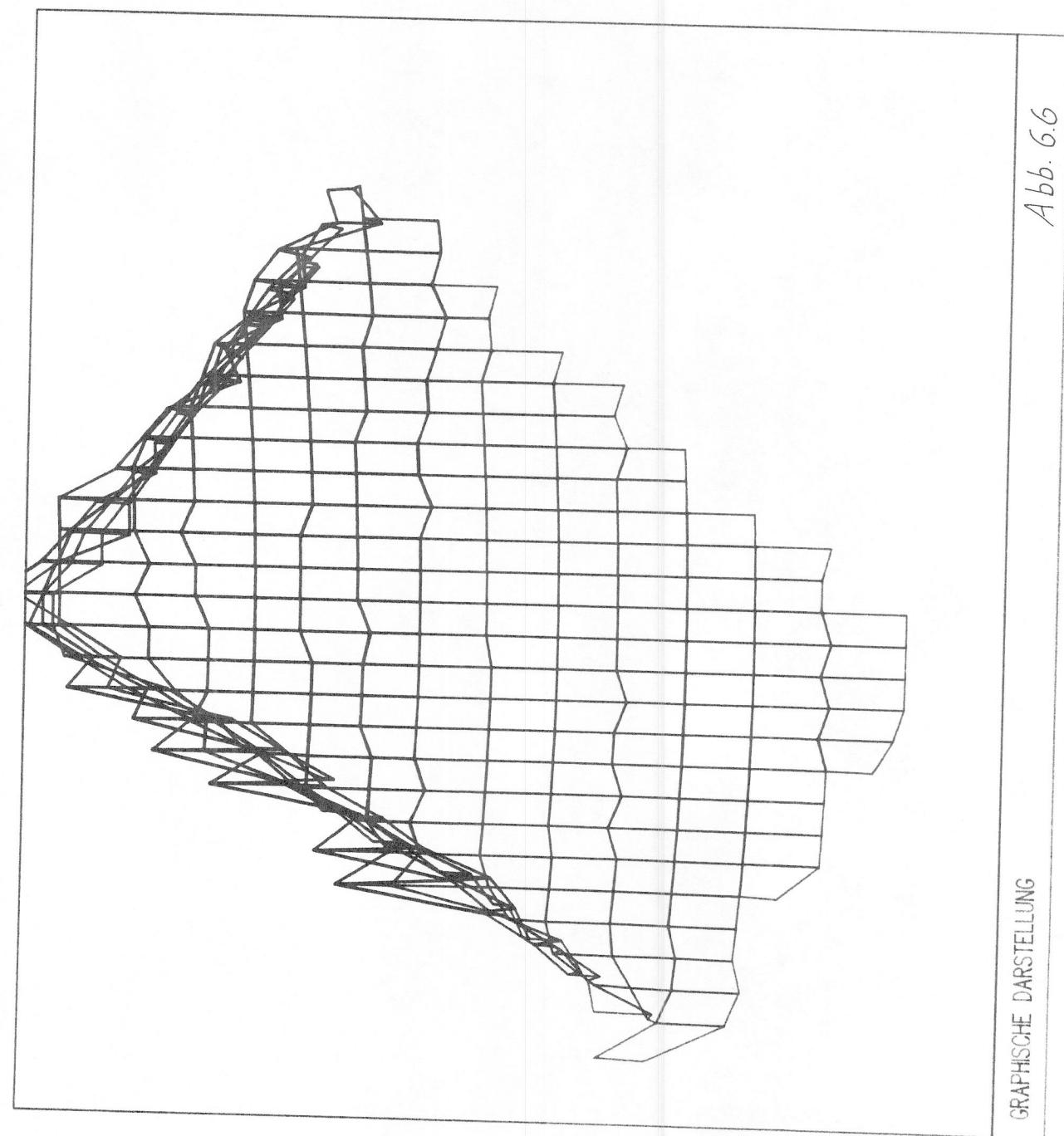


Abb. 6.7

```

invoke database filename argus
wuerfel nummer 14
wuerfel nummer 14
^

%RDO-F-L00K_FOR syntax error, looking for a valid RDO statement, found WUERFEL instead
for e in ebenenparameter
print e.*
end_for
EBENENNR      ALPHA        BETA         GAMMA
 1  1.6139636E+00  1.5469452E+00  4.0325034E-02  -7.1115219E+01
 2  1.5927424E+00  1.5968177E+00  3.1075494E+00   6.3511542E+01
 3  1.6078017E+00  4.2206705E-02  1.5490393E+00   1.1540932E+02
 4  1.5121098E+00  7.6182477E-02  1.5222753E+00   1.3667482E+02
 5  2.4557176E+00  1.1104556E+00  2.0301510E+00  -1.0295764E+02
 6  2.4038262E+00  1.1285715E+00  1.1351603E+00  -9.2571686E+01
 7  1.3043542E-02  1.5568753E+00  1.5593177E+00   4.0249237E+02
 8  1.7722142E-01  1.7326314E+00  1.6271739E+00   3.8652194E+02
 9  6.7354947E-01  1.1344852E+00  2.0474332E+00   7.5696536E+01
10  1.8475571E+00  2.7726237E-01  1.5545412E+00  -3.0019177E+02
sh filde

```

Abb. 6.8

### 6.3. GENERIERUNG EINES KARTHESISCHEN UMWELTPLANES AUS DEM MSS ARGUS

Wie in Kapitel 3 beschrieben, soll ein MSS jeder Ebene des Planungs- und Entscheidungssystems der MAKROBE (s. Abb. 1.3) genau die Daten, und nur die Daten, die sie benötigt, zur Verfügung stellen. Für die Fahrentscheidungen des Piloten ist es zum Beispiel völlig ausreichend, wenn er eine karthesische Landkarte seiner unmittelbaren Umwelt erhält. Da diese Daten aber relevant für die Sicherheit des Fahrzeuges und der Umgebung sind, sollen sie durch möglichst viele Meßpunkte gestützt sein. Eine einfache Abfrage an ARGUS liefert die Daten, die obige Anforderungen erfüllt.

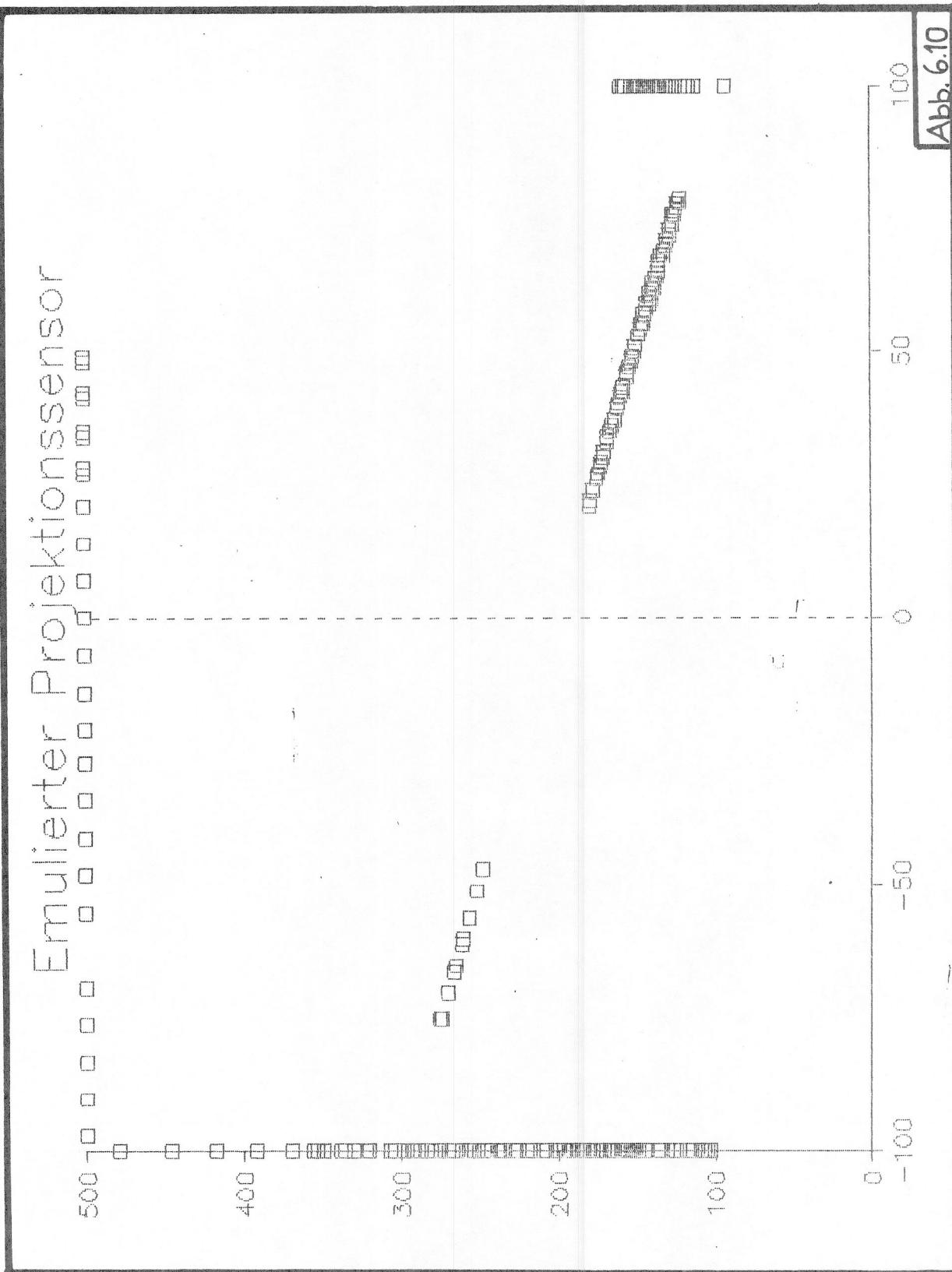
Suche die xyz-Koordinaten aller Meßpunkte, die gleichzeitig Kantenpunkte und nicht nur Begrenzungen des Fußbodens sind.

Abb. 6.9 zeigt diese Bedingungen in der Abfragesprache der RDB/VMS auf das MSS ARGUS, sowie einen Ausschnitt der Meßpunkte, die daraufhin ausgegeben werden. Als Meßszene ist die aus Abb. 4.22 gewählt.

Abbildung 6.10 zeigt eine Graphik der von ARGUS gelieferten Daten.

```
invoke database filename argus
edit
for k in kantenpunkt cross x in kartes
with (k.kantenpnr = x.pnr) and x.z > -50
print x.x, x.y
end_for
      X.X      X.Y
      92       100
      104      -100
      103      -100
      107      -100
      105      -100
      109      -100
      108      -100
      110      -100
      117      -100
      135       66
      132       68
      112       100
      139       61
      137       67
```

Abb. 6.9



## 7. ZUSAMMENFASSUNG UND AUSBLICK

### 7.1. ERGEBNISSE

Ziel meiner Arbeit war es, die Grenzen des Systems LSR.E. von der meßtechnisch-physikalischen Seite bis zu fortgeschrittenen Methoden der Entfernungs bildverarbeitung für autonom mobile Fahrzeuge abzustecken. Innerhalb dieser Grenzen galt es Werkzeuge zu entwickeln, die es erlauben, die Meßergebnisse der LSR.E. zu bewerten, zu validieren und zu bearbeiten, und offene Schnittstellen zu anderen Komponenten der MAKROBE, sowie für weiterführende Arbeiten zu schaffen. Im einzelnen wurden folgende Aufgaben gelöst:

- a) Die meßtechnischen Eigenschaften der LSR.E. wurden vermessen.
- b) Zur Validierung der Meßdaten und Klärung bis dahin offener meßtechnischer Effekte ist ein Sensormodell programmtechnisch realisiert worden, welches die Simulation der LSR.E. für verschiedene Eingangsszenen System und Startparameter erlaubt.
- c) Mit Hilfe des Sensormodells konnten insbesondere das Meßverhalten der LSR.E. an Objektkanten, sowie die zyklische Schwankung des Entfernungsmeßfehlers geklärt werden.
- d) Die softwaremäßige digitale Filterung der an Kanten auftretenden Meßfehler wurde realisiert, eine Hardwarelösung dieses Problems vorgeschlagen.
- e) Basierend auf ersten, in der Literatur vorgeschlagenen Ansätzen zum Multisensorproblem und einer Analyse der Struktur einer meist flächigen industriellen Umwelt, ist ein MSS Entwicklungsrahmen diskutiert und auf der Basis einer relationalen Datenbank im MSS ARGUS realisiert worden.

- f) Die Auswahl, Entwicklung und Implementation einer Grundmenge von EBV-Algorithmen erlaubt es nun, das MSS ARGUS von der LSR.E mit Daten zu versorgen, sprich die flächige Umwelt in Ebenen, Kantenpunkten, Flächen und Objektkanten zu segmentieren.
- g) Schnittstellen zu den Bilddaten von der LSR.E., zum LSR.E. Graphiksystem und RDB/VMS wurden geschaffen.
- h) Anhand einiger Beispiele konnten die Flexibilität und die Möglichkeiten der realisierten Komponenten gezeigt werden.

Noch nicht voll zufriedenstellend arbeitet die mit Hilfe der Houghtransformation realisierte Objekt-Kantensegmentierung. Obwohl in der Literatur als die "am besten geeignete Methode zur Linienverfolgung" gepriesen (/23/, /25/), habe ich während meiner Arbeit den Eindruck gewonnen, daß manche Autoren /40/ dem Hauptproblem dieses Verfahrens, der korrekten Segmentierung paralleler und orthogonaler Kanten, aus dem Weg gehen.

Ein Arbeiten des EBV Systems unter Echtzeitbedingungen der MAKROBE konnte nicht erfüllt werden, war aber auch nicht Ziel meiner Arbeit. Die errechneten EBV-Rechengeschwindigkeiten im 30 s Rahmen sind für das Erkunden der Umwelt aber ausreichend.

## 7.2. AUSBLICK

Die Sensordatenverarbeitung erachte ich als weitgehend abgeschlossen. Die mit geringen Eingriffen in die LSR.E. Auswertelektronik realisierbare Hardwarelösung zur Elimination des Kanteneffektes wäre hier also noch lohnenswert, liefert sie doch, quasi als Abfallprodukt, ein Signal zur Sprungkantendetektion.

Bei der EBV bedarf die Objektkantensegmentierung noch einer Erweiterung. Die Houghtransformation wählt unter allen möglichen Kantenrichtungen die wahrscheinlichste aus, was zu Problemen führt (s. o.). Da in dem von mir realisierten Verfahren die Flächen schon vor den Objektkanten segmentiert werden, wäre es sinnvoll Strecken, die "durch Objekte hindurch" laufen, von vornherein als mögliche Objektkanten auszuschließen.

Die Erweiterung des Multisensorrahmen ARGUS wird in 2 Zielrichtungen verlaufen müssen:

### 1. Steigerung der Geschwindigkeit

Mit Hilfe von ARGUS ist es möglich, verschiedenste Geometriedaten aus einem LSR.E. Bild mit verschiedensten Strategien und Selektionsmechanismen zu gewinnen (s. Kap. 6). Nicht alle dieser Daten und Möglichkeiten werden für die Entscheidungsebenen Pilot und Navigator notwendig sein. Hier ist es sinnvoll, einfache und schnelle logS zu implementieren, also Information weg von der Datenrepräsentation in die Datenverarbeitung zu verlagern.

### 2. Anbindung weiterer KI-Techniken an das MSS

Hier bieten sich 2 Wege an, das MSS ARGUS mit zusätzlicher Intelligenz auszustatten. Zum einen die Einführung von Fuzzy-Werten zur Bewertung der Aussagen in den einzelnen RDB Relationen, sowie Mechanismen zur Verarbeitung und Verknüpfung dieser Werte (s. /41/).

Einen weiteren Weg legt der direkte Zusammenhang zwischen den Aussagen einer Relation (Tupel) und clauses and facts nahe, wie sie z. B. in der Programmiersprache PROLOG Verwendung finden. Hier besteht die Verwendungsmöglichkeit Aussagen, die in ARGUS vorliegen, intelligent weiterzuverarbeiten.

Eine Betrachtung über den Sinn der Bildverarbeitung soll die Arbeit abschließen.

Nach Durchlaufen einer ganzen Galerie von EBV Algorithmen reduziert sich die Datenflut wenig, die die LSR.E. produziert. Geändert hat sich die Darstellung. Deshalb ist an dieser Stelle ein klaren Trennstrich zwischen der Verarbeitung von Bildern und der Aufgabe des "Sehens" zu ziehen. Erst die Einbettung von Bildern in einen kausalen Zusammenhang gestattet den Schritt vom Bild zum Sehen /43/.

Die gezielte Nachfrage nach Bildinhalten, gestützt auf das Gesamtbild Vorwissen und Aufgabe des Fragestellers wird es erlauben, auf dem Weg vom Bild zum Sehen weiterzukommen.  
Ein kleiner Schritt ist hiermit getan.

## LITERATUVERZEICHNIS

- /1/ Voitenleitner, A.: 3-D Betrieb der MAKROBE Laserkamera mit Graphikschnittstelle für die Sensordatenverarbeitung.  
Diplomarbeit am LSSR (1988)
- /2/ Stöckl, J.: Zweifrequente Entfernungssignalgewinnung beim 3-D Laserscanner.  
Diplomarbeit am LSSR (1986)
- /3/ Nitzan, D.; Brain, A. E.; Duda, R. O.: The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis.  
Proceedings of the IEEE, Vol. 65 (1977) 2, S. 206-220
- /4/ Schröder, G.: Technische Fotografie - Grundlagen und Anwendungen.  
2. Auflage, Vogel-Verlag, Würzburg 1981
- /5/ Fröhlich, Ch.: Praktikumsbericht am LSSR (1987)
- /6/ Henderson, T.; Fai Wu So; Hansen, Ch.: MKS: A Multisensor Kernel System.  
IEEE Trans on Systems Man and Cybernetics, Vol. 14 (1984) 5, S. 784-791
- /7/ Henderson, T.; Bhandu Bir; Hansen, C.: Distributed Control in the Multisensor Kernel System.  
Intell. Rob. and Comp. Vision, Casasent D. P.; Hall E. L., Proc. SPIE. 579 (1985), S. 253-255
- /8/ Henderson, T; Bhandu Bir; Hansen, C.: The Synthesis of Logical Sensor Specifications.  
Intell. Rob. and Comp. Vision, Casasent D. P. ed., Proc. SPIE. 579 (1985), S. 442-445

- /9/ Kamat, S. J.: Value Funktion Structure for Multiple Sensor Integration.  
Intell. Rob. and Comp. Vision, Casasent D. P. ed., Proc. SPIE. 579 (1985),  
S. 433- 435
- /10/ Almand, B.: Sensor Information Fusion and a Robot Decisinmakung Methologie.  
Intell. Rob. and Comp. Vision, Casasent D. P. ed., Proc. SPIE 579 (1985),  
S. 436-441
- /11/ Meier, R. J. jr.; Nasa: Fuzzy Vision: Multiple Input Speed Image Understanding.  
Intell. Rob. and Comp. Vision, Casasent D. P. ed., Proc. SPIE 579 (1985),  
S. 196-199
- /12/ Tuijnman, F.; Beemster, M.; Duinker, W.; Hertzberger, L. O.;  
Kuijpers, E.; Müller, H.: A Modell for Control Software and Sensor Algorithms for an Autonomous Mobile Robot.  
Intell. Autonomous Systems, Int. Conf. Amsterdam NL Preprints (1986),  
S. 617-622
- /13/ Elfes, A.: Sonar-Based Real-World Mapping and Navigation.  
IEEE Journ of Robotics and Automation Vol. 3 (1987) 3, S. 249-265
- /14/ Cudhea, P. W.; Brooks, R. A.: Coordinating Multiple Goals for a Mobile Robot.  
Intell. Autonomous Systems, Int. Conf. Amsterdam NL Preprints (1986)
- /15/ Lowrie, J. W.; Thomas, M.; Gremban, K.: The Autonomous Land Vehicle (ALV) Preliminary Road Following Demonstration.  
Intell. Rob. and Comp. Vision, Casasent D. P. ed, Proc. SPIE. 579 (1985),  
S. 336-351
- /16/ Wolfe, W. J.; White, G. K.; Pinson, L. J.: A Multisensor Robotic Locating System and the Camera Cali' Bration Problem.  
Intell. Rob. and Comp. Vision, Casasent D. P. ed, Proc. SPIE. 579 (1985),  
S. 420-421

- /17/ Diemer, W. R.: Relationale Datenbanken kurz und bündig.  
Vogel-Verlag, Würzburg 1987
- /18/ Meier, A.: Erweiterung Relationaler Datenbanksysteme für  
Technische Anwendungen.  
Informatik Fachberichte; 135, Springer-Verlag, Tokyo 1987
- /19/ Pohlmann, G.: Rechnerinterne Objektdarstellung als Basis  
Integrierter CAD-Systeme.  
Hanser-Verlag, München-Wien 1982
- /20/ Eberlein, W.: CAD-Datenbanksysteme.  
Kap. 1.3. Integrierte Ingenieur-Systeme/Produktmodelle, Springer-  
Verlag, Berlin, Heidelberg, New York, Tokyo 1984
- /21/ Cohen, F. S.; Rimey, R. D.; Cayula, J. F.: Real Time 3-D Object  
Recognition from Range or Visible Light Image.  
Comp. Vision for Robots, Faugeras O. D. ed, Kelley R., Proc. SPIE 595  
(1985), S. 2-11
- /22/ Cohen, F. S.; Cayula, J. F.: 3-D Object Recognition from a Single  
Image.  
Intell. Robots and Comp. Vision, Casasent D. P., Hall E. L. ed, Proc.  
SPIE 521 (1985), S. 7-15
- /23/ Faith, J. P.: Intelligente Bildsensoren zum Sichteten, Handhaben,  
Steuern und Regeln.  
Springer-Verlag, Berlin, Heidelberg, New York 1982
- /24/ Crowley, J. L.: The Constructions and Maintenance of a Composite  
Surface Model.  
Comp. Vision for Robots, Faugeras O. D. ed, Kelley R., Proc. SPIE 595  
(1985), S. 125-134
- /25/ Hesse, H.; Goser, K.: Die Hough-Transformation.  
Robotersysteme 4 (1988), S. 27-32

- /26/ Vemuri, B. C.; Aggarwal, J. K.: 3-Dimensional Reconstruction of Objects from Range Data.  
Proceedings of the Seventh Int. Conf. on Patt. Rec. Montreal (1984), S. 752-754
- /27/ Kosko, B.; Limm, J. S.: Vision al Causal Activation and Association. Intell. Robots and Comp. Vision, Casasent D. P. ed, Proc. SPIE 579 (1985), S. 104-109
- /28/ Ailisto, H.; Hakalahti H.; Manninen M.: Dynamic Split and Merge Method for three Dimensional Scene Analysis.  
Intell. Rob. and Comp. Vision, Casasent D. P. ed, Proc. SPIE 579 (1985), S. 488-492
- /29/ Oshima M.; Shirai, Y.: Object Recognition Using Three- Dimensional Information.  
IEEE Trans on Pattern Analysis and Machine Intelligence Vol. 5 (1983) 4, S. 353-361
- /30/ Gil, B.; Mitishe, A.; Aggarwai, J. K.: Experiments in Combining Intensity and Range Edge Maps.  
Computer Vision graphics and Image Processing 21 (1983), S. 395-411
- /31/ Bogaert, M; Cornez, P.: Calibration and Segmentationalgorithms for an accurate and fast rang finder.  
Comp. Vision for Robots , Faugeras O. D. ed, Kelley R., Proc. SPIE 595 (1985), S. 214-224
- /32/ Fan Ting-Jan; Medioni, G.; Nevatia, R.: Segmented Description of 3-D Surfaces.  
IEEE Journal of Robotic and Automation Vol. 3 (1987) 6, S. 527-538
- /33/ Boyter, B.; Aggarwal, J. K: Recognition of Polyhedra from Range Data.  
IEEE Expert Vol. 1 (1986) 1, S. 47-59
- /34/ Rosenfeld, A.: Picture Processing.  
Computer Vision, Graphic and Image, Proc. Vol. 38 (1986), S. 147-225

- /35/ Pratt, W. K.: Digital Image Processing.  
John Wiley & Sons Inc. (1987), S. 523-525
- /36/ Lumia, R.; Shapiro, L.; Zunig, O.: A New Connected Components Algorithm for Virtual Memory Computers.  
Computer Graphics and Image Processing Vol. 22 (1983), S. 287-300
- /37/ Eckhard, P.: Die Hough-Transformation und ihre Anwendung  
Vortrag am 11.02.1988 bei der DFVLR Oberpfaffenhofen
- /38/ Ballard, D. B.: Generalizing the Hough-Transform to Detect Arbitrary Shapes.  
Pattern Recognition Vol. 13 (1981) 2, S. 191-122
- /39/ Illingworth, J.; Kittler, J.: The Adaptive Hough Transform  
IEEE Trans on Pattern Analysis and Machine Intelligence Vol. 9 (1987)  
5, S. 690-698
- /40/ Hesse, H. J.: Konturverfolgung zur Analyse von Grenzwertbildern.  
Dissertation Fernuniversität Hagen 1986
- /41/ Zemankova-Leech, M.: Fuzzy Relational Data Bases: A Key to Expert Systems.  
Köln-Verlag, TÜV Rheinland 1984
- /42/ Brodie, M. L.; Jarke, M.: On Integrating Logic Programming and Databases.  
Expert Database Systems, 1. Int. Workshop on Expert Database Systems  
1984, Kerschberg L. ed, B. Cummings Publishing Company 1986