

Bausteinsystem

Komplexe Automatisierung

Baustein

Automatisierte Bilderkennung

Lehrmaterial 2

System- und Anwendungssoftware

YCDT.net

Dipl.-Phys. A. Neumann
Dr. W. Schoenborn
Dr. Ch.-M. Westendorf
Dr. K. Wloka

Baustein
Automatisierte Bilderkennung

Lehrmaterial 2
System- und Anwendungssoftware

KAMMER DER TECHNIK
Praesidium
Sekretariatsbereich Weiterbildung
Wissenschaftlich-technische Gesellschaft fuer
Mess- und Automatisierungstechnik in der KDT

Berlin 1988



System- und Anwendungssoftware : Lehrmaterial / Neumann, Alfred ; Schoenborn, Werner ; Westendorf, Christian-Michael. - Berlin : Praesidium d. Kammer d. Technik, 1988. - 49 S.

(Bausteinsystem Komplexe Automatisierung)

1. Auflage

© by Praesidium der KDT, Berlin 1988

I 12 4 Ag 238/111/88

Printed in the German Democratic Republic

Redaktionsschluss: 18. Dez. 1987

Internes Lehrmaterial der Kammer der Technik.

Jede Vervielfaeltigung - auch auszugsweise - ist nur mit Genehmigung des Herausgebers gestattet.

Inhaltsverzeichnis

Seite

3.	System- und Anwendungssoftware	4
3.1.	Rahmenkonzept fuer BES 2000-Software	4
3.1.1.	Grundlegende Gesichtspunkte	4
3.1.2.	FORTH als System	5
3.1.3.	FORTH als Sprache	8
3.1.4.	Grundstruktur der Software-Loesung	10
3.1.5.	Das IPC-FORTH-System im Steuerrechner	10
3.1.6.	Das FORTH-System der BEE 1010 (VS-FORTH)	14
3.1.7.	Gleitkomma-Paket	16
3.2.	Datenstrukturen	17
3.2.1.	Elementare Datentypen	17
3.2.2.	Definition von Datenobjekten	17
3.2.3.	Datenstrukturen zur Bildrepraesentation	19
3.2.4.	MAIL-Bereich	19
3.2.5.	Parameter-Records	22
3.3.	Verfahrensloesungen	24
3.3.1.	Kalibrierung (CALIB)	24
3.3.2.	Mustervergleich (PATMAT)	27
3.3.3.	Lage- und Orientierungsbestimmung (BINPOS)	31
3.3.4.	Szenenanalyse (BINSC)	36
3.4.	Hardware-Treiber	44
3.4.1.	Bildaufnahmefunktionen	44
3.4.2.	Graphik fuer GDM 183	48
3.5.	Vertriebsvarianten Systemsoftware	48
3.6.	Literaturverzeichnis	49

RM

3. System- und Anwendungssoftware

3.1. Rahmenkonzept fuer BES 2000-Software

3.1.1. Grundlegende Gesichtspunkte

Die Software fuer das Bildererkennungssystem BES 2000 wurde unter Beruecksichtigung folgender a. gemeiner Gesichtspunkte konzipiert:

- Die prozessnah installierte Bilderkennungseinheit BEE 1010 verfuegt ueber kein Externspeichermedium; Tastatur, Bildschirm und Grafik-Display sind optional.
- Eine autonome Arbeitsweise sowie ein Restart muss moeglich sein ("stand-alone"-Betrieb).
- In einer Reihe von Situationen ist eine interaktive Arbeitsweise sowie der Externspeicherzugriff unumgaenglich, z.B.
 - in Trainings- bzw. Lernphasen, die den Eingriff des Menschen erfordern,
 - bei der manuellen Einstellung von Betriebsparametern (z.B. Kalibrierung),
 - bei der Ueberwachung einer oder mehrerer BEE,
 - bei der Fehlersuche,
 - fuer eine Datensammlung,
 - bei der Entwicklung, Anpassung und Erprobung von Verfahren und Algorithmen.
- Es ist nicht moeglich, komplette Verfahrensloesungen ohne Kenntnis der spezifischen Einsatzbedingungen zu programmieren. Es lassen sich jedoch Verfahrensbausteine herausloesen, die mit hoher Sicherheit in einer Anwendungsloesung enthalten sind.
- Das Software-Konzept soll genuegend flexibel sein, um
 - einen Transport auf andere Rechnertypen,
 - die Steuerung weiterer Spezialhardware,
 - die Integration "systemfremder" Software-Komponenten,
 - die Modifikation von Verfahrensloesungen oder deren voellige Neuprogrammierungzu ermoeeglichen.

Alle diese Anforderungen koennen weitgehend erfuehlt werden, wenn

- die System- und Anwendungssoftware konsequent modular und hierarchisch strukturiert wird,

- eine interaktive Steuer- und Programmierbarkeit durch die Systemsoftware unterstuetzt wird,
- die System- und Anwendungssoftware zu grossen Teilen in einem rechnerunabhaengigen, problemorientierten und transparenten Code geschrieben wird.

Ein allgemein anwendbares Software-Konzept, das entsprechend den genannten Gesichtspunkten aufgebaut ist, existiert in Form von FORTH. Es bildet auch die Grundlage der im folgenden beschriebenen Software-Loesung fuer das Bilderkennungssystem BES 2000.

FORTH bezeichnet sowohl

- ein interaktives Software-System als auch
- die Kommando- und Programmiersprache dieses Software-Systems.

3.1.2. FORTH als System

FORTH ist in erster Linie ein interaktives, programmierbares Software-System (vergleichbar mit einem Betriebssystem). Es besteht aus folgenden Basiskomponenten:

- Ein FORTH-System operiert mit einheitlich aufgebauten Grundelementen, den sogenannten FORTH-Worten (Bild 3.1). Im Namensfeld ist eine beliebig waelhbare alphanumerische Mnemonik enthalten (max. 31 ASCII-Zeichen), ueber die das FORTH-Wort extern ansprechbar ist. Der Link-Pointer dient zur Einordnung des FORTH-Wortes in die Link-Kette des Woerterbuchs. Der Code-Pointer verweist auf Maschinencode, der die Ausfuehrung des FORTH-Wortes steuert. Den eigentlichen Inhalt des FORTH-Wortes (Programm oder Daten) bildet das Parameterfeld.

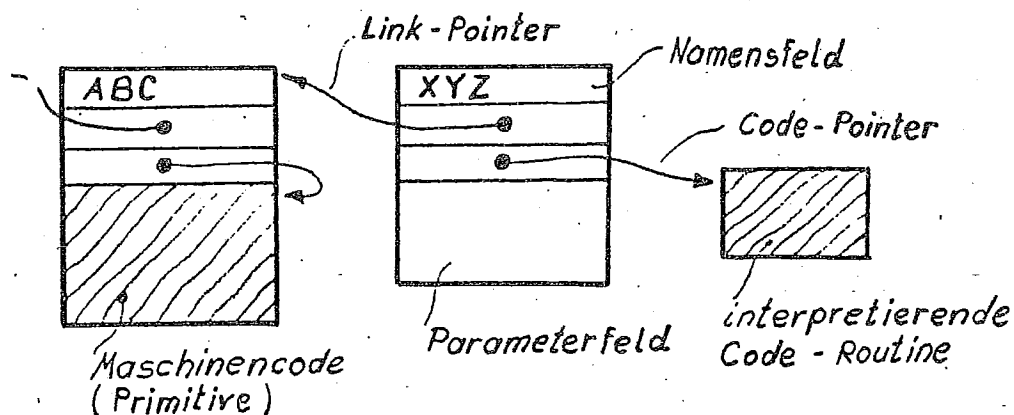


Bild 3.1: Grundstruktur von Woerterbucheintraegen (FORTH-Worte)

- Direkt ausfuehrbare FORTH-Worte (Primitives) enthalten im Parameterfeld Maschinencode. In dieser Weise sind alle Grundfunktionen des Systems sowie Hardware-Treiber und algorithmische Kernprogramme realisiert. Nach der Eingabe des Namens des entsprechenden FORTH-Wortes wird der Maschinencode abgearbeitet.
- Indirekt ausfuehrbare FORTH-Worte (Secondaries) enthalten im Parameterfeld den sogenannten Fadencode als Sequenz von Pointern auf andere, bereits vorhandene FORTH-Worte. Damit ist eine beliebige hierarchische Verknuepfung vorhandener zu neuen FORTH-Worten moeglich. Bei der Aktivierung eines indirekt ausfuehrbaren FORTH-Wortes wird der Fadencode "interpretiert", d.h. bis zu den Primitives zurueckverfolgt, die in der angeordneten Reihenfolge ausgefuehrt werden. Die Generierung des Fadencodes aus FORTH-Quelltexten wird als Compilation bezeichnet. Bild 3.2 zeigt ein Beispiel fuer die interne Repraesentation und die korrespondierende FORTH-Notation eines arithmetischen Ausdrucks.

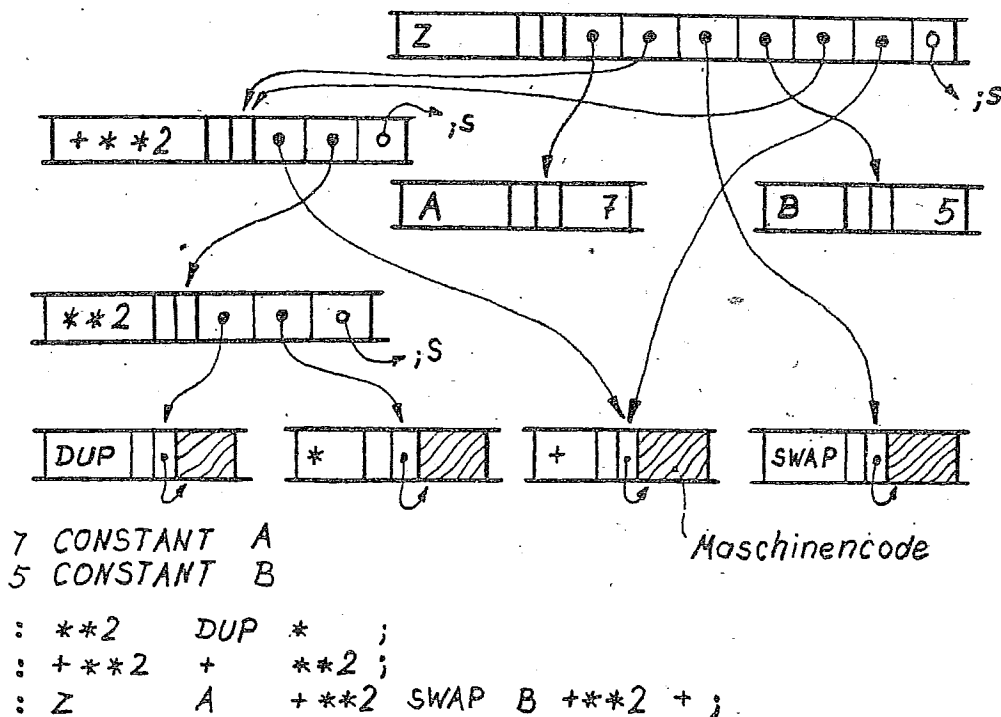


Bild 3.2: Beispiel fuer die interne Fadencode-Repraesentation und die FORTH-Notation des Ausdrucks $Z=(A+X)**2 + (B+Y)**2$. Vor der Ausfuehrung des FORTH-Wortes Z werden Y und X im Datenstack erwartet, der Ergebniswert liegt danach im Datenstack.

- Die FORTH-Worte sind ueber Link-Pointer zum Woerterbuch (dictionary) miteinander verkettet. Das Woerterbuch ist "offen", d.h., es kann interaktiv um benutzereigene FORTH-Worte erweitert werden.
- Die Vermittlung von Werten und Adressen zwischen den FORTH-Worten erfolgt ueber den Datenstack (LIFO-Prinzip). Der Returnstack sichert die gewuenschte Reihenfolge der Abarbeitung der FORTH-Worte durch eine entsprechende Speicherung der Rueckkehradressen.
- Der "aeussere Interpreter" (Textinterpreter) interpretiert FORTH-Texte, die ueber das Terminal oder von einem Massenspeicher eingelesen werden. Die Texte werden in Einheiten (token) zerlegt, die durch mindestens ein Leerzeichen voneinander getrennt sind. Die Interpretation erfolgt mit Hilfe des Woerterbuches, indem versucht wird, das jeweilige token mit der Mnemotik eines FORTH-Wortes (Programm oder Daten) zu identifizieren. Ist dies nicht moeglich, wird eine Konvertierung als 16- oder 32-Bit-INTEGER-Zahl versucht. Bei positivem Verlauf (alle Zeichen sind zulaessig) gelangt diese Zahl ins Datenstack, anderenfalls erfolgt eine Rueckweisung der gesamten Eingabe.
- Der "innere Interpreter" steuert die Abarbeitung des Fadencodes der indirekt ausfuehrbaren FORTH-Worte.
- Die Massenspeicherverwaltung ist nach dem Prinzip des virtuellen Speichers organisiert. Der Inhalt des Massenspeichers kann blockweise in einen Pufferspeicher gelesen werden.
- Quelltexte werden i.allg. auf dem Massenspeicher abgelegt und editiert. Der Quelltext wird in Form von "Screens" (1-KByte-Einheiten) verwaltet, die in der Regel in 16 Zeichen zu je 64 Bytes aufgeteilt werden. Die Bearbeitung der Screens erfolgt mit einem residenten Editor. Screens koennen "geladen" werden (LOAD-Kommando), wobei der in ihnen enthaltene Text dem Textinterpreter zur Verarbeitung uebergeben wird.
- Betriebssystemfunktionen werden ueber entsprechende Grundworte realisiert, die entweder vorhandene Betriebssystem-Schnittstellen nutzen, oder die Treiber vollstaendig in Form von Primitives enthalten (in diesem Fall ist kein weiteres Betriebssystem erforderlich).
- Ein FORTH-System beinhaltet alle Werkzeuge, die zur Programmentwicklung und zur Verwaltung des Woerterbuchs erforderlich sind (z.B. Editor, Decompiler, Assembler, Anzeige Woerterbuch-

inhalt, Entfernen von FORTH-Worten, Auffinden von Woerterbucheintraegen u.a.), so dass i.allg. keine weitere Entwicklungsumgebung erforderlich ist.

3.1.3. FORTH als Sprache

Die im FORTH-Woerterbuch niedergelegte Mnemonik, ueber die die einzelnen FORTH-Worte aktivierbar sind, bildet das Grundvokabular der Sprache FORTH. Dieses Grundvokabular wurde mehrfach standardisiert (FIG-FORTH, FORTH79, FORTH83). Die vorliegende Loesung baut auf dem FIG-Standard auf, der um Sprachelemente von FORTH79 sowie zahlreiche anwendungsspezifische Worte erweitert wurde. FORTH ist sowohl eine Kommando- als auch eine (interaktive) Programmiersprache. Wesentliche Merkmale der Sprache FORTH sind:

- Jede eingegebene Folge von FORTH-Worten wird grundsaeztlich sequentiell abgearbeitet. Dies bedingt eine "postfix"-Notation von Ausdruecken, d.h. die Operanden muessen vor dem Operator spezifiziert werden (UPN=umgekehrte polnische Notation).

Beispiel: `3 5 + . <cr> 8 #OK`

Die Zahlen 3 und 5 gelangen in den Datenstack. Sie werden durch das Kommando + addiert. Das Ergebnis gelangt in den Datenstack und wird anschliessend mit dem Kommando . auf dem Terminal ausgegeben. #OK ist die Bestaetigung fuer die fehlerfreie Ausfuehrung der Eingabe.

- Die Uebernahme bzw. Uebergabe von Werten bzw. Adressen zwischen den FORTH-Worten erfolgt ueber den Datenstack und wird durch das "Stack-Verhalten" beschrieben.

Beispiel: Das FORTH-Wort + (16-Bit-Addition) besitzt das Stack-Verhalten $(n_1 n_2 \rightarrow n)$ mit $n = n_1 + n_2$. n_2 ist der Inhalt der oberen Stack-Zelle (TOS - Top of Stack), n_1 der darunter liegenden Zelle (NOS). Diese Werte werden durch die Operation + aus dem Stack entnommen und durch ihre Summe ersetzt.

- Neue FORTH-Worte koennen unter Verwendung bereits vorhandener mit Hilfe der "colon"- (Doppelpunkt-)Definition erzeugt werden. Sie werden mit der vom Programmierer frei waehibaren Mnemonik in das FORTH-Woerterbuch aufgenommen und stehen als Bausteine fuer die Definition weiterer FORTH-Worte bzw. als Kommando zur Verfuegung.

Beispiel: Das nutzereigene FORTH-Wort += soll zwei Zahlen aus dem Datenstack entnehmen, diese addieren und das Ergebnis anzeigen. Es wird durch Eingabe von

```
: += + SPACE . ; definiert.
```

Die zwischen : += und ; angegebenen FORTH-Worte werden nicht sofort ausgeführt, sondern in die Definition des neuen Wortes += "hineincompiliert". Sie werden erst beim Aufruf des erzeugten Wortes wirksam, z.B.:

```
3 5 += <cr> 8 #OK
```

- Für die Erzeugung von "Datenobjekten" stehen sog. Meta-Worte (vgl. 3.2.2.) zur Verfügung. Solche Meta-Worte (wie z.B. VARIABLE, CONSTANT, ARRAY*1) können mit einer colon-Definition gebildet werden.
- Der Programmablauf kann wie in höheren Programmiersprachen durch Steueranweisungen gesteuert werden. Sprünge im Sinne einer GOTO-Anweisung sind nicht möglich; die in FORTH programmierten Abläufe entsprechen den Prinzipien der strukturierten Programmierung. Bei der Verwendung von Steueranweisungen ist ebenfalls die UPN-Struktur der Sprache zu berücksichtigen.

Beispiele:

```
: TEST DUP @< IF ." KLEINER " DROP <cr>
      ELSE @= IF ." GLEICH " <cr>
            ELSE ." GRÖßER " ENDIF. <cr>
      ENDIF ." NULL " ; <cr> #OK
```

```
-1 TEST <cr> KLEINER NULL #OK
```

Mit dem Wort TEST wird getestet, ob der in TOS enthaltene Wert kleiner, gleich oder größer 0 ist und eine entsprechende Ausschrift erzeugt. Der Test erfolgt mit zwei verschachtelten IF...ELSE...ENDIF-Konstruktionen.

```
: TEST1 0 DO I . LOOP ; <cr> #OK
```

```
10 TEST <cr> 0 1 2 3 4 5 6 7 8 9 #OK
```

TEST1 beinhaltet eine DO-Schleife mit variabler oberer Schranke (ihr Wert wird um 1 erhöht in TOS erwartet). Der aktuelle Wert des Schleifenindex wird auf dem Terminal ausgegeben.

FORTH bietet darüberhinaus zahlreiche weitere Programmierkonzepte, die effektive programmtechnische Lösungen ermöglichen. Zur Einführung in FORTH soll insbesondere auf die Literatur /3.1/, /3.2/ und /3.3/ verwiesen werden. Die Dokumentation /3.4/ enthält im Teil IPCF-2 ebenfalls Hinweise zur FORTH-Programmierung.

Der Befehlsumfang (ca. 400 Befehle) orientiert sich an der Programmierung interaktiver Verfahrensloesungen mit numerisch-algorithmischen Elementen. Die Worte lassen sich folgenden Gruppen zuordnen:

- Mathematisch-logische Operationen:
 - 16- und 32-Bit-INTEGER-Arithmetik
 - Gleitkomma-Arithmetik
 - Vergleichsoperationen
 - Logische Operationen
- Speicherzugriff:
 - Datenstack-Manipulation
 - Wahlfreier Speicherzugriff (8-, 16-, 32-Bit)
- Wort-Definitionen:
 - funktionelle Worte (Prozeduren)
 - Variable und Konstanten
 - Felder und Strings /
- Steueranweisungen:
 - Schleifen
 - bedingte Anweisungen (IF....)
 - Fallauswahl (CASE-Konstrukt), Vektor-Exekution
- String - Behandlung:
 - allgemeine String-Manipulation
 - Konvertierung
- Kommunikation:
 - Terminal-Ein/Ausgabe und Bildschirm-Gestaltung
 - Tabellenanzeige
 - Drucker-Ausgabe
 - Frage-Antwort-Programmierung
 - Menue-Technik
- Systemfunktionen:
 - Woerterbuch-Manipulation
 - Status- und Fehlerbehandlung
- Externspeicher-Bedienung:
 - Quelltext-Verwaltung (Screen-Organisation)
 - Sequentieller und wahlfreier Diskettenzugriff
- Kommunikation mit VS-FORTH ueber SIO-Kopplung:
 - Kommando-Uebermittlung
 - Datenaustausch
 - Programmtransfer

- Programmentwicklung:

Editor

De-Compiler

PROM-Generierung fuer VS-FORTH

Von den FIG-FORTH-Worten sind nur diejenigen zugaenglich, die ohne Spezialkenntnisse hinsichtlich der Arbeitsweise des FORTH-Systems anwendbar sind (ca. 100 Worte).

Bis auf die Steuerkonstruktionen koennen fast alle Befehle interaktiv, d.h. im ausfuehrenden Modus benutzt werden. Insbesondere ist es moeglich, Kommandos ueber die SIO-Strecke an das VS-FORTH-System der BEE 1010 zu uebertragen und dort auszufuehren. Damit wird der gesamte Kommando-Vorrat von VS-FORTH, einschliesslich der Befehle zur Steuerung der Spezialhardware, ueber den Steuerrechner zugaenglich.

3.1.5.2. Programm-Entwicklung in IPC-FORTH

Die Programmierung des Steuerrechners bzw. der Gesamt-Konfiguration erfolgt vorrangig in FORTH durch die Definition von neuen Befehlen und von "Datenobjekten" (Variablen, Feldern, Texten u.a.). Diese Befehle realisieren Teilfunktionen der Anwendungsloesung bzw. diese insgesamt. Sie werden entsprechend dem FORTH-Konzept "bottom-up" aus den Grund-Befehlen bzw. den schon definierten Befehlen aufgebaut.

Die Quelltexte dieser Definition werden i.allg. auf Disketten in Form von Screen-Files (.SCR) abgelegt. Zur Bearbeitung dieser Screen-Files steht ein FORTH-Editor resident zur Verfuegung.

Jede neue FORTH-Definition kann nach dem Laden interaktiv getestet werden; beim Auftreten von Fehlern kann der Quelltext sofort korrigiert und ein erneutes Laden durchgefuehrt werden. Der "Entwicklungszyklus" Editieren - Uebersetzen - Testen kann in kurzer Zeit oftmals wiederholt werden. Dieser und andere Vorteile der FORTH-Programmierung werden jedoch nur dann voll wirksam, wenn der Definition neuer Befehle ein einwandfreies logisches Konzept zugrunde gelegt wird. Grundregeln sind:

- kurze Definition (i.allg. 2...4 Zeilen Quelltext, maximal 1 Screen = 16 Zeilen).
- klar definierter funktioneller Inhalt
- multivalente Anwendbarkeit
- Unabhaengigkeit der Funktionen (Vermeidung von Seiteneffekten)

- sparsamer Umgang mit Steuerkonstruktionen (z.B. max. 1 Schleife je Definition), explizite Angabe von Bedingungen
- geringe Anzahl von Parameteruebergaben (max. 3 Parameter).

Die im Steuerrechner definierten FORTH-Worte koennen auch Kommandos enthalten, die in der BEE 1010 ausgefuehrt werden sollen. Die Abarbeitung solcher Befehle setzt eine aktive SIO-Kopplung voraus. Die fuer die BEE bestimmten Befehle werden zur Laufzeit ueber die SIO-Strecke als ASCII-Strings uebertragen und dort interpretativ abgearbeitet. Entsprechende Moeglichkeiten existieren weiterhin fuer die bidirektionale Datenuebertragung sowie fuer die Uebertragung von Screens.

3.1.6. Das FORTH-System der BEE 1010 (VS-FORTH)

3.1.6.1. Allgemeine Angaben

In der BEE 1010 uebernimmt ein angepasstes PROM-residentes FORTH-System (7 KByte) die Rolle eines Betriebssystems (VS-FORTH). Es umfasst ca. 125 Befehle, die eine Untermenge von IPC-FORTH bilden bzw. dem FIG-FORTH-Standard entsprechen.

Wenn die BEE 1010 mit Tastatur und Bildschirm ausgestattet wird, ist eine interaktive Arbeit ohne Steuerrechner moeglich.

Das FORTH-Basis-System wird durch folgende, ebenfalls im PROM-Bereich angeordnete Erweiterungsmodule ergaenzt:

- Kamerasteuerung / Bildeingabe	2 KByte
- Gleitkomma-Paket	2 KByte
- Grafik-Funktionen fuer GDM 183	1 KByte
- Kalibrierung	1 KByte
- anwendungsspezifische Funktionen	
darunter:	
Mustervergleich (s. 3.3.2.)	4 KByte
Lage- und Orientierungsbest. (s. 3.3.3.)	5 KByte
Szenenanalyse (s. 3.3.4.)	6 KByte

3.1.6.2. FORTH-Programmierung der BEE 1010

Fuer die Entwicklung bzw. Anpassung von Anwendungsloesungen ist vorrangig eine Programmierung der BEE 1010 in FORTH vorgesehen.

Der Quelltext der FORTH-Definitionen wird dabei in Form von Screen-Files auf dem Steuerrechner bearbeitet. Die Uebersetzung (Compilation) der Quelltexte erfolgt jedoch nach deren Uebertragung ueber die SIO-Strecke in der BEE 1010. Der interne FORTH-Code wird im RAM-Speicher der BEE 1010 abgelegt. Ein interaktiver Test jeder einzelnen FORTH-Definition ist sowohl vom Steuerrechner als auch ueber ein an der BEE 1010 direkt angeschlossenes Terminal moeglich.

Mit Hilfe eines Service-Programms im Steuerrechner (VSPROM) ist es moeglich, den internen FORTH-Code PROM-faehig aufzubereiten. Damit koennen ausgetestete, anwendereigene FORTH-Kommandos auf EPROM gebracht und als echte Systemerweiterung resident installiert werden. Auf diese Weise kann eine autonom arbeitende vollstaendige Anwendungsloesung auf EPROM abgespeichert und nach dem Systemstart ohne weitere Kommando-Eingabe abgearbeitet werden.

3.1.6.3. Maschinen-Programmierung der BEE 1010

Wenn eine maximale Ausnutzung der Rechengeschwindigkeit erforderlich ist, muessen die entsprechenden Programm-Passagen (FORTH-Kommandos) in Maschinencode geschrieben werden. Hierfuer stehen folgende Moeglichkeiten zur Verfuegung:

- Der funktionelle Inhalt von FORTH-Worten kann unmittelbar als Hexadezimal-Code (innerhalb der FORTH-Screens) definiert werden.
- Maschinenprogramme koennen unabhaengig vom FORTH-System entwickelt und als .COM-File in das FORTH-Woerterbuch integriert werden.
- Unter Beachtung der Systemstandards koennen PROM-Bausteine als Erweiterungen des Basis-Systems hergestellt werden.

Maschinencode-Programmierung sollte auf die Erstellung von

- Hardware-Treiberprogrammen
 - algorithmischen Basisfunktion zur Bildverarbeitung
- beschraenkt werden. Wegen des relativ hohen Aufwandes, insbesondere bei der Programmaenderung, ist auf eine sorgfaeltige funktionelle Definition besonderer Wert zu legen (Wiederverwendbarkeit).

3.1.7. Gleitkomma-Paket

Als Woerterbuecherweiterung steht sowohl im FORTH-System der BEE 1010 als auch des Steuerrechners ein relativ schnell arbeitendes Gleitkomma-Paket zur Verfuegung. Es enthaelt folgende Operationen (FORTH-Mnemonic):

- F+ , F- , F* , F/ - Grundrechenarten
- F< , F> , F= - Vergleichsoperationen
- N->F , F->N , - Umwandlung Festkommazahlen in Gleitkomma-
- D->F , F->D zahlen und umgekehrt
- FSQRT , FABS - Quadratwurzel und Absolutbetrag
- FSIN , FCOS , FATAN - Winkelfunktionen

Die Genauigkeit bei den Gleitkommaoperationen betraegt 4 Dezimalstellen. Die Winkelfunktionen verarbeiten die Winkelwerte im Bogenmass. Die interne Darstellung der Zahlen ist:

$$\underline{x = a ** b} \quad \begin{array}{l} a = \text{Mantisse, 2 Byte} \\ b = \text{binaerer Exponent, 1 Byte} \end{array}$$

Fuer die Mantisse $0,5 \leq a < 1$ wird die Zweierkomplement-Darstellung benutzt, d.h. Bit 15 ist das Vorzeichenbit. Die Zahlen liegen immer normiert vor, d.h. der Binaerpunkt wird hinter dem Vorzeichenbit angenommen und das nachfolgende Bit ist immer 1. Der Exponent ist stets ganzzahlig und um $40H = 64$ erhoehrt. Eine Gleitkommazahl muss bei der Anwendung der oben genannten Funktionen ueber 2 Stackplaetze uebergeben werden, der untere Stackplatz (NOS) enthaelt den Exponenten und der obere (TOS) die Mantisse. Beispiele fuer die Zahlendarstellung:

NOS	TOS	dargestellte Zahl
40H	4000H	= 0,5
40H	C000H	= -0,5
41H	4000H	= 1

Die Programmlaenge fuer alle Gleitkommaoperationen betraegt 2 KByte. Die Ein/Ausgabekontvertierung wird nur im Steuerrechner unterstuetzt. Die Funktionen stehen sowohl als FORTH-Worte zur Verfuegung, koennen aber auch ueber eine Sprungtabelle direkt aufgerufen werden (speziell von Assembler-Programmen).

3.2. Datenstrukturen

3.2.1. Elementare Datentypen

FORTH ist eine typfreie Sprache. Die Spezifik einer Operation, z.B. der Addition, wird nicht durch die Deklaration der daran beteiligten Datenobjekte bestimmt, sondern muss bei der Wahl des Operators (z.B. + fuer 16-Bit- und D+ fuer 32-Bit-Addition) beruecksichtigt werden.

In IPC-FORTH wurden dennoch eine Reihe von Bezeichnern elementarer Datentypen aufgenommen, die allerdings nur auf die Ein/Ausgabekonvertierung wirken (Tabelle 3.1.). Diese Bezeichner werden auch bei der formalen Beschreibung von Datenrecords (vgl. 3.2.5.) verwendet; Record-Definitionen sind in dieser Weise in IPC-FORTH jedoch nicht moeglich.

Tabelle 3.1: Bezeichner elementarer Datentypen

ASCII-Code	HEX-Code	Datentyp
HBYTE	10	Halbbyte ohne Vorzeichen
BYTE	01	Byte ohne Vorzeichen
DBYTE	02	Doppelbyte ohne Vorzeichen
INT*1	11	1-Byte-INTEGER
INT*2	32	2-Byte-INTEGER
INT*4	34	4-Byte-INTEGER
FIX*4	54	4-Byte-Festkommazahl
REAL*4	44	4-Byte-Gleitkommazahl
CHAR*1	21	1-Byte-ASCII-Zeichen
CHAR*2	22	2-Byte-Element, 2 ASCII-Zeichen
ADDRH	42	2-Byte-Adresse (Pointer)

3.2.2. Definition von Datenobjekten

In IPC-FORTH und - in eingeschaenktem Masse - auch in VS-FORTH sind elementare und zusammengesetzte Datenobjekte definierbar.

Die Definition erfolgt mit Hilfe sogenannter Meta-Worte, die das Datenobjekt mit einem vom Programmierer wählbaren Namen erzeugen, d.h. in das FORTH-Wörterbuch aufnehmen.

- Definition elementarer Datenobjekte

n VARIABLE xxx (16-Bit-Variable)
d DVARIABLE xxx (32-Bit-Variable)
fp FVARIABLE xxx (32-Bit-Gleitkomma-Variable)
n CONSTANT xxx (16-Bit-Konstante)
d DCONSTANT xxx (32-Bit-Konstante)
fp FCONSTANT xxx (32-Bit-Gleitkomma-Konstante)
xxx - Name des Datenobjektes
n - 16-Bit-Anfangswert
d - 32-Bit-Anfangswert
fp - 32-Bit-Gleitkomma-Anfangswert

- Definition von Zeichenketten (Strings)

VTEXT xxx "cccc....c" (xxx enthaelt String cccc....c)
V.TEXT xxx "cccc....c" (selbstanzeigender String)
n VTEXT0 xxx (leerer String der Laenge n)

- Definition von Intervallen

n1 n2 RANGE xxx (16-Bit-Intervall)
d1 d2 DRANGE xxx (32-Bit-Intervall)
fp1 fp2 FPRANGE xxx (32-Bit-Gleitkomma-Intervall)

- Definition eindimensionaler Felder

n ARRAY*1 xxx (Feld mit n 8-Bit-Elementen)
n ARRAY*2 xxx (Feld mit n 16-Bit-Elementen)
n ARRAY*4 xxx (Feld mit n 32-Bit-Elementen)

- Definition zweidimensionaler Felder

m n PLANE*H xxx (Feld mit 4-Bit-Elementen)
m n PLANE*1 xxx (Feld mit 8-Bit-Elementen)
m n PLANE*2 xxx (Feld mit 16-Bit-Elementen)
m Zeilenzahl
n Spaltenzahl

3.2.3. Datenstrukturen zur Bildrepräsentation

Die Bildrepräsentation erfolgt primär in Form des von der Puffer-Codier-Platine erzeugten modifizierten Run-Length-Codes (RLCN), der für die Darstellung von Binärbildern mit beliebiger Zeilen-, Spalten- sowie Uebergangszahl anwendbar ist (vgl. Punkt 2.3.3., Tab. 2.2.). Die bildverarbeitenden Funktionen sind jedoch an den einfachen Run-Length-Code (RLCA) gebunden. Die maximale Zeilenlänge beträgt 256, wobei jeder Uebergang in der Bildzeile direkt adressierbar ist.

Das erste Code-Byte jeder Zeile ist die Adresse des ersten "dunklen" Pixels, das nächste Code-Byte ist die Adresse des nächsten "hellen" Pixels. Die Zählung beginnt mit 1. Eine durchgehend dunkle Zeile ist durch

```
1 FFH 0 0 ... 0
```

codiert, während eine durchgehend helle Zeile nur Nullen enthält.

Im Gegensatz zum Code RLCN, bei dem die Anzahl der Code-Bytes je Zeile durch die Zeilenlänge und die Anzahl der Uebergaenge bestimmt wird, ist die RLCA-Darstellung regulär, d.h. jede Bildzeile wird durch die gleiche Anzahl von Code-Bytes dargestellt (Standard-Wert: 16). Treten weniger Uebergaenge auf, so werden die nicht benötigten Bytes gleich Null gesetzt. Sind mehr als die vorgesehene Anzahl von Uebergaengen vorhanden, ist eine Bildung des Codes nicht möglich.

Im Bildaufnahmeprogramm PICPC (vgl. 3.4.1.) wird dann eine Fehlermeldung erzeugt. Die maximal mögliche Uebergangszahl kann durch MNRL im MAIL-Bereich gewählt werden.

3.2.4. MAIL-Bereich

Der MAIL-Bereich in der BEE 1010 dient zur Parameter-Versorgung bzw. zur Kopplung der Hardware-Treiberprogramme und einiger dieser Gruppe zugeordneter Funktionen. Der Zugriff erfolgt seitens dieser Programme durch direkte Adressierung, während durch VS-FORTH ein symbolischer Zugriff mit

```
(MAILAD) xxx  
möglich ist.
```

Die Namen xxx der Elemente (alles 16-Bit-Werte) des MAIL-Bereichs, ihr Standard-Wert und ihre Bedeutung sind in Tabelle 3.2. beschrieben. Spalte 2 enthaelt den Standard-Wert, der durch das Kommando INIMAIL aus dem PROM-Abbild des MAIL-Bereichs in den entsprechenden RAM-Bereich uebertragen wird. Spalte 3 gibt ob es sich um einen einzugebenden Wert (I) oder auszulesenden Wert (O) handelt.

Tabelle 3.2.: Inhalt und Schluesselworte des MAIL-Bereichs

ASCII-Code	Standard-Wert		Inhalt	Bemerkung
XMIN	0	O	x-Koordinate Objektbeginn	1
XMAX	0	O	x-Koordinate Objektende	1
YMIN	0	O	y-Koordinate des linken Objektrandes (Minimum)	1
YMAX	0	O	y-Koordinate des rechten Objektrandes (Maximum)	1
LPIU	80H	I	obere Binarisierungsschwelle des Fensterkomparators	2
LPIL	0	I	untere Binarisierungsschwelle	2
NPBP	3	I	minimal erforderliche Pixelzahl je Zeile fuer die Detektion des Objektanfangs	
IBEG	18H	I	Beginn der Integration	3
IEND	30H	I	Ende der Integration	4
NLIP	0	O	Anzahl der Zeilen des Objektes	
BYTN	0		Anzahl der Bytes, die ein abgespeichertes Bild im Bildspeicher belegt	
NZEL	0	O	Zeilenzahl laufend bei Bildaufnahme	
LERR	0	O	Fehlerbyte	
MNRL	1	I	maximal zulaessige Anzahl der RLCA-Elemente je Zeile	
MPIL	255	I	maximale Bildzeilen-Anzahl	
MIPL	3	I	Mindest-Zahl von Bildzeilen fuer Objektdetektion	
MIBO	3	I	Mindest-Abstand des Objektes vom Bildrand (Pixel)	

PICA	B000H	I	Adresse fuer die Ablage des Bildes im Speicher
PNTR	B000H	O	Speicheradresse des zuletzt abgelegten Code-Elementes
PNTL	B000H	O	Speicheradresse der aktuellen Bildzeile
ABUF	CF00H	I	Speicheradresse eines Zeilenpuffers
ADBI	C000H	I	Adresse des Parametervektors BINPAR
NMCA	1	I	Nummer der aktuellen Kamera
TYCA	1	I	Typ der aktuellen Kamera
SWIW	0	I	Schalter Teil/Hintergrund
KMUX	F9H	I	Port-Adresse MUX
KOMP	FAH	I	Port-Adresse Komparator
KCTC	FCH	I	Port-Adresse CTC
BUIN	30H	I	Port-Adresse PCE
KADC	10H	I	Port-Adresse ADU

Bemerkungen:

- (1) Die Koordinatenzaehlerung beginnt mit $x = 1$, $y = 1$; y ist die Richtung der Zeile, x die Bewegungsrichtung.
- (2) Die von der Kamera an die Steuereinheit gelieferten Analogwerte werden durch einen Fensterkomparator binarisiert. Werte innerhalb des Fensters werden mit 1, Werte ausserhalb mit 0 bewertet. Die Schwellen des Komparators werden durch LPIU und LPIL festgelegt. FFH legt die Schwelle auf den hoechsten Wert, 0 auf den tiefsten. $LPIU > LPIL$ muss stets gewaehrleistet werden.
- (3) IBEG * 16 :
Anzahl der Rechnertakte fuer Beginn der Empfindlichkeit der CCD-Zeile nach dem Start der Zeilenaufnahme
- (4) IEND * 16 * 64 :
Anzahl der Rechnertakte fuer das Ende der Empfindlichkeit der CCD-Zeile nach dem Start der Zeilenaufnahme.
Es muss gewaehrleistet werden, dass $IEND * 64 > IBEG$ ist!
Fuer einen Rechnertakt von 2,5 MHz kann die Zeit bis zum Integrationsende und damit die Wiederholrate fuer eine periodische Zeilenabtastung durch $IEND * (100/256)$ ms bestimmt werden.

(5) heller Untergrund mit dunklen Objekten:

SWIW = 1 ; der erste Schwarz-Weiss-Übergang tritt an der Grenze Untergrund-Teil auf

SWIW = 0 ; der erste Übergang tritt am Beginn des Bildfeldes auf (falls das Teil den Rand nicht ueberdeckt)

dunkler Untergrund mit hellen Objekten: umgekehrte Zuordnung.

3.2.5. Parameter-Records

3.2.5.1. BINPAR

Der Record BINPAR enthaelt Merkmale sowie Lage- und Orientierungsangaben eines einzelnen binaren Objektes. Er dient vorrangig zur Zusammenfassung der Ergebnisse verschiedener Programme und als Ausgangspunkt der Zusammenstellung problemspezifischer Parameterrecords.

Mehrere Records koennen zu einer Szenentabelle zusammengefasst werden. BINPAR besitzt folgende Struktur:

BINPAR = RECORD

XMIN	: INT*2	; minimale x-Koordinate
XMAX	: INT*2	; maximale x-Koordinate
YMIN	: INT*2	; minimale y-Koordinate
YMAX	: INT*2	; maximale y-Koordinate
XCON	: INT*2	; x-Koordinate Konturanfang
YCON	: INT*2	; y-Koordinate Konturanfang
AREA	: INT*4	; Flaechen
MC0X	: INT*4	; x-Komponente des Momentes 1. Ordnung
MC2Y	: INT*4	; 2*y-Komponente des Momentes 1. Ordnung
XCEN	: INT*2	; x-Koordinate Schwerpunkt
YCEN	: INT*2	; y-Koordinate Schwerpunkt
MCXY	: INT*4	; zentriertes Moment 2. Ordnung mxy
MCXX	: INT*4	; zentriertes Moment 2. Ordnung mxx
MCYY	: INT*4	; zentriertes Moment 2. Ordnung myy
LCON	: INT*2	; Konturlaenge
NCNP	: INT*2	; Anzahl Konturpunkte
RMAX	: INT*2	; max. Abst. Kontur-Schwerpunkt
RMIN	: INT*2	; min. Abst. Kontur-Schwerpunkt

RXMA : INT*2 ; Koordinaten des Randpunktes mit dem
 RYMA : INT*2 ; groessten Abstand zum Schwerpunkt
 RXMI : INT*2 ; Koordinaten des Randpunktes mit dem
 RYMI : INT*2 ; kleinsten Abstand zum Schwerpunkt
 ARCN : INT*4 ; Flaechen innerhalb Aussenkontur
 ARHL : INT*4 ; Lochflaechen
 XHOL : INT*2 ; x-Koordinaten Lochschwerpunkt
 YHOL : INT*2 ; y-Koordinaten Lochschwerpunkt
 NHOL : INT*2 ; Lochanzahl
 DCCH : INT*2 ; Abstand: Lochschwerpunkt - Teileschwerpunkt
 FCON : INT*2 ; Formfaktor * 100
 ALPH : INT*2 ; Winkel Traegheitsachse - x-Achse
 BETA : INT*2 ; Winkel RMAX-Vektor - x-Achse
 GAMA : INT*2 ; Winkel Lochvektor - x-Achse

END-REC

Bemerkungen:

- Alle Koordinaten werden in Pixel-Anzahlen angegeben. Das erste Pixel der ersten Zeile hat die Koordinaten x=1, y=1.
- Alle Laengen werden in der Einheit der Pixelabmessungen angegeben. Dies setzt in einigen Faellen (RMAX, DCCH, LCON) eine quadratische Rasterung voraus.
- Flaechen werden in Pixelanzahlen angegeben.

3.2.5.2. Kalibrierungstabelle AMCAL

Die Kalibrierungstabelle enthaelt fuer jede Kamera (1...3) die Faktoren der geometrischen Kalibrierung, Objektivkenngrößen, einige Aufnahmebedingungen sowie Kamera-Steuergrößen aus dem MAIL-Bereich. Sie dient zur Abspeicherung des Kalibrierungsergebnisses bzw. zur Reproduktion von Einstellungen. Die Kalibrierungstabelle AMCAL besteht aus 3 Records mit folgendem Aufbau:

MCAL = RECORD

NMCA : BYTE ; Nummer der Kamera
 LCAL : BYTE ; Kalibrierungsart
 CALFX : INT*4 ; Kalibrierungsfaktor x-Richtung
 CALFY : INT*4 ; Kalibrierungsfaktor y-Richtung
 CALFA : INT*4 ; Kalibrierungsfaktor Pixelflaechen

RXMA : INT*2 ; Koordinaten des Randpunktes mit dem
 RYMA : INT*2 ; groessten Abstand zum Schwerpunkt
 RXMI : INT*2 ; Koordinaten des Randpunktes mit dem
 RYMI : INT*2 ; kleinsten Abstand zum Schwerpunkt
 ARCN : INT*4 ; Flaechen innerhalb Aussenkontur
 ARHL : INT*4 ; Lochflaechen
 XHOL : INT*2 ; x-Koordinaten Lochschwerpunkt
 YHOL : INT*2 ; y-Koordinaten Lochschwerpunkt
 NHOL : INT*2 ; Lochanzahl
 DCCH : INT*2 ; Abstand: Lochschwerpunkt - Teileschwerpunkt
 FCON : INT*2 ; Formfaktor * 100
 ALPH : INT*2 ; Winkel Traegheitsachse - x-Achse
 BETA : INT*2 ; Winkel RMAX-Vektor - x-Achse
 GAMA : INT*2 ; Winkel Lochvektor - x-Achse

END-REC

Bemerkungen:

- Alle Koordinaten werden in Pixel-Anzahlen angegeben. Das erste Pixel der ersten Zeile hat die Koordinaten x=1, y=1.
- Alle Laengen werden in der Einheit der Pixelabmessungen angegeben. Dies setzt in einigen Faellen (RMAX, DCCH, LCON) eine quadratische Rasterung voraus.
- Flaechen werden in Pixelanzahlen angegeben.

3.2.5.2. Kalibrierungstabelle AMCAL

Die Kalibrierungstabelle enthaelt fuer jede Kamera (1...3) die Faktoren der geometrischen Kalibrierung, Objektivkenngrößen, einige Aufnahmebedingungen sowie Kamera-Steuergrößen aus dem MAIL-Bereich. Sie dient zur Abspeicherung des Kalibrierungsergebnisses bzw. zur Reproduktion von Einstellungen. Die Kalibrierungstabelle AMCAL besteht aus 3 Records mit folgendem Aufbau:

MCAL = RECORD

NMCA : BYTE ; Nummer der Kamera
 LCAL : BYTE ; Kalibrierungsart
 CALFX : INT*4 ; Kalibrierungsfaktor x-Richtung
 CALFY : INT*4 ; Kalibrierungsfaktor y-Richtung
 CALFA : INT*4 ; Kalibrierungsfaktor Pixelflaechen

BLZL : BYTE ; Blendenzahl
 ABST : INT*2 ; Objektabstand in mm
 IBEG : INT*2 ; Integrationsbeginn
 IEND : INT*2 ; Integrationsende
 LPIL : INT*2 ; untere Fenstergrenze
 LPIU : INT*2 ; obere Fenstergrenze
 SWIW : BYTE ; Schalter Objekt/Hintergrund
 VTRANS: INT*2 ; Transportgeschwindigkeit
 FOCUS : INT*2 ; Brennweite in mm
 OTYP : DBYTE ; Objektivtyp

END-REC

- Die Kalibrierungsfaktoren CALFX, CALFY geben die Pixelabmessungen in mm/100 an (x-, y-Richtung). CALFA bezeichnet die Pixelflaeche in mm**2/100.
- Die Transportgeschwindigkeit wird in mm/s angegeben.
- Die Kodierung der Blendenzahl und des (numerischen!) Objektivtyps kann anwendungsspezifisch erfolgen.
- Zur Bildung, Anzeige, Externspeicherung und Manipulation der Kalibrierungstabelle dient das Programm CALIB (vgl. 3.3.1.).

3.3. Verfahrensloesungen

3.3.1. Kalibrierung (CALIB)

3.3.1.1. Funktionsumfang

Im Rahmen eines menuegesteuerten Programms werden einige einfache Funktionen zur Einstellung der Bildaufnahmeparameter bereitgestellt. Von den rechnergesteuerten Bildaufnahme-Parametern sind folgende Werte manipulierbar (vgl. 3.2.4., MAIL-Bereich):

IBEG - Integrationsbeginn
 IEND - Integrationsende
 LPIL - untere Binarisierungsschwelle
 LPIU - obere Binarisierungsschwelle
 SWIW - Schalter Teil/Hintergrund

Die Bildaufnahmebedingungen werden durch

- BLZL - Blendenzahl
- ABST - Objektabstand
- VTRANS - Transportgeschwindigkeit
- FOCUS - Brennweite
- OTYP - Objektivtyp
- TYCA - Kameratyp

bestimmt. Diese Groessen koennen in der Kalibrierungstabelle (vgl. 3.2.5.2.) abgespeichert werden. Im Rahmen des Menues werden folgende Funktionen angeboten:

- INIVS - Initialisierung des MAIL-Bereiches und der Kalibrierungstabelle
- INF - Eingabe Kameranummer und -typ
- MODC - Anzeige eines Kalibrierungsrecords mit Eingabemoeglichkeit
- ORM - Einstellung quadratisches Rastermass
- RCAL - radiometrische Kalibrierung
- GCAL - geometrische Kalibrierung
- WR - Disketten-Ausgabe der Kalibrierungsrecords
- RD - Eingabe der Kalibrierungsrecords von Diskette

3.3.1.2. Geometrische Kalibrierung (GCAL)

Die geometrische Kalibrierung beruht auf der Abtastung einer Kreisscheibe mit bekanntem Durchmesser D. Dabei werden die Groessen

- XMIN, XMAX - minimale und maximale Koordinate in x-Richtung
- YMIN, YMAX - minimale und maximale Koordinate in y-Richtung

bestimmt. Hieraus werden die Kalibrierungsfaktoren

$$CALFX = D * 100 / (XMAX - XMIN)$$

$$CALFY = D * 100 / (YMAX - YMIN)$$

berechnet (D in mm, CALFX, CALFY in mm/100). Dabei wird eine gleichfoermige Bewegung sowie eine scharfe Abbildung der Kreisscheibe vorausgesetzt. CALFX, CALFY sowie der Flaechenfaktor $CALFA = CALFX * CALFY * 100$ werden in die Kalibrierungstabelle eingetragen.

3.3.1.3. Radiometrische Kalibrierung (RCAL)

Die radiometrische Kalibrierung unterstuetzt die Einstellung der Blende bei gegebenen Bildaufnahmebedingungen. Die Einstellung der Blendenzahl basiert auf der Detektion des Auftretens von Hell-Dunkel-Uebergangsen bei langsamer Oeffnung der Blende. Die Kamera tastet dabei eine Flaechе ab, die den Betriebsbedingungen entsprechend beleuchtet ist (heller Untergrund bzw. helles Objekt bei dunklem Untergrund). Die Binarisierungsschwelle (untere Komparatorschwelle) liegt an der Uebersteuerungsgrenze. Die Abtastung wird beendet, sobald das Ausgangssignal der Kamera fuer ein beliebiges Pixel diese Grenze ueberschreitet.

Damit ist gewaehrleistet, dass bei einer unterhalb der Uebersteuerungsgrenze liegenden Schwelle eine Binarisierung sicher moeglich ist. In Bild 3.4 ist der Spannungsverlauf des Analogausgangs der CCD-Zeile beim seriellen Auslesen auf dem Oszillographenbildschirm dargestellt.

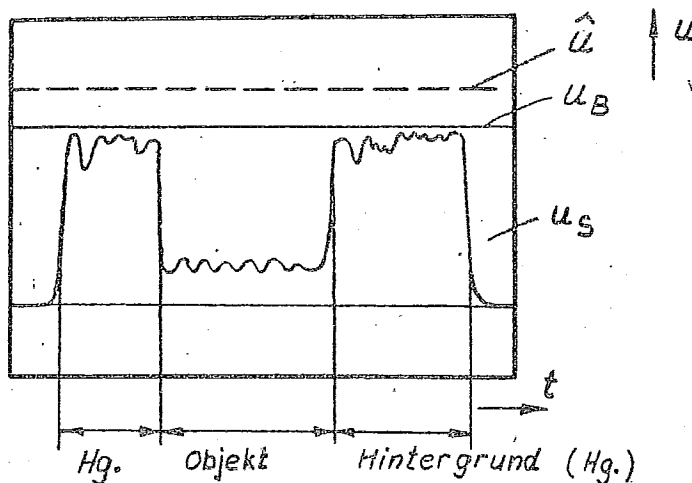


Bild 3.4: Kamera-Ausgangssignal und Binarisierungsschwelle bei radiometrischer Kalibrierung

3.3.1.4. Quadratisches Rastermass (QRN)

Eine quadratische Rasterung ist naeherungsweise erreichbar, wenn der Wert IEND (Integrationsende, vgl. Tabelle 2.2.) gemass
$$IEND = (10 \cdot \text{Bildbreite}) / VTRAN$$
 berechnet wird.

Die Bildbreite ist die Breite des Gesichtsfeldes der Kamera in

der Objektebene in mm, VTRAN ist die Transportgeschwindigkeit des Objektes in mm/s.

Die Ueberpruefung des Ergebnisses kann mit der Funktion GCAL erfolgen. Eine Veraenderung der Integrationszeit wirkt sich auf die radiometrische Kalibrierung aus, die entsprechend zu wiederholen ist.

3.3.2. Mustervergleich (PATMAT)

3.3.2.1. Zielstellung und Anwendungsbereich

Der Mustervergleich (PATMAT) ist fuer die Identifikation von Objekten implementiert worden, wobei folgende Voraussetzungen erfuellt sein muessen:

- Von den Objekten koennen relevante Schwarz-Weiss-Bilder (Binnaerbilder) mit der Kamera erzeugt werden.
- Das zu identifizierende Objekt ist ausgerichtet, d.h. es liegt stets mit der gleichen Orientierung zur Kamera vor.
- Das zu identifizierende Objekt ist vollstaendig im Bildfenster enthalten (sind mehrere Objekte im Bildfenster enthalten, so sind die einzelnen Objekte mit dem Modul Szenenanalyse (BINSC) im Speicher zu isolieren).

Solche Objekte sind beispielsweise gedruckte Zeichen oder ausgerichtete Teile, deren Silhouette von der Kamera erzeugt wird.

Die Identifikation der Objekte erfolgt anhand abgespeicherter Repraesentanten (Muster und Kennwerte, vgl. /3.6/, S.86, /3.7/). Der Mustervergleich ist translationsinvariant. Fuer die Identifikation eines Objektes werden die Muster nacheinander mit dem Objektbild verglichen, wobei Muster und Objektbild mit dem Flaechenschwerpunkt (in Raestereinheiten) zur Deckung gebracht werden. Der Vergleich von Objektbild und Muster wird nur dann ausgefuehrt, wenn die Lage des Flaechenschwerpunkts zum umschreibenden Rechteck vom Objektbild gegenueber dem Muster eine vorgegebene Toleranz nicht ueberschreitet, um die Anzahl erforderlicher Vergleiche zu reduzieren.

Als Abstandsmass fuer den Vergleich wird die Anzahl unterschiedlicher Rasterpunkte zwischen Muster und Bild verwendet. Geringfuegige Verschiebungen (<1 Rasterpunkt) des Objektes gegenueber dem Raster erzeugen unterschiedliche Rasterbilder. Dementsprechend ergeben sich bei dem Vergleich von Rasterbildern

zum gleichen, ausgerichteten Objekt unterschiedliche Distanzen. Fuer die naeherungsweise Bestimmung des zulaessigen Abstandes wird das Rasterbild auf die B benachbarten Plaetze des Schwerpunkts verschoben und mit der Ursprungslage verglichen. Die maximale Distanz - gegebenenfalls mit einem Faktor etwas groesser als 1 zur Kompensation von Stoerungen bei der Bildaufnahme multipliziert - wird dem Muster als zulaessiger Abstand zugeordnet.

3.3.2.2. Programmtechnische Loesung

Das Verfahren PATMAT (vgl. Bild 3.5) ist strukturiert in

- T - Trainingsphase
- P - Pruefphase
- A - Arbeitsphase
- SP - Speichern von Referenzinformation
- LD - Laden von Referenzinformation

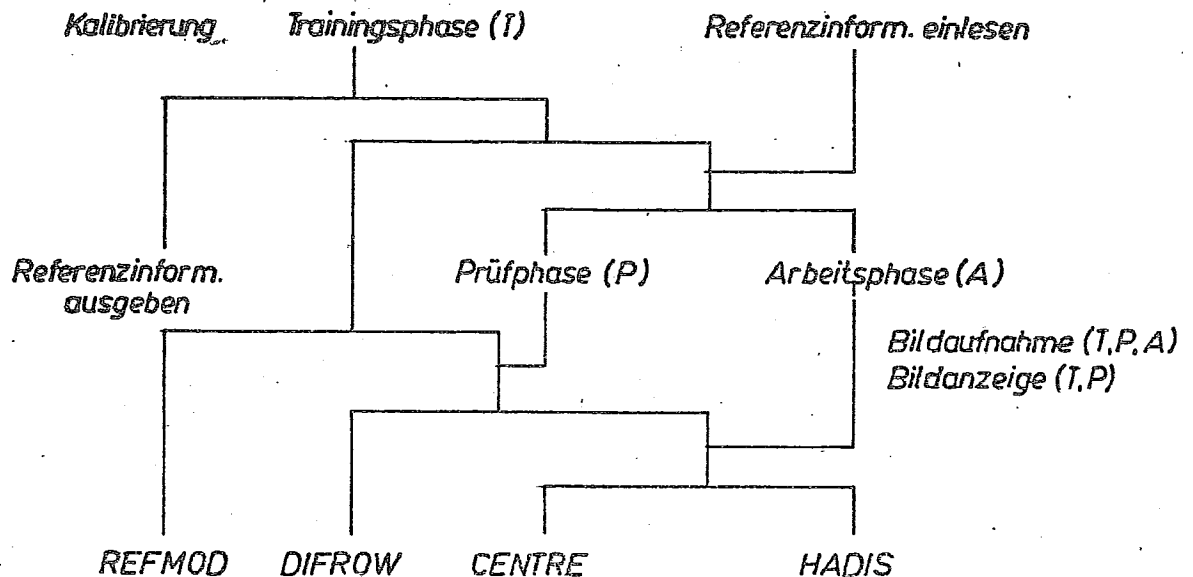


Bild 3.5: Strukturuebersicht zu PATMAT

Neben diesen Funktionen wird auch die Kalibrierung und Anzeige der Fehlertabelle im Menue angeboten. Mit START wird das Hauptmenue aufgerufen, das KALIBRIERUNG, LADEN VON REFERENZINFORMATION und PATMAT BEENDEN als Alternativen enhaelt.

Laden von Referenzinformation wird aufgerufen, wenn mit extern gespeicherten Repraesentanten die Arbeitsphase oder die Pruefphase durchzufuehren ist; ansonsten ist mit dem Aufruf der Kali-

brierung zu beginnen. Nach Abschluss einer Funktion erscheint das naechste Menue, das auf die vorher ausgefuehrten Funktionen abgestimmt ist.

Die Trainingsphase kann aufgerufen werden, wenn das System kalibriert vorliegt. Zu Beginn der Trainingsphase werden fuer die Erkennungsaufgabe typische Parameter auf dem Monitor abgefordert und es wird zur Bereitstellung des Objekts aufgefordert. Wird die Bereitstellung bestaetigt, so wird die Kamera aktiviert und bei Ueberschreitung einer vorgegebenen Anzahl von Objektrasterpunkten in der Zeile eine vorgegebene Anzahl von Bildzeilen in den Bildspeicher eingelesen. Ist das Objektbild vollstaendig im Bildfenster enthalten, so werden die Objektbildparameter auf dem Monitor angezeigt und der Vergleich mit bereits gespeicherten Repraesentanten wird durchgefuehrt. Die Differenzbilder koennen auf dem Rasterdisplay angezeigt werden. Vollstaendige Objektbilder und deren Kennwerte koennen als Repraesentanten gespeichert werden. Bei der externen Auslagerung der Repraesentanten werden auch diejenigen Bildaufnahmeparameter gespeichert, mit denen die Bildaufnahmebedingungen reproduziert werden koennen.

In der Pruefphase koennen die Kennwerte auf dem Monitor und Muster auf dem Rasterdisplay der in der Trainingsphase erzeugten bzw. der von externen Speichern eingelesenen Repraesentanten angezeigt und ein Probetrieb mit Anzeige der Abstaende und Differenzbilder durchgefuehrt werden.

In der Arbeitsphase arbeitet nur der Kamerarechner ohne Rasterdisplay. Liegt das Objektbild innerhalb des zulaessigen Abstandes zu einem Muster, so werden die Klassen-Nr., Flaechen, Schwerpunktkoordinaten sowie die relative Verschiebung des Schwerpunkts gegenueber dem Repraesentanten auf Speicherplaetzen bereitgestellt. Bei unvollstaendigem Objektbild (s. Fehlerzelle) oder Abstaenden groesser als die zulaessigen Abstaende erfolgt Rueckweisung mit Klassen-Nr. 0.

Das in FORTH programmierte Verfahren greift auf folgende Assembler-Moduln zurueck (2 * 1 KByte EPROM):

- CENTRE - Schwerpunktzentrierung
- HADIS - Distanzberechnung zwischen Bild und Muster
- REFMOD - Abspeicherung der Repraesentanten
- DIFROW - Differenzbildanzeige.

Die Assembler-Moduln koennen unabhaengig voneinander verwendet oder durch andere ersetzt werden. Zum Beispiel kann HADIS zusam-

durchgefuehrt werden. Beim Traegheitsellipsoid wird erkannt, ob die Symmetrieachse mehrzaehlig ist, beim Lochvektor, ob der Betrag des Vektors einen vorgegebenen Wert unterschreitet und beim maximalen Abstandsvektor, ob sich die Richtung des Vektors stark aendert, wenn der Schwerpunkt um einen geringen Betrag (3 Rasterpunkte) verschoben wird. Ein weiterer Test ist, ob die Differenz der zwei notwendigen Winkel zur Auflagenseitenbestimmung einen vorgegebenen Wert ueberschreitet. Wenn die Signifikanztests ergeben haben, dass die bestimmten Richtungen stabil sind, koennen in der Arbeitsphase diese Winkel zur Berechnung von Richtungen und Auflagenseiten herangezogen werden. Zum Verfahren gehoert eine Identifikation der Teile durch morphometrische Merkmale. Folgende Merkmale werden dafuer verwendet:

- Teileflaechen
- Quotient aus den 2 Haupttraegheitsmomenten
- Quotient aus der Summe der Haupttraegheitsmomente und der Teileflaeche
- Abstand zwischen Teileschwerpunkt und entferntestem Randpunkt
- Summe der Lochflaechen
- Abstand zwischen Teileschwerpunkt und Lochschwerpunkt
- Umfang der Kontur

Waehrend der Trainingsphase werden Ober- und Untergrenze fuer diese Merkmale in einem Klassenquader abgespeichert. In der Arbeitsphase dient dieser Klassenquader zur Identifikation des Teils. Mit den fuer diese Klasse markierten Winkelnummern kann dann die Winkellage (=Orientierung modulo 180°) und die Auflagenseite bestimmt werden.

Die in der Trainingsphase ermittelten Klassenquader und Referenzwinkel koennen ueber den Steuerrechner auf externen Speichern abgelegt und von dort fuer die Arbeitsphase wieder geladen werden. Fuer die Arbeitsphase ist kein Steuerrechner erforderlich.

In der Pruefphase koennen die aktuell ermittelten Merkmale eines abgebildeten Teils zusammen mit dem Klassenquader auf dem Bildschirm des Steuerrechners dargestellt werden. Die morphometrischen Merkmale sind nur lageunabhaengig, wenn eine quadratische Rasterung des Teils gewaehrleistet ist. Deshalb muss vor Beginn des Verfahrens in der Kalibrierungsphase ein solches quadratisches Rastermass eingestellt werden (vgl. 3.3.1.4.).

3.3.3.3. Programmtechnische Loesung

Fuer das Verfahren sind Transformationen zwischen folgenden Datenstrukturen notwendig:

SIGPAR	Signifikanztestparameter
APRIWI	eingegebenes A-priori-Wissen
FEVE	Merkmalsvektor
ANGVE	Winkelvektor
CLQU	Klassenquader
ANGRE	Winkelreferenzmatrix
MASK	Testmaske fuer benutzte Winkel
BINPAR	Binaerparameter
BILD	RLCA-Code des Bildes
UML	Koordinaten aller Punkte des Konturumlaufes
ROBVE	Bestimmungsgroessen fuer Roboter

Die Datenstrukturen BILD, UML und ROBVE sind nur in der BEE 1010 erforderlich, alle anderen sowohl in der BEE 1010 als auch im Steuerrechner.

Die Transformationen waehrend der Trainings-, Pruef- und Arbeitsphase werden im Bild 3.6 veranschaulicht.

In der Trainingsphase werden zunaechst auf dem Bildschirm des Steuerrechners die Signifikanztestparameter angezeigt; sie koennen veraendert (DISPLSIG) werden. Diese werden dann zur BEE 1010 transportiert. Danach wird das A-priori-Wissen zum angebotenen Teil durch ein Frage-Antwort-Programm (APRI-INP) eingegeben. Hierzu gehoeren die Teilenummer als "Name" des Teils, die Information ueber Auflagenseite, Symmetrie u.s.w. Diese Werte werden wieder zur BEE 1010 uebertragen. Danach beginnt die Bildaufnahme (PICPC). Bildaufnahmefehler werden dabei im Steuerrechner angezeigt.

Sofort nach der Bildaufnahme werden alle Berechnungen zum abgebildeten Teil durchgefuehrt (EBERT). Dies ist eine sehr komplexe Funktion. Sie beinhaltet die Berechnung aller 7 Merkmale und der 3 Winkel einschliesslich des Signifikanztests fuer die Winkel und den Aufruf des Klassifikators, um ein schon eingelerntes Teil signalisieren zu koennen. Die berechneten Werte werden zum Steuerrechner uebertragen und dort zur Anzeige (DISFE , DISAG) gebracht.

Datenstrukturen und Transformationen

im
Steuerrechner

Daten -
übertragungen

in der
BEE 1010

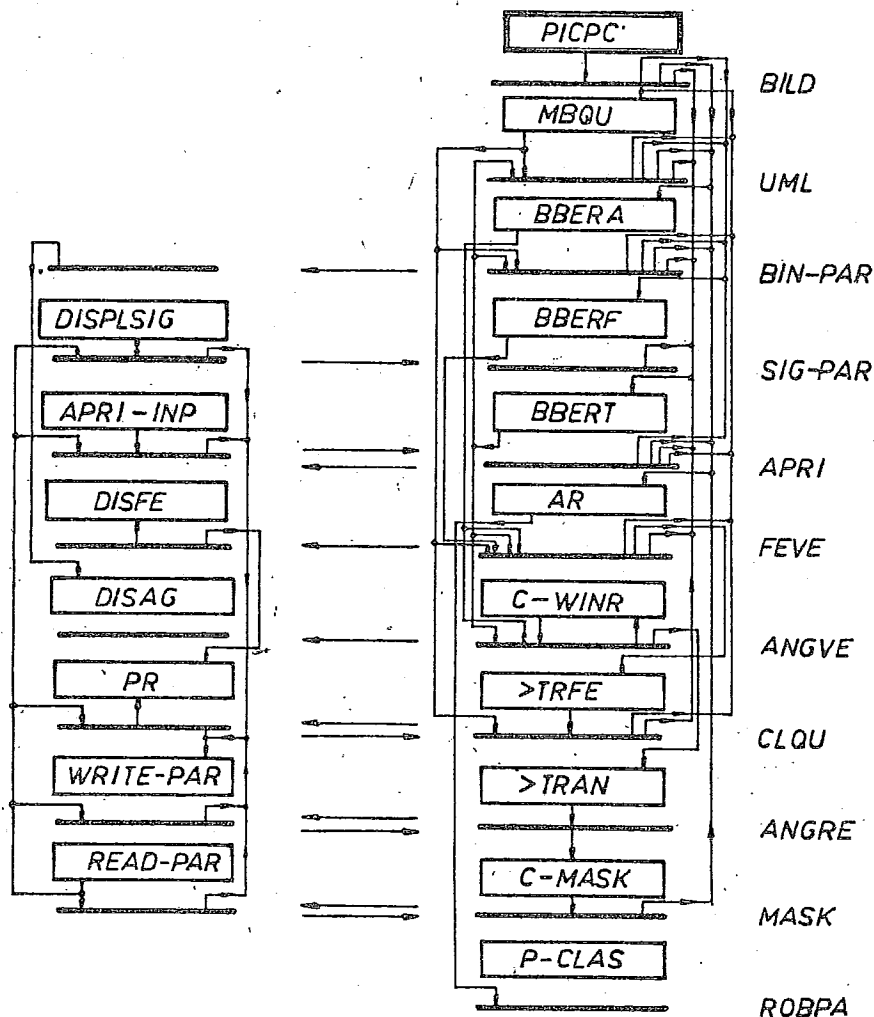


Bild 3.6: Datenflussplan des Verfahrens BINPOS

Wenn es ein bisher unbekanntes Teil ist, werden die Merkmalswerte mit einem "Aufweitungsfaktor" multipliziert und als nächste Klasse in den Klassenquader übertragen (>TRFE). Von den berechneten Winkeln werden mit C-WINR die Nummern ermittelt, die nach dem Signifikanztest Werte unter 360 Grad haben und zusammen mit einem Fehlerbyte, falls nicht genügend Winkel fuer die eindeutige Orientierungsbestimmung vorhanden sind, mit >TRAN in die Winkelreferenzmatrix übertragen.

Das Angebot und die Abspeicherung mehrerer Teile erfolgt in einer Schleife, bis maximal 10 Teile angeboten wurden oder bis der Bediener die Belehrungsschleife in dem dafuer vorgesehenen Frage-

Antwort-Programm vorzeitig abbricht. Nach Beendigung der Belegungsschleife wird mit C-MASK geprüft, welche Winkel bei keiner Klasse zur vollständigen Orientierungsbestimmung herangezogen werden müssen. Diese Winkel werden in einer Maske markiert.

Abschliessend kann in der Trainingsphase der Klassenquader durch ein Mehrfachangebot der bereits eingelernten Teile verbessert werden. Dazu werden die Teilenummern der angebotenen Teile angegeben. Mit BBERF werden nur die morphometrischen Merkmale berechnet. Wenn sich in MBQU zeigt, dass die mit dem Aufweitungsfaktor multiplizierten Merkmale kleiner oder die oberen grösser als die im Klassenquader vorhandenen sind, werden die neuen abgespeichert. Die Mehrfachbelehrung kann jederzeit abgebrochen werden.

Die Trainingsphase wird mit der Frage beendet, ob die eingelernten Parameter abgespeichert werden sollen. Für diese Parameter muss vorher ein File unter SCP auf einer Diskette angelegt worden sein. Mit WRITE-PAR werden nacheinander der Klassenquader, die A-priori-Werte, die Winkelreferenzmatrix, die Signifikanztestparameter und die Winkelmaske auf der Diskette abgelegt. Die Gesamtlänge dieses Datenfiles beträgt 1 KByte.

In der Arbeitsphase können diese abgespeicherten Parameter mit READ-PAR zunächst in die BEE 1010 geladen werden. Danach erfolgt nur ein zyklischer Aufruf der Bildaufnahme und mit BBERA die Berechnung der Merkmale und Winkelwerte unter Berücksichtigung der Maske. P-CLAS ergibt die Klassennummer, aus der die Teilenummer bestimmt wird. Mit C-WINR wird schliesslich der auszugebende Winkel bestimmt. Die Werte, die Lage und Orientierung des Teils charakterisieren, werden im Roboterparametervektor ROBVE zusammengefasst und zur Unterstützung des Tests auf dem Display, das an der BEE 1010 angeschlossen ist, angezeigt. Der zyklische Ablauf der Arbeitsphase kann mit ^D unterbrochen werden.

In der Prüfphase ist die Übertragung des aktuellen Merkmalsvektors und des Klassenquaders von der BEE 1010 zum Steuerrechner vorgesehen. Im Steuerrechner werden die Klassenquadergrenzen neben dem zugehörigen Merkmal schrittweise für jede eingelernte Klasse angezeigt.

Folgende Funktionen stehen als Assembler-Module zur Verfügung (3 * 1 K Byte EPROM):

MDM2 - Berechnung der Momente 2. Grades

CIRC - Berechnung und Abspeicherung der Konturpunkte und der von der Kontur umschlossenen Fläche

- RMAX - Bestimmung des groessten Abstandes zwischen Schwerpunkt und Kontur
- KOMO - Berechnung der konturbezogenen Momente 1. Grades
- UMF - Berechnung der Konturlaenge
- CBINP - Berechnung der Teileflaeche und der Momente 1. Grades

3.3.4. Szenenanalyse (BINSK)

3.3.4.1. Zielstellung und Anwendungsbereich

Das Verfahren BINSK dient zur Analyse von Binaerbildern im Run-Length-Code RLCA mit dem Ziel, die in einer Szene vereinigte Objektgruppe fuer eine sequentielle Einzelobjektverarbeitung verfuegbar zu machen.

Bei einer Bildaufnahme von Einzelobjekten koennen durch Verschmutzung, Lichtreflexe u.a. weitere fiktive Objekte erfasst werden. Aus der so entstandenen Szene kann mit dem Verfahren das zu erkennende Einzelobjekt selektiert werden, sofern mit den Selektionsmerkmalen Flaeche, Umfang oder Formfaktor eine Unterscheidung gegenueber den Stoerobjekten moeglich ist.

Die Anwendung des Verfahrens zur Bildbereinigung ist fuer die Verfahren wie PATMAT und BINPOS dann relevant, wenn mit Stoerobjekten zu rechnen ist, da von beiden Verfahren nur Bilder mit einem Objekt verarbeitet werden. Bei einem gleichzeitigen Angebot mehrerer Objekte in einer Szene koennen mit einer Bildaufnahme alle relevanten Objekte verarbeitet werden.

Bei der Verarbeitung werden mit verschiedenen Merkmalskriterien (Flaeche, Umfang oder Formfaktor) sowie mit gesetzten Merkmalsgrenzwerten alle oder spezielle Objekte aus der Szene sequentiell selektiert.

Mit BINSK koennen u.a. Objekte gezaehlt und selektiert werden. Hoehere Verarbeitungsstufen (wie: Positions- und Orientierungsbestimmungen, Klassifikation) koennen in Verbindung mit einzelnen Berechnungsfunktionen aus dem Verfahren BINPOS ausgefuehrt werden.

3.3.4.2. Struktur der Verfahrensloesung

Das Verfahren besteht aus den Verarbeitungsstufen (Bild 3.7):

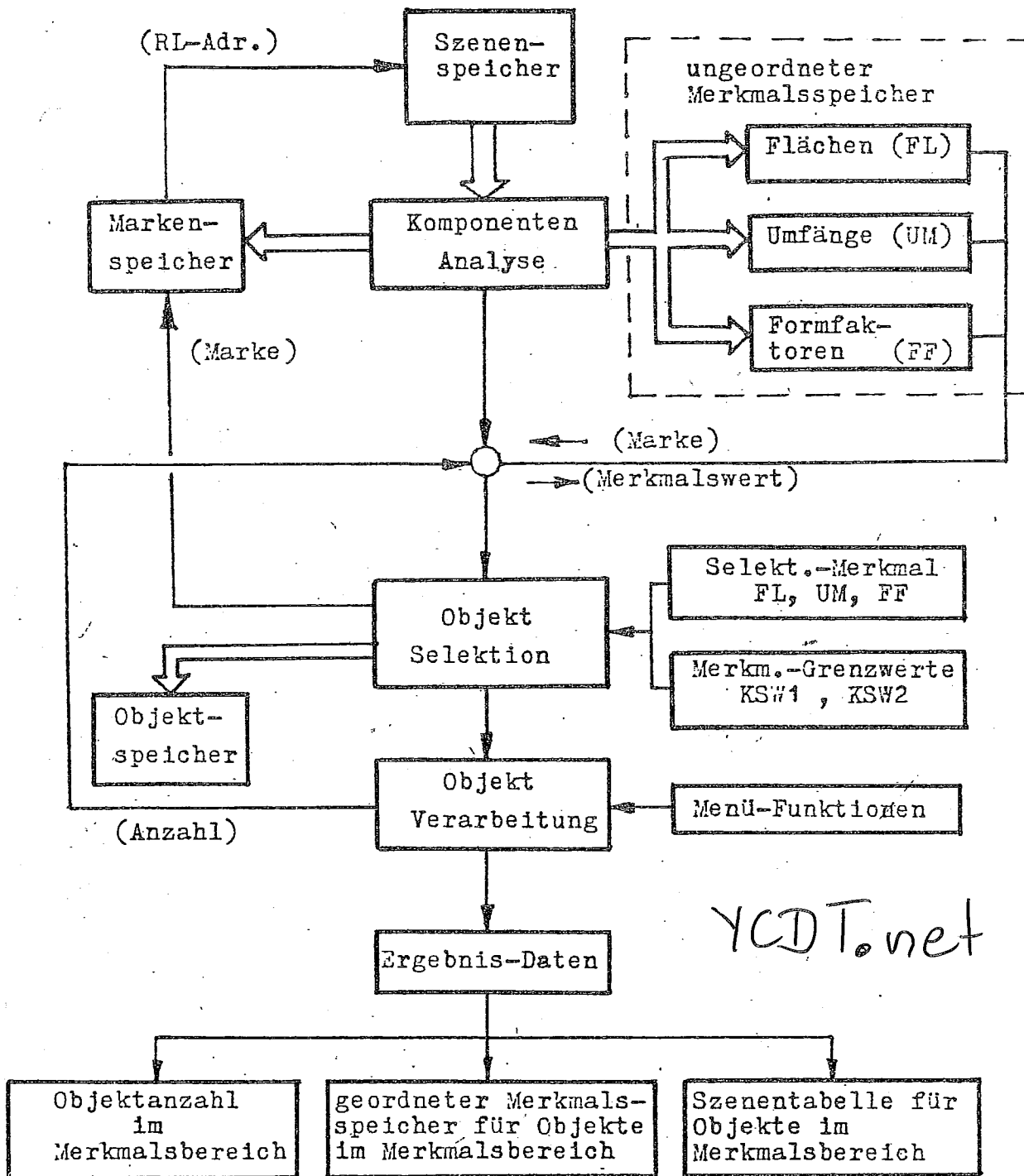


Bild 3.7: Schematische Darstellung der Szenenanalyse

- Komponentenanalyse
- Objektselektion
- Objektverarbeitung.

Bei der Komponentenanalyse werden zeilenweise Koinzidenzen (Zusammenhaenge) zwischen einzelnen Objektsegmenten aus zwei benachbarten Bildzeilen analysiert und mit Marken gekennzeichnet. Gleiche Marken bezeichnen Segmente eines Objektes. Gleichzeitig erfolgt eine inkrementelle Berechnung der Flaechen und Umfaenge fuer alle Objekte der Szene (vgl. /3.6/, S.61-68, /3.8/).

Nach der Komponentenanalyse wird die Markierung aller zusammenhaengenden Objektteile mit unterschiedlichen Marken auf eine einheitliche Marke korrigiert. Gleichzeitig werden entsprechend den Objektkorrekturen die Merkmalsspeicher korrigiert.

Aus den objektbezogenen Werten fuer Flaechen und Umfang werden abschliessend die entsprechenden Formfaktoren berechnet.

Bei der Objektselektion erfolgt der Zugriff zum Einzelobjekt im Bildspeicher ueber die Position der Marke im Markenspeicher fuer jedes einzelne Objektsegment. Das so selektierte Objekt wird fuer die weitere Verarbeitung in einen separaten Bildspeicher transferiert.

Bei Zaehl- oder Ordnungsaufgaben ohne Objektselektion erfolgt lediglich eine Auswertung der waehrend der Komponentenanalyse generierten Merkmalsspeicher unter Beruecksichtigung der gesetzten Merkmalsgrenzwerte.

Die Objektverarbeitung beinhaltet neben der genannten Zaehl- und Ordnungsfunktion auch eine Funktion zur Positionsbestimmung sowie Anschlussmoeglichkeiten fuer Funktionen zur Orientierungsbestimmung und Klassifikation.

Die Ergebnisse werden in einer Szenentabelle zusammengefasst.

3.3.4.3. Programmtechnische Loesung

Das Programm zur Szenenanalyse BINSC wird unterteilt in :

- Parameterphase (SZ-PAR)
- Arbeitsphase (SZ-RUN)

(Bild 3.8).

Die Parameterphase ist vollstaendig im Steuerrechner realisiert. Dagegen existieren fuer die Arbeitsphase zwei separate Varianten (unter Beruecksichtigung der Kommunikationsmoeglichkeiten) mit und ohne Einbeziehung des Steuerrechners.

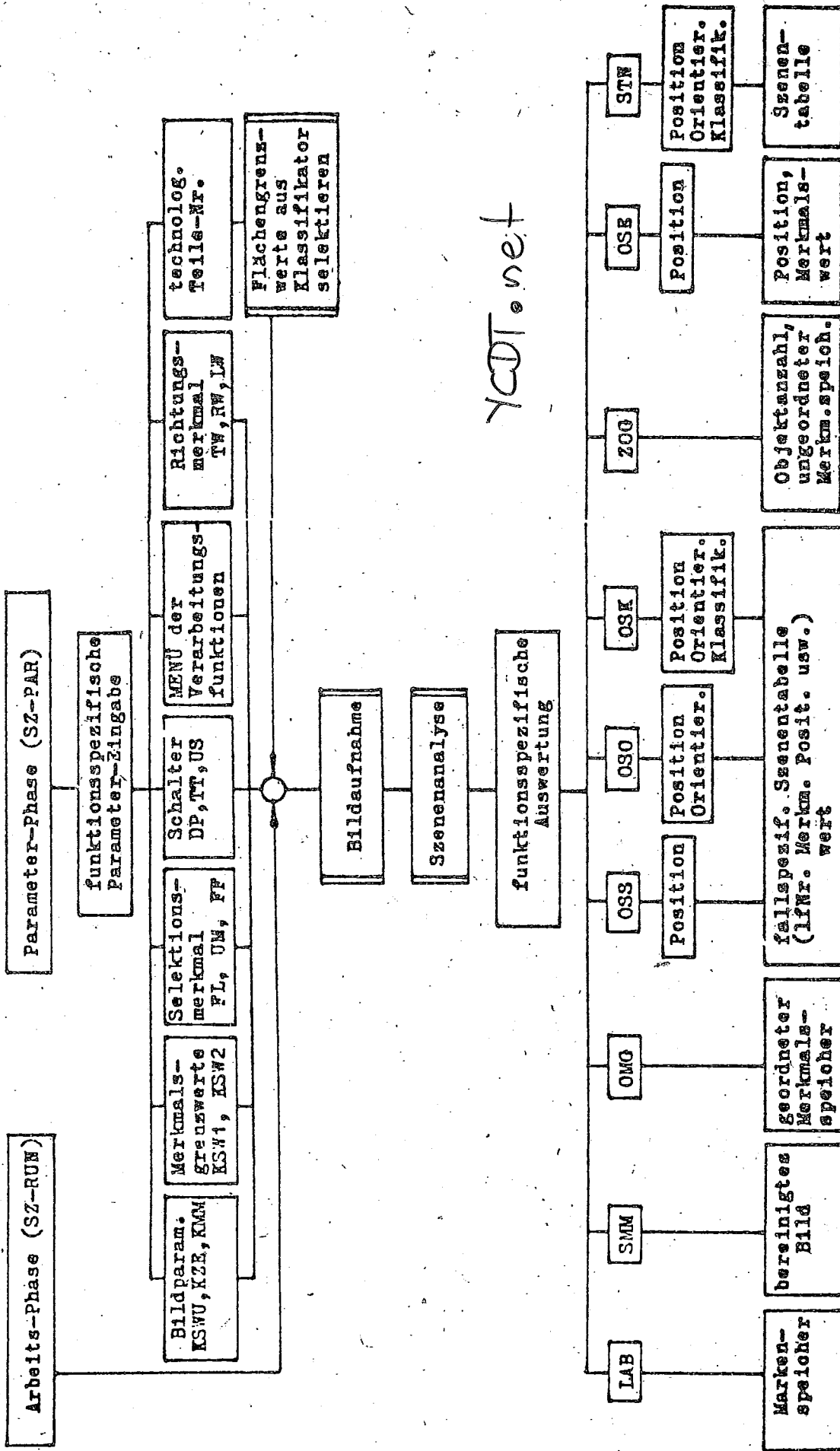


Bild 3.8: Funktionelle Darstellung des Programms BINSC

In der Parameterphase besteht die Moeglichkeit der interaktiven Manipulation an Bildparametern und Steuergroessen fuer den Ver-
fahrensablauf. Die entsprechende Spezifizierung erfolgt durch
Menue-Angebote fuer die Selektionsmerkmale (FL-Flaeche, UM-Um-
fang, FF-Formfaktor) und 3 verschiedene Richtungsmerkmale sowie
fuer das Spektrum der Verarbeitungsfunktionen. Gemaess der ausge-
waehlten Verarbeitungsfunktion (siehe 3.3.4.4.) werden die er-
zielten Ergebnisse tabellarisch angezeigt.

In der Arbeitsphase sind Interaktionen nicht vorgesehen. Es gel-
ten die spezifizierten Parameter. Bei der Aktivierung der Ar-
beitsphase seitens des Steuerrechners werden die Ergebnisse wie
in der Parameterphase angezeigt.

Die Arbeitsphase in der BEE 1010 enthaelt keine Ergebnisanzeige.
Die ermittelten Werte stehen zur weiteren Auswertung im Speicher
bereit.

Beide Phasen des Verfahrens arbeiten unabhaengig voneinander. Sie
enthalten den Aufruf zur Bildaufnahme, setzen jedoch die richtige
Belegung von Aufnahmeparametern zur Integrationszeitsteuerung und
Binarisierung voraus.

Bei bestimmten Verfahrensfunktionen, die eine Orientierungsbe-
stimmung und Klassifikation erfordern, sind zusaetzliche Funktio-
nen z.B. aus dem Verfahren BINPOS zu verwenden.

In BINSK sind folgende Verarbeitungsfunktionen wahlbar:

LAB - Komponenten-Markierung

SMM - Objektselektion mit max. Merkmalswert (Bildbereinigung)

OMG - Objekte nach Merkmalsgroesse ordnen

ZOG - Objekte zaehlen

OSE - Objekte selektieren

OSS - Objektselektion mit Schwerpunktberechnung

OSO - Objektselektion, Schwerpunktberechnung und
Orientierungsbestimmung

OSK - Objekte selektieren und klassifizieren

STN - Objektselektion mit technologischer Teile-Nr.

SYS - Arbeit beenden, zurueck zum System

3.3.4.4. Anwendungshinweise

Nach dem Laden muss das Programm grundsaeztlich mit der erweiter-
ten Parameterphase (SZ-PAR0) gestartet werden, um Anfangswerte
fuer die Programmvariablen zu setzen.

Bei der EPROM-Version in der BEE 1010 sind Anfangswerte vorgegeben, so dass die Arbeitsphase sofort aufgerufen werden kann.

Die während der Komponentenanalyse erzeugten Umfangs-Inkremente können grafisch dargestellt werden, falls eine Grafik GDM 183 angeschlossen ist. Zur Aktivierung der Anzeigefunktionen ist in der Parameterphase der Schalter "UM-INC. ANZEIGEN" mit - j - zu belegen. Bei den Funktionen OSE, OSS, OSO u. OSK werden dann die selektierten Objekte mit ihrer vollen Fläche in die konturhafte Szenendarstellung eingeblendet.

Die im Menue angebotenen Funktionen können wie folgt verwendet werden:

- Komponenten-Markierung (LAB)

Zur Überprüfung der Durchführbarkeit einer Szenenanalyse wird mit der Funktion LAB getestet, ob die Anzahl der Objekte, die (aus der geometrischen Objektstruktur resultierende) Komponentenanzahl und die Run-Length-Anzahl je Zeile die zulässigen Grenzwerte nicht überschreiten. Andernfalls müssen die entsprechenden Parameter innerhalb ihrer Variationsmöglichkeiten verändert werden.

- Objekt mit max. Merkmalswert selektieren (SMM)

Mit dieser Funktion können Operationen zur Bildbereinigung durchgeführt werden, wobei für die Unterdrückung von Störobjekten in der Szene die Merkmale Fläche, Umfang oder Formfaktor (normiert auf Kreisscheibe) verwendet werden können. Bei der Bildbereinigung wird der ursprüngliche Bildspeicherinhalt gespeichert.

- Objekte nach Merkmalsgrösse ordnen (OMG)

Die in der Szene enthaltenen Objekte werden mit ihrem Merkmalswert (entsprechend dem aktuell definierten Selektionsmerkmal Fläche, Umfang oder Formfaktor) in einem Merkmals-Grenzwertbereich (min. und max. Merkmalswert) auf Zulässigkeit überprüft und gegebenenfalls mit ihrem Merkmalswert in einem speziellen Speicherbereich geordnet gesammelt. Die Objektzuordnung zu den Merkmalswerten erfolgt durch die jeweilige Marke des Objektes aus dem Merkmalspeicher. Für eine grössengeordnete Objektverarbeitung kann das gewünschte Objektspektrum mit Hilfe der Selektionsmerkmale und der Merkmalsgrenzwerte zusammengestellt werden. Der geordnete Merkmalspeicher wird ausgegeben.

- Objekte zaehlen (ZOG)

Zur Ueberpruefung einer geforderten Anzahl von Objekten in einer Szene kann der Zaehlbereich durch die Merkmalsgrenzwerte entsprechend manipuliert werden. Der Merkmalswert, als zu zaehlende Groesse innerhalb der Grenzwerte, ist dabei mit den Selektionsmerkmalen Flaechen, Umfang oder Formfaktor in einen passenden Bezug zum Objekt zu bringen. Die ermittelte Objektanzahl wird ausgegeben.

- Objekte selektieren (OSE)

Durch Vorgabe geeigneter Merkmals-Grenzwerte, bezueglich des gesetzten Selektionsmerkmals Flaechen, Umfang oder Formfaktor, kann gezielt ein Einzelobjekt (oder eine spez. Objektgruppe) aus einer Szene selektiert werden. Das erfasste Objekt wird angezeigt und steht zur weiteren Verarbeitung in einem gesonderten Einzelobjekt-Speicher bereit. Bei mehreren Objekten innerhalb des Grenzwertebereichs koennen die Objekte sequentiell in den Einzelobjektspeicher uebertragen werden.

- Objekte selektieren und Schwerpunkte berechnen (OSS)

Die durch die Merkmals-Grenzwerte ausgewaehlten Objekte werden groessengeordnet und sequentiell aus dem Szenenbildspeicher in einen Einzelobjektspeicher transferiert. Dabei werden die jeweiligen Schwerpunktkoordinaten des Objektes bezueglich der Szene berechnet und gemeinsam mit den Merkmalswerten in eine Szenentabelle eingetragen. Die Objekte werden einzeln angezeigt. Abschliessend wird die Szenentabelle ausgegeben. Die Eintragungen in der Szenentabelle (hier: Merkmale und Objektpositionen) koennen einerseits zur Anwesenheitskontrolle bestimmter Objekte in geforderten Positionen verwendet werden, eignen sich andererseits auch fuer Zugriffsoperationen durch einen Manipulator.

Es lassen sich auch Relationen zwischen Objektpositionen auswerten, um geforderte Abstaende zu ueberpruefen. Die Merkmalswerte lassen sich auch im Sinne einer messtechnischen Auswertung der selektierten Objekte verwenden.

- Objekte selektieren, Schwerpunkte u. Orientierungen berechnen (OSO)

Der Verfahrensablauf OSS und moegliche Anwendungsrichtungen werden durch Hinzunahme einer Orientierungsbestimmung nach den Richtungsmerkmalen "Traegheitswinkel, Radiuswinkel oder Lochwinkel" ergaenzt. Die Szenentabelle wird mit den Orientierungs-

angaben erweitert (vgl. BINPOS, 3.3.3.).

Die Richtungsangaben koennen sowohl fuer eine geometrische Vermessung der Objekte als auch fuer einen gerichteten Manipulatorzugriff verwendet werden.

- Objekte selektieren und klassifizieren (OSK)

Durch Vorgabe von Grenzwerten fuer die Selektionsmerkmale Flaeche, Umfang oder Formfaktor werden aus einer Szene die zugelassenen Objekte selektiert und sequentiell einem Erkennungsverfahren zugefuehrt. Die dabei berechneten Angaben zur Objektposition (Schwerpunktkoordinaten), Orientierung, Klassennummer (technol. Teile-Nr.) und Auflageseite werden zusammen mit dem Merkmalswert zeilenweise in einer Szenentabelle aufgefuehrt. Die Reihenfolge der Objektverarbeitung erfolgt groessengeordnet bezueglich des Selektionsmerkmals. Mit dem Klassifikationsergebnis koennen Aussagen ueber die Anwesenheit/Verfuegbarkeit bestimmter Objekte bzw. Objektlagen verwertet werden (sowohl fuer Kontrollfunktionen als auch im Manipulationsprozess zur Steuerung der Bewegungsablaeufo eines Roboters).

- Objekt mit technologischer Teile-Nr. selektieren (STN)

Durch die gezielte Vorgabe einer Teile-Nr. (z.B. entsprechend dem geforderten Manipulationsablauf) soll das bekannte Objekt aus einer Szene selektiert und in seiner Position und Orientierung sowie Auflageseite ausgewiesen werden.

Mit der Teile-Nr. werden aus den im Klassenquader des Klassifikators abgespeicherten Angaben objektspezifische Grenzwerte fuer das Selektionsmerkmal "Flaeche" ermittelt. Aus den einzelnen Merkmalswerten der Objekte einer Szene wird der fuer die Grenzwerte passende Merkmalswert gesucht und damit die zugehoerige Objekt-Marke ermittelt. Ueber die Objekt-Marke wird das Objekt aus der Szene selektiert und in den Einzelobjektspeicher uebertragen.

Die nachfolgende Objekterkennung liefert die genannten Daten fuer die auszugebende Tabelle.

Durch mehrfache Anwendung der Funktion koennen die Zugriffsdaten fuer eine festgelegte Objektreihenfolge ausgewiesen werden. Aufgrund des beschraenkten Speicherplatzes koennen nicht beliebig komplexe Szenen verarbeitet werden.

Fuer die Datenspeicherung bestehen folgende Anforderungen (in Byte):

$$R = 3/2 * (KZE * KSWU) + 10 * KMM + 12 * KOZ$$

mit:

KZE - Bildzeilenanzahl

KSWU - Run-Length-Anzahl pro Bildzeile

KMM - max. Markenzahl fuer Szenenkomponenten

KOZ - max. Objektzahl in einer Szenentabelle

Fuer eine Szenengestaltung gilt als Orientierung, moeglichst solche Objekte zu verarbeiten, deren Silhouetten eine geringe Run-Length-Anzahl (< 20) pro Bildzeile hervorbringen.

Dadurch benoetigen die Bild- und Markenspeicher < 7,5 KByte.

Die zusaetzlichen Speicheranforderungen fuer die Merkmalstabellen (10*KMM) und die Szenentabelle (12*KOZ) sind dann von untergeordneter Bedeutung.

Bei der Nutzung der Funktionen OSK und STN sind die Anschuesse aus dem Verfahren BINPOS zu realisieren.

Messgenauigkeit: - Positionsbestimmung +- 1 Bildpunkt
- Orientierungsbestimmung +- 1,4 Grad

3.4. Hardware-Treiber

3.4.1. Bildaufnahmefunktionen

Entsprechend der Ausruestung der BEE 1010 mit der Kamerasteuereinheit (STE), der Puffer- und Codiereinheit (PCE) sowie der Multiplexeinheit (MPE) fuer den Betrieb von maximal 3 Zeilenkameran der Typen ZFK 1021 und ZFK 1040 ist ein Paket von Treiberprogrammen zu einer Woerterbuecherweiterung zusammengefasst worden. Diese Woerterbuecherweiterung steht nur im VS-FORTH der BEE 1010 zur Verfuegung. Die Programme sind zusammen 2 kByte lang. Der Datenaustausch mit uebergeordneten Programmen erfolgt ueber den MAIL-Bereich (siehe 3.2.4.) auf der Grundlage von vereinbarten vierbuchstabigen Adressnamen. Die Erweiterung enthaelt folgende Grundfunktionen:

INIMAIL Erzeugung einer Anfangsbelegung des MAIL-Bereiches durch Verschieben von Werten aus dem PROM-Bereich in den RAM-Bereich von MAIL.

MAILAD Zugriff zum MAIL-Bereich. Als Eingangsgroesse verlangt

diese Funktion die Anfangsadresse von 4 ASCII-Zeichen. Als Ausgangsgroesse wird die Adresse des Namens geliefert oder eine 0, falls die 4 ASCII-Zeichen nicht zu den vereinbarten Adressnamen gehoeren.

CASEL

Auswahl des Kameratyps und der Kameranummer. Aus den Eingabewerten fuer Kameratyp (1 oder 2) und Kameranummer (1, 2 oder 3) wird ein Steuerwort erzeugt und ueber die Portadresse der MPE ausgegeben. Das Steuerwort wird im MAIL-Bereich abgelegt. Alle Portadressen werden ebenfalls dem MAIL-Bereich entnommen. Dadurch ist die Software leicht an eventuell geaenderte Hardware-Adressen anpassbar.

LINRC

Aufnahme einer Bildzeile im RLCN-Code und Ablage im Rechnerspeicher. Die Speicheranfangsadresse wird dem MAIL-Bereich (ABUF siehe Tabelle 3.2) entnommen. Die Aufnahme erfolgt entsprechend den im MAIL-Bereich eingetragenen Parametern. Es werden unabhaengig von der Zeileninformation 128 Byte aus dem PCE-Puffer in den Rechnerspeicher uebertragen.

BUFST

Start einer zyklischen Zeilenabtastung, wobei die Daten nur im PCE-Puffer seriell abgelegt werden. Wenn eine Pufferseite mit Daten von der Kamera gefuellt ist, wird automatisch auf die andere Pufferseite umgeschaltet. Diese Funktion verlangt als Eingangsgroesse eine Adresse. Der Wert auf dieser Adresse wird beim Umschalten des Puffers geaendert. Damit kann die Verarbeitung der Daten mit den asynchron einlaufenden Kameradaten synchronisiert werden.

CSTOP

Beenden der zyklischen Zeilenabtastung.

PICRAM

Auslesen einer Pufferseite (2KByte) der PCE in den Arbeitsspeicher der BEE 1010. Die Zieladresse fuer den Arbeitsspeicher wird dabei dem MAIL-Bereich entnommen.

Die folgenden Worte dienen der Erzeugung flaechenhafter Abbildungen mit der Kamera ZFK 1021. Dazu ist eine gleichmaessige Bewegung der aufzunehmenden Objekte notwendig.

PICPCE

Bildaufnahme mit Bildanfangserkennung. Als Bildanfang gilt, wenn mindestens ein Uebergang von hell nach dunkel oder umgekehrt innerhalb einer Zeileninformation festgestellt wurde. Nach der Bildanfangserkennung wird

ein Bild aus einer Zeilenzahl < 256 zusammengesetzt, die dem MAIL-Bereich entnommen wird.

PICDEC Die Bilddaten im RLCN-Code werden aus dem Puffer der PCE ausgelesen und in den RLCA-Code umgewandelt, der eine im MAIL-Bereich vorgegebene maximale Anzahl von Run-Length-Daten pro Zeile enthaelt.

PICPC Kombination der Funktionen PICPCE und PICDEC.

Die Bildanfangserkennung in PICPCE erfolgt, wie in Teil I, 2.3.3 beschrieben, mittels der Pufferseitenumschaltung auf der PCE. Es wird nach einer Einzelzeilenabtastung ein OUT-Befehl zur Pufferseitenumschaltung gegeben. Danach kann die CCD-Zeileneinformation aus der vom Rechner zugreifbaren Pufferseite ausgelesen und ausgewertet werden (RLCN-Code). Bei der Auswertung wird nur geprueft, ob das 2. Byte den Code fuer Zeilenende (FDH) enthaelt. In diesem Fall handelt es sich um eine voellig dunkle Zeile. Falls das 1. Byte den Code fuer den Uebergang zum Hellsignal im 1. Rixel der Zeile enthaelt (80H), wird zusaetzlich das 3. Byte auf den Zeilenende-Code geprueft. Wenn der Test positiv ist, handelt es sich um eine voellig helle Zeile. Auf diese Weise kann der Teileanfang gesucht werden, unabhaengig davon, ob es sich um ein helles Teil auf dunklem Grund oder ein dunkles Teil auf hellem Grund handelt. Falls der Teileanfang gefunden wurde, wird nicht mehr umgeschaltet und die Zeileneinformationen werden seriell in die aktuelle Pufferseite geschrieben. Die erste Zeile, die Informationen ueber das Teil enthaelt, geht nach diesem Verfahren verloren.

Grundlage der Bildaufnahmefunktionen ist die Ansteuerung der CCD-Zeile durch den CTC-Baustein auf der STE (siehe Teil I, 2.3.1.) sowie die Programmierung des Fenster-Komparators der STE fuer die Binarisierung. Die Werte fuer die untere und fuer die obere Schwelle des Komparators koennen in 256 Stufen von 0 bis zum Maximalwert durch OUT-Befehle veraendert werden. Um die Komparatorschwelle fuer uebergeordnete Programme variabel zu halten, werden die Werte fuer die OUT-Befehle dem MAIL-Bereich entnommen. Um eine aequidistante Abtastung eines bewegten Teils zu gewaehrleisten, wird mit dem CTC-Baustein ein Interruptbetrieb der Kamera realisiert. Der CTC-Baustein wird so programmiert, dass die Zaehler 0 und 1 Interrupts ausloesen. Die Impulse dieser beiden Zaehler steuern Integrationsanfang und -ende und damit die Em-

pfindlichkeit der CCD-Zeile.

Zaehler 2 wird als Zeitgeber programmiert. Dessen Impulse werden durch Zaehler 1 registriert. Dafuer ist eine Hardware-Verbindung zwischen dem Ausgang von Zaehler 2 und dem Eingang von Zaehler 1 hergestellt. Zaehler 1 ist als Impulzahaehler programmiert. Das Integrationsende wird durch den Nulldurchgang von Zaehler 1 hardwaremaessig ausgeloeset. Der Zaehler 1 kann mit einem Wert aus dem MAIL-Bereich voreingestellt werden. Jeder Impuls von Zaehler 2 setzt den Zaehler 1 um einen Wert zurueck. Der Zeitgeber Zaehler 2 ist mit einem solchen Wert eingestellt, dass seine Zeitimpulse zusammen mit dem groessten Wert fuer Zaehler 1 (255) eine Integrationszeit von 100 ms bei einem Rechnertakt von 2,5 MHz liefern. Dies ist etwa die obere Grenze fuer die CCD-Zeile. Der kleinste Wert, mit dem Zaehler 1 voreingestellt werden kann, ist hardwarebedingt 2. Damit wird eine Integrationszeit von ca 0,5 ms erreicht. Fuer noch kuerzere Zeiten muss in dem Programm PICPCE der fest vorgegebene Wert (40H) fuer den Zaehler 2 auf einen entsprechend kleineren Wert gesetzt werden. Alle Bildaufnahme-funktionen setzen voraus, dass die Miniaturschalter auf der STE und der PCE entsprechend der Betriebsanleitung so geschaltet sind, dass die Codierung der binaren Bilddaten auf der PCE im Run-Length-Code erfolgt. Eine Einzelzeilenaufnahme erfolgt durch Programmierung und Starten der 3 Zaehler. Der Zaehler 0 muss dabei so eingestellt sein, dass er den Nulldurchgang frueher erreicht als Zaehler 1. In der Interruptroutine von Zaehler 0 wird dieser Zaehler gestoppt. Eine zyklische Abtastung wird dadurch erreicht, dass in der Interruptroutine von Zaehler 1 alle 3 Zaehler wieder neu gestartet werden. Mit der Funktion BUFST ist eine ununterbrochene zyklische Abtastung moeglich. Zur Synchronisation der Kamera mit dem Verarbeitungsprogramm fuer die Daten muss das Umschalten der Pufferseiten erkannt werden. Dies wird bei der Funktion BUST durch eine Interruptroutine fuer den PIO-Baustein auf der ZRE-Platine geloest. Die ZRE ist die K 1520-Standardkarte mit der zentralen Recheneinheit der BEE 1010. Dafuer ist eine Hardwareverbindung zwischen der PCE und der ZRE (BIT 7 von Kanal 1 des PIO-Bausteins) notwendig. In der Interruptroutine wird geprueft, ob der Inhalt der uebergebenen Adresse 0 ist. Falls der Wert 0 ist, wird eine 1 eingetragen. Wenn beim naechsten Durchlaufen der Interruptroutine noch die 1 enthalten ist, wird 40H eingetragen. Auf diese Weise kann ein uebergeord-

netes Programm erkennen, ob die Pufferseiten ein- oder zweimal durch die Kamera umgeschaltet wurden.

3.4.2. Graphik fuer GDM 183

Die Graphik-Funktionen fuer das Binaerbild-Rasterdisplay GDM 183 sind nur in der 2. Variante der Systemsoftware verfuegbar. Der Koordinatenursprung ($x=0$, $y=0$) liegt in der linken unteren Ecke des Bildschirms; x verlaeuft nach rechts, y nach oben. Es stehen folgende Funktionen zur Verfuegung:

DPORT Uebergabe der Portadresse (oberes Halbbyte) und Untergrundfarbe (unteres Halbbyte) fuer das Rasterdisplay, z. B. $n=B0H$ fuer die Portadresse B und Untergrund dunkel und $n=B1H$ fuer Portadresse B und Untergrund hell. Die Displayfunktionen setzen die vorherige (einmalige) Uebergabe von Portadresse und Untergrundfarbe voraus.

DCLEAR Rasterdisplay wird auf Untergrundfarbe gesetzt.

DISP Aufruf der Bildausgabe mit Fehleradresse, maximaler Bildzeilen-Nummer und minimaler Bildzeilen-Nummer in x -Richtung, Zahl der reservierten Bytes/Zeile fuer den Run-Length-Code und der RLCA-Bildspeicheradresse. Bei erkennbaren Fehlern in RLCA wird das Bit 5 (20H) der Fehlerzelle gesetzt.

DISPC Hintereinanderausfuehrung von DCLEAR und DISP.

DRAWL Die Punkte mit den Koordinaten (x_1 , y_1) und (x_2 , y_2) mit $0 \leq x_1$, y_1 , x_2 , $y_2 \leq 255$ werden durch eine zum Bild invertierte Linie verbunden.

3.5. Vertriebsvarianten Systemsoftware

Die Systemsoftware wird wie folgt vertrieben (zur Hardware-Ausstattung der 1. Variante vgl. 2.1.7.):

1. File IPCF.COM : FORTH-Kernsoftware fuer Steuerrechner
2. File IPCREL.SCR : zur Herstellung von PROM-Modulen fuer die BEE 1010
3. EPROM 1-3 fuer ZRE K2521 : Sprungtabelle und Bildaufnahme
4. EPROM 1-7 fuer PFS K3820 : FORTH-Kernsystem
- 5.1. EPROM 13-16 fuer PFS K3820 : Gleitkommapakete, Kalibrierung

Die 2. Variante (vgl. 2.1.7.) enthaelt anstelle von 5.1. :

5.2. EPROM 2-6 fuer DFS K3620 : Gleitkommapaket, Kalibrierung,
Graphikpaket

(alle Angaben entsprechend /9/).

3.6. Literaturverzeichnis

1. Zech, R.: Die Programmiersprache FORTH, Franzis Verlag GmbH, Muenchen, 1984
2. Brodie, L.: Programmieren in FORTH, Carl Hanser Verlag, Muenchen und Wien; Prentice Hall International Inc., London 1984
3. Brodie, L.: In FORTH denken, Carl Hanser Verlag, Muenchen und Wien; Prentice Hall International Inc., London 1985
4. Westendorf, C.-M.; Schroeder, U.; Neumann, A.: Anwendungsbeschreibung. Systemsoftware fuer Bilderkennungseinheit BEE1010. VEB Studioteknik Berlin, 1987. ZKI der AdW der DDR, Berlin, 1987
5. Vack, G.-U.: Die erweiterbare Mikrorechner-Programmiersprache FORTH. Dokumentation. ZKI d. AdW d. DDR, Dresden, 1985
6. Fritsch, K.: Visuelle Sensoren, Akademie-Verlag, Berlin 1987
7. Schoenborn, W.; Neumann, A.; Schroeder, U.: Teile- und Lageerkennung mit MAVIS. Bild und Ton, Leipzig, 38(1985)12, S. 357-360
8. Wloka, K.: Ein Verfahren zur Selektion von Einzelobjekten mittels Flaechenwert. Bild und Ton, Leipzig, 37(1984)5, S. 145-148
9. Bedienungsanleitung fuer Bilderkennungseinheit BEE 1010, Ausgabe 1/1987. VEB Studioteknik Berlin, 1987