
Archiv-Ex.:

FZR 93 - 17

Juli 1993

E. J. Wagenaar

**Beschreibung der Software zur Gewinnung
und Bearbeitung von Bilddaten
der Mikrostrahlanlage IMSA-100**

***Beschreibung der Software zur
Gewinnung und Bearbeitung von
Bildaten der Mikrostrahlanlage
IMSA-100***

E.J. Wagenaar
Fachhochschule Enschede, Niederlande

Betreuer Dr. J. Teichert
Forschungszentrum Rossendorf e.V.

Inhaltsverzeichnis

1.	Das Forschungszentrum Rossendorf	3
1.1	Das Zentralinstitut für Kernforschung	3
1.2	Das Forschungszentrum Rossendorf	3
1.2.1	Institut für Ionenstrahlphysik und Materialforschung	5
2.	Imsa-100 Ionenmikrostrahlgerät	5
2.1	Beschreibung	5
2.2	Der Scanfeldaufbau	6
2.3	Der Scanprozessor	7
2.3.1	Einleitung	7
2.3.2	Anschlüsse des Prozessors	7
2.3.3	Register des Prozessors	9
2.4	Die Bildspeicherkarte	10
2.4.1	Einleitung	10
2.4.2	Anschlüsse der Bildspeicherkarte	11
2.4.3	Das Register der Karte	11
3.	Das Programm Iscan	13
3.1	Funktion des Programms	13
3.2	Die Menüs	13
3.2.1	Das Hauptmenü	13
3.2.2	Das Scanmenü	14
3.3	Die Bedienung des Programms	15
3.3.1	Allgemeines	15
3.3.2	Die Statusangabe	16
3.3.3	Einlesen einer Scandatei	17
3.3.4	Selektieren eines Scans	17
3.3.5	Einen Scanvorgang starten	17
3.3.6	Sichern der Bilddaten	18
3.3.7	Darstellung der Daten	18
3.3.8	Einlesen einer Bilddatei	19
3.4	Programmdateien	20
3.4.1	Der Scanfile	20
3.4.2	Der Statusfile	21
3.4.3	Die Startupdatei	22
3.4.4	Die Datei cross.scn	22
3.4.5	Die Bilddatei	23
3.5	Der Kommandomodus	23
4.	Bildverarbeitung	27
4.1	Einleitung	27
4.2	Spatial Domain Methods	27
4.2.1	Filtern	27
4.2.2	Histogrammfunktionen	30

5.	Aufbau des Programms	32
5.1	Das Einlesen der Scandatei	32
5.2	Verarbeitung der Scandaten	33
5.2.1	Programmierung des Scanprozessors	34
5.3	Die Menüsteuerung	34
5.4	Einlesen aus dem Bildspeicher	35
5.5	Abbildung der Daten	35
5.6	Sichern und Laden der Bilddaten	36
5.7	Dateikomprimierung	37
5.8	Verarbeitungsfunktionen	37
5.8.1	Farben eliminieren	37
5.8.2	Kontur	37
5.8.3	Andere Programme starten	38
5.8.4	Filtern	38
5.8.5	Multiple Scan	39
5.9	Statuswiederherstellung	39
5.10	Allgemeine Funktionen	40
	Quellenverzeichnis	41

1. Das Forschungszentrum Rossendorf

1.1 Das Zentralinstitut für Kernforschung [1]

Im Jahre 1956 wurde nach einem Regierungsbeschluß, die Kernforschung in der damaligen DDR zu beginnen, das ZfK gegründet. Es sollte die Forschung auf den damaligen aktuellen Gebieten der Physik der Atomkerne und der Nutzung der Kernenergie durchgeführt werden. Dazu wurde 1957 ein Forschungsreaktor, und ein Jahr später ein Zyklotron installiert. Später kamen ein Van-de-Graaf-Generator und ein Tandembeschleuniger dazu. Um diese Großgeräte bildete sich die Arbeit des ZfK. Als Großforschungseinrichtung war es Mitglied der Akademie der Wissenschaften der DDR. Im Institut befanden sich fünf verschiedene Bereiche.

1. Reaktortechnik und Neutronenphysik
Einige Jahren zuvor nahm das erste Atomkraftwerk der Welt in der UdSSR den Betrieb auf. Die Aufgabe dieses Bereiches war es, Prozesse an Reaktoren zu entwickeln und zu optimieren.
2. Physik der Atomkerne
Der Bereich untersuchte den Aufbau des Atoms. Am Anfang lag der Schwerpunkt der Untersuchung beim niederenergetischen Bereich. Später erweiterte sich in Zusammenarbeit mit dem Vereinigten Institut für Kernforschung in Dubna die Forschung nach höheren Energieregionen.
3. Radiochemie
In diesem Bereich wurden radioaktive Präparate zur Nutzung in der Medizintechnik entwickelt und produziert.
4. Werkstoffe und Festkörper
Hier wurde der Einfluß von Kernstrahlung auf Materie untersucht. Weiterhin wurde u.a. neutronendotiertem Silicium zur Nutzung in der Halbleitertechnik untersucht.
5. Technik
In diesem Bereich wurden Geräte, die in anderen Bereichen gebraucht wurden, entwickelt und gebaut. Ein Beispiel eines solchen Gerätes ist die Rechenanlage ZRA-2.

1.2 Das Forschungszentrum Rossendorf [2]

Nach der Wiedervereinigung Deutschlands endete am 31. Dezember 1991 die Existenz des ZfK. Am 1. Januar 1992 wurde das Forschungszentrum Rossendorf (FZR) gegründet. In diesem Zentrum gibt es die fünf folgenden Institute :

1. Institut für Ionenstrahlphysik und Materialforschung
2. Institut für Bioanorganische und Radiopharmazeutische Chemie
Nuklearmedizinische Forschung klärt ohne Alternative bestimmte biochemische Vorgänge im lebenden Organismus, gewährt Einblicke in bestimmte Krankheitsursachen und sucht nach neuen diagnostischen Möglichkeiten oder Behandlungsmethoden.

3. **Institut für Radiochemie**
Das Institut untersucht die Migration und Diffusion von Isotopen in verschiedenen Umgebungen.
4. **Institut für Kern- und Hadronenphysik**
Dieses Institut untersucht die Atomkerne und seine Bausteine sowie das Verhalten beim hochenergetischen Kern-Kern-Stoß. Aus diesen Arbeiten ergeben sich experimentelle Methoden zur Anwendung in der Materialforschung und Biomedizin.
5. **Institut für Sicherheitsforschung**
Ziel der Arbeiten dieses Instituts ist die Bewertung der Auslegungssicherheit, Erhöhung der Betriebssicherheit von technischen Systemen und die Forschung nach Maßnahmen für die Begrenzung der Auswirkung von Betriebsstörungen und Störfällen.

1.2.1 Institut für Ionenstrahlphysik und Materialforschung

Die Hauptziele des Instituts umfassen die Optimierung von Werkstoffeigenschaften durch Oberflächebeeinflussung, die Entwicklung ionenstrahlgestützter Technologien zur Oberflächebeschichtung und Strukturierung von Materialien, den Einsatz von Ionenstrahlverfahren in der Mikrosystemtechnik und die Ionenstrahlanalyse von Oberflächen und dünnen Schichten. [3] Die Forschung ist so eingerichtet, daß die Resultate leicht von der Industrie übernommen werden können. Sie kann damit auch als industrieorientierte Vorlaufforschung bezeichnet werden. Weitere Ziele sind das Aufbauen enger Zusammenarbeiten mit anderen Instituten und Universitäten und die Ausbildung von Studenten und Wissenschaftlern im Bereich der Materialforschung.

Heutzutage erfolgen die Entwicklung von Materialien, die spezifischen Anwendungszwecken genügen, vielfach durch die Einstellung von Oberflächeneigenschaften. Durch Oberflächen- und Strukturtechnologie lassen sich Oberflächen in ihrer Zusammensetzung, ihrer Gitterstruktur und ihrer Topographie gezielt verändern. Weiterhin können dünne Schichten anderer Stoffe, welche die Eigenschaften des Materials verändern, aufgebracht werden. Gegenüber herkömmlichen Methoden bieten Ionenstrahlen neue Möglichkeiten. Durch die Wahl der Ionenenergie kann der Tiefbereich der Materialmodifikation gezielt eingestellt werden. Auf dieser Weise lassen sich sowohl oberflächennahe als auch tiefer liegende, leitfähige oder isolierende Schichten herstellen. In Ionenquellen können verschiedene Elemente oder Legierungen von Elementen benutzt werden.

2. Imsa 100 Ionenmikrostrahlgerät

2.1 Beschreibung

Das Mikrostrahlgerät hat im Wesentlichen zwei verschiedene Funktionen. Erstens ist es möglich, Strukturen in Oberflächen zu schreiben. Benutzt wird eine Flüssigmetallionenquelle. Das durch Erhitzung flüssig gemachte Metall wird mit einer hohen Spannung (Ziehspannung) aus der Quelle gelöst und ionisiert. Mit einem Ablensystem wird die genaue Stelle der Oberfläche des Targets getroffen. Der geringe Durchmesser des Strahls (ca 1 μm) bewirkt eine hohe Auflösung und eine große Präzision. Hierdurch können sehr kleine Strukturen im Halbleitermaterial erzeugt werden. Außerdem ist es möglich, die Implantation der Ionen ohne Maske durchzuführen. Daher ist es möglich, Fehler in bereits geschriebenen Strukturen zu reparieren.

Die zweite Möglichkeit besteht darin, daß Strukturen gelesen werden können. In der Nähe der Probe befinden sich vier Detektoren zur Messung geladenen Teilchen die von der bestrahlten Oberfläche wegfliegen. Bei herkömmlichen Bildaufnahmen werden die Meßdaten sofort und analog einem Monitorssystem zur Verfügung gestellt. Das hat zur Folge, daß ein Scan solange wiederholt werden muß, solange man das Bild sehen will. Jedes Mal aber wenn die Probe mit dem Strahl in Berührung kommt, bleiben Ionen aus dem Strahl in der Probe zurück. Bei größeren Scanzeiten wird die Probe also mit diesen Ionen verunreinigt oder die Struktur wird zerstört. Darum besteht der Wunsch, die Daten von einem Scan in einen Speicher zu schreiben, damit nach nur einem Scan immer über sie verfügt werden kann. Ein weiteres Vorteil ist, daß die aufgenommenen Bilder verbessert werden können. Dabei kann man z.B. an eine Verbesserung des Kontrastes oder die Eliminierung des Rauschens denken. Die digitalisierten Daten, die vom Detektor-Verstärker kommen, werden von einer für diesen Zweck entworfenen Bildspeicherkarte gesammelt. Die gesamte Scansteuerung und damit die Positionierung des Strahles wird von einem Scansteuermodul übernommen. Abb. 2.1 zeigt den globalen Aufbau des Systems.

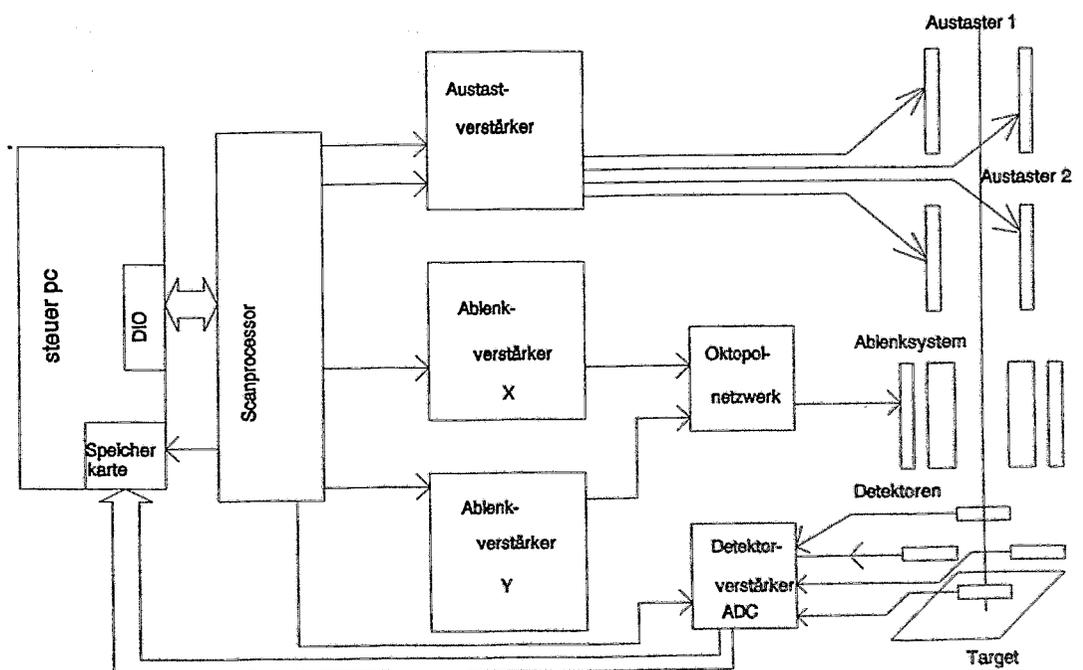


Abb. 2.1 Der globalen Aufbau des Scansystems

Das System wird mit einem 486 PC gesteuert. Die Kommunikation mit dem Scanprozessor findet mit einer IO-Karte statt. Diese Karte und die Bildspeicherkarte sind im Rechner eingebaut. Der Scanprozessor steuert den gesamten Scanvorgang und wird über den Rechner programmiert. Der Austastverstärker steuert die Austaster, die es ermöglichen, die Bestrahlung der Probe zu unterbrechen. Statt zur Probe zu gelangen, wird der Strahl zu den Austastplatten abgelenkt. Damit wird es ermöglicht die Probe zu wechseln oder den Strahl zu einer anderen Stelle zu lenken. Die beiden Ablenkverstärker bestehen aus einem langsamen Verstärker mit einem großen Spannungsbereich und einem schnellen Verstärker mit einem kleinen Spannungsbereich. Dieser Aufbau hängt mit dem Scanfeldaufbau zusammen (Siehe Abschnitt 2.2).

Das Oktopolnetzwerk führt die Spannungen der Verstärker an das Ablenkensystem weiter.

Die Ströme, die von den vier Detektoren kommen, werden verstärkt, in Spannungen umgewandelt und digitalisiert bevor sie in den Zwischenspeicher eingelesen werden. Die Auflösung pro Detektorplatte ist im Moment drei Bits. Vorgesehen ist eine Erhöhung auf vier Bits pro Platte. Alle Bits werden auf einer sechzehn Bit breiten Leitung zum Pufferspeicher in der Bildspeicherkarte geführt. Das Einlesen der Daten darf erst geschehen, wenn sie stabil an den Ausgängen des Detektorverstärkers vorliegen. Für diese Synchronisation sorgt auch der Scanprozessor. Er gibt einen Einlesetakts an die Bildspeicherkarte.

2.2 Der Scanfeldaufbau

Die Positionierung der Probe erfolgt mit einem beweglichen Tisch der in zwei Richtungen gefahren werden kann. Die Koordinaten zwischen $x,y = -80 \text{ mm}$ und $x,y = 80 \text{ mm}$ sind möglich. Mit dem Tisch wird ein Scanfeld ausgewählt. Ein Scanfeld ist eine Fläche, die vollständig mit der Ablenkung des Ionenstrahls erreicht werden kann. Die Größe ist ca. $200 \mu\text{m} \times 200 \mu\text{m}$ bei einer minimalen Schrittweite von 50 nm . Das bedeutet, daß es insgesamt maximal 4000×4000 Scanpunkte gibt.

Ein Scanfeld wird in 31×31 Subscanfelder unterteilt. Diese Felder sind ca. $6.4 \mu\text{m} \times 6.4 \mu\text{m}$ groß und entsprechen 128×128 Scanpunkten. Die Abmessung eines Scanfeldes ist klein gegenüber der Ablenkung wenn ein Subscanfeld ausgewählt wird. Darum übernimmt der langsame Verstärker die Auswahl der Felder, und der schnelle Verstärker den Scan innerhalb eines Subfeldes. Die Aufbau wird in Abb. 2.2 gezeigt.

Ein unabgelenkter Strahl trifft die Mitte eines Subscanfeldes, so daß die Koordinaten auch negativ sein können. Für die interne Ansteuerung des Scanprozessors sind die Subfeldnummern und die Subfeldkoordinaten anzugeben.

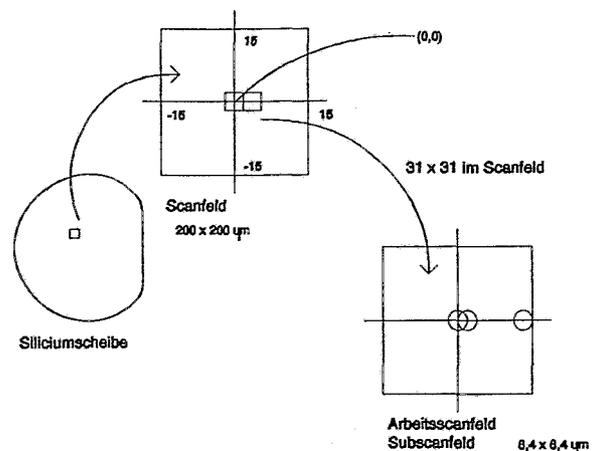


Abb. 2.2 Der Scanfeldaufbau

2.3 Der Scanprozessor [4]

2.3.1 Einleitung

Der Scanprozessor ist ein Teil der gesamten Steuerung für die Ionenmikrostrahlanlage. Die Daten erhält der Scanprozessor vom Parallelport des Rechners. Sechs Bits werden für die Adressierung der verschiedenen Register des Prozessors benutzt, und acht Bits für die Übertragung der Daten. Die Ausgänge sind für die Austaster, das Ablenkungssystem, die Detektor-Verstärker und die Bildspeicherkarte.

Um ein Rechteck zu scannen, sind verschiedene Scanformen möglich. Die einfachste Methode ist ein Linienscan über die ganze Oberfläche. Ein großer Nachteil ist aber, daß nach jedem Ende einer Zeile der Strahl zum Anfang einer neuen Zeile zurückspringen und während dieser Zeit ausgetastet werden muß. Eine bessere Lösung ist die Benutzung einer Meanderform. Das Subscanfeld wird dann im Zick-zack abgescannt. Nur während eines Wechsels des Subscanfeldes muß jetzt noch ausgetastet werden. Die beiden Scanformen werden in Abb. 2.3 und Abb. 2.4 gezeigt.

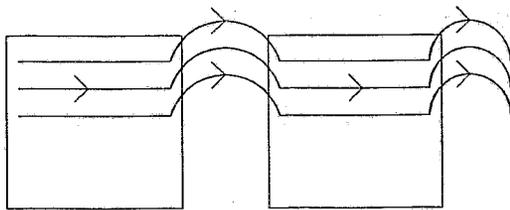


Abb. 2.3 Linienscan

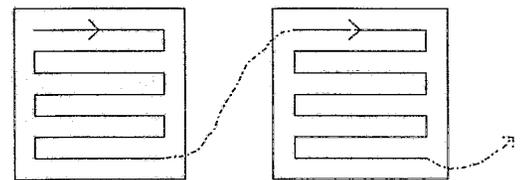


Abb. 2.4 Meanderscan

Wenn alle erforderlichen Daten an den Scanprozessor übergeben sind, kann ihm das Signal zum Scanstart gegeben werden.

2.3.2 . Anschlüsse des Prozessors

In der Kommunikation mit dem Rechner sind die folgenden Signale wichtig:

D0-D7	Bidirektionale Datenleitung zum Beschreiben und Lesen der Register des Scanprozessors.
A0-A5	Adressenleitungen für die Adressierung der verschiedenen Register.
CLEARHI	Ein Hochpegel auf diesem Eingang löscht alle Register und versetzt die Steuerung in den Anfangszustand.
READHI	Mit diesem Signal lassen sich die Register lesen. Der Inhalt wird auf die Datenleitung gelegt. Welches Register gelesen wird, wurde vorher über die Adressenleitungen eingegeben.

WRITECLKHI	Steuersignal zum Beschreiben des Registers, das ist über die Adressenleitungen angegeben ist.
ASFHI	Wenn dieses Signal aktiv wird, wird die Scanfeldadresse für den langsamen Verstärker weitergegeben. Es wird also ein neues Subscanfeld ausgewählt.
STARTHI	Dieses Signal startet einen Scanvorgang. Bevor es aktiv gemacht wird, müssen alle erforderlichen Register beschrieben sein.
ENDHI	Dies ist ein Ausgangssignal und zeigt durch Hochpegel an, daß alle programmierten Scanvorgänge abgearbeitet sind. Es wird auch durch ein aktives Clearhi hochgefahren. Starthi setzt dieses Signal auf Tiefpegel.
SCANHI	Scanhi zeigt das Ende der einzelnen Scanzyklen an.

Neben diesen Signalen gibt es noch einige, die für die Kommunikation zwischen dem Scanprozessor und der Bildspeicherkarte wichtig sind. Das sind die Signale

SCANHP	Dieses Signal ist ähnlich wie Scanhi, doch hier zur Übertragung zum Bildspeicher.
ENDLP	Wie ENDHI, jedoch hier low-aktiv zur Übertragung an die Bildspeicherkarte.
TDVH	Torimpuls für die Analog-digital-Wandlung des Detektorverstärkers Die Breite ist 100ns.
IBTH	Dieses Signal informiert den Bildspeicher, daß die Daten zur Verfügung stehen. Der Speicher wird dann die Daten übernehmen.
STEPXL	Schritimpuls in X-Richtung für die Bildspeicherkarte.
STEPYL	Schritimpuls in Y-Richtung für die Bildspeicherkarte. Wenn Stepxl und Stepyl gleichzeitig auf Tiefpegel gelegt werden, markiert das den Anfang eines Scanvorgang.

2.3.3 Register des Prozessors

Damit der Scanprozessor genügend Informationen hat, um einen Scan durchzuführen, verfügt er über 13 Register, die sowohl beschrieben als auch gelesen werden können. Die Breite jedes Registers ist acht Bits. Die Register sind:

- SWRX** Hier werden die Schrittweite und die Richtung für die X-Koordinaten eingetragen. Die Bits 0 bis 5 enthalten die Schrittweite im Zweierkomplement. Bit 6 steuert die Scanrichtung. Eine '0' bedeutet einen Scan in X-Richtung, eine '1' einen Scan in Y-Richtung. Bit 7 ist ein '*enable' bit für die Inkrementierung. Das bedeutet, daß der Adressenzähler wenn es auf Hochpegel liegt. In diesem Fall wird nur die Anfangskoordinate belichtet.
Adresse : 00H
- SWRY** Schrittweite und Richtung für die Y-Koordinaten.
Adresse 01H
- SKX** Dieses Register enthält die Anfangsadresse für den schnellen Ablenkverstärker der X-Richtung. Die Adresse wird bei jedem Scanstart an den Verstärker weitergegeben.
Adresse 02H
- SKY** Anfangsadresse für den schnellen Y-Verstärker.
Adresse 03H
- ASLX** Hier werden die Daten für den langsamen X-Verstärker bereitgestellt. Sie versorgen die Auswahl der Arbeitsscanfelder und die Linearitätskorrektur für die X-Richtung.
Adresse 04H
- ASLY** Wie ASLX, jedoch in Y-Richtung.
Adresse 05H
- SSZX** Dieses Register enthält die Anzahl der Schritte in X-Richtung. Bei Eingabe einer '0' wird also nur ein Punkt in X-Richtung bestrahlt.
Adresse 06H
- SSZY** Wie SSZX, jedoch in Y-Richtung.
Adresse 07H
- PBZ** Hier wird die Belichtungszeit für einen Pixel eingetragen. Die Belichtungszeit ist:
 $T = (PBZ + 1) * 0,5 \mu s$
Adresse 08H
- SCZ** Es besteht die Möglichkeit, einen Scan zu wiederholen. Hier wird dann die Anzahl der Wiederholungen eingetragen. Bei einem Startsignal wird der programmierte Scanzklus <SCZ>-mal ausgeführt.
Adresse 10H

TAT2V	Bei Austastung des Strahles, muß der zweite Austaster etwas später als der erste eingeschaltet werden. Diese Verzögerung wird in diesen Register eingetragen. Sie beträgt $(TAT2V + 1) * 50$ ns. Adresse 18H
VAS	Verzögerung in der Ablenkspannung, die gegenüber Austaster 1 bestimmt wird. Die Verzögerung beträgt $(VAS + 3) * 50$ ns. Adresse 20H
DATAU	Verzögerungszeit der Datenaustastung. Es wird den Zeitpunkt gegenüber die Veränderung der Ablenkspannung bestimmt. Sie beträgt $(DATAU - 5) * 50$ ns. Adresse 28H

Bei der vorhandenen Hardware gibt es einige Einschränkungen, die eingehalten werden müssen :

1. VAS muß immer größer als '5' sein, wenn mehrere Zyklen programmiert sind.
2. Bei einem Taktfrequenz von 20 Mhz gilt :
PBZ ist kleiner als 1FH (16 μ s),
VAS ist kleiner als 1FH (1,7 μ s)
Wenn unbedingt längere Zeiten notwendig sind, muß eine niedrigere Taktfrequenz gewählt werden.
3. Für die Adressen 00H..07H müssen bei Programmierung die Werte 80H..87H benutzt werden.

2.4 Die Bildspeicherkarte [5]

2.4.1 Einleitung

Während eines Scans müssen die gewonnenen Daten aufbewahrt werden. Das sofortige Einlesen in den Rechner würde ziemlich viel Zeit in Anspruch nehmen, und damit die maximale Scanfrequenz verringern. Darum werden die Daten hardwaremäßig in einem Zwischenspeicher geschrieben. Zu diesem Zweck enthält die Karte einen Adressenzähler, der das Füllen des Speichers koordiniert. Dieser Zähler ist so eingerichtet worden, daß bei einem meanderförmigen Scan die Daten so geordnet werden, daß sie sofort nach Beendigung des Scans in den Rechner eingelesen werden können. Wenn z.B. ein Scan in X-Richtung stattfindet, wird bei jeder Step_x-Impuls der Zähler mit 1 inkrementiert. Wenn der Scan zur nächsten Y-Zeile hüpf, wird die Anzahl der Schritte in einer X-Zeile addiert, und die Zählrichtung des Adressenzähler umgekehrt. Auf diese Weise werden die Daten in die richtige Reihenfolge geordnet. Der Inhalt des Speichers bleibt so lange vorhanden, bis ein neuer Scan durchgeführt wird. Der aktuelle Zustand ist der, daß den Signalen, die von den Detektoren kommen und drei Bits groß sind, vier Bits im Speicher zur Verfügung stehen. Es ist vorgesehen, daß später noch ein Bit hinzugefügt wird. Jedes Speicherelement enthält also $4 \times 4 = 16$ Bits. Die Größe des Speichers ist 128 KB, was bedeutet, daß maximal 16384 Meßwerte gespeichert werden können.

Der Speicher kann über den Rechner sofort gelesen werden. Die Basisspeicheradresse liegt bei der Adresse d000H und die Adresse des Kontrollregisters ist 220H.

2.4.2 Anschlüsse der Bildspeicherkarte

Da sich die Karte im Rechner befindet, stehen Adressen- und Datenleitungen zu Verfügung. Auf der Karte kann mit Hilfe von Schaltern eine Basisadresse gewählt werden, mit der der Speicherbereich anfängt. Auf der Prozeßseite sind folgende Anschlüsse zu finden:

SCANH	Hier wird das Statussignal des Scanprozessors angeschlossen, zur Kontrolle des Ablaufs des Scanprozesses.
ENDL	Wie SCANH
STEPXL	Hier wird das Signal für die Inkrementierung in X-Richtung angeschlossen. Das Signal kommt vom Scanprozessor. Es dient zur Steuerung des Adressenzählers bei der Umwandlung der Meanderstruktur in die normale Zeilenstruktur. Welches von den beiden Signalen STEPXL oder STEPYL zuerst kommt, bestimmt die Scanrichtung.
IBTH	Übernahmetakt vom Scanprozessor für die Übernahme der bereitgestellten Daten vom Detektor-Verstärker.
MESDA01	Hier kommen die Meßdatensignale des Prozesses vom Detektor-Verstärker.
MESDA16	

2.4.3 Das Register der Karte

Die Karte kann sowohl mit dem Scanprozessor zusammenarbeiten, als auch selbständig Meßwerte einlesen. Auf welcher Art und Weise der Speicher funktioniert, kann mit Hilfe eines Registers eingestellt werden. Zur Zeit ist das Register im Speicher vom Rechner auf die Adresse 210H gelegt. Das Register kann beschrieben und gelesen werden. Die Bitbelegung beim Schreiben ist folgende:

RES	Bit 0	Ein Hochpegel generiert einen Resetimpuls für die Karte und löscht sich automatisch nach fünf Takten des Clocksignals.
INDADR	1	Wenn 0, wird das Zählen des inneren Adressenzählers gestoppt, und es werden die Adressenleitungen auf den Speicher geschaltet. Der Inhalt des Speichers kann jetzt in den Rechner eingelesen werden. Wenn 1, wird der innere Adressenzähler als Speicheradresse gewählt und die Auswertung des Übernahmetaktes IBT ist erlaubt.
STEPX	2	Ein Hochpuls auf diesem Eingang hat dieselbe Funktion, wie ein Puls STEPXL vom Scanprozessor. Die Pulsbreite ist vier Taktzeiten.
STEPY	3	Wie STEPX, jedoch für die Y-Richtung.

ENSTX	4	Mit diesem Bit kann die Inkrementierung in X-Richtung ein- oder ausgeschaltet werden. Es ist hoch-aktiv.
ENSTY	5	Wie ENSTX, jedoch für die Y-Richtung.
A15	6	Wenn Bit 1 auf Tiefpegel liegt, ist dieser Anschluß das fünfzehnte Bit der Speicheradresse. Wenn es auf Hochpegel liegt, wird mit diesem Bit der ADD-Impuls des Zählers ein- oder ausgeschaltet.
A16	7	Wenn Bit 1 auf Tiefpegel liegt, ist dieser Anschluß das sechzehnte Bit der Speicheradresse. Wenn es auf Hochpegel liegt, setzt es über eine ODER Funktion das sechzehnte Bit des Adressenzählers.

Beim Lesen stehen die folgenden Informationen zur Verfügung :

ENDL	0	Der Scan ist beendet.
INDADR	1	Dieselbe Funktion wie beim Schreiben.
VZADD	2	Hier kann der Status des inneren Zählers ermittelt werden. Wenn VZADD = 1, zählt der Zähler rückwärts, und wenn VZADD und ADDZ beide auf Tiefpegel liegen, zählt der Zähler vorwärts.
ADDZ	3	Wenn ADDZ = 0, wird mit VZADD die Zählrichtung ermittelt, wenn ADDZ = 1, wird der Inhalt eines Hilfsspeichers addiert. Dieser Hilfsspeicher enthält die Länge einer Scanzeile.
Bits 4,5		Wie beim Schreiben.
Bits 6,7		A15, A16

3. Das Programm Iscan

3.1 Funktion des Programms

Die Hauptaufgabe des Programms ist es, die Daten aus dem Hilfsspeicher der Karte einzulesen und auf dem Bildschirm darzustellen. Um Wünschen des Benutzers entgegen zu kommen, gibt es einige Funktionen mit denen die Darstellung der Daten geändert werden kann. Es ist z.B. denkbar, daß in einem Fall größere Objekte im Bild deutlich gezeigt werden müssen, und in einem anderen Fall möchte man kleine Elemente hervorheben. Diese Wünsche erfordern Funktionen, die die Bilddaten so bearbeiten, daß die wichtigen Punkte sichtbar werden. Für diese Bearbeitung stehen unter anderem Filter zur Verfügung mit den man z.B. den Rauschanteil verkleinern oder Intensitätsübergänge detektieren kann.

Wenn die Daten im Speicher des Rechners zur Verfügung stehen, können sie auch auf der Festplatte gesichert werden, damit man sie jeder Zeit wieder laden kann. Wie später erklärt wird, gibt es verschiedene Dateiformate, um den benötigten Speicherplatz möglichst klein zu halten.

Neben diesen Funktionen, die für die Bearbeitung der Daten nützlich sind, gibt es Hilfsmittel, die bei der Vorbereitung eines Scans benutzt werden, z.B. wenn der Strahl fokussiert werden muß.

3.2 Die Menüs

Die Hauptfunktionen des Programms, wie das Sichern der Bilddaten oder das Starten eines Scans, werden mit Hilfe von Menüs gesteuert. Beim Aufruf wird nach der Initialisierung des Programmstatus das Hauptmenü gezeigt.

3.2.1 Das Hauptmenü

Beim Aufrufen des Programms erscheint auf dem Bildschirm das Hauptmenü aus dem verschiedene Funktionen zur Verarbeitung der Bilddaten gewählt werden können. Das Menü gibt auch den Zugang zum zweiten Menü und zum Kommandomodus, in den das Programm gesetzt werden kann.

In dem Hauptmenü stehen die folgenden Funktionen zur Verfügung:

Menü 2 SCAN	Das zweite Menü mit den Scanfunktionen wird dargestellt.
Read memory	Der Inhalt des Speichers der Bildspeicherkarte wird in den Rechner eingelesen. Solange kein neuer Scan durchgeführt oder der Rechner ausgeschaltet wird, bleiben die Daten im Speicher vorhanden.
enter Filename	Ein neuer Name für die Bilddatendatei kann eingegeben werden. Der Name wird beendet mit <CR>.
Write file	Die Bilddaten werden gesichert. Die Datei hat die Erweiterung 'pic' oder 'bld'.

cOmpress file	Die Datei wird ohne Informationsverluste komprimiert. Der Algorithmus sucht die Änderungen im Signalpegel. Wenn sich ein Signal nicht oft ändert, kann die Komprimierung zu einer erheblichen Speicherreduzierung führen. Andererseits wenn das Signal viele Änderungen hat, z.B. durch Rauschen, kann es passieren, daß die komprimierte Datei größer ist als die normale Datei. Das passiert am schnellsten bei '.pic' Dateien.
Expand file	Eine komprimierte Datei wird wieder in eine normale Datei umgewandelt.
Clear screen	Der Bildschirm wird gelöscht. Das Menü und das Statusfeld erscheinen neu.
Show picture	Das dritte Menü mit dem die Darstellungsart (Farben / Schwarz-weiß) ausgewählt werden kann, wird hervorgerufen. Nach der Eingabe erscheint das Bild.
commanD	Das Programm stellt für die Verarbeitung von Bilddaten noch einige Kommandos zur Verfügung. Diese Funktion versetzt das Programm in den Kommandomodus. Ein Zurückspringen zum Menümodus geschieht mit dem Befehl 'quit' oder mit der <ESC>-Taste.
Add data parts	Die Daten der verschiedenen Detektorplatten können hier addiert werden. Nach der Auswahl werden die Koeffizienten für die Addition über die Tastatur eingegeben. Diese dürfen auch negativ sein. Wenn das Resultat kleiner als '0' ist, wird eine Meldung ausgegeben.
Quit	Das Programm wird ohne Sicherung der Daten verlassen.

3.2.2 Das Scanmenü

Zur Ausführung eines Scans muß der Scanprozessor alle erforderlichen Daten enthalten. Sie werden aus dem Scanfile selektiert und in den Prozessor übertragen. Das Scanmenü bietet dazu die folgenden Funktionen:

Clear screen	Der Bildschirm wird gelöscht. Das Menü und das Statusfeld werden neu gedruckt.
enter Filename	Ein neuer Name für die Scandatei kann eingegeben werden. Die Eingabe wird abgeschlossen mit <CR>.
Select scan	Bei einem Scanvorgang werden nur die Bilddaten vom letzten Scan im Hilfsspeicher der Bildspeicherkarte zur Verfügung stehen. Darum kann für die Aufnahme bei Select scan ein bestimmtes Subfeld ausgewählt werden. Beim Aufrufen dieser Funktion wird die erste Zeile des Scanfiles auf dem Schirm gezeigt. Mit der Maus oder mit den Cursortasten kann die aktuelle Auswahl geändert werden. Wenn die linke Maustaste oder die Leertaste betätigt wird, werden die Parameter des ausgewählten Befehls in den Prozessor übertragen. Zu beachten ist, daß nicht nur der 'scan' Befehl

gezeigt wird, sondern alle möglichen Befehle gezeigt werden. Diese können jedoch nicht selektiert werden, weil sie keine Scan-Information enthalten. Das Programm gibt in diesem Fall eine entsprechende Meldung aus.

Do all scans	Diese Funktion sorgt dafür, daß alle Daten im Scanfile abgearbeitet werden. Dabei wird immer gezeigt, welcher Befehl zur Zeit ausgeführt wird. Vorher fragt das Programm, wie oft der ganze Scanvorgang durchgeführt werden soll.
Print actual scan	Zeigt welcher Scanbefehl zuletzt selektiert wurde.
Measure & read	Hier erfolgt der eigentliche Scan. Dem Scanprozessor wird den Befehl zum Start gegeben und das Programm läuft erst weiter, wenn der Scan beendet ist.
Add to scandata	Es kann passieren, daß die Scanbefehle aus der Datei nicht optimal sind, und z.B. die Pixelbelichtungszeit angepaßt werden muß. Mit 'Add to scandata' kann ein neues Element hinzugefügt werden. Dieses Element erscheint dann immer am Ende der Befehlsreihe. Das Speichern dieser neuen Struktur ist möglich mit:
Write scanfile	Hier kann die ganze Scandatenstruktur auf Festplatte gespeichert werden. Die alte Datei wird überschrieben.
repeat One scan	Um eine Überprüfung von Signalen z.B. bei einem Hardwaretest zu ermöglichen, müssen sie periodisch sein. Das wird durch eine ständige Wiederholung eines Scans erreicht. Die 'unendliche' Wiederholungsschleife endet, wenn die Leertaste gedrückt wird.
Exit to main Menu	Zurück zum Hauptmenü.

3.3 Die Bedienung des Programms

3.3.1 Allgemein

Das Programm kann mit 'iscan' aufgerufen werden. Nach dem Start wird das erste Menü hervorgerufen das links auf dem Bildschirm erscheint. Rechts erscheint eine Statusangabe zur Überprüfung einiger Parameter. Unter dem Menü erscheint die Meldung: 'Your choise ->' Für die Auswahl aus einem Menü gelten die folgenden Möglichkeiten:

1. Die Auswahl wird über Tastatur eingegeben. Die Eingabe des Buchstaben am Ende der Zeile aktiviert die Funktion.
2. Die mit einer unterschiedlichen Farbe angegebene Auswahl kann mit den Cursorstasten geändert und mit <return> aktiviert werden.

3. Das Bewegen der Maus ändert die Auswahl und die Betätigung der linken Maustaste aktiviert diese.

Nach der Abarbeitung einiger Funktionen meldet das Programm, daß es fertig ist, und daß man eine Taste drücken soll. (Ready...hit key) Diese Taste darf auch die linke Maustaste sein.

3.3.2 Die Statusangabe

Beim Programmablauf können mehrere Einstellungen z.B. Dateinamen geändert werden. Um die Verwaltung der Einstellungen zu erleichtern, werden sie auf der rechten Bildschirmseite im Statusfeld abgebildet. So hat der Benutzer einen Überblick über die Dateinamen und andere Programmvariablen. Nach jeder Operation werden die Statusvariablen entsprechend angepaßt. Die folgenden Einstellungen werden gezeigt.

1. Datafilename Dieser Name zeigt mit welcher Datei ab jetzt gearbeitet wird.
2. Loaded file Hier steht der Name der Datei die zuletzt gesichert oder geladen wurde.
3. Kind of file Die Dateien haben verschiedene Formate. Hier kann abgefragt werden, ob eine Datei komprimiert worden ist oder nicht.
4. Array Im Programm wird für die Speicherung der Bilddaten ein Array (eine Matrix) benutzt. Bei einer Dateikomprimierung wird dieses Array benutzt, um ein anderes Format zu erstellen. Die Matrix enthält dann keine Bilddaten sondern Werte, die bei der Komprimierung entstanden sind.
5. Scan direction Die Scans können in X-oder Y-Richtung durchgeführt werden. Siehe Kapitel 2.3.1 Hier wird die Richtung gezeigt.
6. Scanfilename Hier steht der Name des verwendeten Scanfiles. Die Erweiterung heißt immer '.scn'.
7. Number of scans Diese Nummer zeigt wieviel, Zeilen der zuletzt geladene Scanfile enthält. Die Anzahl der Zeilen entspricht der Anzahl der Befehle.
8. Selected scan Hier wird die Nummer, des zur Zeit im Prozessor programmierten Scanbefehls gezeigt.
9. Loaded scanfile Der Name des zuletzt geladenen Scanfiles.
10. Auto startup Gezeigt wird eine der beiden Möglichkeiten 'ENABLED' oder 'DISABLED' Bei 'ENABLED' werden beim Aufruf des Programms die Kommandos in der 'startup.txt' Datei automatisch ausgeführt. Wenn das nicht erwünscht ist, kann der Benutzer es in 'DISABLED' umwandeln.

11. **Picture scale** Die Bilddaten können in verschiedenen Größen auf dem Bildschirm dargestellt werden. Wenn die Skalierung '1' beträgt, stimmt jeder Pixel mit einem Meßpunkt überein.

3.3.3 Einlesen einer Scandatei

Um eine schnelle Abarbeitung der verschiedenen Scans zu ermöglichen, werden die einzelnen Befehle nicht während eines Scans, sondern vorher in eine Datei gespeichert. Bevor ein Scan jetzt durchgeführt werden kann, muß diese Datei eingelesen werden. Diese Funktion ist im zweiten Menü unter dem Namen 'Get file' vorhanden. Dazu muß also zuerst mit 'Menü 2 SCAN' aus dem ersten zum zweiten Menü gesprungen werden. Wenn das Programm den Namen der Datei noch nicht kennt, wird dieser zuerst abgefragt. Die Erweiterung der Scandatei '.scn' braucht nicht eingegeben zu werden, sondern wird vom Programm hinzugefügt. Die Werte aus Dateien die eventuell vorher schon geladen worden sind, werden gelöscht. Der Programmstatus auf der rechten Bildseite übernimmt den Dateinamen und zeigt, wieviel Zeilen der Scanfile enthält.

3.3.4 Selektieren eines Scans

Wenn eine Bildaufnahme gemacht werden soll, wird nicht die ganze Scandatei abgearbeitet, sondern ein Scanbefehl ausgesucht und ausgeführt. Dazu kann mit 'Select scan' im zweiten Menü ein Befehl aus der Datei selektiert werden. Bei einer senkrechten Bewegung der Maus oder mit den Cursorstasten ↑ und ↓ werden die verschiedenen Befehle gezeigt. Wenn der gewünschte Scanbefehl erscheint, wird er mit der Leertaste oder mit der linken Maustaste aktiviert. Das Programm sorgt nun dafür, daß die Daten in den Scanprozessor übertragen werden.

Es muß darauf geachtet werden, daß nicht nur Scanbefehle, sondern auch andere Befehle die nicht selektiert werden können, gezeigt werden. Wenn der Benutzer eine Zeile anders als eine mit dem 'scan ...' zu selektieren versucht, gibt das Programm eine Fehlermeldung aus. Das Selektieren muß wiederholt werden, um einen Scan erfolgreich durchführen zu können.

3.3.5 Einen Scanvorgang starten

Nachdem ein bestimmter Scan selektiert worden ist, steht alles bereit, um einen Scan auszuführen. Mit 'Measure & read' im zweiten Menü wird dem Scanprozessor der Befehl zum Scanstart gegeben. Das Programm zeigt die Meldung, daß gerade gescant wird und fragt den Prozessor ob er schon fertig ist. Wenn nicht, wird nochmals gefragt. Wenn ja, werden die Bilddaten aus dem Bildspeicher in den Rechner übertragen, und es kann mit dem Programm weiter gearbeitet werden.

Neben diesem Einzelscan ist es möglich die ganze Scandatei abzuarbeiten. Dazu wird im zweiten Menü 'Do all scans' gewählt. Bevor der Ablauf gestartet wird, muß die Anzahl der Wiederholungen eingegeben werden. Bei Eingabe einer negativen Zahl wird der Scanvorgang abgebrochen. Während der Abarbeitung der Datei werden die Scandaten auf dem Bildschirm angezeigt. Am Ende sind die Bilddaten vom letzten Scan im Bildspeicher vorhanden, müssen aber noch mit 'Read memory' aus dem ersten Menü in den Rechner übertragen werden.

3.3.6 Sichern der Bilddaten

Die Bilddaten, die sich im Speicher des Rechners befinden, können auf der Festplatte gespeichert werden. Hierfür ist die Funktion 'Write file' zuständig. Wenn der Dateiname, der im Statusfeld bei 'scanfilename' zu lesen ist, noch unbekannt (NOFILE) ist, wird das Programm den Namen abfragen. Er kann ohne Erweiterung eingegeben werden, da das Programm noch das geeignete Datenformat bestimmen muß und dann dementsprechend eine Erweiterung benutzt. Das hat den folgenden Grund: Die gelesenen Bilddaten sind pro Element zwölf Bits breit und erfordern entsprechend ziemlich viel Speicherplatz. An den Namen wird dann die Erweiterung '.bld' zugefügt. Die Wortbreite der verarbeiteten Daten dagegen ist nur vier oder fünf Bits. Diese Daten werden in ein anderes Format umgewandelt und gespeichert. Hier wird die Erweiterung '.pic' an den Namen angefügt. Das bedeutet, daß abhängig von der Größe der Bilddatenelemente ein geeignetes Datenformat benutzt wird, damit die Dateien so wenig wie möglich Speicherplatz belegen. Wenn die Datei unter einem anderen Namen als dem aktuellen gesichert werden soll, kann über 'enter filename' aus dem ersten Menü ein neuer Name eingegeben werden.

3.3.7 Darstellung der Daten

Die Meßpunkte vom Scan können auf dem Bildschirm dargestellt werden. Dabei liefert jeder Meßpunkt ein Pixel. Abhängig vom Meßwert werden die Pixel unterschiedliche Helligkeiten besitzen. In Abb. 3.1 wird der Zusammenhang zwischen den Intensitätswerten und dem Inhalt des Bildes gezeigt. Weil Intensitäten immer größer als Null sind, haben negative Werte keine Bedeutung. Negative Werte können bei der Verarbeitung der Bilddaten entstehen und werden dann gleich Null gesetzt (schwarz). Die Daten, die aus dem Bildspeicher in den Rechner übertragen werden, enthalten noch die einzelnen Meßwerte der vier Detektorplatten. Diese Werte können beliebig addiert werden. Dazu muß im ersten Menü 'Add data parts' gewählt werden. Nach der Aktivierung sind noch die verschiedenen Koeffizienten, mit denen die Werte der einzelnen Platten multipliziert werden, einzugeben. Diese Koeffizienten dürfen negativ sein, was zur Folge hat, daß die Daten der betreffenden Detektorplatten subtrahiert werden. Hierbei ist zu beachten, daß negative Resultate (Pixelintensitäten) keine Bedeutung haben und vom Programm auf '0' gesetzt werden. Wenn dies der Fall ist, wird eine entsprechende Meldung ausgegeben.

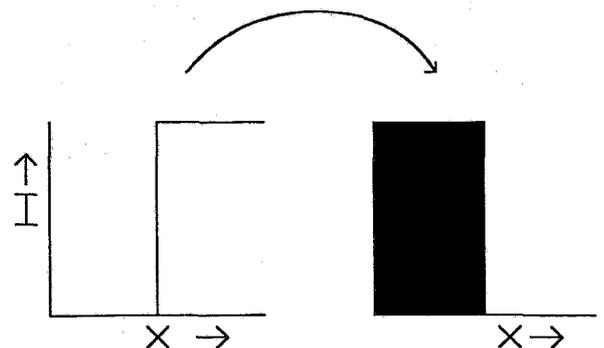


Abb 3.1 Intensität und Bild

Die eigentliche Darstellung der Werte als Intensitäten geschieht, indem im ersten Menü 'Show picture' aktiviert wird. Über ein drittes Menü wird der Anwender gefragt, welche Darstellungsart, Farben oder Graustufen, er sehen möchte. Sofort nach der Eingabe werden die Daten dargestellt. Die Größe des gezeigten Bildes kann mit dem Kommando SCALE geändert werden (Siehe Abschnitt 3.5). Die Höhe und Breite des Bildes wird vom Programm aus dem Scanbefehl übernommen.

Manchmal besteht der Wunsch Abstände im Bild, oder Intensitäten einzelner Pixel zu messen. Dazu werden zwei Cursorlinien eingeblendet. Eine Linie in X-Richtung und eine in Y-Richtung. Die Koordinaten dieser Linien und die Intensität des Pixels (x,y) werden linksoben auf dem Bildschirm gezeigt. Die Linien können mit den Cursortasten oder mit der Maus verschoben werden. Dabei ist noch zu beachten, daß die Bewegungsgeschwindigkeit der Linien nicht die der Maus entspricht. Eine schnellere Bewegung der Maus hat kein Einfluß auf die der Linien. Die Maus langsam zu bewegen, funktioniert am besten. Die Koordinaten der Linien werden entsprechend angepaßt und zeigen immer den Abstand in Nanometer von der aktuellen Position der Cursorlinie bis zum Nullpunkt der Graphik (linksoben). Um einfach senkrechte und waagerechte Abstände messen zu können, wird nach Betätigung der linken Maustaste oder der <CR> Taste ein zweites Linienkreuz eingeblendet. Diese Linien können auch wieder verschoben werden, und ihre Koordinaten und die Intensität werden links-unten auf dem Bildschirm gezeigt. Beim Einblenden des zweiten Paars werden auch die X- und Y-Abstände rechts im Bild dargestellt, bei 'dx' und 'dy'. Durch eine erneute Betätigung der linken Maustaste oder <CR> können die ersten beiden Linien wieder verschoben werden. Ein Beispiel einer Abbildung mit zwei Cursorlinien wird in Abb. 3.2. gezeigt. Mit der rechten Maustaste oder der Leertaste wird diese Darstellungsfunktion verlassen, und man gelangt wieder ins Hauptmenü.

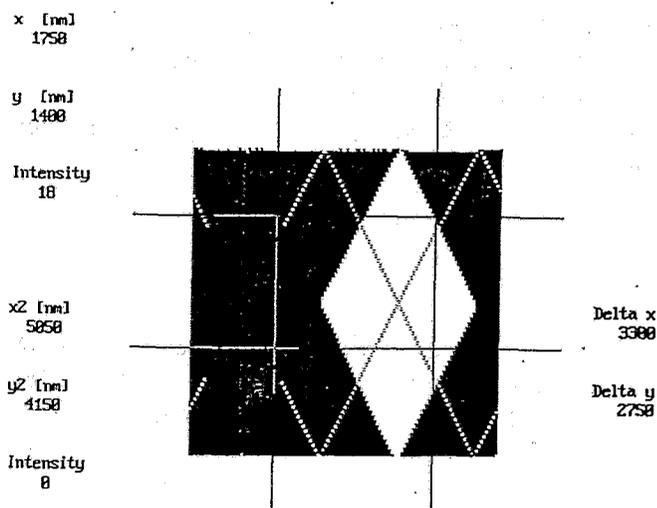


Abb. 3.2 Darstellung der Bilddaten

Wenn die Meßwerte der einzelnen Detektorplatten nicht vorher addiert worden sind, werden auf dem Bildschirm nur die Daten von der ersten Platte gezeigt. Das kommt daher, daß bei der Programmierung einer Intensität nur die ersten vier Bits Bedeutung haben. Die anderen werden nicht beachtet. Es können nur sechzehn Intensitäten oder Farben von schwarz bis weiß gleichzeitig dargestellt werden.

3.3.8 Einlesen einer Bilddatei

Daten, die auf der Festplatte gespeichert sind, können immer wieder eingelesen werden. Als Name wird der benutzt, der im Statusfeld gezeigt wird. Wenn man eine andere Datei laden möchte, muß zuerst der neue Namen unter 'enter filename' im Hauptmenü eingegeben werden. Wenn keine Erweiterung angegeben wird, gibt das Programm dem Namen die Erweiterung '.bld'. Dann wird die Datei geladen, indem 'Get file' aktiviert wird. Die Bilddaten können in verschiedenen Formaten gespeichert sein. Erstens gibt es die '.bld' und '.pic' Dateien die sich darin unterscheiden,

daß die '.pic' Datei nur 4 Bit Elemente hat, und die '.bld' Datei 16 Bit Elemente. Wenn ein neuer Dateiname ohne Erweiterung eingegeben wird, wird davon ausgegangen, daß eine '.bld' Datei gemeint wird. Das Programm versucht also, 'filename.bld' zu laden. Falls dieser File nicht existiert, wird versucht, ob die Datei vielleicht eine '.pic' Erweiterung hat. Das Programm erkennt automatisch den Unterschied zwischen den beiden Typen. Wenn auch dieser File nicht existiert, wird eine Meldung ausgegeben.

Daneben gibt es von den genannten Dateitypen jeweils noch komprimierte Versionen, die ohne Probleme eingelesen werden können. Sie werden beim Einlesen vom Programm ausgedehnt. Nach dem Einlesen der Datei können die verschiedenen Eigenschaften im Statusfeld abgelesen werden.

3.4 Programmdateien

Das Programm benutzt verschiedene Dateien. In einigen davon können Parameter eingestellt werden, die für einen Scanvorgang notwendig sind. Die wichtigste Datei ist die Scandatei in der mehrere Scanbefehle geschrieben sind. Bei der Durchführung eines Kreuzscans, wird ein spezielles Scanfile benutzt. Weiterhin gibt es noch die Möglichkeit, das Statusfeld zu sichern, damit bei einem erneuten Aufruf des Programms, ganz schnell ein neuer Scan gemacht, oder ein gesichertes Bild gezeigt werden kann. Danach können beim Aufruf noch Befehle aus einer zusätzlichen Befehlsdatei ausgeführt werden. Die Bilddaten können, wie schon erwähnt, in verschiedenen Formaten gespeichert werden.

3.4.1 Der Scanfile

Um eine Probe an einer bestimmten Stelle zu belichten, müssen Position und Größe in den Scanprozessor eingetragen werden. Wenn die Daten während eines Scans eingegeben werden müssen, würde das ziemlich viel Zeit in Anspruch nehmen. Darum liest das Programm die Daten aus einer Datei, die zuvor mit einem Editor geschrieben wurde. Damit die Scandaten gut verarbeitet werden können, ist ein spezielles Dateiformat vorgeschrieben. Jede Zeile der Datei beginnt mit einem Kommando, und jedes Kommando ist in eine neue Zeile zu schreiben. Die folgenden Befehle kann das Programm verstehen:

1. scan x1 y1 dx dy nx ny dir tp nw
Mit dem Befehl 'scan' wird ein Rechteck innerhalb eines Subfeldes bestrahlt.
x1, y1 -> Anfangskoordinaten des Rechtecks.
dx, dy -> Schrittweite in X- und Y-Richtung. Ein Schritt entspricht 50 nm.
nx, ny -> Anzahl der Schritte in X- und Y-Richtung.
dir -> Scanrichtung (x oder y). Der Strahl bewegt sich zuerst in diese Richtung.
tp -> Belichtungszeit pro Meßpunkt in μ s.
nw -> Anzahl der Wiederholungen dieses Scans.
2. loop n
Hier fängt eine Schleife an, die n -Mal wiederholt wird. Alle Befehle zwischen LOOP und dem dazugehörigen LEND werden in die Schleife aufgenommen. Es ist möglich, mehrere Schleifen in einander zu programmieren.
3. lend
Die Schleife, die mit dem LOOP Befehl begann, wird abgeschlossen.

4. `field xs ys`
Hier erfolgt der Wechsel des Arbeitsscanfeldes (Subfeld). Die Parameter sind die neuen Subfeldnummern.
5. `wait t`
Der Scanvorgang wird für eine gewisse Zeit angehalten. Die Wartezeit beträgt $t \cdot 5 \cdot 10^{-2}$ s.
6. `break`
Der Scanvorgang wird gestoppt. Das Weiterarbeiten erfolgt nach Betätigung einer Taste.
7. `end`
Dieser Befehl markiert das Ende der Datei. Weitere Befehle werden ignoriert.

Hier folgt ein Beispiel einer Datei, genannt 'scan3.scn'

```

loop 3
scan 1 0 1 1 127 127 x 1 1
loop 4
scan 2 0 1 1 127 127 y 1 1
loop 5
scan 3 0 1 1 127 127 x 1 1
loop 6
scan 4 0 1 1 127 127 y 1 1
field 4 1
wait 20
lend
scan 5 0 1 1 127 127 x 1 1
lend
scan 6 0 1 1 127 127 y 1 1
lend
lend
end

```

Man bedenke, daß die Prozessorregister alle 8 Bit breit sind. Hierdurch sind die Werte der Parameter beschränkt auf höchstens 255. Bei einigen Registern gelten noch weitere Einschränkungen (Siehe Kapitel 2.3.3).

3.4.2 Der Statusfile

Bei einem Neustart des Programms, muß der Benutzer normalerweise jedes Mal den Scanfile einlesen und den gewünschten Scanbefehl selektieren. Um ihm diese Arbeit zu ersparen, sucht das Programm beim Aufruf immer nach einer Datei mit Namen 'status.stt' im aktuellen Verzeichnis. Diese Datei enthält genau die Variablen, die im Statusfeld gezeigt werden. Entsprechend diesen Daten sucht das Programm die gewünschten Dateien und selektiert den entsprechenden Scanbefehl. Wenn der File nicht vorhanden sein sollte, werden die Anfangswerte für die Variablen benutzt. Versucht man ohne einen Scanfile geladen und einen Scanbefehl selektiert zu haben, einen Scan zu starten, wird eine Fehlermeldung über den Bildschirm ausgegeben.

Der Statusfile wird erzeugt, indem das Kommando STA benutzt wird. Sofort wird dann der Inhalt des Statusfeldes gespeichert. Nach der Eingabe des Kommandos 'BYE' wird neben dem Statusfeld auch noch der Scanfile und der Bilddatenfile gesichert. Danach wird das Programm beendet.

Im Programm können als Bilddateinamen nur die Erweiterungen '.pic' und '.bld' und als Scanfilenamen nur '.scn' verwendet werden. Man kann das umgehen, indem man in der Datei 'status.stt' mit einem Editor die Dateinamen ändert. Vorsicht ist dann allerdings geboten. Wenn man die Namen der Scanfiles ändert, muß beachtet werden, daß man zusätzlich die Anzahl der Dateizeilen (number of scans) und die Nummer des selektierten Scanbefehls (selected scan) entsprechend anpaßt. Die Nummer des selektierten Scanbefehls soll nicht über die Anzahl der Scanbefehle übersteigen, da sonst der Rechner abstürzen kann.

Neben dieser Statusdatei gibt es noch eine Datei namens 'status.tmp', die nur existiert während ein Kreuzscan gemacht wird, weil für diesen Scan einige Einstellungen geändert werden müssen. Nach Beendigung des Scans wird der alte Zustand wiederhergestellt.

3.4.3 Die Startupdatei

Es besteht die Möglichkeit nach dem Programmstart, eine Reihe von Befehlen automatisch ausführen zu lassen. Diese Befehle sind zuvor in eine Datei namens 'startup.txt' zu schreiben. Weiterhin muß im Statusfeld die Variable 'Auto startup' ENABLED sein. Als Befehle können alle Kommandos des Programms verwendet werden. Es gibt Kommandos wie z.B. ADS (ADd to Scanstructure) die eine Eingabe von Parametern verlangen. Diese können nur über die Tastatur und nicht aus einer Datei eingegeben werden. Möglich ist z.B. sofort ein Scan zu starten, dann die Daten der einzelnen Detektorplatten zu addieren, zu filtern und auf dem Bildschirm zu zeigen. Die Befehlsreihe würde in diesem Fall lauten : MES ADD MED SHI. Die Befehle aus der Startupdatei können auch mit dem Kommando ECF (Execute Command File) aktiviert werden.

3.4.4 Die Datei cross.scn

Beim Einregeln des Mikrostrahlgerätes wird um den Strahl zu fokussieren ein Kreuzscan gemacht. Das bedeutet, daß ein Gitter abwechselnd in X-Richtung und in Y-Richtung abgescant wird. Dabei werden die Intensitäten in einer Graphik auf dem Bildschirm dargestellt. Die Linien sollten dann so steil wie möglich sein. Je steiler die Linie, je schärfer ist der Strahl eingestellt. Die Datei 'cross.scn' enthält zwei Scanbefehle die einen Scan in X-Richtung und nach Betätigung einer Taste einen Scan in Y-Richtung ausführen. Diese Datei wird automatisch nach Eingabe des Kommandos 'CSC' geladen. Nach dem Verlassen des Kreuzscans wird die alte Scandatei wieder geladen.

3.4.5 Die Bilddatei

Die Datei, die nach der Sicherung der Bilddaten entstanden ist, enthält zusätzliche Informationen die beim Einlesen wichtig sind. Erstens muß dem Programm mitgeteilt werden, ob es sich hier um eine normale oder komprimierte Datei handelt, und ob die Daten als Zahlen (.bld) oder als Charakter (.pic) gespeichert wurden, um die Daten korrekt einzulesen. Die Unterscheidung der beiden letzten Möglichkeiten erhält man aus dem Dateinamen. Den Komprimierungszustand findet man in der ersten Zeile dieser Datei.

Unabhängig vom Format sind alle Dateien als ASCII-Datei gesichert. Sie können also mit einem Textbetrachter gelesen werden.

3.5 Der Kommandomodus

Alle Funktionen, die mit den Menüs aktiviert werden können, lassen sich auch über die Eingabe von Kommandos durchführen. Das hat unter anderem den Vorteil, daß die Kommandos in dem 'startup.txt' oder in einer anderen Befehlsdatei aufgenommen werden können. So besteht die Möglichkeit über einen Texteditor Makros zu schreiben. Diese Makros enthalten dann eine bestimmte Zusammenstellung von Kommandos, die eine gewünschte Aufgabe ausführen. Zum Beispiel könnte man die Befehle : MES ADD MED SHI in einer Datei schreiben. Wenn die 'Startup' aktiv gemacht wurde, wird beim Starten des Programms sofort eine Messung, Addition, Filterung und Darstellung durchgeführt, vorausgesetzt die benötigten Dateien sind im Verzeichnis vorhanden.

Wenn der Kommandomodus über das Menü mit 'command mode' aufgerufen wurde, wird das Menü ausgeblendet. Das Statusfeld bleibt vorhanden. Das Kommando 'quit' oder die Betätigung der <ESC>-Taste bewirkt, daß das Programm wieder in den Menümodus versetzt wird.

Vor allem bei der Bearbeitung von Bildern kann man nicht auf Kommandos verzichten, weil diese wegen der großen Anzahl der Möglichkeiten nicht im Menü aufgenommen sind. Die folgenden Kommandos sind möglich:

ADD	ADD Data parts from detectors. Die einzelnen Detektordaten werden addiert. Als Koeffizienten für die Addition wird für alle Detektoren '1' benutzt.
ADS	ADD to Scanstructure Die Scandatenstruktur wird erweitert. Der Befehl wird vom Programm abgefragt und kann über die Tastatur eingegeben werden.
BYE	Das Statusfeld wird gespeichert und das Programm wird beendet.
CDF <filename>	Change Data Filename Der Name der Bilddatendatei wird geändert in <filename>.
CFR <number>	Color Filter Der Grauwert mit der Nummer <number> wird aus den Daten entfernt.
CLS	CLear Screen Löscht den Bildschirm und druckt das Statusfeld neu.
CON	CONtour Bei einem Schwarzweißbild werden die Konturen hervorgehoben.
CSC	CrossSCAN Ein Kreuzscan wird ausgeführt. Zuerst in X-Richtung und bei Betätigung einer Taste in Y-Richtung. Diese Einstellung wird beendet mit der <ESC> Taste.
CSF <filename>	Change Scan Filename Der Name der Scandatei wird geändert in <filename>

DEF	DEfine Filter Der Benutzer kann hier selber einen 3 x 3 Filter definieren. Der Anwender wird aufgefordert die Koeffizienten über die Tastatur einzugeben.
DLF <filename>	DeLete File Die Datei mit dem Namen <filename> wird gelöscht.
ECF <filename>	Execute Command File Die Befehle, die in der Datei <filename> geschrieben sind, werden ausgeführt.
EXE <filename>[p]	EXEcute programm Ein anderes DOS Programm wird aufgerufen. Es ist möglich, Parameter mitzugeben.
FDX <turns>	Filter Derivative X
FDY <turns>	Filter Derivative Y
HIC	HISTogram Contrast enhancement. Mit Hilfe der Histogrammverteilung wird der Kontrast des Bildes geändert.
HIG <turns>	HIGHpass filter Dieser Filter sorgt dafür, daß einzelne Pixel stärker hervorgehoben werden.
HIS	HISTogram Dieser Befehl erzeugt eine Histogrammverteilung der Intensität.
HV2	Das Hochspannungsprogramm HV2.exe wird gestartet.
INT [number]	INTensity Die Intensität der ersten <number> Pixel werden graphisch gezeigt. Wenn keine Pixelanzahl mitgegeben wird, werden die Pixel von Anfang bis Ende gezeigt. Beachtet werden muß, daß einige zwischenliegenden Pixel nicht gezeigt werden, wenn es mehr als 512 Bilddaten gibt.
INV	INVert Invert invertiert die Intensität des Bildes.
LDD [filename]	LoaD Datafile. Es wird eine Bilddatei geladen. Als Option kann der neue Dateiname eingegeben werden. Wenn nicht, wird der Name aus dem Statusfeld benutzt.

LDS	LoaDScanfile Die Scandatei mit dem Namen der im Statusfeld steht, wird geladen. Die alte Scandatenstruktur im Rechner wird dabei überschrieben.
LOW	LOWpass filter Dieser Filter ist ein Smoothingfilter und sorgt dafür, daß starke Intensitäts-übergänge abgeschwächt werden.
LO2	Lowpass filter 2 Wie LOW aber jetzt mit anderen Koeffizienten.
LSD	LiSt Data files Zeigt alle Bilddateien mit der Erweiterung '.bld' oder 'pic' im aktuellen Verzeichnis.
LSS	LiSt Scan files Zeigt alle Scandateien mit Erweiterung '.scn' im aktuellen Verzeichnis.
MED	MEDian filter Dieser Filter ersetzt die Intensität durch den mittleren Wert eines 3 x 3 Rechtecks.
MES	MEaSure & read Der Scanprozessor startet den Scan. Nach Beendigung liest der Rechner die Daten aus dem Hilfsspeicher.
NOR	NORmalize Die Daten werden so normiert, daß der größte Wert 15 ist.
NUL	Dieser Befehl macht den kleinsten Datenwert gleich Null.
PAS	Print Actual Scan Die aktuelle Scanauswahl wird gezeigt.
QUIT	QUIT Der Kommandomodus wird verlassen und man gelangt wieder ins Menümodus.
RDM	ReaD Memory Der Rechner liest den Inhalt des Hilfsspeichers. Die Anzahl der gelesenen Daten entspricht der Anzahl der Schritte in der X-Richtung und in der Y-Richtung, die im Scanbefehl angegeben sind.
SAL	Scan All Die ganze Scandatei wird abgearbeitet.
SCL <scale>	SCaLe Hier kann man die Vergrößerung eingeben, mit der die Bilddaten dargestellt werden sollen. Es sind nur Ganzzahlen möglich.

SCN <turns>	Es wird <turn> mal gescant. Die Daten der Scans werden gemittelt. Zu beachten ist dabei, daß das Programm nach einem Scan die einzelnen Daten der Detektorplatten sofort addiert. Nach dem Scan sind die Daten also nicht mehr als Meßwerte sondern nur als Pixel vorhanden.
SEL	SElect scan Ein bestimmter Scanbefehl kann mit der Maus oder mit den Cursortasten selektiert werden.
SHI [c/bw]	SHow Image Die Bilddaten werden auf dem Bildschirm dargestellt. Als Option kann eingegeben werden, ob die Darstellung in schwarz-weiß oder in Farben geschehen soll. Ohne Eingabe erfolgt die Darstellung in schwarz-weiß.
STA	store STatus Die Werte im Statusfeld werden in der Datei 'status.stt' gesichert. Diese Datei wird bei einem Neustart des Programms gelesen, sodaß dieselben Dateien wieder geladen werden, und derselbe Scanbefehl selektiert wird.
STI	Show Titel Image Dieser Befehl zeigt das Titelbild dieses Programms.
STR	toggle STaRtup on/off Dieses Kommando schaltet den Autostartup-Schalter um. Im Statusfeld steht ENABLED für eingeschaltet und DISABLED für ausgeschaltet.
THR <level>	THReshold Alle Intensitäten größer als <level> ändern sich in weiß, und alle Intensitäten kleiner als oder gleich <level> ändern sich in Schwarz. Es entsteht also ein Binärbild. Wenn als Schwelle '0' eingegeben wird, nimmt das Programm den Durchschnittswert als Schwelle.
UFI <turns>	User Filter Die Bilddaten werden vom benutzerdefinierten Filter gefiltert. Dieser Filter wurde zuvor mit dem Kommando DEF definiert.
WRD [filename]	WRite Datefile Die Daten werden gespeichert. Als Option kann der Dateiname eingegeben werden. Wenn nicht, wird der Name im Statusfeld benutzt.
WRS	WRite Scanfile Die Scandatenstruktur wird unter dem Namen, der im Statusfeld steht, gespeichert.

4. Bildverarbeitung

4.1 Einleitung

Das Hauptziel der Bildverarbeitung besteht darin, ein Bild so zu verändern, daß es für eine bestimmte Verwendung besser benutzbar ist als das Original. Dabei muß man sich vor der Verarbeitung entscheiden, welche Elemente oder Eigenschaften eines Bildes wichtig sind. Es kann zum Beispiel passieren, daß man große Objekte aus dem Hintergrund hervorheben will, und ein anderes Mal ein kleines Element isolieren möchte. Diese verschiedenen Wünsche erfordern verschiedene Verarbeitungswege. Es gibt Bearbeitungen die sehr oft gebraucht werden. Hierzu gehören unter anderem das Eliminieren des Rauschens oder eine Kontrastvergrößerung.

Die Bilddaten können auf unterschiedliche Art und Weise bearbeitet werden. Erstens gibt es eine eins-zu-eins Verarbeitung. Das bedeutet, daß der neue Intensitätswert eines Pixels nur vom alten Wert abhängt. Jeder Pixel liefert in der neuen Abbildung also einen neuen Pixel. Ein Beispiel solcher Bearbeitung ist das Invertieren der Intensitäten, wobei die neue Intensität durch Subtraktion der alten Intensität von einem Maximalwert entsteht. Weiter gibt es Prozesse, wobei neben einem Pixel noch die Nachbarpixel in die Berechnung aufgenommen werden. Die Berechnungsfunktion wird dann Maske oder Filter genannt. Diese beiden Methoden, die mit Pixelintensitäten arbeiten heißen 'Spatial Domain Methods'. Darüber hinaus gibt es noch die 'Frequency Domain Methods', wobei die Pixelwerte vor der Verarbeitung transformiert werden (z.B. Fourier Transformation). In diesem Kapitel werden einige Funktionen beschrieben, die zur Verarbeitung benutzt werden können.

4.2 Spatial Domain Methods

Der Begriff 'Spatial Domain' bezieht sich auf dem Zustand der Pixel, die zusammen ein Bild formen. Ihre Intensitäten werden meistens als $f(x,y)$ bezeichnet, wobei x und y die Bildkoordinaten sind. Spatial Domain Methods ändern die Intensitäten der Pixel, ohne daß eine Transformation benutzt wird. Sie werden benutzt, um die Grauwerte eines Bildes zu ändern. Bei thresholding z.B. bekommen die Pixel abhängig von ihrem alten Wert einen neuen Wert (schwarz oder weiß). Bei Bearbeitung mit einer Maske ist die Form dieser Maske wichtig. In den meisten Fällen ist sie rechteckig und enthält neun Elemente.

4.2.1 Filtern

Filter gibt es in verschiedenen Arten. Sie können linear oder nichtlinear sein, Objekte wegfiltern oder aber hervorheben. Ein Lowpass-Filter sorgt dafür, daß hohe Frequenzen schwächer werden. Hohe Frequenzen treten dann auf, wenn die Intensität der Pixel sich in einer bestimmten Richtung schnell ändert. In einem Bild, wo es einen Übergang von schwarz nach weiß gibt, wird es entsprechend hohe Frequenzen geben. Nach eine Lowpass-Filterung werden an diesem Übergang noch Zwischenwerte sichtbar. Sehr oft werden als Filter 3×3 Masken benutzt. So eine Maske wird in Abb. 4.1 gezeigt.

w1	w2	w3
w4	w5	w6
w7	w8	w9

Abb. 4.1 Felder eines Filters

Wenn es Lowpass-Filter gibt, gibt es natürlich auch Highpass Filter. Hier werden dann die hohe Frequenzen nicht gesperrt, sondern verstärkt und die niedrigen Frequenzen geschwächt. Niedrige Frequenzen repräsentieren langsame Intensitätsübergänge. Die Berechnung des neuen Bildes erfolgt in beiden Fällen durch die Addition der Produkte von Maskenkoeffizienten und Pixelintensitäten. Wenn man die Koeffizienten der Maske aus Abb. 4.1 betrachtet, wäre das Resultat der Maske

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9. \quad (4.1)$$

Der Wert R wird dem Pixel, der in der Mitte der Maske liegt, zugeordnet. Die Maske liegt z.B. am Anfang eines Bildes auf die Pixel (0,0) bis (3,3). Dann wird die Maske um einem Pixel verschoben. Das geht so weiter, bis alle Pixel in die Berechnung aufgenommen worden sind. Der neuen Wert kommt am Anfang im neuen Bild auf Platz (2,2). Das hat zur Folge, daß die Ränder des Bildes verschwinden. Es kann vorkommen, daß das Resultat einer Filterung mit diesem Filter negativ wird. In diesem Fall begrenzt man es auf Null.

Lowpass-Filter, auch Smoothing-Filter genannt, werden oft benutzt, um scharfe Intensitätsübergänge abzuschwächen. Das kann nützlich sein, um kleine Objekte in einem Bild verschwinden zu lassen, oder kleine Lücken zu schließen. Auch ist es möglich, mit diesen Filtern Rauschen zu vermindern. Wenn man einen 3 x 3 Filter betrachtet, hat man die einfachste Form wenn alle Koeffizienten gleich '1' sind. Um das Resultat noch sinnvoll zu machen, muß die Summe R noch durch '9' geteilt werden, weil man sonst über die Intensitätsgrenze hinaus kommt. Bei dieser Methode wird also für das Pixel der Mittelwert des alten Pixels und der umliegenden Pixel genommen. Das Resultat einer Lowpass-Filterung wird in Abb. 4.2 gezeigt. Das Bild links zeigt ein Kreuz mit abrupten Übergängen. Im rechten Bild sind die Übergänge schwächer geworden. Das Kreuz wurde im Programm zweimal mit dem Befehl 'LOW' gefiltert.



Abb. 4.2 Das Resultat eines Lowpass-Filters

Highpass-Filter sind in der Form nicht anders als Lowpass-Filter. Nur die Koeffizienten sind unterschiedlich. Das zentrale Pixel hat einen positiven Koeffizient, und die umliegenden Pixel negativen Koeffizienten. Nach dem Filtern sind in einem konstanten Hintergrund kleine Objekte besser sichtbar geworden. Die benutzten Bilder sollten aber kein Rauschen haben. Weil das auch kleine Objekte mit scharfen Übergänge sind, werden sie auch verstärkt. Das Kreuz aus Abb. 4.3 wurde mit einem Highpass-Filter gefiltert. Nachher ist das Kreuz nur noch sehr schlecht im Rauschen zu erkennen. Wird andererseits ein Kreuz ohne Rauschen gefiltert, werden nur die Übergänge sichtbar gemacht. Ein solcher Filter kann also auch als Differentialfilter für beide Richtungen X und Y benutzt werden.

Eine andere Möglichkeit besteht darin, das Resultat einer Lowpass-Filterung vom Originalbild zu subtrahieren. So werden niedrige Frequenzen teilweise wieder ins Bild eingefügt, wo sie beim Highpass-Filter ganz verschwunden wären. Diese Methode wird 'High-Boost Filtering' genannt.

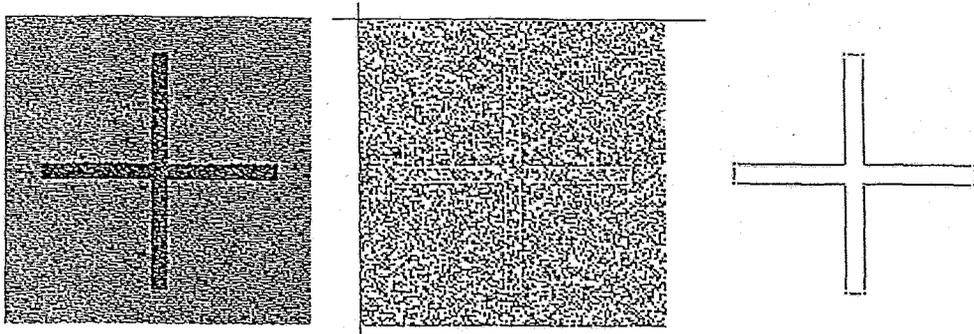


Abb. 4.3 Die Resultate eines Highpass-Filters

Es ist schon erwähnt, daß High-Filter auch als Differentialfilter benutzt werden können. Durch Anpassung der Koeffizienten können die Richtungen X und Y unterschieden werden. Eine Eigenschaft dieser Filter ist, daß die Summe der Koeffizienten gleich Null ist. Abb. 4.4 zeigt ein Beispiel. Der linke Filter bemerkt Änderungen in der Y-Richtung und der rechte Filter in X-Richtung. Dieser Filter hat den Name 'Sobel Filter'. Im Programm sind fünf Filter vordefiniert. Zwei Lowpass-Filter (LOW, LO2), ein Highpass Filter (HIG) und Zwei Differentialfilter (FDX und FDY). Weiter ist es möglich, mit dem Kommando 'DEF' ein zusätzliches Filter zu definieren. Zuerst werden die Koeffizienten von z_1 bis z_9 eingegeben, und danach der Multiplikationsfaktor. Dieser Filter bearbeitet die Daten, indem man das Kommando 'UFI' eingibt.

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Abb. 4.4 Sobelfilter

Wenn man Rauschen wegfiltern möchte, kann man einen Lowpass-Filter benutzen. Ein Nachteil ist aber, daß alle scharfen Intensitätsübergänge verwaschen. Eine andere Möglichkeit Rauschen zu beseitigen, ist die Benutzung eines Mittelwertfilters. Er berechnet den statistischen Mittelwert des Pixels und seiner acht Nachbarn. Erst werden die Werte auf Größe sortiert und danach wird der Mittelwert für das neue Pixel benutzt. Wenn die Werte z.B. 10, 15, 20, 20, 20, 20, 20, 25, 100 betragen, ist der Mittelwert 20. Im Gegensatz zu den bisher besprochenen Filter ist dieser nicht linear. Das bedeutet, daß Information verloren geht. Jetzt ist gut einzusehen, daß bei dieser Filtermethode Pixel, die sich von einem gleichmäßigen Hintergrund unterscheiden, weggefiltert werden. Sie bekommen denselben Wert wie der Hintergrund. Abb. 4.5 zeigt ein Beispiel.

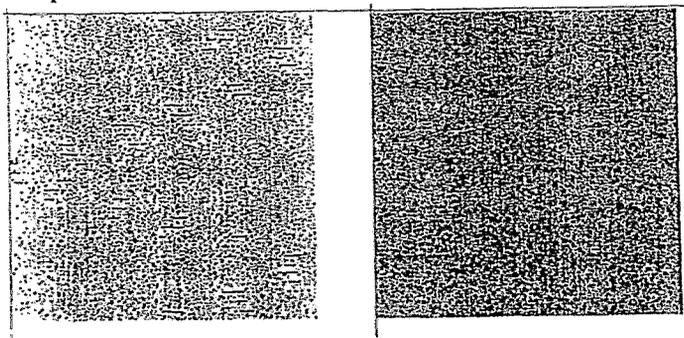


Abb. 4.5 Das Resultat eines Mittelwertfilters

4.2.2 Histogrammfunktionen

Ein Histogramm zeigt, wie oft ein bestimmter Grauwert auftritt. Wenn z. B. der Wert '10' hundert Mal vorkommt, enthält die Histogrammklasse '10' den Wert '100'. Auf dieser Art kann man einen Eindruck von der Grauverteilung in einem Bild bekommen. Diese Verteilung sagt etwas über die Charakteristik des Bildes aus. Wenn die Klassen am Anfang des Histogramms viele Elemente haben, wird das Bild ziemlich dunkel sein, haben die Klassen der hohen Werte viele Elemente, ist es ziemlich hell. Wenn ein Bild einen schlechten Kontrast hat, zeigt sich das im Histogramm. Die Form ist dann eng. Auch wenn diese Darstellung nur allgemeine Information enthält, kann man mit ihr den Kontrast eines Bildes verbessern. Diese Methode der Kontraständerung heißt Histogram Equalisation.

Das Ziel dieser Bearbeitung ist, möglichst eine gleichmäßige Verteilung der Grauwerte zu schaffen. Weil ein Histogramm die Anzahl eines bestimmten Wertes enthält, kann man die Wahrscheinlichkeit berechnen mit der ein bestimmter Wert im Bild auftritt. Die Wahrscheinlichkeit ist

$$p_r(r_k) = \frac{n_k}{n}$$

Wobei p_r die Wahrscheinlichkeit ist, daß der Wert r_k auftritt, n_k ist die Anzahl Elemente mit diesem Wert und n die Gesamtzahl aller Elemente. Die neuen Grauwerte werden jetzt aus diesen Wahrscheinlichkeiten berechnet. Dabei repräsentieren die kumulativen Wahrscheinlichkeiten die neuen Grauwerte, die dann zwischen '0' (schwarz) und '1' (weiß) liegen. In einer Formel sieht dies so aus:

$$S_k = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j)$$

Der größte Wert von k ist dann die Anzahl der Grauwerte. Abb. 4.6 zeigt den Effekt eines derartigen Prozesses. Das erste Bild ist ziemlich hell, was am Histogramm zu sehen ist. Das zweite Bild hat eine andere Verteilung und entsprechend ein anderes Histogramm.

Das Programm stellt das Kommando 'HIS' zur Verfügung um ein Histogramm darzustellen, und das Kommando 'HIC' um eine Histogram Equalisation durchzuführen.

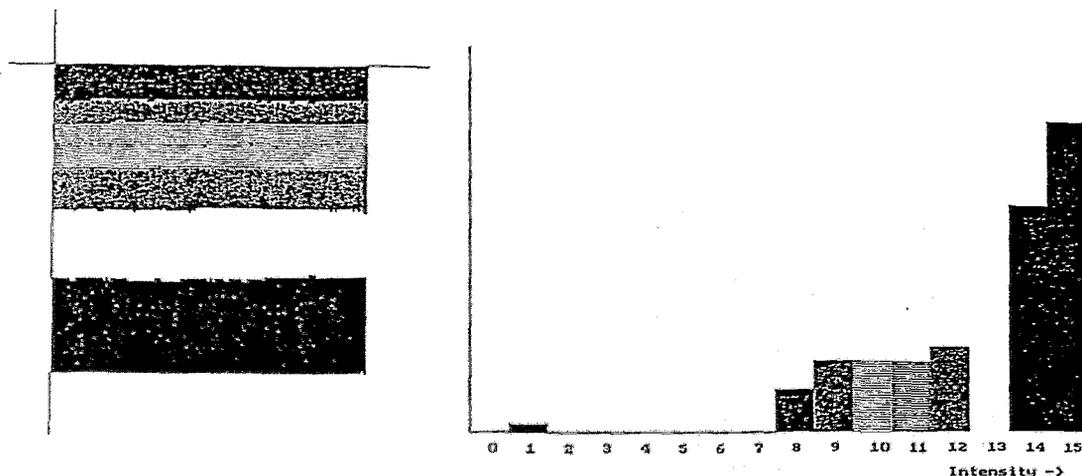


Abb. 4.6a Ein Bild und die Histogrammverteilung

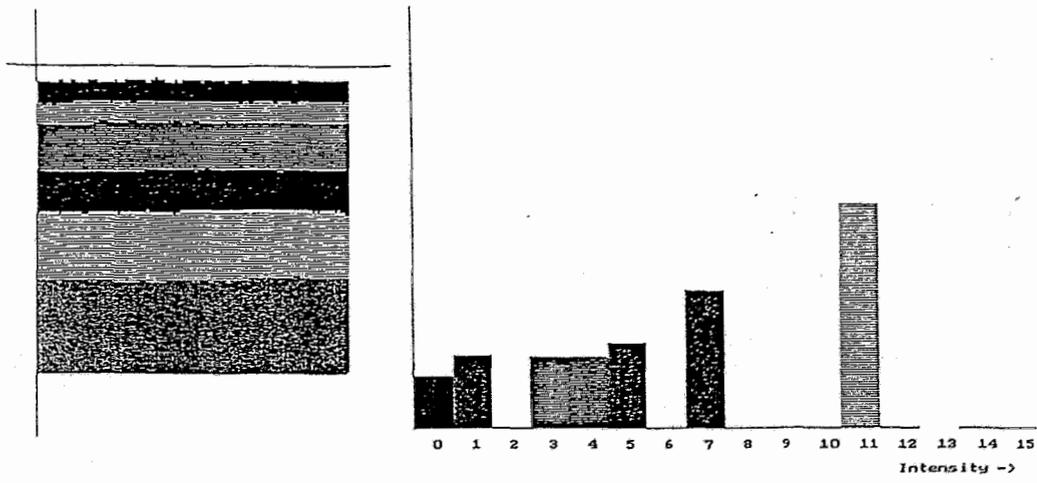


Abb. 4.6b Eine neue Verteilung

5. Aufbau des Programms

5.1 Das Einlesen der Scandatei

Die Daten, die aus der Scandatei gelesen werden, werden in eine Variablenstruktur gespeichert. Weil die Menge der Daten unterschiedlich groß sein kann, weiß man im voraus nicht, wieviel Speicherplatz im Rechner reserviert werden soll. Das ist der Grund, weshalb für die Verwaltung der Scandaten eine sogenannte Zeigerstruktur aufgebaut wird. Jeder Befehl aus der Datei wird dabei in einer Struktur gespeichert. Bei der Verwaltung dieser Struktur benutzt das Programm nicht die Elemente selbst, sondern die Anfangsadressen wo sich diese Elemente befinden. Eine Variable, die diese Anfangsadresse enthält, heißt 'Zeiger'. Abb. 5.1 zeigt den Zusammenhang zwischen einer Variablen und dem dazugehörenden Zeiger.

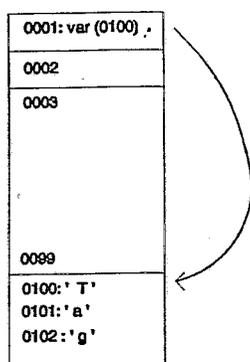


Abb. 5.1 Zeiger und Variablen

Auf der Adresse '0001' befindet sich eine Variable 'var' die den Wert '100' enthält. Das ist die Adresse, wo das Wort 'Tag' anfängt. Die Variable 'var' ist ein Zeiger auf 'Tag' und wird statt 'Tag' vom Programm benutzt. Genau so werden Zeiger definiert, die auf die Strukturelemente zeigen, nur zeigen sie nicht auf ein Wort, sondern auf eine Struktur aus Charakterketten und Zahlen.

Um die Datenelemente zu verknüpfen, werden die Adressen des vorigen und des nächsten Elementes an die Struktur hinzugefügt. Auf dieser Weise bekommt man eine Kette von Strukturen. Der Anfang und das Ende der Kette wird markiert mit einem 'NULL-Zeiger', was bedeutet, daß dieser Zeiger den Wert '0' enthält. Beim Aufbau der Kette, reserviert das Programm für jedes neue Element einen gewissen Speicherbereich. So wird erreicht, daß abhängig von der benötigten Speichergröße, Speicherplatz reserviert wird, und man nicht von einem festen Speicherbereich abhängig ist. Bei Beendigung des Programms werden diese Bereiche wieder freigegeben. Diese Art der Speicherverwaltung heißt 'dynamische Speicherverwaltung'. Wie die Kette in diesem Fall aussieht, zeigt Abb. 5.2

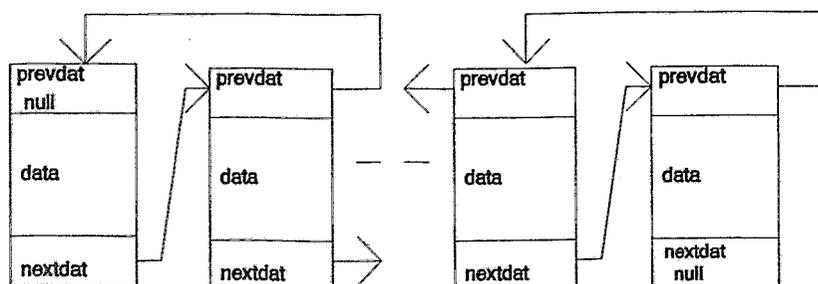


Abb. 5.2 Die Scandatenstruktur

Das Programm benutzt einige globale Variablen, welche die Adressen vom ersten, letzten und aktivierten Element enthalten. Wenn eine neue Datei eingelesen wird, kann das Programm mit diesen Variablen, die alte Struktur löschen. Nach dem Einlesen der neuen Datei werden ihnen neue Werte zugeordnet. Weil für alle Elemente in der Struktur der Speicherplatz einzeln reserviert wurde, muß die Freigabe des Speichers auch pro Element stattfinden. Das Einlesen der Datei macht die Funktion 'readfilesca', wobei abhängig vom eingelesenen Befehl auch noch die benötigten Parameter gelesen werden. Als Ergebnis gibt die Funktion den Anfangszeiger und markiert mit einer Variablen das Ende der Struktur.

Nachdem die Scandaten in den Rechner eingelesen worden sind, kann die Struktur erweitert werden. Hierzu braucht man nicht die Datei zu ändern und neu einzulesen. Die Variable 'lastscan' enthält die Adresse vom letzten Element, das immer den Befehl 'end' enthält. Dieses Element wird gelöscht, und das neue Element kann an das Ende der Kette angebunden werden. Diese Prozedur wiederholt sich, bis keine Daten mehr eingegeben werden. Nach jeder Eingabe eines Befehls wird die Variable 'lastscan' angepaßt, damit sie immer auf das letzte Element zeigt. Die Erweiterung der Struktur findet mit in der Funktion 'fill_scan_structure' statt.

5.2 Verarbeitung der Scandaten

Bei der Ausführung der Befehle gibt es die Möglichkeit, entweder die gesamten Scanbefehle auszuführen, oder einen einzelnen Scan zu selektieren und diesen zu starten. Bei der ersten Möglichkeit wird mit der Maus eine Auswahl getroffen. Das Programm fragt ständig die Koordinaten der Mausposition ab. Der aktuelle Zustand der Maus kann dann aus einer Variablen gelesen werden. Eine Änderung in der Y-Koordinate des Mausursors beim Bewegen der Maus bewirkt das Durchlaufen der Struktur. Nachdem ein Befehl selektiert ist, soll er in den Scanprozessor übertragen werden. Das erfolgt mit der Funktion 'progre'. Danach kann der Scan gestartet werden.

Bei der zweiten Möglichkeit erfolgt die Abarbeitung der Scandatei vom Anfang bis zum Ende. Diese Arbeit wird von der Funktion 'test_all_scan' übernommen. Wenn das Programm den Befehl 'LOOP' begegnet, muß der dazugehörige Befehl 'LEND' gesucht werden. Wenn mehrere Schleifen programmiert sind, befinden sie sich immer ineinander und dürfen sich nicht kreuzen. Das bedeutet, daß beim Suchen nach dem zugehörigen Schleifenende, immer vom Ende der Datei aus gesucht werden muß. Der zuerst gefundene 'LEND'-Befehl markiert dann das Ende der Schleife. Alle Befehle zwischen dem Anfang und dem Ende der Schleife werden dann als Subdatei behandelt. Sie werden mit einem rekursiven Aufruf der Funktion 'test_all_scan' verarbeitet. Diese Methode ermöglicht es mehrere Schleifen zu programmieren. Die Tabelle 5.1 zeigt ein Beispiel wie diese Methode funktioniert.

Aufruf test all scan	Erstes Mal	Zweites Mal	Drittes Mal
Dateiabschnitt	scan 1.... loop 2 scan 3.... scan 4.... loop 5 scan 6.... scan 7.... lend scan 8.... lend end	scan 3.... scan 4.... loop 5 scan 6.... scan 7.... lend scan 8....	scan 6.... scan 7....
Anfang der Schleife	scan 1	scan 2	scan 6
Ende der Schleife	end	scan 8	scan 7
Anzahl der Wiederh.	1	2	5

Tabelle 5.1 Verarbeitung von Schleifen in einer Scandatei.

5.2.1 Programmierung des Scanprozessors

Die Kommunikation zwischen dem Scanprozessor und dem Rechner findet mit einer Interface-Karte statt. Im normalen Betriebsmodus gibt die Karte die Daten des Rechners an den Scanprozessor weiter. Im Reset-Modus ist es möglich, einzelne Bits des Scanprozessors auf Hoch- oder Tiefpegel zu legen. Mit Hilfe dieser beiden Modi werden auch die Scandaten in den Prozessor übertragen. Zuerst werden im normalen Modus die Werte der Register bereitgestellt und danach wird im Reset-Modus die Leitung 'WRITECLKHI' des Scanprozessors eine kurze Zeit aktiviert. Diese Programmierung übernimmt die funktion 'writeaddr'. Auch die anderen Steuersignale des Prozessors werden im Reset-Modus der Interfacekarte durchgeführt. Der Scan startet, wenn an dem Pin 'STARHI' des Prozessors kurzzeitig einen Hochpegel gelegt wird.

Die Interfacekarte kann auch Daten vom Prozessor in den Rechner übertragen, was auch nach einem Scanstart passiert. Das Programm liest ständig die Ausgangssignale des Prozessors und kontrolliert, ob das Pin 'ENDHI' auf Hochpegel liegt. Das ist dann der Fall wenn der Scan zu Ende ist.

Wenn ein Wechsel des Arbeitsscanfeldes erfolgen soll, wird nach der Datenübertragung zum Prozessor der Pin 'ASFHI' kurz auf Hochpegel gelegt. Die neuen Koordinaten werden damit in die beiden langsamen Verstärker eingetragen. Das passiert mit dem Befehl 'field ...'.

5.3 Die Menüsteuerung

Das Programm wird hauptsächlich mit Menüs betrieben. Die Auswahl kann auf dreierlei Weise erfolgen. Erstens ist es möglich, mit der Maus auszuwählen. Dazu wurde über den Mausinterrupt die Bewegungsbereich des Mausursors so programmiert, daß sie über den ganzen Menübereich bewegt werden kann. Bei der Abfrage des Cursors wird dann die Auswahl, die sich auf dieser y-Koordinate befindet, hervorgehoben. Gleichzeitig wird der Zustand der linken Maustaste geprüft. Wird sie gedrückt, wird die Auswahl aktiviert. Dazu wird im Programm die Y-Koordinate weiterverarbeitet. Zunächst wird geprüft, welches Menü zur Zeit aktiv ist. Abhängig davon wird der Buchstabe, welcher die gewählte Funktion markiert, weitergegeben. Dazu hat jedes Menü eine eigene Funktion. Als letztes werden die Buchstaben in den Funktionen 'main', 'menu2' oder 'image' weiterverarbeitet.

Die zweite Möglichkeit besteht darin, daß der Menücursor mit den Cursortasten verschoben werden kann. Bevor das Programm den Mauszustand abfragt, wird daher geprüft, ob gerade eine Taste gedrückt wurde. Sollte das eine Cursortaste sein, wird der Mauscursor angepaßt. Ist es die Eingabetaste, wird die Auswahl aktiviert.

Das Abfragen der Tastatur ermöglicht auch die dritte Methode. Wenn sofort der Buchstabe, der im Menü groß geschrieben wurde, gedrückt wird, kann die Auswahl sofort durchgeführt werden.

5.4 Einlesen aus dem Bildspeicher

Der Pufferspeicher ist 128KB groß und besteht aus 128K Speicherplätzen von je 8 Bits. Diese Wortbreite des Speichers entspricht der halben Breite vom Datenwort des Detektor-verstärkers. Ein Meßwert von den vier Detektorplatten nimmt also zwei Speicherplätze ein und befindet sich daher immer auf einer geraden Adresse. Wenn der Wert '0' ins Kontrollregister der Bildspeicherkarte geschrieben wird, kann der Speicher wie ein normaler Speicher im Rechner behandelt werden. Das Kontrollregister befindet sich auf der Rechneradresse 300H. Die Basissegmentadresse des Speichers ist D000H. Bei der Benutzung dieser Verschiebung können die Daten mit einer einfachen Schleife eingelesen werden. Im Programm werden sie als ganze Zahlen in eine Matrix gespeichert. Die vier Werte der Detektoren befinden sich nebeneinander und bilden zusammen eine Zahl in einem Matrixelement. Um die verschiedenen Daten wieder aus der Zahl zu filtern, werden die Bits so verschoben, daß die vier benötigten Bits auf die vier niedrigsten Bitstellen gelangen. Das erfolgt bei der Addition der Daten der einzelnen Detektorplatten in den Funktionen 'firstnibble' bis 'fourthnibble'.

5.5 Abbildung der Daten

Die Meßwerte, die von den Detektorplatten kommen, repräsentieren Intensitäten und können direkt auf dem Bildschirm dargestellt werden. Wird nicht das Kommando 'SHI' benutzt, um die Daten darzustellen, fragt das Programm den Benutzer in einem dritten Menü, ob er das Bild in Graustufen oder Farben sehen will. In einem C-Programm können insgesamt sechzehn Farben gleichzeitig dargestellt werden. Jede Farbe hat dabei eine Nummer. Für die Darstellung in Grauwerten müssen die Farbregister der Graphikkarte neu definiert werden. Erst dann erhält man die richtigen Graustufen von schwarz bis weiß. Die Daten können mit der C-Funktion 'putpixel' auf einer bestimmten Stelle auf den Bildschirm plaziert werden. Die Breite des Bildes entspricht der Anzahl der Schritte in X-Richtung und die Höhe der Anzahl der Schritte in Y-Richtung. Diese beiden Größen können dem Scanbefehl entnommen werden.

Diese 1:1-Darstellung benutzt maximal 128 x 128 Bildpunkte von insgesamt 640 x 480 Bildschirmpunkten. Das Bild ist darum sehr klein. Die Variable 'scale' ermöglicht daher eine vergrößerte Darstellung des Bildes. Wenn Breite und Höhe zweimal so groß sein sollen, erscheint ein Meßwert als ein Rechteck von vier Pixel auf dem Bildschirm. Sind Breite und Höhe drei mal so groß, muß ein Wert aus neun Pixel bestehen. Das erreicht man, indem erstens die Werte der Bildzeile 'scale' mal hinter einander auf dem Bildschirm plaziert werden, und zweitens diese gesamte Zeile auch 'scale' Mal geschrieben werden, sodaß jeder Meßwert ein Rechteck der Größe 'scale * scale' bildet. Diese Arbeit übernimmt die Funktion 'image'.

Nach der Darstellung der Daten auf dem Bildschirm, zeichnet die Funktion 'distance measure' zwei Cursorlinien. Die Linien verschieben sich, wenn der Mauscursor sich ändert, oder wenn die Cursortasten gedrückt werden. Das Verschieben besteht darin, daß die alte Linie gelöscht und die neue Linie geschrieben wird. Da sie ins Bild geschrieben werden, muß beim Löschen darauf geachtet werden, daß der Bildteil, der sich 'unter' der Linie befand, wieder sichtbar wird. Das erreicht man, indem man die Linien nicht mit einer festen Farbe zeichnet, sondern sie als XODER Funktion mit der ursprünglichen Bildfarbe verknüpft. Wenn jetzt eine Linie zum zweiten Mal gezeichnet wird, verschwindet sie, und das alte Bild ist wieder sichtbar. Ein zusätzlicher Vorteil ist, daß die Linie immer eine andere Farbe als die Bilddaten hat, und daher immer sichtbar ist. Wenn die Koordinaten eines Bildpunktes bekannt sind, kann man damit die dazugehörige Meßwertnummer in der Matrix finden. Die Nummer ist dann :

$$Y * X_{\text{laenge}} + X$$

Der Intensitätswert des Meßpunktes kann jetzt auch auf dem Bildschirm angezeigt werden. Die Koordinaten der zwei Linienpaare werden in verschiedenen Variablen gespeichert. So können durch Subtraktion die Abstände in X- und Y-Richtung berechnet werden.

5.6 Sichern und Laden der Bilddaten

Bevor die Daten auf die Festplatte gesichert werden, wird geprüft, welches Format am besten benutzt werden sollte. Wenn die Werte nicht die Zahl 255 übersteigen, können sie als ASCII-Zeichen in eine Datei gesichert werden. Jedes Element braucht dann nur 1 Byte. Werte die neu aus dem Pufferspeicher gelesen werden, sind jedoch größer und erfordern mehr Platz. Sie werden dann als Ganzzahlen in Hexadezimalform gespeichert und brauchen in der Datei vier Byte pro Wert. Nachdem bekannt ist, welches Format passend ist, ergänzt das Programm die Erweiterung. Die Information über das benutzte Format wird in einer Struktur gesichert. Aus dieser Struktur kann dann die Funktion 'writefile', die die Daten sichern soll, das Format erkennen.

Beim Einlesen einer Datei sind Steuerzeichen, wie das 'End Of File'-Zeichen, auch Charakterzeichen. Um zu verhindern, daß diese Charakterzeichen in der Datei das Einlesen beeinflussen, werden alle Werte beim Schreiben mit der Zahl 65 addiert. Das ist möglich weil Charakterzeichen auch durch Zahlen repräsentiert werden. Wenn die Daten wieder eingelesen werden, muß noch 65 subtrahiert werden damit die Bilddaten ihren alten Wert haben.

Die Dateien können in Normalform oder in komprimierter Form existieren. Dieser Unterschied erfordert auch eine andere Methode beim Einlesen der Daten. Bevor die ersten Bilddaten eingelesen werden soll, der Rechner also wissen, um welches Dateiformat es sich hier handelt. Dazu wird diese Information beim Schreiben zuerst gesichert. Als zweites werden die Breite und Höhe gesichert, denn wenn eine neue Datei eingelesen wird, soll diese Information für die Darstellung vorhanden sein.

Man kann zusammenfassen, daß es vier verschiedene Dateiformate gibt. Das Programm muß in der Lage sein, alle vier Dateiformaten einzulesen. Um das ohne Probleme zu schaffen, wird, nachdem die Datei geöffnet ist, untersucht um was für eine Datei es sich handelt. Die Funktion 'get_file_type' schließt aus dem Namen, ob es eine Charakter- oder Ganzzahldatei ist. Das erste Wort in der Datei bestimmt dann, ob es sich um eine komprimierte oder normale Datei handelt. Diese Informationen werden in die Struktur 'file_info' gespeichert und von der Funktion 'readfile' benutzt, um die Daten einzulesen. Wenn es sich um eine komprimierte Datei handelt, wird diese beim Einlesen expandiert. Die genannten Methoden zur Komprimierung und Expandierung werden im Kapitel 5.7 beschrieben. Weil mit dem Mikrostrahlgerät experimentell bisher nur einmal ein Bild gelesen werden konnte, wurden zum Testen des Programms softwaremäßig Bilddateien

erzeugt. Diese Dateien enthalten jedoch keine Information über die Größe und über die Form der Datei. Bei der Ermittlung des Dateityps wird erkannt, daß die erste Zeile in diesen Dateien nicht das Wort 'compressed' oder 'normal' enthalten. Daraus folgt, daß es sich hier um Testdateien handelt. Das Einlesen der Werte für den Zustand (komprimiert oder nicht) und die Größe des Bildes werden dann übersprungen.

5.7 Dateikomprimierung

Ein Bild mit der Breite und Höhe von 128 hat 16384 Elemente. Bei Sicherung in Form von Ganzzahlen wird eine Speicherplatz von 64KB gebraucht. Obwohl die Festplatten heutzutage immer mehr Speicherkapazität haben, lohnt es sich zu überlegen, ob nicht weniger Speicherplatz ausreichen. Wenn man ein egales Bild betrachtet und dieses beschreiben will, fällt schnell auf, daß wenige Worte schon genügen. Alle 16384 Pixel haben dieselbe Intensität. Die Komprimierungsmethode nutzt diese Eigenschaft. Alle Pixel von Anfang bis Ende werden untersucht. So lange die Intensität sich nicht ändert, wird gezählt, wie viele Pixel nacheinander diesen Wert haben. Bei einer Änderung werden die Intensität und die Anzahl der Pixel mit dieser Intensität gespeichert. Wenn z.B. 100 aufeinanderfolgende Pixel den Wert 10 haben, wird in der komprimierten Datei geschrieben: 0064000a. Im Fall, daß es sich um eine Charakterdatei handelt wird geschrieben: 0064J. Die Einsparung an Speicherplatz ist im ersten Fall am größten weil ein Element hier 4 Byte braucht und bei der '.pic' Datei nur 1 Byte, und damit nur ein Viertel des Speicherraums einer normalen Datei einnimmt. Diese Methode der Komprimierung führt zu keinen Informationsverlust. Die Komprimierung erledigt die Funktion 'reduce_file'. Sie unterscheidet dabei, ob es sich um eine Charakter- oder Ganzzahldatei handelt.

Wenn eine komprimierte Datei eingelesen wird, werden die Intensität und die Anzahl der Pixel in einer Schleife wieder umgewandelt. Die Schleife sorgt dafür, daß wieder 100 aufeinanderfolgende Pixel den Wert 10 bekommen. Auf diese Weise entsteht wieder das ursprüngliche Bild. Die zur Komprimierung benutzte Methode heißt: Running Length Compression.

5.8 Verarbeitungsfunktionen

5.8.1 Farben eliminieren

Den auf dem Bildschirm dargestellten Grauwerten ist jeweils einen Nummer zugeordnet. Mit dem C-Übersetzer Turbo C++ ist es möglich, die Farben 0 bis 15 gleichzeitig zu benutzen. Das Kommando 'CFR' ersetzt den Grauwert mit der angegebenen Nummer durch einen Wert für schwarz (0). Dabei wird jeder Pixel kontrolliert.

5.8.2 Kontur

Wenn die Kontur der Bilddaten mit dem Kommando 'CON' berechnet wird, bleiben nur die Ränder der Objekte im Bild. Die Funktion geht bei der Berechnung von der Änderungen der Daten aus. Jedes Mal, wenn der Wert sich ändert, kommt beim Pixel im neuen Bild ein positiver Wert, wenn der Wert sich nicht ändert wird an der Stelle der Wert 0 eingetragen.

5.8.3 Andere Programme starten

Nach der Eingabe des Kommandos 'exe....' zusammen mit dem Namen eines Programms, wird dieser Name von der Funktion 'get_string_param' aus dem Befehl abgeleitet und in eine Variable gespeichert. In der Standardbibliothek der Sprache C gibt es die Funktion 'spawnl' um andere MS DOS Programme zu starten. Dazu muß diese Funktion aufgerufen werden, wobei als Parameter der Programmname und die Programmparameter auftreten. Der letzte Parameter der Funktion muß zur Erkennung des Endes des neuen Programmaufrufs eine 'NULL' sein. Das aufgerufene Programm wird geladen und gestartet. Das alte Programm bleibt im Speicher vorhanden und wird angehalten, bis das neue Programm beendet ist. Auf dieser Weise wird auch das Hochspannungsprogramm HV2 aufgerufen.

5.8.4 Filtern

Die Methode des Filterns ist bei jeder Art von Filter gleich. Es wird eine Maske über die Bildpixel gelegt, und die neue Werte werden mit dem Formel 4.1 in Abschnitt 4.2.1 berechnet. Die Daten stehen in einer eindimensionalen Matrix, sodaß die Nummern der Pixel zuerst berechnet werden müssen. Die Reihe fängt bei 0 an und endet beim Wert $X_RANGE * Y_RANGE$, welche die Anzahl der Pixel (Werte) in X- und in Y-Richtung enthalten. Die folgende Tabelle zeigt welches Feldelement der Maske mit der Nummer der Matrix übereinstimmt. Die Anfangssituation wird gezeigt.

0	1	2
X_RANGE	X_RANGE+1	X_RANGE+2
2*X_RANGE	2*X_RANGE+1	2*X_RANGE+2

Tabelle 5.2 Koordinaten in der Bilddatenmatrix.

Das Resultat der Berechnung kommt dann auf den Platz mit der Nummer $X_RANGE+1$, also in die Mitte des Vierecks der Maske. Die Resultate werden in einer neuen Matrix gespeichert, weil die alten Werte mehrmals benutzt werden. Die gesamte Berechnung wird in zwei Schleifen gemacht. Eine Schleife für die X-Richtung und eine Schleife für die Y-Richtung.

Beim Filtern mit einem Mittelwertfilter wird der Wert des neuen Pixels statistisch ermittelt. Wenn die Maske am Anfang liegt, werden die Matrixnummern aus der Tabelle zuerst benutzt. Die Werte dieser Nummern müssen in die richtige Reihenfolge gesetzt werden, d.h. sie müssen auf Größe sortiert werden. Das kann schnell erreicht werden, indem man in der Matrix zählt, wie oft ein Wert auftritt. Die Anzahl der Elemente dieser Matrix stimmt mit dem größtmöglichen Wert überein. Danach muß jedes Mal wenn ein Wert auftritt, das dazugehörige Matrixelement inkrementiert werden. Im Programm sieht das so aus : $M2[M1[i]]++$. M2 ist die neue Matrix, M1 die Bilddatenmatrix und M1[i] entspricht den Wert des Pixels. Für den Buchstaben i müssen die Koordinaten aus der Tabelle eingetragen werden.

5.8.5 Multiple Scan

Das Rauschen hat die Eigenschaft, daß der Durchschnittswert der Pegel gleich Null ist. Daher ist eine Möglichkeit, das Rauschen im Bild zu verringern, indem man mehrere Scans macht, und als Endresultat den Durchschnitt der gemessenen Werte nimmt. Ein Problem ist dabei, daß jeder Scan 64 KB Speicherraum braucht. Will man den Durchschnittswert von 'n' Scans, braucht man $n \cdot 64\text{KB}$ Speicherplatz. Im Programm sind drei Matrizen von je 64KB definiert worden. Um es möglich zu machen mehr als drei Scans zu benutzen, werden nach jedem Scan die Daten der Detektorplatten sofort addiert. Die Resultate können in einen Raum von 8Bits gespeichert werden. Weil eine Matrix 16 Bit-Elemente enthalten kann, ist es möglich zwei Bilder in einer Matrix zu speichern. Nach dem ersten Scan befinden sich die Datenbits auf den ersten 8 Bitpositionen. Diese Bits können dann um 8 Positionen nach links verschoben werden. Die Resultate des zweiten Scans können dann einfach zu den ersten addiert werden. Die Werte des ersten und des zweiten Scans befinden sich also nebeneinander in der Matrix. Es gibt drei Matrizen, aber da die letzte Matrix beim Einlesen aus dem Pufferspeicher vollständig gebraucht wird, können maximal fünf Scans durchgeführt werden, von denen der Durchschnittswert berechnet wird. Das Kommando hierzu heißt 'scn <n>', wobei n die Anzahl der Scans ist.

5.9 Statuswiederherstellung

Bevor ein Scan gestartet werden kann, muß der Benutzer alle erforderlichen Dateien laden und dazu über die Menüs oder mit den Befehlen die Dateinamen eingeben. Bei einem Einzelscan für die Bildaufnahme muß er auch noch den gewünschten Scanbefehl selektieren. Im Fall, daß er dieselben Dateien und Einstellungen wie bei einem vorhergehenden Scan benötigt, kann diese Arbeit vom Rechner übernommen werden. Es besteht die Möglichkeit, das Statusfeld in eine Datei zu sichern. Die Funktion 'writestatus' schreibt dabei die Werte genau so in die Datei 'status.stt', wie sie auf dem Bildschirm stehen. Wenn jetzt das Programm erneut gestartet wird, liest die Funktion 'startup' das Statusfeld wieder ein. Danach wird kontrolliert, ob die Namen der beim letzten Mal benutzten Dateien, im Statusfeld standen. Wenn das nicht der Fall ist, erscheint als Dateiname : NOFILE im Feld. Es passiert dann nichts. Es wird keine Datei eingelesen. Wenn ein Name gefunden wird, wird die Datei geladen, vorausgesetzt sie ist im aktuellen Verzeichnis vorhanden. Wurde die Scandatei gefunden, wird nach dem Laden der selektierte Scanbefehl wiederhergestellt. Es wird in der Scanstruktur dann solange zum nächsten Element gesprungen, bis die richtige Scannummer erreicht ist. Das passiert in der Funktion 'restore_selected_scan'.

Als letztes versucht das Programm, eine Datei namens 'startup.txt' zu öffnen und die Befehle aus dieser Datei auszuführen. Die Abarbeitung geschieht auf der gleichen Weise wie im Kommandomodus, wo abhängig vom Befehl die verschiedenen Funktionen ausgeführt werden. Die Befehle werden gelesen und dann verarbeitet. Danach wird der nächste Befehl gelesen. Die Startupdatei ist genau so aufgebaut wie eine Befehlsdatei, die mit dem Kommando 'ECF' ausgeführt werden kann.

5.10 Allgemeine Funktionen

Das Programm benutzt einige Funktionen, die es ermöglichen die Maus zu benutzen und Funktionen die Charakterketten (strings) bearbeiten. Die Funktion 'declare_mousemem' reserviert einen Speicherbereich für die Variablen, die beim Aufruf des Mausinterrupts benutzt werden. Die Funktion 'init_mouse' setzt den Mauscursor auf eine bestimmte Stelle auf dem Bildschirm. Es hat sich herausgestellt, daß nicht immer alle Koordinaten zu erreichen sind. Daher liest diese Funktion die Koordinaten nach dem Programmieren nochmal. Sie stehen dann der aufrufenden Funktion zur Verfügung, die sie weiter nutzen kann. Die Funktion 'mouse_int' ermittelt die Position des Mausursors und den Status der Maustasten. Es kann passieren, daß eine mit der Maus gewählte Funktion nur wenig Zeit in Anspruch nimmt, sodaß die linke Maustaste nach dem Ende der Funktion noch immer gedrückt ist. Um zu verhindern, daß das Programm sofort darauf reagiert, läßt die Funktion 'mouserest' das Weiterlaufen des Programms solange anhalten, bis die linke Maustaste wieder losgelassen wird. Diese Funktionen reichen aus, um die Maus so zu steuern, daß sie bei der Menüauswahl usw. benutzt werden kann.

Die Funktion 'wait_for_key' wartet bis eine Taste gedrückt oder die linke Maustaste betätigt und wieder losgelassen wurde. Sie wird meistens benutzt, um eine Meldung auf dem Bildschirm zu schreiben, wenn eine gewählte Funktion beendet ist.

Im Modul 'ownlib' befinden sich noch die folgenden Funktionen:

wait_cs(n)	Diese Funktion läßt das Programm <n>hundertstel Sekunden warten.
wait_a_second(n)	Diese Funktion läßt das Programm <n> Sekunden warten.
strcpy_to_extension(*d,*s)	Wenn der eingegebene Dateiname eine Erweiterung enthält, wird sie beim Kopieren nicht mitgenommen. Es wird kopiert von <s> nach <d>.
fileexist(*n)	Diese Funktion prüft, ob eine Datei mit Namen <*n> existiert.
get_string_param(*p,*c)	Diese Funktion sucht den Parameter aus der Kommandozeile <*c> und speichert ihn in <*p>.
read_command(*c,*fp)	Scanf liest nur Charakter bis ein Leerzeichen auftritt. Damit ist scanf ungünstig für das Einlesen von Kommandos mit Parametern. Die Funktion read_command liest die Eingabe bis zum Ende der Zeile.

Quellenverzeichnis

- [1] 30 Jahre Zentralinstitut für Kernforschung
G. Flach, H. Ullmann, M. Bonitz
ZfK Rossendorf Oktober 1986
- [2] Das Forschungszentrum Rossendorf; Darstellung des Profils
FZR 1993
- [3] Das Forschungszentrum Rossendorf; Darstellung des Profils
FZR 1993 Seite v-2
- [4] Kurzbeschreibung des Scanprozessors
D. Hentschel, M. Kahlenbach
FZR
- [5] Kurzbeschreibung der Bildspeicherkarte
M. Kahlenbach
FZR