

DESy DV-81/01
Juli 1981

K o n z e p t e i n e s i n t e r a k t i v e n
R a s t e r d i s p l a y - S y s t e m s a u f
d e r B a s i s l o g i s c h e r I n d e x b i l d e r

von

Grigore C. Nicolae

Zusammenfassung

Die vorliegende Arbeit befaßt sich mit dem strukturellen Entwurf von Rastercandisplay-Systemen als Werkzeuge für die Kommunikation zwischen Mensch und informationsverarbeitenden Maschinen mit Hilfe von Bildern. Eine effiziente Bildkommunikation, die die unbeschwerte Interpretation komplexer Begriffe auf Grund ihrer graphischen oder bildlichen Darstellung ermöglichen soll, setzt voraus, daß die betrachteten Begriffe durch Verwendung umfangreicher optischer Attribute anschaulich dargestellt und außerdem in beliebiger Weise interaktiv manipuliert werden können. Ein wesentlicher Aspekt beim Entwurf solcher Rastercandisplay-Systeme stellt die gemeinsame Betrachtung der Datenstruktur der Bilder und der informationellen Struktur des Display-Systems dar. Die erste legt ein Bildmodell für die Codierung der Bilder fest, während die zweite die Algorithmen für die Erzeugung, Darstellung und Manipulation solcher Datenstrukturen bestimmt.

Gegenstand der Arbeit ist die Entwicklung eines neuen Konzeptes für ein farbiges Rastercandisplay-System, das sich für die interaktive Bildkommunikation auf der Ebene logischer Begriffe besonders eignet. Für die Datenstruktur wurde ein hierarchisches Bildmodell entwickelt, das die transparente, logische Beschreibung komplexer graphischer Bilder ermöglicht. Es kann mit gewissen Einschränkungen auch auf die Videobilder übertragen werden. Die informationellen Strukturen, die als Ergebnis eines quasisystematischen Entwurfs resultieren, sind durch die Darstellung der Daten- und Operationsflußebene beschrieben. Dabei wurde eine Reihe neuer Konzepte, wie das des logischen Indexbildes oder das der Flächengruppe, entwickelt, die eine wesentliche Leistungssteigerung des Display-Systems bewirken, ohne seine informationelle Transparenz zu mindern. Die abschließende Untersuchung der wesentlichen Implementierungsaspekte der entwickelten informationellen Strukturen weist auf systematische und technologisch moderne Realisierungsmöglichkeiten solcher Rastercandisplay-Systeme hin.

INHALTSVERZEICHNIS

	<u>Seite</u>
0. Einleitung	1
1. Grundbegriffe	4
1. Allgemeine Prozesse in Display-Systemen	4
2. Allgemeine Strukturen von Display-Systemen	6
1. Bilddarstellung	9
2. Bilderfassung	20
3. Bildmanipulation	22
3. Problematik im Bereich der Display-Systeme	23
1. Bestandsaufnahme	23
2. Entwicklungstendenzen	24
2. Neue Konzepte	25
1. Das Konzept des logischen Indexbildes	25
1. Definition optischer Attribute	26
2. Relation Begriff/Optik	28
3. Konzept der dynamischen Index/Optik-Zuordnung	31
2. Die Flächengruppe als neues Strukturelement	35
1. Flächengenerierung	36
2. Flächenoperatoren	38
3. Konzept der Flächengruppe	39
3. Kachelstruktur des Weltbildes	42
3. Datenstruktur	43
1. Grundbegriffe der Datenstrukturdarstellung	43
2. Hierarchisches Bilddatenstruktur-Modell	44
1. Datenstruktur der Graphikinformation	47
1. Datenstruktur der Vektorzüge	48
2. Datenstruktur der Flächen	51
3. Datenstruktur der Dias	54
4. Datenstruktur der Kacheln	56
2. Datenstruktur der Bildinformation	61
3. Codierungsvorschlag für die graphischen Datenobjekte	61

	<u>Seite</u>
4. Informationelle Strukturen	69
1. Bilddarstellungsinstanz	69
1. Datenstrukturabwicklungsinstanz	76
2. Bildtransformationsinstanz	78
1. Mathematisch-geometrische Transformation	80
2. Auswertung optischer Attribute	91
3. Auswertung freier Attribute	107
4. Kontrollinstanz	111
3. Bildpunktinstanz	111
1. Vektorgenerator	114
2. Flächengenerator	115
3. Bildverarbeitungsinstanz	116
4. Monitorinstanz	117
1. Bildrefresher	119
2. Optik-Transformationsinstanz	120
2. Bilderfassungsinstanz	121
5. Implementierung	125
1. Multiprozessor-Systemarchitektur	125
1. Funktionelle Systembeschreibung	127
2. Implementierungstechniken	129
2. Struktur der Systemkomponenten	130
1. Asynchroner Bus	130
2. Bildspeicher	132
3. Bildgeneratorprozessor	135
4. Monitorprozessor	138
1. Struktur des Bildrefresherwerks	138
2. Programmierbare Echtzeit- Transformationseinheit	141
6. Anhang	147
1. Algorithmus für Vektorgenerierung	147
2. Algorithmus für Flächengenerierung	148
7. Literaturverzeichnis	150

0. Einleitung

Informationsverarbeitende Maschinen stellen für den Menschen, im weitesten Sinne, ein Kommunikationsmedium dar. Obwohl die Verarbeitung von Datenobjekten schnell und genau von der Maschine ausgeführt wird, bleibt die Interpretation dieser so resultierenden Datenobjekte oder deren Verarbeitungsvorschriften nur dem Menschen vorbehalten, der allein ihre begriffliche Bedeutung versteht. Daher hat die Kommunikation zwischen Mensch und Maschine eine zentrale Bedeutung beim Umgang mit informationsverarbeitenden Maschinen.

Die grundsätzliche Problematik des Dialogs zwischen Mensch und Maschine liegt in den unterschiedlichen Codierungsformen der Datenobjekte, die sich einerseits für die Maschine, andererseits für den Menschen eignen müssen. Die für die maschinelle Verarbeitung geeignete Codierungsform wird i. a. durch eine Programmiersprache erzeugt, die wiederum durch einen Satz syntaktischer und lexikalischer Regeln festgelegt wird. Bei der Festlegung dieser Regeln ist man vornehmlich darauf bedacht, unter Berücksichtigung der realen Maschinenarchitektur eine optimale Codierungsform für die Maschine, nicht jedoch für den Menschen, zu erreichen. Die Einführung höherer Programmiersprachen, die die Architektur einer, für den Anwender leichter zu verstehenden, virtuellen Maschine berücksichtigen, haben eine, wenn auch umstrittene, Transparenz allein bei der Codierung von Verarbeitungsvorschriften, nicht aber bei der Codierung von Datenobjekten selbst, gebracht. Eine für den Menschen zweckmäßige Beschreibung der Objekte eines Datenverarbeitungsprozesses und deren gegenseitiger Relationen, die eine unbeschwerte begriffliche Interpretation ermöglichen soll, kann nur mit den bewährten Mitteln der zwischenmenschlichen Kommunikation erfolgen: natürliche Sprache für die Begriffsbeschreibung und bildliche Darstellung für die Begriffsvorstellung. Während die Verwendung der natürlichen Sprache für den Dialog zwischen dem Menschen und der Maschine wegen der bis vor kurzem unüberwindbaren technologischen Schwierigkeiten bei der Spracherkennung bzw. Spracherzeugung nicht stattfinden konnte, kann die bildliche Darstellung, die i. a. unter dem Namen Computer-Graphik bekannt ist, auf beinahe zwei Jahrzehnte Geschichte zurückblicken. Die Effektivität der Bildkommunikation zwischen Mensch und Maschine rührt von der Tatsache her, daß das menschliche Gehirn die unvergleichbare Fähigkeit besitzt, die die Erkennung von Bildobjekten und deren Eigenschaften sowie die Ableitung sehr komplexer Zusammenhänge zwischen den Bildobjekten schnell und (meistens) korrekt ermöglicht. Da der Mensch mit Si-

cherheit die leistungsfähigste "Maschine" für die Interpretation von Bildern darstellt, liegt es nahe, diese Fähigkeit in den Dialog mit den informationsverarbeitenden Rechenmaschinen einzusetzen. Die technischen Systeme, die die Bildkommunikation zwischen Mensch und Maschine in beiden Richtungen durchführen, werden i. a. Display-Systeme genannt. Die einseitige Dialogrichtung, die durch den Namen suggeriert wird - nämlich nur von der Maschine zum Menschen - kommt daher, daß am Anfang die Display-Systeme ausschließlich die symbolische graphische Darstellung von Datenobjekten unterstützten konnten. Heute ist es unbestritten, daß die Leistungsfähigkeit eines Display-Systems hauptsächlich von der Kommunikationsgestaltung in beiden Richtungen abhängt. In der einen geht es um die Informationsmenge über die Datenobjekte, die durch Verwendung von Bildattributen - z. B. Form, Größe, Helligkeit, Farbe - dem Menschen anschaulich zur Interpretation angeboten werden kann. In der anderen geht es um die Manipulationsmöglichkeiten mit den Datenobjekten, die der Mensch - ohne Kenntnis ihrer maschinenspezifischen Codierung, ohne Zeitverzug und in beliebiger Reihenfolge, d. h. interaktiv, vornehmen kann.

Technologische Fortschritte, die in den letzten Jahren, besonders auf dem Gebiet der Halbleiterspeicher und der bipolaren Mikroprozessoren, erzielt worden sind, haben die technischen Voraussetzungen gebracht, neuartige, leistungsfähigere und kostengünstigere Display-Systeme zu realisieren.

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung eines neuen Konzeptes für Rastercandisplay-Systeme, die sich für die interaktive Bildkommunikation auf der Ebene informationeller Datenobjekte besonders gut eignen.

Da in der Literatur, trotz kürzlich gestarteter Bemühungen zwecks Standardisierung auf dem Gebiet der Computer-Graphik /ACM-79, NEUMANN-78 /, keine brauchbare systemtechnische Begriffswelt vorhanden ist, werden im ersten Kapitel zunächst einmal die Grundbegriffe der Display-Systeme einheitlich erläutert und anschließend, an Hand einer Bestandsaufnahme, die Entwicklungstendenzen auf diesem Gebiet angedeutet.

Das zweite Kapitel beschäftigt sich mit drei neuen Beiträgen zum Problembereich der Display-Systeme: Zuerst wird das Konzept des logischen Indexbildes mit seinen Konsequenzen für die informationelle Struktur des Display-Systems vorgestellt. Dann wird ein neues Strukturelement - die Flächengruppe - durch

eine mit den Vektorbildern vollkompatible Definition eingeführt. Schließlich wird die Kachelstruktur des Weltbildes erläutert.

Die Struktur des Rasterdisplay-Systems auf der Basis logischer Indexbilder wird unter zwei mit einander stark verkoppelten Gesichtspunkten ausführlich dargestellt: Im dritten Kapitel wird die Datenstruktur der darzustellenden Bilder untersucht. Ein hierarchisches Datenstrukturmodell, das gleichzeitig für die Codierung der Graphik- und Bildinformation verwendet werden kann, wird systematisch aufgebaut. Im vierten Kapitel werden die informationellen Strukturen, die zur Erzeugung, Abwicklung und Manipulation dieser Datenstruktur notwendig sind, detailliert behandelt. Als informationelle Struktur wird hier die präzise Beschreibung sowohl der Daten- als auch der Operationsflüßebene bezeichnet, wobei die Codierungsform und damit auch die physikalische Realisierung solcher Strukturen irrelevant sind. Die Struktur des entwickelten Display-Systems wird als Ergebnis eines quasisystematischen Entwurfsprozesses dargestellt. Besonderer Wert wird - wie auch überall in dieser Arbeit - auf die Transparenz der dargestellten Strukturen gelegt. Dies wird ermöglicht durch die konsequente Verwendung geeigneter Darstellungsmittel, die sowohl den Datenfluß als auch den Aktionsablauf anschaulich beschreiben können. Für die Darstellung der Datenflüßebene wird das neue Instanz/Zustandsraum-Konzept /WENDT-79/ verwendet, das mit wenigen und daher überschaubaren Grundbegriffen komplexe informationelle Strukturen modellieren kann. Bei der Darstellung der Aktionsflüßebene, bei der die Dynamik des Prozeßablaufes möglichst durch ein statisches Bild veranschaulicht werden soll, werden "klassische" Darstellungsmittel wie Ablaufdiagramme, Zustandsgraphen und Petri-Netze verwendet /PETRI-62/.

Schließlich werden im fünften Kapitel die Implementierungsaspekte dieser informationellen Strukturen, die in Form eines Multiprozessor-Systems realisiert wurden, praxisnah und unter Verwendung fortgeschrittener technologischer Mittel präsentiert.

Die praktische Grundlage für diese Arbeit bildet ein Projekt für die Auswertung medizinischer Bilder, das beim Deutschen Elektronen-Synchrotron DESY in Hamburg durchgeführt wurde und an dessen praktischem Einsatz die Nutzbarkeit der entworfenen Strukturen überprüft werden konnte.

1. Grundbegriffe

Das nachfolgende Kapitel beschäftigt sich mit der Festlegung der zu den Display-Systemen gehörenden Begriffe. Diese klärende Diskussion soll zunächst einmal die von der Realisierungsform unabhängige Wirkungsweise der in Display-Systemen ablaufenden Prozesse und der damit verbundenen informationellen Strukturen systematisch beschreiben. Anschließend wird, gestützt auf diese Ergebnisse, gezeigt, welche Aufgaben die Display-Systeme dem Menschen überhaupt abnehmen und zweckmäßig durchführen können.

1.1. Allgemeine Prozesse in Display-Systemen

Die Prozesse der Bildkommunikation finden i. a. zwischen dem Menschen und technischen Systemen statt (s. Bild 1.1). Die optischen Bilder entstehen durch speziell dafür gebaute physikalische Systeme. Solche Systeme - wie Videokameras, Röntgenanlagen u. a. - bilden unsere Umwelt unter Verwendung verschiedener physikalischer Gesetzmäßigkeiten in ein meistens kontinuierliches Bild ab. Dieses so entstandene optische Bild wird zuerst zwecks Verarbeitung in ein informationsverarbeitendes System durch den Bilderfassungsprozeß übertragen. Die erfaßten Bilder werden, bevor sie in einen digitalen Speicher abgelegt werden, zweckmäßigerweise so codiert, daß die logische Bildstruktur, möglichst ohne Informationsverlust und bei minimalem Speicherbedarf, erhalten bleibt. Sie können daher als "logische Bilder" betrachtet werden. Die informationsverarbeitende Maschine stellt eine programmierte Instanz dar, d. h., daß sie die logischen Bilder entsprechend einer vorgegebenen Aufschreibung, dem Programm, verändern kann. Die Rücktransformation vom logischen zum optischen Bild wird durch den Bilddarstellungsprozeß realisiert. Um reproduzierbare Bilder zu erzeugen, muß diese Transformation eindeutig und ohne Informationsverlust stattfinden. Dies bedeutet, daß einem beliebigen Codestück in dem logischen Bild eine eindeutige Erscheinung auf dem Bildschirm entspricht. Man kann die logischen Bilder im weitesten Sinne als "abstrakte Dias" betrachten, die einerseits von einer speziellen Fotokamera erfaßt und andererseits von einem speziellen Projektor dargestellt werden. An dieser Stelle soll zu diesen technischen Systemen ein nicht technisches System - der Mensch - "angeschlossen" werden. Die Fähigkeit des menschlichen Gesichtssinnes, sich an die Semantik der Bilder anzupassen sowie seine höchst parallele Struktur erklären die extrem hohe Leistungsfähigkeit und Schnelligkeit des Menschen, Bilder zu interpretieren /GROSSER-78/. In der der Interpretation entgegengesetzten Richtung, also bei der Codierung und Ausführung

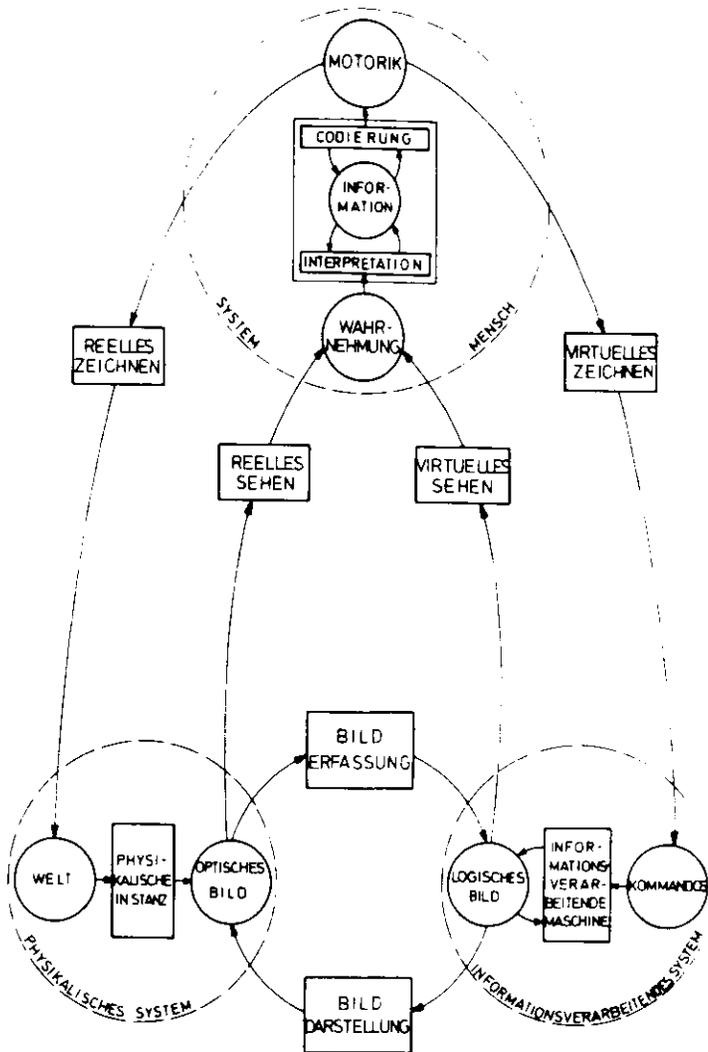


Bild 1.1 Allgemeine Prozesse in Display-Systemen

verschiedener Handlungskommandos durch die Motorik, zeichnet sich der Mensch zwar durch eine hohe Leistungsfähigkeit, jedoch, wegen des sequentiellen Ausführungsvorganges, durch eine geringe Schnelligkeit aus.

Die Frage nach dem zweckmäßigen Anschluß des Menschen an technische Systeme wie informationsverarbeitende Maschinen kann nach dieser Betrachtungsweise eindeutig beantwortet werden: In der zum Menschen hin gerichteten Kommunikationsrichtung ist es sicherlich zweckmäßig, die hervorragenden Eigenschaften des menschlichen reellen Sehens optimal auszunutzen. Die direkte Interpretation der maschinengerecht codierten abstrakten Dias, die einem "virtuellen Sehen" gleichkommt, kann wegen der im menschlichen Gehirn notwendigen aber umständlichen Umcodierung nicht in Betracht gezogen werden. In der entgegengesetzten Kommunikationsrichtung, vom Menschen her, kommt es besonders darauf an, die nach der Interpretation vom Menschen getroffenen Entscheidungen und ihre (Rück)wirkungen auf die optischen Bilder auf einfache und gleichzeitig schnelle Art entfalten zu können. Daher ist es zweckmäßig, daß die informationsverarbeitende Maschine, entsprechend den vom Menschen ausgestellten Kommandos, die logischen Bilddaten direkt und daher effektiv verändern kann. Das Erzeugen von Kommandos, die sich letzten Endes als Veränderung des optischen Bildes auswirken, kann im weitesten Sinne als virtuelles Zeichnen verstanden werden. Die wunschgemäße Veränderung des logischen Bildes, die im Dialog und möglichst verzögerungsfrei ausgeführt werden soll, ist eine spezifische Form der vorher erwähnten Bildverarbeitung und wird als interaktiver Bildmanipulationsprozeß bezeichnet.

Reelles Sehen und virtuelles Zeichnen bilden zusammen die geeignetste Interaktionsart zwischen Mensch und Maschine, die in Verbindung mit den Bilddarstellungs-, Bilderfassungs- und Bildmanipulationsprozessen die Grundlage einer effektiven Kommunikation durch Bilder ergeben.

1.2. Allgemeine Strukturen von Display-Systemen

Die allgemeine Struktur eines Display-Systems ist in Bild 1.2 dargestellt. Die gezeigte strukturelle Zerlegung in Bilddarstellungs-, Bilderfassungs- und Bildmanipulationsinstanz entsteht durch die Zuordnung jeweils einer dedizierten Instanz zu jedem in einem Display-System ablaufenden Prozeß. Durch die vorangegangene Diskussion sind auch die in Bild 1.2 enthaltenen Interaktionen zwischen den Instanzen und den Zustandsräumen klargeworden, so daß hier darauf nicht

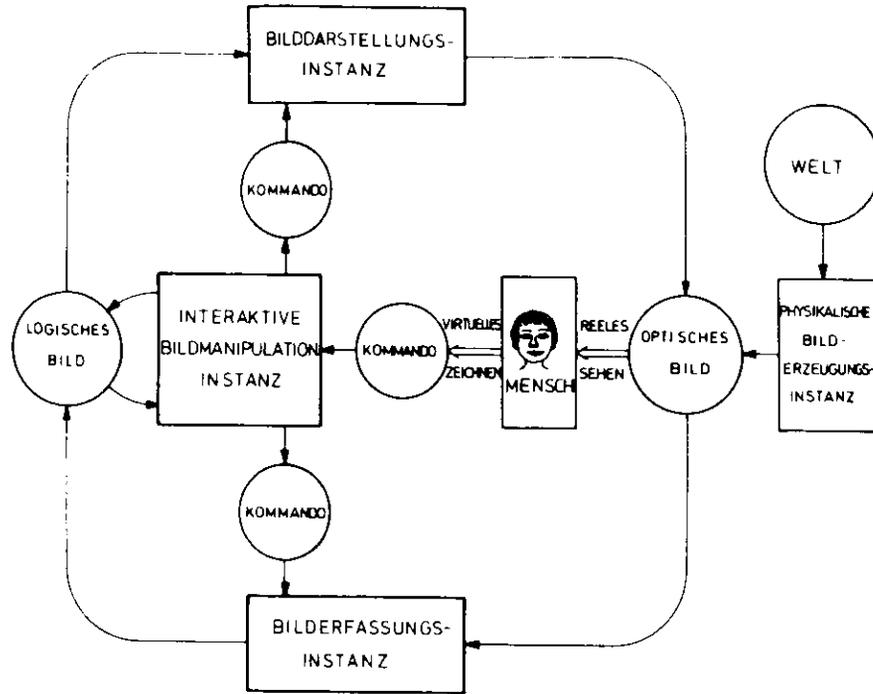


Bild 1.2 Allgemeine Struktur eines Display-Systems

näher eingegangen zu werden braucht. Wenn die gezeigte Aufteilung dem Leser im Hinblick auf die physikalische Realisierungsform willkürlich erscheinen sollte, muß gesagt werden, daß die in dieser Arbeit vorgestellten Strukturen überwiegend für die transparente Darstellung der Daten und Operationen gedacht und dadurch auf dieser Betrachtungsebene relativ einfach optimierbar sind. Die Frage nach der optimalen Realisierung solcher Strukturen, in die die wesentlichen Entscheidungen bezüglich der Codierungs- und Implementierungsform eingehen, ist eine Zweckmäßigkeitsfrage und kann deshalb nicht allgemein beantwortet werden. Unumstritten ist jedoch, daß die Festlegung einer (optimierten) informationellen Struktur zu einer besseren Implementierungslösung hinsichtlich Effektivität, Kostenaufwand und Dokumentation führt als eine direkte, modellfreie Realisierung.

Bevor die detaillierten Strukturen der Instanzen aus Bild 1.2 präsentiert werden, sollen zunächst die relevanten Eigenschaften der beiden wichtigen Zustandsräume in Display-Systemen - das logische und das optische Bild - erläutert werden.

Das logische Bild beinhaltet in codierter Form die logische Beschreibung der Bildinformation. Eine sinnvolle logische Beschreibung kann nur dann erreicht werden, wenn zuvor einerseits ein Bildmodell als Beschreibungsmodell und andererseits ein Codierungsmodell als Organisationsform der Bilddaten gewählt worden sind. Das Bildmodell enthält die Summe aller Informationen über die Bildobjekte und deren gegenseitige Beziehungen, die durch das logische Bild beschrieben werden können oder sollen, und bestimmt damit den maximalen Informationsgehalt des logischen Bildes. Das Codierungsmodell der Bilddaten schreibt die Codierungs- und Zugriffsregel für die Bilddaten vor und entscheidet damit über die Möglichkeiten zur Manipulation des logischen Bildes. Bildmodell und Codierungsmodell der Bilddaten ergeben zusammen die Datenstruktur, die durch zwei wesentliche schon angedeutete Merkmale - maximaler Informationsgehalt und leichte Manipulierbarkeit - gekennzeichnet ist. Ein Bildmodell, das sich für die Beschreibung der in der realen Welt auftretenden Bilder eignen würde, ist wegen der großen Zahl der möglichen Bildobjekte und deren schwer strukturierbarer gegenseitiger Relationen bis jetzt nicht bekannt und, wenigstens in absehbarer Zeit, nicht zu erwarten. Eine logische Bildbeschreibung ist daher nur durch die Vereinfachung der realen Weltbilder möglich. Dafür wird nur ein bestimmtes Repertoire, z. B. relative geometrische Position, in den Bildern zugelassen. Diese eingeschränkte aber beschreibbare symbolische Darstellung wird als Graphik verstanden. Im Gegensatz dazu soll unter dem Begriff Abbild die bloße zweidimensionale Verteilung von Chrominanz verstanden werden. Graphik und Abbild werden wegen ihrer unterschiedlich komplexen Bildmodelle auch mit verschiedenen Codierungsmodellen in vollkommen verschiedenen Datenstrukturen modelliert. Diese benötigen wiederum verschiedene Strukturen für die Display-Systeme. Da die Trennung zwischen Abbildung und Graphik nicht a priori vorausgesetzt werden muß, wird in dieser Arbeit der allgemeinere Begriff "Bild" verwendet, womit sowohl Abbild als auch Graphik verstanden werden kann.

Das optische Bild stellt die Erscheinungsform des logischen Bildes dar. Abgesehen von dem Informationsgehalt ist die an der Darstellungskraft gemessene Brauchbarkeit der optischen Bilder stets durch die sogenannte Bildqualität bestimmt. Die Bildqualität wird einerseits durch technische Maßstäbe, wie die

räumliche und zeitliche Auflösung, definiert und andererseits wird die Bildqualität subjektiv bei der Betrachtung mit dem menschlichen Auge beurteilt. Die bekannten psycho-physiologischen Eigenschaften des menschlichen Gesichtssinnes, wie Sehschärfe, Visionspersistenz, Helligkeitsunterscheidungsvermögen und Farbreizfunktion, legen die sinnvollen, wenngleich auch begrenzenden Anforderungen an die technischen Maßstäbe der Bildqualität fest /SCHÖNFELDER-72/. Die Korrelation zwischen den technischen und physiologischen Aspekten der Bildqualität ergibt, daß ein optisches Bild, das für die menschliche Wahrnehmung nicht mehr verbesserungsfähig ist, folgende technische Merkmale aufweist: räumliche Auflösung $\geq 4096 \times 4096$ Punkte, zeitliche Auflösung ≥ 50 Bilder/Sekunde, Helligkeitsauflösung ≥ 512 Graustufen, Farbauflösung ≥ 4096 Farbwerte. Maßgeblich für die Bildqualität ist das Auflösungsprodukt

$$\text{räumliche Auflösung} \times \text{zeitliche Auflösung} \times \text{Helligkeitsauflösung}$$

das gleichzeitig den Informationsquader des Bildübertragungskanal darstellt und dadurch dessen Bandbreite und Kapazität bestimmt. Die extrem hohen Bandbreiten, die für die Erzeugung von Bildern mit brauchbarer Bildqualität notwendig sind - z. B. 50 Mega-Punkte/Sekunde für ein 1024×1024 Bild bei 50 Bildwiederholungen/Sekunde - berühren heute die Grenzen des technologisch Möglichen.

1.2.1. Bilddarstellung

Die Bilddarstellungsinstanz soll aus logischen Bildern anschauliche optische Bilder erzeugen. Da einerseits die Darstellung optischer Bilder auf dem Bildschirm nur sequentiell erfolgen kann und andererseits die Datenstruktur der Bilder beliebig komplex gestaltet werden kann oder muß, ist es zweckmäßig, die logischen Bilder zuerst in die einfache Form einer Befehlsfolge umzuwandeln und danach die optischen Bilder zu generieren (s. Bild 1.3). Die Befehlsfolge enthält zwei Klassen von Befehlen: Bildelementbefehle, die Darstellungsanweisungen für mehr oder weniger komplexe Bildelemente bedeuten - z. B. ZEICHNE VEKTOR, ZEICHNE KREIS - und Bildattributbefehle, die Veränderungsanweisungen für den Bildstatus bedeuten - z. B. ROTIERE BILD MIT 90° , SKALIERE BILD MIT FAKTOR 2. Es ist nicht spezifiziert, ob im Kommunikationsraum "Befehlsfolge" in Bild 1.3 jeweils nur ein Befehl zur Ausführung übergeben wird oder ob hier die gesamte Befehlsliste eines Bildes gepuffert wird. Die Struktur der Datenstrukturabwicklungsinstanz hängt wesentlich von der Organisationsform der logischen Bilder ab

und kann daher ohne vorherige Festlegung der Datenstruktur allgemeingültig nicht näher beschrieben werden.

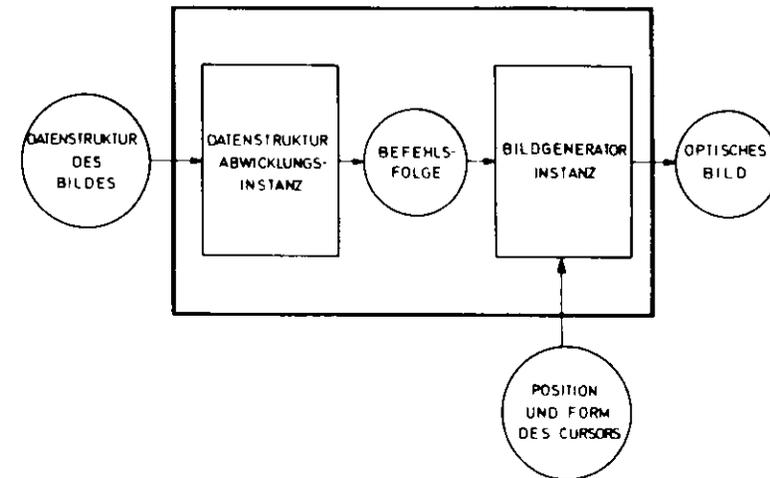


Bild 1.3 Allgemeine Struktur der Bilddarstellungsinstanz

Die Bildgeneratorinstanz dagegen kann durchaus allgemein gültig näher strukturiert werden. Der Einfachheit halber sei angenommen, daß der Bildgeneratorinstanz einzelne Befehle übergeben werden. Bei der Bildgenerierung müssen drei miteinander gekoppelte Prozesse unterstützt werden: Bildtransformation, Bildpunktgenerierung und Bildpunktdarstellung. Während die Ausführungsreihenfolge der Bildtransformation und der Bildpunktgenerierung noch wählbar ist, muß die Bildpunktdarstellung selbstverständlich am Ende der Bildgenerierungskette liegen. Eine in der Literatur übliche Struktur des Bildgenerators, bei der die Bildtransformation nach der Bildpunktgenerierung für jeden einzelnen Bildpunkt durchgeführt wird, ist als Datenflußgraph und als Aktionsablauf in Bild 1.4 bzw. Bild 1.5 dargestellt. Die zweite Möglichkeit, bei der die Bildtransformation vor der Bildpunktgenerierung für ganze Bildelemente stattfindet, wird im Kapitel 4 eingehend beschrieben (s. Bild 4.1).

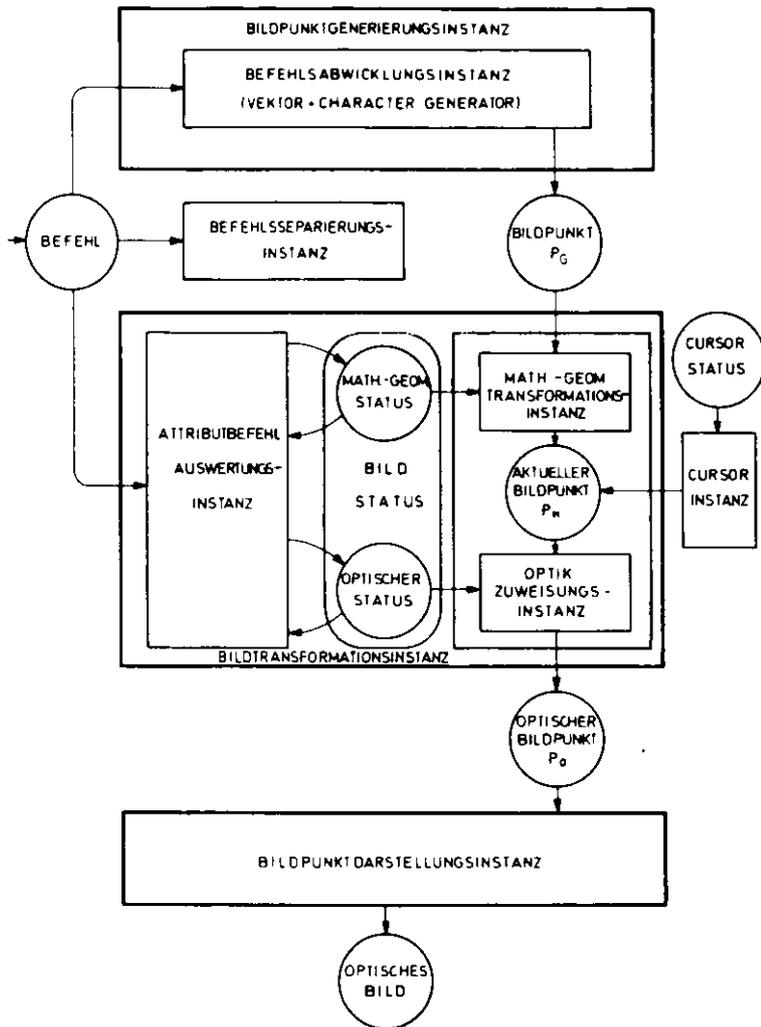


Bild 1.4 Datenflußgraph der Bildgeneratorinstanz

Die Bildpunktgenerierung findet durch die Abwicklung der Bildelementbefehle statt, d. h. die Bildelementbefehle werden unter Berücksichtigung ihrer Definition bezüglich eines Generierungskordinatensystems in entsprechende Punktmengen

transformiert. Die Bildelementabwicklungsinstanzen werden i. a. als Generatoren bezeichnet - z. B. Vektor-, Character- oder Kreisgenerator. Es muß bemerkt werden, daß die Bildpunktgenerierung naturgemäß eine beträchtliche Vergrößerung der von den nachfolgenden Instanzen zu verarbeitenden Datenmenge verursacht, wobei die Durchsatzanforderungen für die gesamte Bildgeneratorinstanz vom Auflösungsprodukt des Bildschirms abzuleiten sind.

Die Bildtransformationsinstanz erfüllt eine doppelte Aufgabe: einerseits werden die Bildattributbefehle für die Veränderung des (globalen) Bildstatus ausgewertet und andererseits werden die Bildpunkte, die von den Generatoren erzeugt wurden, entsprechend dem geltenden Bildstatus transformiert.

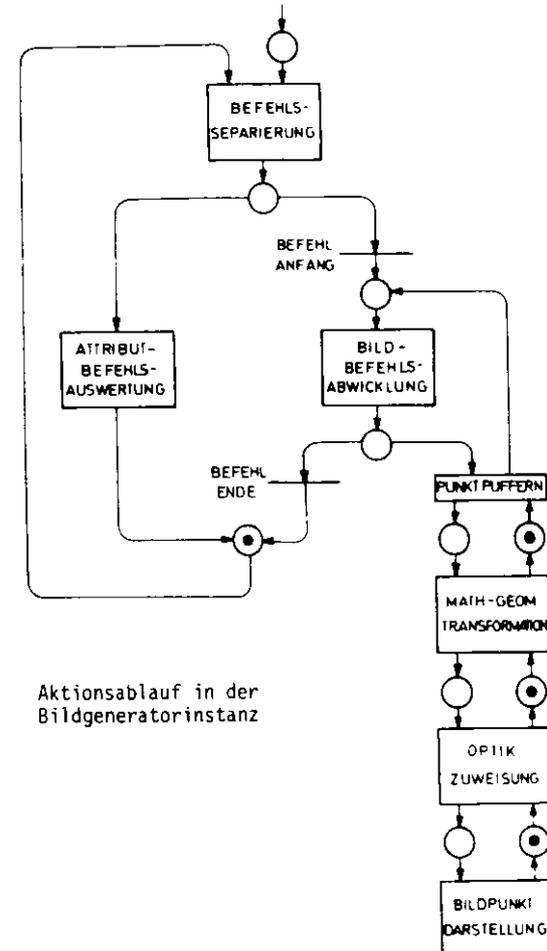


Bild 1.5 Aktionsablauf in der Bildgeneratorinstanz

Der Bildstatus beschreibt die gesamte Erscheinungsform des Bildes und setzt sich aus zwei Komponenten zusammen:

- Der geometrische Aspekt des Bildes wird durch den mathematisch-geometrischen Status beschrieben. Dieser bestimmt die relative Lage des Koordinatensystems der Bildpunktgenerierung (KS_g) in dem normalliegenden Bildschirmkoordinatensystem (KS_B). Der mathematisch-geometrische Status kann formell als ein Vektor aufgefaßt werden:

$$\underline{STATUS}_M := [POS (X_{OS}, Y_{OS}), ROT (\theta_S), OA (\phi_S), SF (S_{XS}, S_{YS}), W_S (P_1, P_2)] \quad (1.1)$$

mit $POS (X_{OS}, Y_{OS})$: Position des KS_g im KS_B ,
 $ROT (\theta_S)$: Rotationswinkel θ der x-Achse im KS_g gegenüber KS_B ,
 $OA (\phi_S)$: Orthogonalitätsabweichung des KS_g ,
 $SF (S_{XS}, S_{YS})$: achsenspezifische Skalierungsfaktoren des KS_g gegenüber KS_B ,
 $W_S (P_1, P_2)$: darzustellender Bildausschnitt, wobei P_1 und P_2 die den Ausschnitt bestimmenden Diagonalepunkte im KS_g sind.

Dieser Vektor definiert die durchzuführenden mathematisch-geometrischen Bildpunkttransformationen eindeutig.

- Der optische Aspekt des Bildes wird durch den optischen Status beschrieben, der mit dem Chrominanzvektor identisch ist:

$$\underline{STATUS}_O := C (R, G, B), \quad (1.2)$$

wobei $C (R, G, B)$: die Chrominanz-Komponenten für die Primitivfarben Rot (R), Grün (G) und Blau (B) sind.

Die Veränderungen des gesamten Bildstatus (STATUS) durch einen Attributbefehl (ATTRBEF) werden nur relativ zum geltenden Bildstatus ausgeführt. Ein solcher Transformationsschritt kann mit der Gleichung

$$\underline{STATUS}^{n+1} = (\underline{STATUS}_M, \underline{STATUS}_O)^{n+1} = f (\text{ATTRBEF}, \underline{STATUS})^n \quad (1.3)$$

beschrieben werden, wobei die Veränderungsvorschriften f von der gewählten Definitionsart der Bildattributbefehle und des Bildstatus abhängen.

Wie aus dem Bild 1.4 ersichtlich, findet die Bildtransformation in zwei Phasen statt: zuerst werden die generierten Bildpunkte (P_g) mathematisch-geometrisch in aktuelle Bildpunkte (P_M) transformiert. Danach werden aus den aktuellen Bildpunkten durch Zuweisung optischer Attribute darstellungsfähige optische Bildpunkte (P_o) erzeugt.

Die Transferfunktion der mathematisch-geometrischen Transformationsinstanz ist durch die Gleichung 1.4 angegeben:

$$P_M = g (P_g, \underline{STATUS}_M) \quad (1.4)$$

Die Funktion g in der Gleichung 1.4 muß in der Funktion f in der Gleichung 1.3 enthalten sein, weil die Lage des Ursprungs des KS_g , der einen besonderen Bildpunkt im KS_B darstellt, Teil des mathematisch-geometrischen Status ist.

Es gibt zwei Arten von mathematisch-geometrischen Transformationen/NEWMAN - 79/:

1. Die Euklidischen Transformationen beinhalten die Regeln für die Durchführung der Bildpunkttransformationen als Sequenz von elementaren Transformationen, d.h. Translation (T), Rotation (R) und Skalierung (S). Für die elementaren Transformationen eines Bildpunktes (X_1, Y_1) in einem neuen Bildpunkt (X_2, Y_2) gelten im Falle der 2D-Bilder folgende Gleichungen:

Translation (T)

$$(X_2, Y_2) = \begin{bmatrix} X_1 \\ Y_1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ X_{op} & Y_{op} & 0 \end{bmatrix} \quad (1.5)$$

Rotation (R)

$$(X_2, Y_2) = \begin{bmatrix} X_1 \\ Y_1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta_p & \sin \theta_p & 0 \\ -\sin \theta_p & \cos \theta_p & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.6)$$

Skalierung (S)

$$(X_2, Y_2) = \begin{bmatrix} X_1 \\ Y_1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} S_{xp} & 0 & 0 \\ 0 & S_{yp} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.7)$$

Die Matrizenkoeffizienten können direkt als Transformationsparameter aufgefaßt werden, die den Translationsbetrag (X_{op}, Y_{op}) , den Rotationswinkel θ_p und die achsenspezifischen Skalierungsfaktoren (S_{xp}, S_{yp}) angeben. Da die Transformationsreihenfolge nicht kommutativ ist, was durch die Matrizenmultiplikation leicht nachgeprüft werden kann, hängt die erzielte Wirkung einer Bildpunkttransformation wesentlich von der Ausführungsreihenfolge der elementaren Transformationen R, T und S ab. Weit verbreitet ist die Konkatenation RoSoT, die wegen ihrer minimalen Anzahl von sechs Multiplikationen einfach zu realisieren ist /ENCARNACAO-75/. Bild 1.6 zeigt die Wirkung der Konkatenation RoSoT bei der Transformation eines einzelnen Bildpunktes P, wobei die Bedeutung der Matrizenkoeffizienten aus den Gleichungen 1.5 bis 1.7 ersichtlich ist.

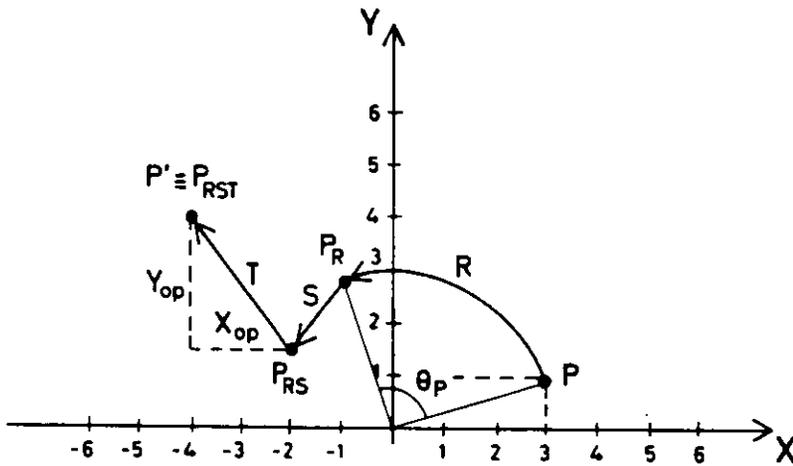


Bild 1.6 Zur Veranschaulichung der Bildpunkttransformation

Für den Anwender, der bei der Bilddefinition umfangreiche geometrische Manipulationen mit Bildelementen oder Teilbildern vornehmen muß, spielt die transparente Gestaltung der Euklidischen Transformationen eine wesentliche Rolle. Dies bedeutet, daß die Angabe der Transformationsparameter für die beabsichtigte Transformation vom Anwender einfach abzuleiten sein muß. Die Transparenz der Euklidischen Transformationen hängt von der Komplexität des Zusammenhangs zwischen den Statusparametern, die die relative Lage zweier Figuren und ihrer Koordinatensysteme anschaulich beschreiben, und den Matrizenkoeffizienten, die

die Transformation einzelner Bildpunkte unmittelbar beschreiben, ab. Da der Anwender hauptsächlich an Manipulationen mit ganzen Figuren interessiert ist, die in eine bestimmte relative Lage zueinander gebracht werden müssen, ist es bei der Gestaltung der Euklidischen Transformation sinnvoll, von gezielten Veränderungen des mathematisch-geometrischen Status auszugehen. Bei gegebenen Matrizenkoeffizienten $[(X_{op}, Y_{op}), \theta_p, (S_{xp}, S_{yp})]$ als Transformationsparameter ergibt sich eine relative Veränderung des mathematisch-geometrischen Status, die durch folgende Gleichungen beschrieben ist:

$$\Delta X_{os} = X_{op} \quad ; \quad \Delta Y_{os} = Y_{op} \quad (1.8)$$

$$\Delta \theta_s = \arctg \left[\frac{S_{yp}}{S_{xp}} \operatorname{tg} \theta_p \right] \quad ; \quad \Delta \theta_s = -\pi/2 - 2 \cdot \Delta \theta_s \quad (1.9)$$

$$\begin{aligned} \Delta S_{xs} &= [S_{xp}^2 \cdot \cos^2 \theta_p + S_{yp}^2 \cdot \sin^2 \theta_p]^{1/2} \\ \Delta S_{ys} &= [S_{xp}^2 \cdot \sin^2 \theta_p + S_{yp}^2 \cdot \cos^2 \theta_p]^{1/2} \end{aligned} \quad (1.10)$$

Die Gleichungen 1.8 bis 1.10 zeigen deutlich, daß zwischen der Statusveränderung und den Matrizenkoeffizienten ein besonders komplexer Zusammenhang besteht, der für den Anwender nicht transparent ist. Die Verwendung der Matrizenkoeffizienten mit ihrer Bedeutung von der Transformation einzelner Bildpunkte würde somit zu unbeabsichtigten Verzerrungen des zu transformierenden Bildes führen, die in vielen Anwendungen nicht hingenommen werden können.

Die Euklidischen Transformationen, die mathematisch eine 3x3-Matrizenmultiplikation bedeuten, stellen eine besonders rechenintensive Aufgabe dar. Überall dort, wo aus vorher erwähnten Durchsatzgründen eine Softwarelösung nicht verwendbar ist, müssen spezielle Hardwarelösungen eingesetzt werden. Solche Hardwarelösungen verwenden für die Matrizenmultiplikation entweder reine Analogtechniken /GILDI-70/ oder spezielle Digitaltechniken /GHEST-71/ - beispielsweise "Binary Rate Multipliers", die für die Transformation eines Punktes zwischen 5 und 15 µsek benötigen. Eine Hybridlösung /ENCARNACAO-72/, die die gute Bandbreite der Analogtechnik mit der hohen Genauigkeit der Digitaltechnik vorteilhaft vereinigt, erreicht eine Transformationsdauer von 10 µsek/Punkt. In der Literatur sind die gerätetechnischen Realisierungsaspekte der Euklidischen Transformationen erschöpfend behandelt worden. Daher werden sie in dieser Arbeit nicht weiter erläutert.

2. Die Domäne-Transformationen beinhalten die Regel der Abbildung eines gewünschten Ausschnittes des logischen Bildes auf den Bildschirm. Es ist zu betonen, daß die Domäne-Transformationen nur deshalb notwendig sind, weil die technisch bedingte räumliche Auflösung des Bildschirms in der Regel erheblich kleiner ist als die des gedachten logischen Bildes. Somit ist ohnehin nur ein Ausschnitt des logischen Bildes auf dem Bildschirm ohne Informationsverlust darstellbar. Infolgedessen müssen die Koordinatensysteme für das logische und das optische Bild hinsichtlich ihrer Koordinatenwertebereiche unterscheiden: Das logische Bild verwendet das Weltkoordinatensystem

$$KS_W := X_W \times Y_W = (0 : 2^{14} - 1) \times (0 : 2^{14} - 1) \quad (1.11)$$

und das optische Bild das Bildkoordinatensystem

$$KS_B := X_B \times Y_B = (0 : 2^{10} - 1) \times (0 : 2^{10} - 1). \quad (1.12)$$

Eine Domäne-Transformation setzt sich nach /NEWMAN-79/ zusammen aus der Window-Transformation:

$$W : KS_W \longrightarrow F, \quad (1.13)$$

wodurch zuerst ein beliebiges Fenster F aus dem Weltkoordinatensystem KS_W ausgeschnitten wird, und aus der Viewport-Transformation:

$$V : F \xrightarrow{R \circ S \circ T} KS_B \quad (1.14)$$

wodurch dann das gewählte Fenster - eventuell Euklidisch transformiert - auf einen Ausschnitt des Bildkoordinatensystems KS_B abgebildet wird. Im allgemeinen spricht man bei der Abbildung des Weltbildes auf dem vollen Bildschirm von Window-Transformation und auf einem Teil des Bildschirms von Viewport-Transformation. Die Zulassung Euklidischer Transformationen bei der Viewport-Transformation bedeutet, daß die ursprünglich nur zur Abbildung gedachten Domäne-Transformationen auch einen in diesem Zusammenhang unerwünschten Verarbeitungsvorgang der Bildinformation darstellen. Beispielsweise führt die Skalierungstransformation zu Veränderungen der Bildinformation, die bei den Domäne-Transformationen vom Anwender eigentlich nicht beabsichtigt wurden: bei Skalierungsfaktoren $SF > 1$ muß durch die SF-fache Darstellung jedes einzelnen Bildpunktes redundante Bildinformation zugeführt werden und bei Skalierungsfaktoren $SF < 1$

geht durch die Darstellung jedes SF-ten Bildpunktes (wichtige) Bildinformation verloren.

Die Algorithmen für die Domäne-Transformationen müssen die Ausschnittzugehörigkeit jedes Bildpunktes überprüfen und die außerhalb befindlichen Bildpunkte austasten. Solche Austastalgorithmen (scissoring) setzen voraus, daß zuerst alle Bildpunkte generiert werden, und sind deshalb so zeitaufwendig, daß eine flimmerfreie Darstellung in der Regel nicht mehr möglich ist. Die Abschneidealgorithmen (clipping) versuchen, diesen Nachteil zu umgehen, indem nicht mehr die einzelnen Bildpunkte, sondern ganze Bildelemente auf ihre Ausschnittzugehörigkeit überprüft werden. Da die Bildelemente auch nur teilweise in dem Bildschirmfenster liegen können und die Ermittlung solcher Teile von der Topologie des Bildes abhängt, entwickeln sich die Abschneidealgorithmen zu sehr komplexen und oft auch fehlerbehafteten Verfahren /SPROUL-68/. Mit dieser unbefriedigenden Situation, die mit der Codierung des Weltbildes zusammenhängt, werden sich die Abschnitte 2.3 und 3.2.1.4 befassen.

Die bisher noch nicht erwähnte Cursor-Instanz (s. Bild 1.4) hat die Aufgabe, auf dem Bildschirm eine leicht erkennbare Marke, den sogenannten "Cursor", zu erzeugen, womit Bildpunkte, Bildelemente oder gar Teilbilder zwecks Identifizierung durch den Anwender markiert werden können. Die jeweilige Position, Form und Optik des Cursors werden zweckmäßigerweise durch den Cursor-Status definiert.

Die Optik-Zuweisungsinstanz ordnet den aktuellen Bildpunkten die in dem optischen Status gültigen optischen Attribute zu. Zweckmäßigerweise geschieht dies durch die Zuweisung eines Codewortes, das in codierter Form die gewünschten optischen Attribute beinhaltet. Die so resultierenden optischen Bildpunkte können unmittelbar dargestellt werden.

Schließlich hat die Bildpunktdarstellungsinstanz die scheinbar einfache Aufgabe, die optischen Bildpunkte auf dem Bildschirm darzustellen. Da der Darstellungsprozeß wesentlich vom verwendeten Bildschirm abhängt, sind folgende Fälle zu unterscheiden:

Bildschirme mit wählbarer Abtastung (Randomscan): Bei diesen können die Bildschirmpunkte in der Reihenfolge ihrer Generierung adressiert und dargestellt werden. Die Bildpunktdarstellungsinstanz muß lediglich die Ablenk- und Strahl-

intensitätssteuerungssignale erzeugen, die sich i. a. durch eine einfache Digital-Analog-Konvertierung realisieren lassen. Die Randomscan-Bildschirme können entweder Refresh-Bildschirme, z. B. CRT-Schirm, oder Speicherschirme, z. B. Speicher- oder Plasmaschirm, sein. Im Falle des Refresh-Schirms stellt die gesamte Bildgeneratorinstanz einen Prozessor dar, der die lineare Befehlsfolge (als Programm) mit der gewünschten Bildwiederholrate selbständig abwickelt. Dagegen kann im Falle des Speicherschirms die Bildgeneratorinstanz als eine befehlsausführende Senke realisiert werden, der einzelne Befehle zur Ausführung übergeben werden.

Bildschirme mit zeilenweiser Abtastung (Rasterscan): Bei diesen werden alle Bildschirmpunkte, unabhängig von ihrer Generierungsreihenfolge und ihren optischen Attributen, zeilenweise dargestellt. Die Rasterscan-Bildschirme, z. B. Fernsehmonitore, benötigen die optischen Bildpunkte in einer ausschließlich nach der Abtastung gerichteten Reihenfolge und können somit die optischen Bildpunkte nicht mehr in der Reihenfolge ihrer Generierung darstellen. Bild 1.7 zeigt, daß in diesem Falle die optischen Bildpunkte zuerst in der Generierungsreihenfolge durch die Zugriffsinstanz in einen zusätzlichen Bildspeicher als optische Codewörter abgelegt werden. Danach werden sie in der Abtastreihenfolge

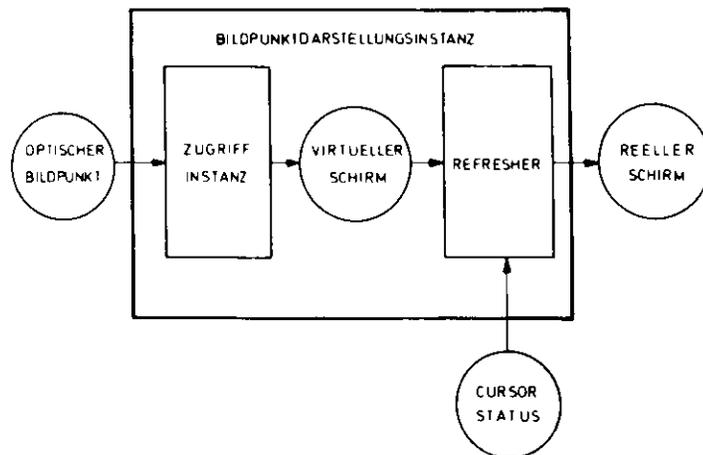


Bild 1.7 Struktur der Bildpunktdarstellungsinstanz für Rasterscan-Bildschirme

durch die Refreshinstanz auf dem reellen Schirm dargestellt. Obwohl es für die Speicherung des optischen Bildes in dem Bildspeicher grundsätzlich mehrere Möglichkeiten gibt /LINDNER-79/, wird hier - wegen seiner praktischen Bedeutung - nur das Konzept betrachtet, wo der Adreßraum des Bildspeichers unmittelbar die Bildschirmgeometrie abbildet. Dieser so organisierte Bildspeicher wird daher im folgenden als "virtueller Schirm" bezeichnet. Die beträchtliche Speicherkapazität des virtuellen Schirmes - bei einer Bildqualität von 1024 x 1024 Punkten und 256 Graustufen ist 1 MByte notwendig - sowie die in der Refreshinstanz notwendigen hohen Digital-Analog-Konvertierungsraten - typisch 10 Megapunkte/Sekunde - haben bis vor kurzem aus technologischen Gründen die Verwendung der Rasterscan-Bildschirme unrealistisch erscheinen lassen. Erst durch Fortschritte auf dem Gebiet der Halbleiterspeicher sowie der Digital-Analog-Konvertierungsbausteine wurde überhaupt die Verwendung der Rasterscan-Bildschirme als Alternativlösung zu den Randomscan-Bildschirmen ermöglicht.

Der in Bild 1.5 gezeigte Aktionsablauf der Bildgeneratorinstanz soll in Verbindung mit dem Datenflußbild als ein möglicher - jedoch nicht der einzige - Aktionsablauf angesehen werden. Die mehrfach erwähnte Durchsatzforderungen sind in Bild 1.5 durch eine weitgehende Parallelisierung des Ablaufes berücksichtigt worden. Bis auf die Attributsbefehlsauswertungsinstanz können alle anderen Instanzen in einer Pipeline-Struktur parallel arbeiten. Die Kommunikation zwischen den Instanzen findet durch bidirektionale Markenübergabe statt, was operationell den Einsatz von Pipeline-Registern als Kommunikationspuffer erfordert.

1.2.2. Bilderfassung

Die Bilderfassungsinstanz hat die Aufgabe, aus optischen Bildern Abbilder in Form logischer Bilder zu generieren. Die allgemeine Struktur der Bilderfassungsinstanz ist im Bild 1.8 dargestellt. Es zeigt, daß ein logisches Bild i. a. durch die Abtastung, Digitalisierung und anschließende Codierung des Abbildes entsteht.

Die Bildabtastung identifiziert die zu erfassenden Bildpunkte möglichst in einer Reihenfolge, die eine spätere Codierung im Rahmen des gewählten Bildmodells erleichtert. So ist es z. B. sinnvoll, ein graphisches Vektorbild durch Entlangfahren der Vektoren abzutasten und dadurch die logische Codierung der Vektoren einfacher zu gestalten. In manchen Fällen muß aus physikalischen Gründen die Abtastung unabhängig von der Bildstruktur nur in einer vorgeschriebenen

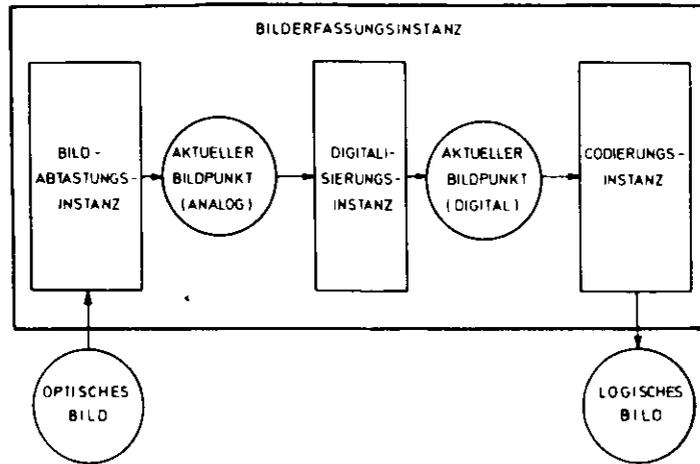


Bild 1.8 Allgemeine Struktur der Bilderfassungsinstanz

Weise stattfinden. Für Videobilder z. B., die wegen ihres weiten Verbreitungsgrades in dieser Arbeit besonders aufmerksam betrachtet werden, ist nur eine zeilenorientierte Abtastung möglich, unabhängig davon, ob dies ein reelles Weltbild oder ein graphisches Bild darstellt.

Die nachfolgende Digitalisierung der abgetasteten Bildpunkte muß in einer Weise geschehen, daß die Bildqualität des optischen Bildes erhalten wird. Für Standard-Videobilder ergeben sich z. B., entsprechend dem Shannon-Abtasttheorem, Digitalisierungsraten von 10 Megapunkten/Sekunde.

Die Codierung der digitalisierten Bildpunkte in eine für die Datenstruktur der logischen Bilder geeignete Form stellt, wie in Abschnitt 1.1 schon angedeutet wurde, die schwierigste, wenngleich eine der interessantesten Aufgaben bei der Bilderfassung dar. Die Komplexität der notwendigen Bildmodelle einerseits und die z. Zt. scheinbar unüberwindbaren Schwierigkeiten bei der durchzuführen-

den Bildanalyse andererseits machen die Codierung allgemeiner, reeller Weltbilder in logische Bilder nicht möglich. Dieser Codierungsvorgang, der eine logische Datenreduktion bedeuten würde, wird in der Praxis vielmehr durch eine physikalische Datenreduktion ersetzt. Damit wird die Bilddatenmenge, ohne die Semantik der Bilder zu berücksichtigen, mit Hilfe von Mustererkennung und anderen speziellen Verfahren nur physikalisch reduziert /ROSENFELD-77, MUSMANN-79/. Die so codierten Bilddaten können eher die physikalischen als die logischen Bildeigenschaften beschreiben, stellen jedoch z. Zt. die einzige Möglichkeit dar, reelle Weltbilder in einer für die Darstellung zweckmäßigen Form zu codieren. Aus Abbildern von graphischen Bildern lassen sich durch geeignete Abtastung leichter logische Bilder ableiten.

1.2.3. Bildmanipulation

Die Bildmanipulation kann nur in Verbindung mit ihrem allgemeinen Verwendungsrahmen präsentiert und verstanden werden. Der Anwender informationsverarbeitender Rechenmaschinen betrachtet die graphische und bildliche Darstellung seiner Datenobjekte nur als Mittel zur symbolischen Präsentation seiner spezifischen Begriffswelt. Da für ihn seine Begriffe eine primäre und ihre Symbole nur eine sekundäre Bedeutung haben oder haben müssen, bezweckt er bei seiner Arbeit hauptsächlich eine Begriffsmanipulation. Die symbolische Darstellung seiner Begriffe ist aber nur dann sinnvoll und hinsichtlich der anschließenden Bildinterpretation effektiv, wenn jede Begriffsmanipulation zweckmäßigerweise durch Bildmanipulation unmittelbar in die Darstellungsform abgebildet werden kann. Die vom Anwender ausgestellten Kommandos können i. a. drei Manipulationsaktionen hervorrufen:

- eine Begriffsmanipulation:
z. B. die Veränderung der Ausgangsfunktion eines Automaten, ohne den dazugehörigen Zustandsgraphen verändern zu müssen.
- eine Bildmanipulation:
z. B. die Zuordnung eines neuen Symbols für ein Gatter, ohne den vorhandenen Begriff "Gatter" verändern zu müssen.
- eine Folge von Begriff- und Bildmanipulationen:
z. B. die Veränderung der Transferfunktion eines Automaten, die auch die Veränderung des zugeordneten Zustandsgraphen erforderlich macht.

Da der Strukturierung der Bildmanipulationsinstanz z. Zt. keine allgemeingültigen Prinzipien zugrundeliegen und dieses Problem auch nicht Gegenstand dieser vorliegenden Arbeit ist, soll dieser Problembereich hier nicht weiter betrachtet werden.

1.3. Problematik im Bereich der Display-Systeme

Die vorangegangene Einführung der Grundbegriffe hat die grobe strukturelle Zerlegung sowie die funktionalen Eigenschaften verschiedener Komponenten von Display-Systemen, ohne Rücksicht auf die vorhandenen Realisierungsmöglichkeiten, deutlich gemacht. Dadurch ist überhaupt klar geworden, welche Aufgaben von Display-Systemen grundsätzlich übernommen werden könnten.

In diesem Abschnitt wird zuerst die heutige Situation auf dem Gebiet der Display-Systeme in Form einer Bestandsaufnahme präsentiert, und anschließend werden Entwicklungstendenzen aufgezeichnet. Dadurch wird auch geklärt, welche realistischen Anforderungen an die Display-Systeme gestellt und für welche Aufgabenbereiche sie derzeit eingesetzt werden können.

1.3.1. Bestandsaufnahme

Die Untersuchung einer für Display-Systeme repräsentativen Gruppe von Systemen /ACM-79/ zeigt, daß die Hauptaufmerksamkeit der Konstrukteure bis jetzt hauptsächlich der Bilddarstellungsinstanz galt. Besonders bei der Bildgenerierung ist, durch Verwendung schneller Vektor-, Character- und Kurvengeneratoren sowie durch mathematisch-geometrische Echtzeittransformationen eine beachtliche Leistungsfähigkeit erreicht worden. Die meistens auf den Randomscan-Bildschirmen dargestellten Bilder haben in der Regel ein sehr gutes Auflösungsprodukt: 1024 x 1024 Bildpunkte und 30 Bilder/Sekunde sind typische Werte. Der relativ geringe Informationsgehalt der logischen Bilder liegt darin, daß die Bilddarstellungsinstanzen relativ primitive Datenstrukturen für die logischen Bilder verwenden. So sind z. B. die Befehlslisten /ABRAMS-71/ heute noch stellvertretend für die Datenstruktur am weitesten verbreitet. Pseudohierarchische Datenstrukturmodelle mit einer begrenzten Anzahl von Hierarchieebenen und Zuweisungsmöglichkeiten für absolute geometrische und optische Bildattribute /GILOI-74/ sind vereinzelt anzutreffen. Komplexere hierarchische Datenstrukturmodelle, die eine unbegrenzte Anzahl von Hierarchieebenen und die Zuweisung relativer geometrischer und optischer Attribute ermöglichen, sind nur in dem wissenschaftli-

chen Bereich vorhanden /KELLER-74, GONAUSER-78, SHAPIRO-79/. Es muß auch gesagt werden, daß der Schwerpunkt der Entwicklung im Bereich der Datenstrukturen eher auf die Organisationsform der Daten als auf das ausschlaggebende Bildmodell gelegt wurde. Es besteht auch eine scharfe Trennung zwischen der Darstellung der Graphik- und Abbildinformation. Die heutigen Display-Systeme vermögen diese beiden Informationsarten nicht gleichwertig zu behandeln.

Die Bilderfassung als wesentlicher abbildgenerierender Prozeß ist nur selten, und dann in Form von Koordinatendigitalisierern für einzelne Bildpunkte, unterstützt.

Schließlich findet die Bildmanipulation in der Regel in Form eines interaktiven Dialogs mit flexiblen Gestaltungsformen statt. Die Trennung der Bildmanipulationsinstanz in einen systemspezifischen und einen anwendungsspezifischen Teil ist bisher (noch) nicht erreicht worden.

1.3.2. Entwicklungstendenzen

Der allgemeine Wunsch nach einer verbesserten optischen Darstellungsform bestimmt eindeutig die aktuellen Entwicklungstendenzen auf dem Gebiet der Display-Systeme. Dies bedeutet, daß einerseits Bildschirme mit umfassenderen Darstellungsmöglichkeiten und andererseits Bildmodelle mit erweiterten Beschreibungsmöglichkeiten verwendet werden müssen. Von allen verfügbaren Bildschirmarten bietet allein der Farbvideomonitor ausreichende Darstellungsmöglichkeiten für Grau- und Farbtöne bei gleichzeitigen niedrigen Herstellungskosten und breitem Verbreitungsgrad. Durch laufende technologische Verbesserungen werden die z. Zt. noch relativ hohen Herstellungskosten der hochauflösenden Farbvideomonitore (1024 x 1024 Bildpunkte) gesenkt werden können. Die Hauptproblematik bei der Verwendung solcher Rasterdisplay-Systeme stellt (s. Kapitel 1.2.1) die notwendige Abtastkonversion dar /NEWMANN-76/. Die Rasterdisplay-Systeme haben den prinzipiellen Nachteil, daß infolge der Speicherung der optischen Eigenschaften pro Bildpunkt im virtuellen Schirm begrifflich zusammengehörige Punktmengen nicht mehr auf einfache Weise als solche identifizierbar sind. Eine reichhaltigere optische Darstellungsform kann also nur durch den kostenintensiven Videobildspeicher und durch die Komplexität der Refresh- und Bildmanipulationsinstanz erkauft werden /BAECKER-79/. In dieser Arbeit werden neue Konzepte beschrieben, die zur Überwindung dieser prinzipiellen Nachteile der Rasterdisplay-Systeme dienen.

2. Neue Konzepte

Drei neue Konzepte, die im Zusammenhang mit Rasterdisplay-Systemen eingesetzt werden können, sind Gegenstand dieses Kapitels: das des logischen Indexbildes als Repräsentationsform der Bildinformation, das der Flächengruppe als neues Strukturelement des logischen Bildes und das der Kachelstruktur des Weltbildes werden eingeführt.

2.1. Das Konzept des logischen Indexbildes

Es besteht die Aufgabe, eine für die Bildkommunikation geeignete Repräsentationsform der Bildinformation im virtuellen Schirm eines Raster-Scan-Display-Systems zu finden. Bei der Suche nach einer solchen Repräsentationsform muß beachtet werden, daß die Effizienz - und damit auch die Brauchbarkeit - der Bildkommunikation zwischen Mensch und Maschine einerseits von der optischen Erscheinungsform und andererseits von der Identifizierbarkeit der einzelnen Begriffe abhängt. Als Begriffe wird hier die Menge der einzelnen Objekte, die Gegenstand der Kommunikation sind oder sein können, definiert. Die Begriffe sind auf einer gegebenen Kommunikationsebene grundsätzlich nicht mehr teilbar, d. h. sie werden betrachtet und manipuliert, ohne ihre weitere Begriffsauflösung zu berücksichtigen.

Für eine anschauliche optische Erscheinungsform der Begriffswelt müssen den einzelnen Begriffen umfangreiche optische Attribute zugeordnet werden, die die Semantik und gegebenenfalls die weiter auflösbare Struktur des Begriffes darstellen. Die so dargestellten Begriffe müssen aufgrund ihrer Optik sowohl vom Menschen als auch von der Maschine eindeutig identifiziert werden können. Diese beiden Forderungen bezüglich der Erscheinungsform und Identifizierbarkeit der Begriffe bedeuten, daß theoretisch der virtuelle Schirm Information sowohl über die Optik als auch über die Begriffszugehörigkeit des einzelnen Bildpunktes beinhalten muß. Die Speicherung der Begriffs- und Optikinformation für jeden Bildpunkt in dem virtuellen Schirm hätte jedoch die Folge, daß die Speicherkapazität des virtuellen Schirmes über das für die Praxis vertretbare Maß hinaus erhöht werden müßte. Der virtuelle Schirm von heutigen Display-Systemen beinhaltet - mehr oder weniger redundant codiert - ausschließlich die Optikinformation des logischen Bildes und wird daher in dieser Arbeit als optisches Indexbild bezeichnet. Ein optischer Indexwert stellt somit einen nicht redundanten Code für

die Optik des Bildpunktes dar. Aus einem optischen Indexbild ist jedoch die Ableitung der Begriffsinformation - wenn überhaupt - nur umständlich möglich, so daß die Begriffe nicht immer eindeutig identifiziert werden können. Aus der Erkenntnis heraus, daß die Bildkommunikation hauptsächlich eine Begriffskommunikation ist, wird hier, im Gegensatz zu der üblichen Technik, in dem virtuellen Schirm grundsätzlich die Begriffsinformation gespeichert. Durch die Indizierung des virtuellen Schirms nach der Begriffszugehörigkeit der einzelnen Bildpunkte ergibt sich das sogenannte logische Indexbild. Ein logischer Indexwert stellt einen nicht redundanten Code für die Begriffszugehörigkeit des Bildpunktes dar. Das Konzept des logischen Indexbildes setzt jedoch voraus, daß die Zuweisung optischer Attribute während der Refresh-Zeit und nur aufgrund der Begriffszugehörigkeit einzelner Bildpunkte stattfinden kann. Ein hierfür geeignetes Verfahren, das eine tabellarische Zuweisung der Optik ermöglicht, wird im Abschnitt 2.1.3 näher erläutert werden.

2.1.1. Definition optischer Attribute

Die intensive Verwendung von Optikinformation wird hier als Anlaß genommen, eine klärende Diskussion über die optischen Attribute durchzuführen.

Das optische Attribut eines Bildpunktes kann im weitesten Sinne als Zuordnung von Grau- bzw. Farbtönen aufgefaßt werden. Ein beliebiger Farbton entsteht durch die Mischung der drei Farbprimitiven Rot (R), Grün (G) und Blau (B) und kann durch den Chrominanzvektor

$$\underline{C} = (R, G, B) \quad (2.1)$$

angegeben werden (s. Bild 2.1a). Der Grauwert, bzw. die Luminanz, des Farbtones ergibt sich aus der sogenannten Luminanz/Chrominanz-Gleichung

$$Y = 0,30 R + 0,59 G + 0,11 B \quad (2.2)$$

und braucht daher nicht zusätzlich angegeben zu werden.

Der Vollständigkeit halber soll erwähnt werden, daß ein neues Farbdarstellungssystem - das FSW-System (Farbwert, (Farb)-Sättigung, (Grau)-Wert) - wegen der einfachen Korrelation mit den physiologischen Eigenschaften des menschlichen Gesichtssinnes (s. Bild 2.1b) in letzter Zeit zunehmend verwendet wird /JOBLOVE-78/. Dies bleibt jedoch im folgenden außer Betracht.

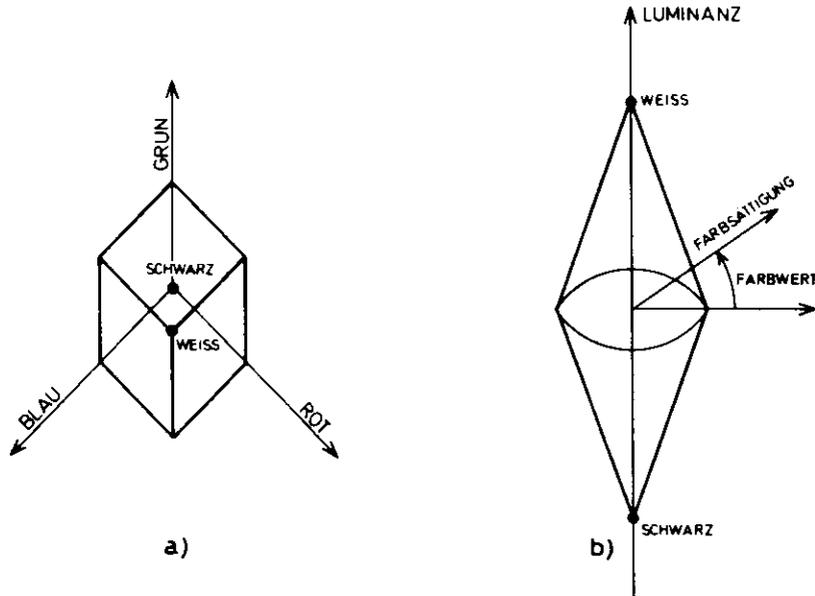


Bild 2.1. Farbwertdarstellung: a) RGB-Farbsystem
b) RSW-Farbsystem

Die Angaben zur Definition eines Farbtones müssen zusätzlich eine Aussage über die Farbtransparenz enthalten. In der reellen Welt findet man Farbtöne, die wegen des endlichen Mischungsverhältnisses zwischen den Transmissions- und Remissionsfaktoren teils Filter-, teils deckende Farben sind. Hier wird auf solche Zwischenstufen verzichtet und die Farbtransparenz nur mit einem binären Farbtransparenzattribut angegeben, dem entweder reine Filter- oder reine deckende Farben entsprechen:

$$FT := \text{Farbtransparenzattribut} \quad (2.3)$$

mit $FT := T$ Filter- oder transparente Farbe

$FT := D$ Aufsicht- oder deckende Farbe

Mit den oben beschriebenen Einschränkungen kann das optische Attribut eines Bildpunktes quantitativ in der folgenden Form dargestellt werden:

$$A_0 = (C, FT) = (R, G, B, FT) \quad (2.4)$$

Operationen mit den optischen Attributen - beispielsweise additive Farbmischung - sind wegen des neu eingeführten Farbtransparenzattributs sequenzabhängig geworden. Man stellt sich deshalb die möglichen Operationen mit den optischen Attributen als eine Sequenz von Farbauftragungen auf einer Glasscheibe vor. Trägt man die Farben beispielsweise auf der Rückseite der Glasscheibe auf, so bewirkt die Auftragung einer transparenten Farbe die additive Mischung mit den nachkommenden Farben, während die Auftragung einer deckenden Farbe alle später aufgetragenen Farben endgültig verdeckt. Für die additiven Mischoperationen auf der Rückseite der Glasscheibe gelten folgende Regeln:

$$(C, FT)^{n+1} = \begin{cases} \text{für } (FT)^n = D & : (C, FT)^n \\ \text{für } (FT)^n = T & \begin{cases} \text{für } (\Delta FT)^n = D : (C \oplus \Delta C, D) \\ \text{für } (\Delta FT)^n = T : (C \oplus \Delta C, T), \end{cases} \end{cases} \quad (2.5)$$

- wobei (C, FT) : vorhandenes optisches Attribut,
- $(\Delta C, FT)$: neu "aufgetragenes" optisches Attribut,
- \oplus : Operator für "Addition mit Sättigung",
- n : diskrete Auftragszeit

bedeuten.

2.1.2. Relation Begriff/Optik

Um eine geeignete Zuweisungsform der optischen Attribute zu den als logische Indexwerte codierten Begriffen finden zu können, soll hier zunächst einmal die Beziehung Begriff/Optik näher betrachtet werden.

Ein optischer Begriff entsteht grundsätzlich durch die Zuordnung optischer Attribute zu der Punktmenge, die die mathematisch-geometrische Abstraktion des Begriffes aus der Datenstruktur darstellt. Das Bild 2.2 zeigt mit einem einfachen Beispiel die Bildung optischer Begriffe, wobei hier hauptsächlich die

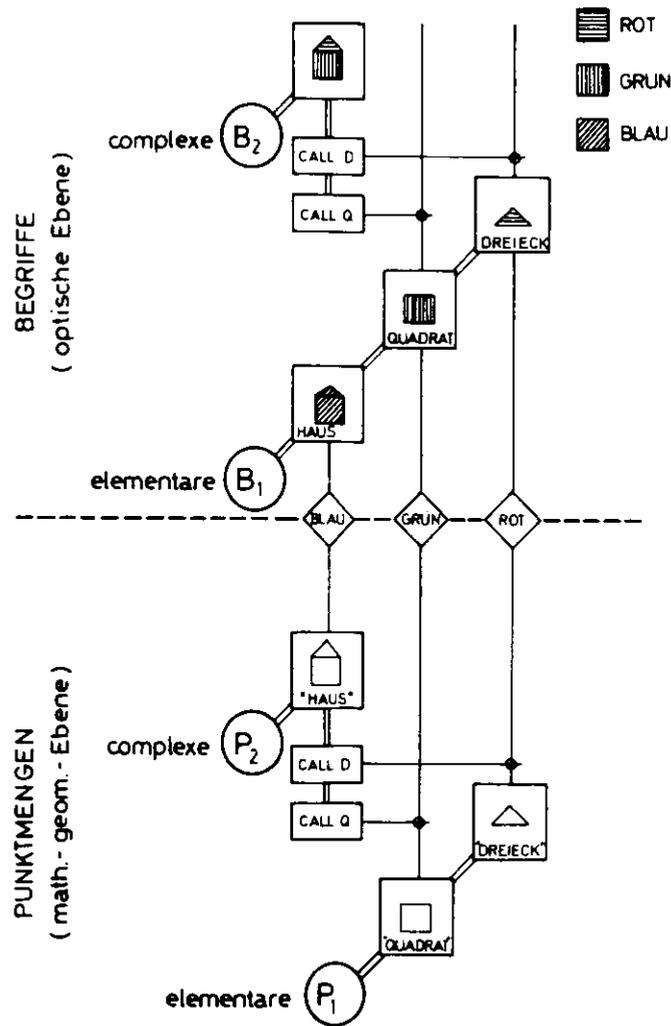


Bild 2.2 Bildung optischer Begriffe

Trennung zwischen der mathematisch-geometrischen und der optischen Betrachtungsebene verdeutlicht werden soll. Allgemeingültig kann die Optik eines Begriffes als eine Verteilung von Chrominanzvektoren über die Punktmenge des Begriffes aufgefaßt werden. Die Verteilungsfunktion ist durch das sogenannte Beleuchtungsmodell bestimmt /BLINN-77, CROW-77, JOBLÖVE-78, WHITTED-79/. Durch die Verwendung eines Beleuchtungsmodells hat ein Begriff normalerweise heterogene optische Attribute. Dies bedeutet, daß zunächst einmal jedem logischen Indexwert eine ganze Teilmenge optischer Indexwerte zugeordnet werden muß, die anschließend für die Zuweisung der tatsächlichen optischen Attribute verwendet wird. In der vorliegenden Arbeit wird das Beleuchtungsmodell absichtlich außer acht gelassen und die Verteilungsfunktion auf eine gleichmäßige Verteilung beschränkt. Dadurch wird jedoch die Gültigkeit des sich hier ergebenden Konzeptes nicht grundsätzlich eingeschränkt. Diese Einschränkung bedeutet, daß die Begriffe nur homogene optische Attribute besitzen können, und somit kann auf die Abbildung der logischen auf die optischen Indexwerte verzichtet werden. Die Optikuweisung geschieht dann direkt aus den logischen Indexwerten durch die tabellarische Zuordnung optischer Attribute (s. Abschnitt 2.1.3).

Als Gegenstände der Kommunikation können zwei Arten von Begriffen betrachtet werden:

(1) die unmittelbaren Begriffe, die sich durch die direkte Zuordnung der spezifizierten optischen Attribute ergeben. Beispielsweise stellen die elementaren Begriffe B1 in Bild 2.2 solche unmittelbaren Begriffe dar. Auch die aus der Datenstruktur resultierenden komplexeren Begriffe - z. B. B2 in Bild 2.2 - müssen auf der entsprechenden Kommunikationsebene als unmittelbare Begriffe betrachtet werden. Die Optik der unmittelbaren Begriffe resultiert aus den Spezifikationen in der Datenstruktur und wird hier als Normaloptik bezeichnet.

(2) die mittelbaren Begriffe, die durch die willkürliche Vereinigung einer Teilmenge von unmittelbaren Begriffen resultieren. Beispielsweise können alle unmittelbaren Begriffe "HAUSER" - also B1 und B2 in Bild 2.2 - unabhängig von ihrer Normaloptik als ein mittelbarer Begriff deklariert werden und durch eine gewünschte Optik - z. B. rot-blinkend - markiert werden. Die Optik des mittelbaren Begriffes ergibt sich allein aus dem Markierungswunsch des Anwenders und wird im folgenden Markierungsoptik bezeichnet.

Diese vielfältige, aber erforderliche Betrachtungsweise der Begriffe, die nur durch unterschiedliche optische Erscheinungen darstellbar ist, macht es notwendig, die Relation Begriff/Optik leicht veränderbar zu gestalten. Eine für diesen Zweck geeignete Zuordnungsform der Optik ist durch das im nächsten Abschnitt beschriebene Konzept der dynamischen Index/Optik-Zuordnung realisiert.

2.1.3. Konzept der dynamischen Index/Optik-Zuordnung

Das Prinzip der dynamischen Index/Optik-Zuordnung ist in Bild 2.3 veranschaulicht. Die angestrebte Entkopplung zwischen den Begriffen und deren optischen Erscheinung ist durch die getrennte Repräsentation einerseits der Begriffe als logisches Indexbild und andererseits der optischen Erscheinung als In-

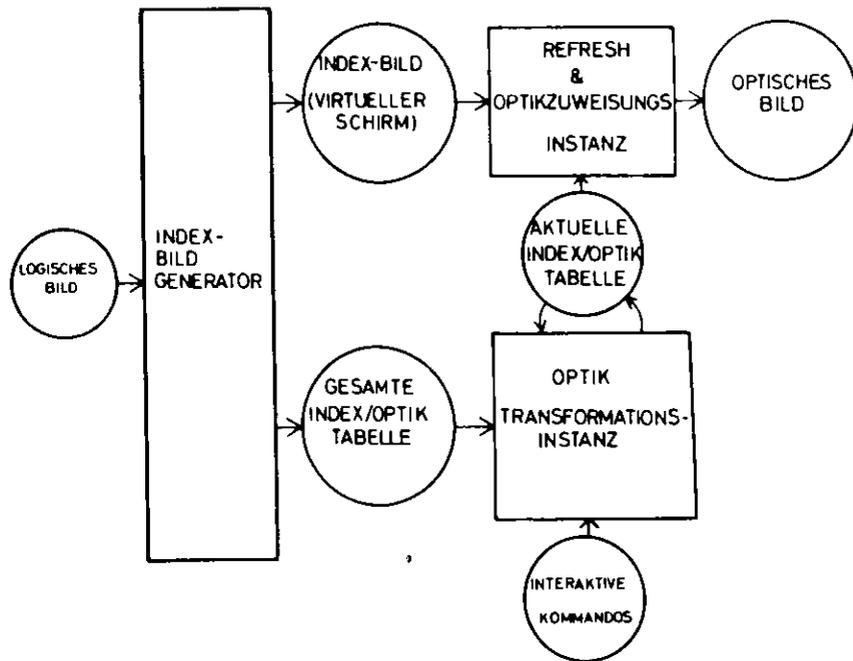


Bild 2.3. Prinzip der Index/Optik-Zuordnung

dex/Optik-Tabelle realisiert: jeder Begriff wird unabhängig von seiner Optik durch einen Indexwert im virtuellen Schirm dargestellt. Für jeden Index wird das optische Attribut des Begriffes in die Index/Optik-Tabelle eingetragen. Da diese Tabelle gewöhnlich alle bei der Abwicklung der Datenstruktur resultierenden unmittelbaren Begriffe und alle durch den Anwender definierten mittelbaren Begriffe beinhaltet, wird sie gesamte Index/Optik-Tabelle genannt. Die Abwicklung der Datenstruktur in das Indexbild und die Gesamt-Index/Optik-Tabelle wird im folgenden Begriff/Index-Transformation bezeichnet. Da die Begriff/Index-Transformation zwar die Zuordnung, nicht jedoch die Zuweisung der optischen Attribute ermöglicht, muß die tatsächliche Zuweisung während des Bildrefreshprozesses stattfinden. Die Refresh-Instanz beinhaltet daher zusätzlich eine Optikzuweisungseinrichtung, die den logischen Indexwerten in Echtzeit die zugeordnete optische Erscheinungsform tabellarisch assoziieren kann. Bestimmend für diese Assoziation ist die Optik-Transformation-Instanz. Sie erzeugt aus der gesamten Index/Optik-Tabelle auf Grund interaktiver Kommandos vom Anwender eine andere, nur für die gewünschte Betrachtungsweise der Begriffe geltende Tabelle, die hier als aktuelle Index-Optik-Tabelle bezeichnet wird. Diese Tabelle enthält gewöhnlich nur einen Teil der optischen Normal- bzw. Markierungsattribute der Begriffe und diese zweckmäßigerweise in einer redundanten Form, so daß die Erzeugung der reellen Optik unmittelbar stattfinden kann.

Mit der Struktur aus Bild 2.3 gelingt es, die optischen Normal- und Markierungsattribute nur durch die Veränderung der aktuellen Index/Optik-Tabelle, d. h. dynamisch, den Begriffen aus der Datenstruktur zuzuordnen.

Der Wirkungsgrad - und damit die Brauchbarkeit - der vorgestellten Struktur hängt von zwei wesentlichen Realisierungsaspekten ab:

Erstens muß durch die Begriff/Index-Transformation eine eindeutige Begriff/Index-Abbildung in beiden Richtungen gewährleistet werden, so daß jeder Begriff anhand des ihm zugeordneten Indexwertes eindeutig identifiziert werden kann. Bei der Abwicklung der Datenstruktur ist die Begriff/Index-Transformation jedoch von der Sequenz der Abwicklung abhängig, so daß eine eindeutige Begriffsidentifikation nur durch die zusätzliche Speicherung der Indexgenerierungssequenz möglich ist. Diese Problematik, die sich prinzipiell im Zusammenhang mit den Raster-Scan-Display-Systemen ergibt, wird im Kapitel 4 eingehend behandelt.

Zweitens muß die Zuweisung der optischen Attribute einfach und in Echtzeit stattfinden. Diese erwünschte Zuweisungsart kann mit Look-up-Tabellen (LUT) realisiert werden (s. Bild 2.4). Eine LUT ist hardwaremäßig in Form von Schreib-Lese-Speichern (RAM) realisiert und wird durch die Indexwerte adressiert.

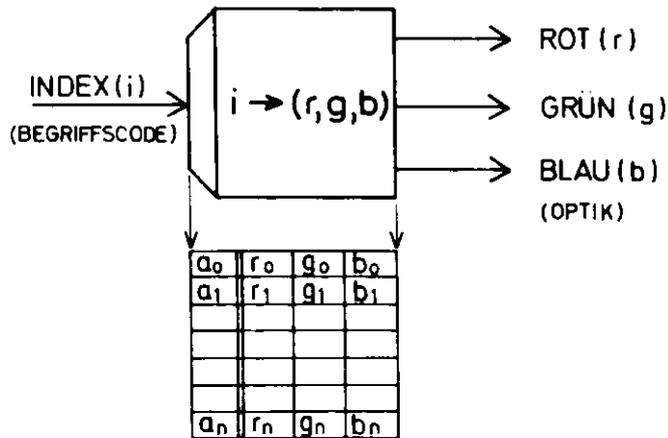


Bild 2.4 Look-up-Tabellen-Prinzip

Die Tabelle enthält die zugehörigen optischen Attribute - beispielsweise in RGB-codierter Form. Während des Bildrefreshprozesses werden somit die Indizes (I) aus dem virtuellen Schirm allein durch die Adressierung der LUT in die optischen Primärfarben (RGB) transformiert:

$$I \rightarrow (RGB) \quad (2.6)$$

Durch die Verwendung der LUT ergeben sich folgende Vorteile /SLOAN-79/:

- Einfache Zuweisung der Optik; die optische Erscheinungsform eines Begriffes wird durch das Beschreiben einer Adresse der LUT mit dem entsprechenden RGB-Wert bestimmt.

- Echtzeit-Zuweisung der Optik; die Transformation (2.6) wird zeitlich nur durch die Zugriffszeit der LUT bestimmt. Dadurch ist eine Echtzeit-Transformation durch Verwendung schneller RAMs (Zugriffszeit \leq Pixeldarstellungszeit) möglich.

- Kleine Speicherkapazität für LUT: die Speicherkapazität der LUT

$$K_{LUT} = 2^i \cdot (r + g + b) \quad (2.7)$$

mit 2^i := Anzahl von Indizes,
 r, g, b := Bits/Primärfarbe,
 i := Bits per Pixel im virtuellen Schirm

ist nicht von der Speicherkapazität des virtuellen Schirms

$$K_{VS} = X_{max} \cdot Y_{max} \cdot i \quad (2.8)$$

mit $X_{max} \cdot Y_{max}$:= räumliche Auflösung

abhängig, so daß in der Regel die Bedingung

$$K_{LUT} \ll K_{VS} \quad (2.9)$$

erfüllt ist. Für einen virtuellen Schirm mit einer Speicherkapazität von 1 Megabyte (1024 x 1024 Pixel x 8 Bits) benötigt man eine LUT mit einer Speicherkapazität von $K_{LUT} = 256 \times 24$ Bits, um 2^{24} Farbwerte darstellen zu können.

Komplexere LUT-Strukturen, die sowohl für die Bestimmung des optischen Aspektes der Datenbegriffe als auch für frei programmierbare Bildpunkttransformationen eingesetzt werden können, werden in den Kapiteln 4 und 5 dargestellt.

Das Konzept der dynamischen Index/Optik-Zuordnung bietet eine realistische Möglichkeit, die Relation Begriff/Optik in einer bisher nicht bekannten Weise sowohl einfach als auch flexibel zu gestalten, und wird deshalb in dieser Arbeit konsequent eingesetzt.

2.2. Die Flächengruppe als neues Strukturelement

Die Darstellung von Flächen kann in den Randomscan-Display-Systemen nur durch Verwendung komplexer mathematischer Modelle für die parametrische Erzeugung von Flächen /CLARK-77, GREENBERG-77/ erreicht werden und ist i. a. umständlich und ineffizient. Die Raster-Display-Systeme bieten im Gegensatz zu den Randomscan-Display-Systemen grundsätzlich die Möglichkeit, Flächen darzustellen. Die Fläche hat als zweidimensionales Bildelement eine doppelte Bedeutung: Erstens ermöglicht die Fläche grundsätzlich eine bessere Darstellung von reellen Bildobjekten. Dies führt im Zusammenhang mit einem geeigneten Beleuchtungsmodell, das beispielsweise perspektivische Schattierungen für abgebildete Gegenstände ermöglicht /JOBBOVE-78/, zu einer optischen Erscheinungsform der Graphikinformation, die kaum von den reellen Bildobjekten unterschieden werden kann. Dadurch erhalten die Graphik- und Abbildinformation mindestens vergleichbare Darstellungsmöglichkeiten, was für die einheitliche Behandlung der logischen Bilder eminent wichtig ist. Zweitens eröffnet die Fläche, die als Argument für den kompletten Satz von Booleschen Operatoren auftreten kann, völlig neue Möglichkeiten für die Erzeugung neuer komplexerer Bilder, was im Laufe dieses Abschnittes näher erläutert wird. Bei der Suche nach einer geeigneten (mathematischen) Definition der Fläche als Bildelement muß man aus historischen Gründen besonders auf die Kompatibilität mit den vorhandenen Vektorbildern achten. Dies bedeutet, daß die Fläche nur mit der Information, die durch den begrenzenden Vektorzug vorgegeben ist, definierbar sein muß (s. Bild 2.5). Eine

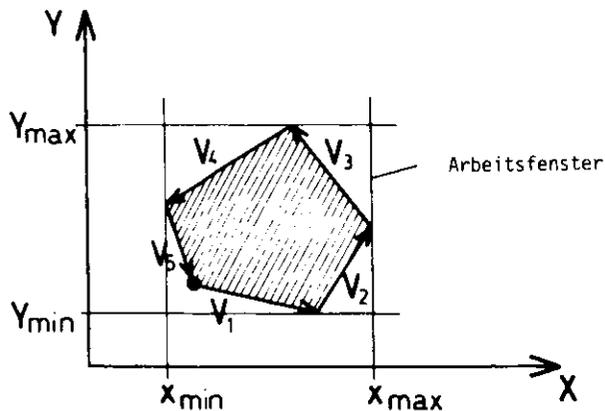


Bild 2.5 Zur Definition der polygonalen Flächen

eindeutige Definition kann deshalb nur durch bestimmte topologische Einschränkungen für den begrenzenden Vektorzug erreicht werden: Der Vektorzug muß (a) geschlossen sein und (b) einen einzigen Flächenbereich separieren, so daß es keine gerade Linie gibt, die von einem beliebigen Innenpunkt startend bei beliebiger Länge und Richtung den Vektorzug in einer geraden Anzahl von Punkten durchschneidet. Solche Flächen werden im folgenden als polygonale Flächen bezeichnet. Die polygonale Fläche kann somit als "die Summe aller Punkte, die sich beim Abfahren des begrenzenden Vektorzuges im Gegen- bzw. Uhrzeigersinn links bzw. rechts vom Vektorzug befinden", definiert werden. Eine effiziente Verwendung der Fläche als Bildelement hängt wesentlich von der Lösung der mit der Flächengenerierung und Flächenoperationen verbundenen Problematik ab.

2.2.1. Flächengenerierung

Algorithmen für die Generierung polygonaler Flächen müssen die Punktmenge einer Fläche korrekt und schnell im virtuellen Schirm erzeugen. Bei der Suche nach der Methodik solcher Algorithmen ist es hilfreich, die Vorgehensweise des Menschen beim Zeichnen als Modell zu nehmen. Demnach ist es sinnvoll, zuerst die Kontur zu generieren und danach die resultierende Fläche zu füllen. Eine Lösungsmöglichkeit besteht darin, die polygonale Fläche zuerst in einfachere benachbarte Teilflächen - gewöhnlich Dreiecke - einzuteilen /FULTON-74/. Diese können dann einfach gefüllt werden. Die korrekte Dekomposition in Dreiecke ist jedoch stark von der Topologie der Fläche abhängig, so daß die Flächenteilungsalgorithmen komplex und damit zeitaufwendig gestaltet werden müssen. Diese eher analytische Lösung hat für die Praxis keine Bedeutung. Eine andere, erstmals von /DUNDANI-76/ vorgeschlagene Lösung strebt die direkte Füllung der gesamten polygonalen Fläche durch die zeilenweise Abtastung des Flächenbereiches an. Hier wird während der Abtastung durch die Identifizierung der Konturpunkte und der Übergangsrichtung - von außen nach innen oder umgekehrt - festgestellt, welche Punkte sich innerhalb und welche außerhalb der Kontur befinden. Die Abtastung kann auf einen durch die Vektorzuggrenzen (x_{min} , x_{max} , y_{min} , y_{max}) definierten Bereich eingeschränkt werden, der im folgenden als Arbeitsfenster bezeichnet wird (s. Bild 2.5). Da die Abbildung der unendlich kleinen geometrischen Vektorzugpunkte auf dem virtuellen (und reellen) Bildschirm wegen der begrenzten Auflösung nur durch quadratische Pixel endlicher Größe stattfinden kann, ist eine eindeutige Identifizierung der Konturpunkte nicht immer möglich. Bild 2.6 zeigt die zwei hauptsächlichen Fehlerquellen, die durch diese unvollkommene Abbildung entstehen können: Konturpunkte vom Typ A, die geometrisch un-

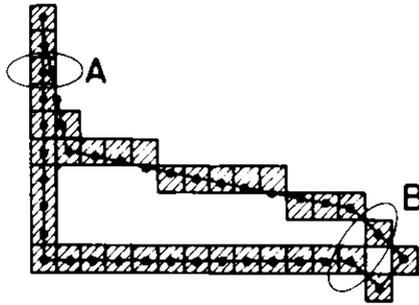


Bild 2.6 Fehlerquellen bei der Flächengenerierung

terschiedlichen Vektoren angehören, werden mit dem gleichen Pixel dargestellt und können dadurch bei der Abtastung nicht mehr getrennt identifiziert werden. Konturpunkte von Typ B, die auch geometrisch unterschiedlichen Vektoren angehören, werden mit benachbarten Pixeln dargestellt und bilden dadurch zusammen einen neuen "diagonalen" Konturvektor. Die Fläche, die so durch den Pixel-Vektorzug vorgegeben ist, erfüllt nicht mehr die topologischen Bedingungen einer polygonalen Fläche. Um diese Abbildungsfehler vermeiden zu können, müssen die Flächengenerierungsalgorithmen i. a. bei der Identifizierung von Konturpunkten nicht nur einen, sondern mehrere benachbarte Pixels gleichzeitig untersuchen. Dies führt aber zwangsläufig - wegen des vielfachen Zugriffs zum virtuellen Schirm - zu einem Geschwindigkeitsverlust für die Flächengenerierungsalgorithmen.

Bild 2.7 zeigt die grobe informationelle Struktur für die Flächengenerierung auf. Das Arbeitsfenster wird zweckmäßigerweise schon bei der Vektorzuggenerierung erzeugt und anschließend vom Flächengenerator verwendet. Eine detaillierte Beschreibung des Flächengenerators und des dazugehörigen Flächengenerationalgorithmus ist in Abschnitt 4.3.2 gegeben.

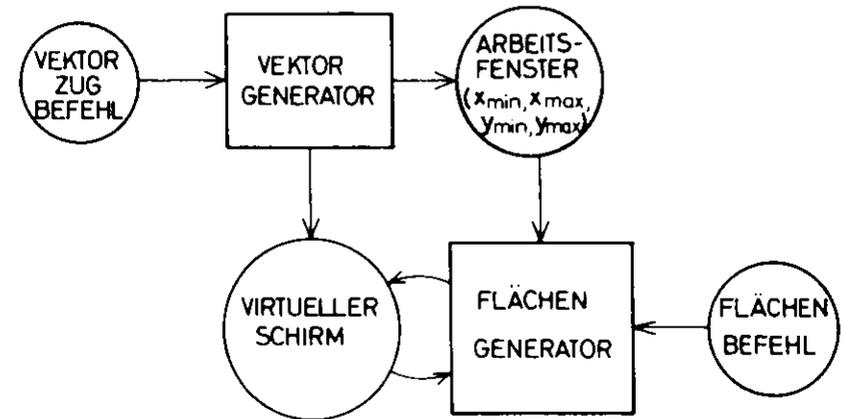


Bild 2.7 Informationelle Struktur des Flächengenerators

2.2.2. Flächenoperatoren

Nach der Generierungsfrage für eine Bildelementart stellt die Frage nach den möglichen, aber auch sinnvollen Operationen mit solchen Bildelementen einen wesentlichen Aspekt für ihre Verwendung dar. Diese Operationen werden i. a. dazu verwendet, um durch logische Verknüpfung aus zwei oder mehreren Bildelementen ein neues, meist komplexeres Bildelement erzeugen zu können. Von besonderem Interesse hinsichtlich der hierarchischen Erzeugung von Bildern (s. Kapitel 3) sind solche logischen Operatoren, bei denen das resultierende Bildelement selbst ein weiterverknüpfbares Bildelement darstellt. Solche Operatoren werden im folgenden als repertoirebildende Operatoren bezeichnet.

Die Vektorbilder, die nur durch Überlagerung einfacher Vektorteilbilder entstehen können, akzeptieren allein die Vereinigung als repertoirebildenden Operator. Da die Überlagerung der Teilbilder im Rahmen der sequentiellen Bildentstehung geschieht, wurde dieser implizierte Vereinigungsoperator bisher auch nicht bewußt als logische Verknüpfung aufgefaßt.

Die Flächenbilder dagegen akzeptieren per Definition den kompletten Satz Boolescher Operatoren - Vereinigung, Intersektion und Komplement - als repertoirebildende Operatoren. Dies eröffnet für den Anwender theoretisch ganz neue und erheblich effizientere Modellierungsmöglichkeiten in der Datenstruktur der Graphikinformation. Dies wird in Kapitel 3 anhand von Beispielen deutlicher zum Ausdruck gebracht. An dieser Stelle soll bemerkt werden, daß die Realisierung des Komplementes als Flächenoperator zweckmäßigerweise durch eine Veränderung der Flächendefinition durchgeführt werden sollte: Das Arbeitsfenster wird durch die Grenzen des virtuellen Schirmes dargestellt. Gesucht werden in diesem Falle die Punkte, die sich außerhalb des Vektorzuges befinden. Die praktische Nutzbarkeit dieser theoretisch verfügbaren logischen Operatoren wird jedoch an der Lösung von zwei wichtigen Verwendungsaspekten gemessen: Einerseits soll die Spezifizierung des gewünschten logischen Operators in der Datenstruktur einfach und gleichzeitig flexibel gestaltet werden, und andererseits soll die Ausführung jedes spezifizierten Operators korrekt, schnell und zweckmäßig durchgeführt werden. Diese beiden Aspekte sind durch das im nächsten Abschnitt dargestellte Konzept der Flächengruppe abgedeckt.

2.2.3. Konzept der Flächengruppe

Es besteht die Aufgabe, N verschiedene Flächen F_i mit $i = 0, 1, \dots, N-1$, womit insgesamt $2^N - 1$ Minterme

$$m_j = \prod_{i=0}^{N-1} F_i^* \quad (2.10)$$

mit $F_i^* \in \{F_i, \bar{F}_i\}$

$$j = \sum_{i=0}^{N-1} f_i \cdot 2^i \quad \text{und} \quad f_i = 1 \text{ für } F_i^* = F_i, \quad f_i = 0 \text{ für } F_i^* = \bar{F}_i$$

bildbar sind, durch einen beliebigen Booleschen Operator

$$F = \sum_{k=1}^M m_k \quad (2.11)$$

mit $1 \leq M \leq 2^N - 1$

$$m_k \in \{m_j \mid j = 0, 1, \dots, 2^N - 1\}$$

logisch zu verknüpfen. Gesucht wird eine geeignete Spezifizierungsform dieser Verknüpfung in der Datenstruktur und die resultierende Fläche F in dem virtuellen Schirm. Eine beliebige logische Verknüpfung kann nur nach der vollständigen Generierung aller beteiligten Flächen stattfinden. Deswegen ist es sinnvoll, die sequentielle Generierung aller Flächen so durchzuführen, daß am Ende die Menge G aller möglichen Minterme, die mit den gegebenen N Flächen bildbar sind, im virtuellen Schirm identifizierbar sind, wobei

$$G = \{m_j \mid j = 0, 1, \dots, 2^N - 1\} \quad (2.12)$$

ist. Diese Menge G wird hier als Flächengruppe bezeichnet. Die Bestimmung einer beliebigen logischen Verknüpfung kann durch Vereinigung einer Untermenge von Mintermen aus der Flächengruppe problemlos erreicht werden.

Das Konzept der Flächengruppe soll mit einem einfachen Beispiel verdeutlicht werden. Das Ziel der in Bild 2.8 dargestellten Verknüpfung ist die Intersektion zwischen einem Quadrat F1 und einem Kreis F2 zu einem Halbkreis F3:

$$F3 = F1 \cdot F2 \quad (2.13)$$

Zuerst wird die Flächengruppe generiert. Um alle Booleschen Komponenten (oder Minterme) der Flächengruppe identifizierbar zu machen, werden die Eingangs-Operand-Flächen durch Index-Addition in dem virtuellen Schirm generiert. Stellt man sich den virtuellen Schirm ursprünglich mit dem Index 0 belegt vor, so kann das Quadrat F1 durch "ADDIERE 1" und der Kreis F2 durch "ADDIERE 2" generiert werden. Der aktuelle Additions-Index muß immer um eins höher als der maximale Index sein, der im virtuellen Schirm vorhanden ist. Die so entstandene Indexbelegung des virtuellen Schirms (s. Detail A) weist jeweils einen Indexwert für jede Boolesche Komponente der Flächengruppe auf. Die erwünschte logische Verknüpfung nach Gleichung 2.13 ist beispielsweise durch den Indexwert I = 3 dargestellt und wird graphisch durch einen logischen Verknüpfungskreis angedeutet. In der Datenstruktur muß eine logische Verknüpfung durch die codierte Eingabe aller an der Verknüpfung beteiligten Minterme spezifiziert werden.

Mit der Flächengruppe gelingt es, jeden logischen Operator unmittelbar nach der Flächengenerierung durch einfache Zuordnung bestimmter, durch die Spezifizierung des Operators vorgegebener Indexbereiche zu realisieren. Diese Einfachheit und Flexibilität des Verfahrens muß jedoch mit einem erhöhten Bedarf an

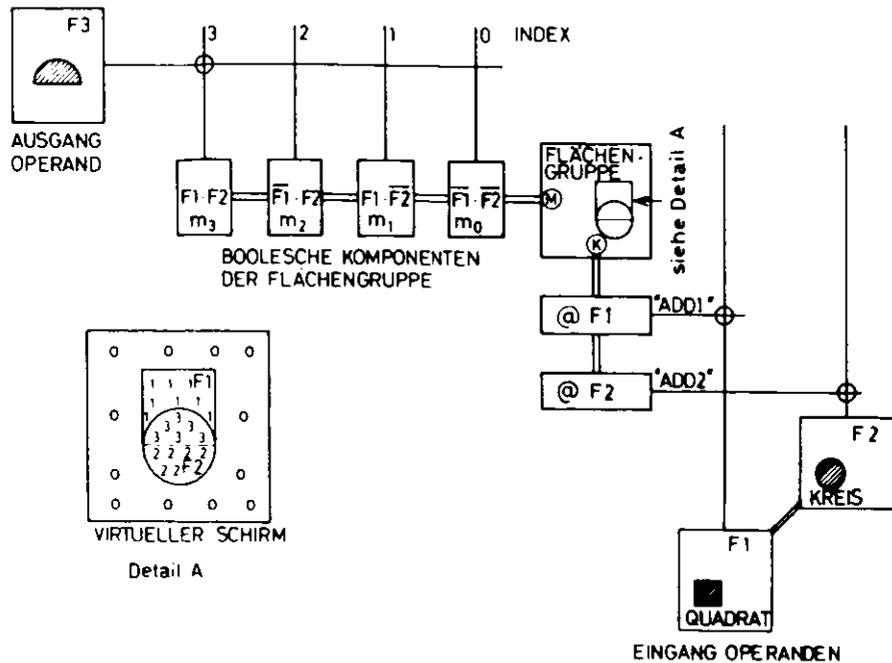


Bild 2.8 Veranschaulichung des Konzeptes der Flächengruppe

Indexwerten bezahlt werden. Es ist auffallend, daß bei der Generierung der Flächengruppe i. a. mehr Indizes erzeugt werden, als für eine gegebene Verknüpfung erforderlich wären. In Bild 2.8 werden beispielsweise für die Flächengruppengerierung drei neue Indizes (1, 2, 3) erzeugt, obwohl für die Begriffscodierung nur ein Index (3) gebraucht wird. Es muß deutlich gesagt werden, daß die für die Codierung der Minterme erzeugten Indexwerte nur durch die Konstruktion der Flächengruppe bedingt sind, und sie dürfen mit den in Abschnitt 2.1 eingeführten logischen Indexwerten nicht verwechselt werden. Bei der Generierung einer Flächengruppe wird der bisher verwendete Indexbereich um die Minterm-Anzahl erhöht. Eine Flächengruppe kann dann nur solange generiert werden, wie die Bedingung

$$N \cdot 2^i \leq 2^i \quad (2.14)$$

mit i = Bits/Pixel im virtuellen Schirm,
 i = Anzahl der zu verknüpfenden Flächen,
 N = belegter Indexbereich

noch erfüllt ist. Die abweichende Bedeutung der bei der Flächengruppengerierung erzeugten Indizes einerseits und deren starkes Wachstum andererseits deuten darauf hin, daß die Verwendung des Flächengruppen-Konzeptes nur in Verbindung mit einer anschließenden Index-Kompression praktikierbar sein kann. Danach muß der virtuelle Schirm konsequenterweise wieder nur logische Indizes beinhalten (s. Abschnitt 4.1.2.2).

2.3. Kachelstruktur des Weltbildes

In Abschnitt 1.2.1 wurde schon auf die Problematik der Domäne-Transformationen als Folge der Codierung des Weltbildes in einem Weltkoordinatensystem KS_W mit einem erheblich höheren Koordinatenwertebereich als der des Bildschirm-Koordinatensystems KS_B hingewiesen. Die Einfachheit der integralen Codierung des Weltbildes bewirkt jedoch in diesem Falle, daß die Algorithmen für die Domäne-Transformationen sehr komplex werden. Um diese Schwierigkeiten zu umgehen, wird in dieser Arbeit das Weltbild nicht mehr im ganzen Weltkoordinatensystem, sondern in mehreren Kacheln des Weltkoordinatensystems codiert. Dafür wird das Weltkoordinatensystem gedanklich in eine Matrix von Kacheln eingeteilt, die die gleichen Koordinatenwertebereiche wie die des Bildschirm-Koordinatensystems haben. Da jetzt die Kachel des Weltbildes und der Bildschirm die gleiche Auflösung aufweisen, sind die Domäne-Transformationen als Abbildungstransformationen prinzipiell überflüssig geworden. In Abschnitt 3.2.1.4 wird gezeigt, welche einfachen Operationen zur Durchführung der Abbildung des Weltbildes auf dem Bildschirm aufgrund der Kachelstruktur des Weltbildes noch nötig sind. Der erheblichen Vereinfachung der Domäne-Transformationen steht die (komplexere) kachelweise Codierung des Weltbildes gegenüber. Hierfür muß ein zusätzlicher Aufwand betrieben werden, um Bildobjekte, die sich über mehrere Kacheln erstrecken, in einer semantisch konsistenten Weise in die Datenstruktur der verschiedenen Kacheln zu codieren. Dieser Aufwand, womit die integrale Codierung des Weltbildes auf der Anwenderseite in die kachelweise Codierung in der Datenstruktur abgebildet wird, ist hinsichtlich der Komplexität der hierfür notwendigen Algorithmen und der benötigten Ausführungsgeschwindigkeit erheblich kleiner als der für die Durchführung der üblichen Domäne-Transformationen in Echtzeit. In dieser Arbeit werden daher nur Weltbilder mit Kachelstruktur betrachtet.

3. Datenstruktur

Dieses Kapitel befaßt sich mit der Festlegung eines Datenstrukturmodells für die Graphik- und Abbildinformation. Die Hauptaufmerksamkeit gilt dabei dem gewählten Bildmodell, dessen Mächtigkeit als Beschreibungsmittel sich entscheidend auf den potentiellen Informationsgehalt der logischen Bilder und somit auf die Brauchbarkeit der ganzen Datenstruktur auswirkt. Die Organisationsform der Datenstruktur, die bei der tatsächlichen Implementierung durch verschiedene maschinenspezifische Randbedingungen beeinflusst wird, ist nur durch einen prinzipiellen Vorschlag für die Codierung und die Zugriffsart auf die Datenobjekte angegeben. Die Datenstruktur wird weniger aus einem formellen, als aus dem anschaulichen Blickwinkel betrachtet. Dabei werden intensiv - wie könnte es anders in einer Arbeit über die Graphik sein - graphische Darstellungsmittel verwendet.

3.1. Grundbegriffe der Datenstrukturdarstellung

Bevor die Beschreibung der spezifischen Datenstruktur für die Graphik- und Abbildinformation begonnen werden kann, sollen hier die allgemeinen Grundbegriffe der Datenstrukturdarstellung kurz erläutert werden. (Weitere Informationen in /FRY-76/ und /KELLER-78/). Eine Datenstruktur wird dargestellt als eine Menge von Datenobjekten

$$\{E_1, E_2, \dots, E_n\}, \quad (3.1)$$

wobei jedes Datenobjekt Attribute aus den Mengen

$$\{A_1, A_2, \dots, A_m\} \quad (3.2)$$

haben und in Relation zu anderen Datenobjekten stehen kann. Es sind zwei Arten von Relationen zu unterscheiden:

Die Attributrelationen

$$R_A \subseteq E_i \times A_1 \times \dots \times A_k \quad (3.3)$$

beschreiben formell die Zuordnung von sogenannten amorphen Attributen zu den Basisdatenobjekten. Da dies bereits durch die Begriffsbildung geschehen ist, stel-

len die Attributrelationen nur eine formelle Umschreibung der vorhandenen Begriffe dar und können nicht zur Bildung neuer Begriffe führen.

Die Strukturrelationen

$$R_S \subseteq E_1 \times E_2 \times \dots \times E_k \quad (3.4)$$

beschreiben gegenseitige Beziehungen irgendwelcher Art zwischen verschiedenen Datenobjekten. Die Elemente einer Strukturrelation können selbst wieder als Datenobjekte aufgefaßt werden und bilden somit eine neue Objektmenge. Die Strukturrelationen führen also im Gegensatz zu den Attributrelationen i. a. zur Bildung neuer Begriffe in der Datenstruktur. Eine besondere Strukturrelation ist die Hierarchierelation, wobei hier die Baumhierarchie und nicht die Netzhierarchie gemeint ist. Die Hierarchierelation, die in der Natur oft vorkommt, drückt die Beziehung "ein Datenobjekt gehört zu einem höheren Datenobjekt" aus. Die Menge der Datenobjekte, die zu einem höheren Datenobjekt in einer Hierarchiebeziehung stehen, wird in /KELLER-78/ als strukturiertes Attribut des höheren Datenobjektes bezeichnet. Die Mächtigkeit eines Datenstrukturmodells wird einerseits an seiner Fähigkeit, komplexe mehrstellige Strukturrelationen in einfacher Weise verknüpfen zu können, und andererseits an der dafür erforderlichen Komplexität seiner Organisationsform gemessen. Allgemein gilt, daß mit dem Umfang der möglichen Strukturrelationen auch der Aufwand für die Codierung und Speicherung der Datenobjekte wächst /FRY-76/, so daß bei der Wahl eines Datenstrukturmodells diesbezüglich ein Kompromiß geschlossen werden muß.

Im nächsten Abschnitt wird am Beispiel der Graphik- und Abbildinformation gezeigt, welche Aspekte für die Festlegung der Datenstruktur relevant sind.

3.2. Hierarchisches Bilddatenstrukturmodell

Hier wird im Gegensatz zu der vorher im Zusammenhang mit dem Begriff des strukturierten Attributes erwähnten Baumhierarchie eine Netzhierarchie betrachtet, wo Elemente, die nicht in benachbarten Hierarchieebenen liegen, in unmittelbarer Hierarchiebeziehung stehen dürfen.

Bei der Suche nach einem geeigneten Bilddatenstrukturmodell für die Graphik- und Abbildinformation spielen zwei Bildbeschreibungsaspekte eine Rolle:

(1) Die Erzeugung beliebig komplexer Bilder, wie sie bei der Bildsynthese gefordert wird, kann sinnvollerweise nur durch iterative Zusammensetzung mehrerer einfacherer Bilder geschehen. Diese hierarchische Aufbauweise von Bildern zeigt, daß die Hierarchierelation zweckmäßigerweise als repertoirebildende Strukturrelation der Datenstruktur gewählt wird. Das Datenstrukturmodell muß infolgedessen die Verknüpfung von Hierarchierelationen, die wegen der dargestellten Beziehung zwischen einem übergeordneten Datenobjekt und mehreren untergeordneten Datenobjekten auch "1 zu N -Relationen" genannt werden, in einfacher, aber effizienter Weise ermöglichen.

(2) Die Darstellung komplexer informationeller Beziehungen zwischen den unmittelbaren Begriffen, wie sie bei der interaktiven Bildmanipulation benötigt wird, kann zweckmäßigerweise durch Erzeugung der im Abschnitt 2.1 eingeführten mittelbaren Begriffe realisiert werden. Die mittelbaren Begriffe resultieren auch durch Verknüpfung von Hierarchierelationen mit den unmittelbaren Begriffen, die aufgrund eines Attributes dem mittelbaren Begriff zugehörig sind. Wegen des breiten Interpretationsrahmens der unmittelbaren Begriffe kann die Hierarchierelation, die zur Bildung mittelbarer Begriffe führt, frei, aber semantisch konsistent assoziiert werden. Die Markierung der so resultierenden mittelbaren Begriffe wird dann assoziative Markierung genannt. Ein Sonderfall für die Verknüpfung einer solchen Hierarchierelation ist die Verwendung der repertoirebildenden Hierarchierelation aus der Datenstruktur selbst, die normalerweise zur Bildung von Oberbegriffen führt. Damit ist es möglich, beliebige Unterbäume in der Datenstruktur zu markieren. Diese Markierungsart wird im folgenden als direktive Markierung bezeichnet. Die beiden Markierungsarten können mit dem Konzept der dynamischen Index/Optik-Zuordnung (s. Abschnitt 2.1.3) vorteilhaft realisiert werden und brauchen daher in der Datenstruktur selbst nicht berücksichtigt zu werden.

Mit der Absicht, das Datenstrukturmodell einerseits transparent zu gestalten und die Organisationsform der Daten andererseits einfach zu halten, wird in dieser Arbeit für die Datenstruktur der logischen Bilder ein hierarchisches Datenstrukturmodell entwickelt. Obwohl dieses hierarchische Modell hauptsächlich für die logische Beschreibung graphischer Bilder gedacht ist, kann es mit gewissen Einschränkungen auf die Videobilder übertragen werden.

Die Einführung der Netzhierarchie als Modell für die Datenstruktur der Graphik- und Abbildinformation kann nur im Zusammenhang mit dem Begriff des Repertoireelementes begründet werden. Ein Repertoireelement stellt ein Datenobjekt dar, das auf Grund seiner Definitionsart zur einfachen und gleichzeitig flexiblen Erzeugung neuer, hierarchisch übergeordneten Datenobjekte verwendet wird. Bei der Definition eines Repertoireelementes wird ein Teil seiner Attribute, beispielsweise die mathematisch-geometrischen oder die optischen Attribute, parametrisch definiert. Bei der Verwendung eines solchen Repertoireelementes werden die parametrisch definierten Attribute durch die tatsächlich gewünschten Werte ersetzt. Diese Attributwerte sind spezifisch für die Verwendung des Repertoireelementes und können daher als Verwendungsparameter aufgefaßt werden. Durch die Zuweisung unterschiedlicher Verwendungsparameter kann ein Repertoireelement, das in der Datenstruktur einmalig vorhanden ist, mit verschiedenen Erscheinungsformen für den hierarchischen Aufbau komplexerer Datenobjekte verwendet werden.

Die Wirkungsweise dieser Vorstellung von Repertoireelementen kann durch den direkten Vergleich mit dem Prozedur-Konzept, das in dem Softwarebereich extensiv angewandt wird, verdeutlicht werden. Dabei entspricht die Prozedur-Deklaration der Definition des Repertoireelementes, während der Prozedur-Aufruf der Verwendung des Repertoireelementes gleichkommt. Im Falle des Repertoireelementes können als Aufrufparameter nur Attribute zugelassen werden, die eine veränderte Erscheinungsform des Repertoireelementes bewirken können. Die Aufrufparameter beinhalten daher die Werte für die mathematisch-geometrischen- und optischen Attribute.

Das Konzept des Repertoireelementes wird in der Datenstruktur hierarchisch verwendet, d. h. die Datenobjekte, die durch Verwendungsaufrufe von Repertoireelementen entstanden sind, stellen ihrerseits auch Repertoireelemente dar und können somit zur Erzeugung neuer Datenobjekte herangezogen werden. Auf dieser Weise wird die Hierarchierelation als repertoirebildende Strukturrelation der Datenstruktur gestaltet.

Die Mächtigkeit dieses netzhierarchischen Datenstrukturmodells, das den einfachen und transparenten Aufbau komplexer Datenobjekte mit einem kleinen Codierungsaufwand erlaubt, wird durch die nachfolgende detaillierte Beschreibung der Datenstruktur verdeutlicht.

3.2.1. Datenstruktur der Graphikinformation

Die graphischen Bilder werden durch Verwendung von vier hierarchisch angeordneten Klassen von Repertoireelementen logisch beschrieben. Diese Klassen oder Datenobjekttypen, die unterschiedliche Begriffsarten graphisch darstellen können, sind in ihrer hierarchischen Reihenfolge im folgenden beschrieben:

(1) Vektorzüge (V), die die Menge aller mit Vektoren erzeugbaren und beschreibbaren Datenobjekte beinhalten. Die Vektorzüge verwenden nur Vektoren als eindimensionales Bildelement und können daher nur zur Erzeugung von Linienbildern eingesetzt werden. Die Vektorzüge beschreiben nur den mathematisch-geometrischen Aspekt der darzustellenden Begriffe und dürfen nur als deren mathematisch-geometrische Abstraktion betrachtet werden.

(2) Flächen (F), die die Menge aller mit Flächen erzeugbaren und beschreibbaren Datenobjekte beinhalten. Die Flächenmenge verwendet die Fläche als zweidimensionales Bildelement und kann somit flächenartige Datenobjekte darstellen (s. Abschnitt 2.1). Die Flächenmenge beschreibt ebenso wie die Vektorzüge auch nur den (jetzt erweiterten) mathematisch-geometrischen Aspekt der darzustellenden Begriffe und darf daher auch nur als deren mathematisch-geometrische Abstraktion gesehen werden.

(3) Dias (D), die die Menge aller mit optischen Attributen versehenen Datenobjekte aus den Mengen der Vektorzüge (V) und Flächen (F) beinhalten. Die Dias stellen die optische Erscheinungsform der Begriffe dar.

(4) Kacheln (K), die die Menge aller Kacheln des Weltbildes (s. Abschnitt 2.3) beinhalten. Die Kacheln stellen jeweils einen Ausschnitt des Weltbildes dar, auf den die Dias (D) projiziert werden. Sie haben daher die gleichen Möglichkeiten für die Darstellung von Begriffen wie die Dias.

Innerhalb jedes Datenobjekttyps existiert wiederum eine zusätzliche Hierarchieordnung, die die hierarchische Bilderzeugung und -darstellung auf dieser Datenobjekttypenebene ermöglicht.

In den nächsten Abschnitten wird mit Hilfe eines einfachen Beispiels die Datenstruktur jedes dieser Datenobjekttypen durch die Beschreibung ihrer relevanten Eigenschaften sowie zugelassener Relationen verdeutlicht.

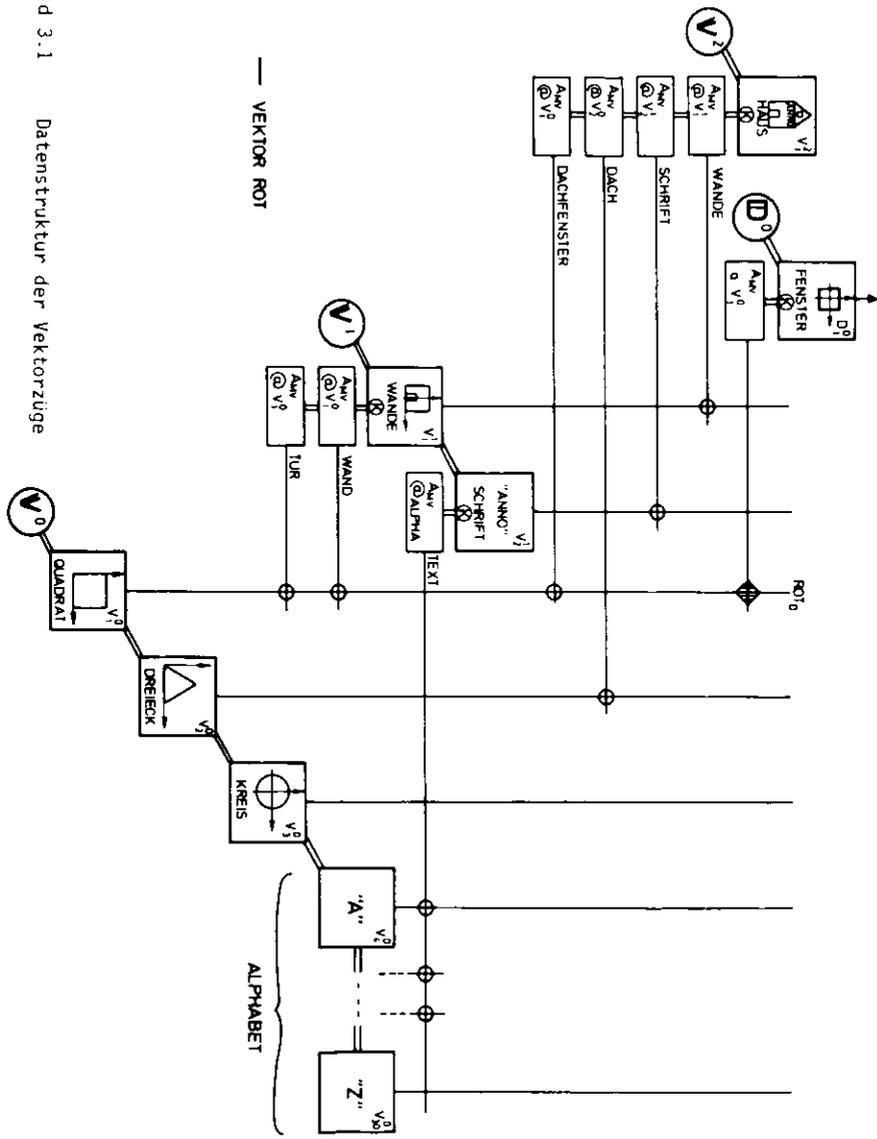
3.2.1.1. Datenstruktur der Vektorzüge

Bild 3.1 stellt graphisch die Modellvorstellung für die Datenstruktur der Vektorzüge dar. In diesem Beispiel soll gezeigt werden, wie aus einfachen Vektorzügen, beispielsweise Quadraten, Dreiecken oder Kreisen, durch ausschließliche Verwendung der Hierarchierelation komplexere Vektorzüge, beispielsweise ein Haus, erzeugt werden können.

Die Menge V^0 der elementaren Vektorzüge stellt das grundsätzliche Repertoire dar, das für die Erzeugung aller übergeordneten Vektorzüge verwendet werden kann. Ein elementarer Vektorzug ist graphisch allein durch die Reihenfolge der am Vektorzug beteiligten Vektoren mit Anfangs- und Endkennung vollständig definiert. Diese für die Definition des elementaren Vektorzuges maßgebende Reihenfolge von Vektoren kann daher als ein amorphes graphisches Attribut (A_{GV}) des elementaren Vektorzuges betrachtet werden. Das Alphabet der alphanumerischen Symbole (s. V_4^0 bis V_{30}^0 in Bild 3.1) wird sinnvollerweise auch als Menge elementarer Vektorzüge aufgefaßt, deren graphische Attribute durch einen Character-Generator vorgegeben sind. Diese einheitliche Betrachtungsweise für die Vektoren und alphanumerischen Symbole ist die Voraussetzung dafür, daß diese beiden Bildelementarten überall in der Datenstruktur, aber auch bei der Bild-darstellung und -manipulation, einheitlich und dadurch einfacher behandelt werden können.

Die Menge V^n der hierarchischen Vektorzüge entsteht durch die Verknüpfung einer Hierarchierelation über eine Menge untergeordneter Vektorzüge, die jedoch mindestens ein Element der Menge V^{n-1} beinhalten muß. Beispielsweise entsteht in Bild 3.1 der Vektorzug V_1^1 , der die Wände und die Tür eines Hauses darstellen soll, durch die doppelte Verwendung des elementaren Vektorzuges V_1^0 (Quadrat), allerdings jeweils mit verschiedenen mathematisch-geometrischen Attributen. Im allgemeinen wird die Verwendung eines untergeordneten Datenobjektes in einer Hierarchierelation durch den sogenannten Verwendungscall spezifiziert. Ein Verwendungscall beinhaltet einerseits einen Zeiger (@) auf das untergeordnete Datenobjekt und andererseits einen für den jeweiligen Datenobjekttyp relevanten Satz von Attributen des untergeordneten Datenobjektes. Diese Attribute, die gewissermaßen Parameter des Verwendungscalls darstellen, können beispielsweise mathematisch-geometrische Transformationsattribute (A_M), optische Attribute (A_O) oder andere frei definierbare Freie Attribute (A_M) - beispielsweise Verwendungsname - sein. Beinhaltet ein Verwendungscall auf einer beliebigen Hierar-

Bild 3.1 Datenstruktur der Vektorzüge



chreebene zusätzlich auch ein optisches Attribut A_{0V} , wie z. B. das Datenobjekt "Fenster" in Bild 3.1, gehört das resultierende Datenobjekt nicht mehr der Vektorzugmenge V , sondern als sichtbares Datenobjekt der Diamenge D^0 an. Ein so resultierendes Datenobjekt kann dann nur in Hierarchierelationen der Dias weiterverwendet werden. Somit ist gewährleistet, daß jedem in der Vektorzughierarchie auftretenden Datenobjekt homogene optische Attribute zugeordnet und somit optische Begriffe erzeugt werden können. Solche Verwendungscalls, die nicht mehr Datenobjekte gleichen Typs, sondern Dias produzieren, werden graphisch statt mit einem Kreis mit einem Rhombus gekennzeichnet. Alle für die Bildung eines übergeordneten Datenobjekts notwendigen Verwendungscalls beschreiben zusammen eindeutig die verwendete Hierarchierelation über die untergeordneten Datenobjekte und stellen daher ein komponentenbeschreibendes Attribut des übergeordneten Datenobjektes dar. Dieses strukturierte Attribut wird im folgenden Komponenten-Attribut genannt und ist graphisch durch die Komponentenmenge \textcircled{k} dargestellt (s. Bild 3.1). Mit so definiertem Verwendungscall und Komponenten-Attribut können beliebig viele hierarchische Vektorzüge einfach und flexibel gebildet werden. Der Wirkungsgrad der hierarchischen Datenstruktur wird an ihrer Fähigkeit gemessen, aus einem kleinen Repertoire von elementaren Datenobjekten alle erdenklichen hierarchischen Datenobjekte durch die entsprechende Verwendung der Attribute als Parameter der Verwendungscalls in einer transparenten Weise zu erzeugen. Dabei hängt die Anzahl der Hierarchieebenen nur von der gewünschten Begriffsauflösung innerhalb der Vektorzüge ab. Ein hierarchisches Datenobjekt soll nur dann gebildet werden, wenn es durch seine Semantik ein eigenständiger Begriff ist, der eventuell in einer weiteren Hierarchierelation verwendet werden kann. Beispielsweise bildet in Bild 3.1 das Datenobjekt V_1^1 , das den Unterbegriff "Wände" in dem Oberbegriff "Haus" repräsentiert, gerade wegen dieser Bedeutung eine eigene Hierarchieebene. Man hätte auch auf das Datenobjekt V_1^1 "Wände" verzichten und seine Komponenten direkt beim Datenobjekt V_1^2 "Haus" erzeugen können. Das Datenobjekt "Haus" würde dann der Hierarchieebene V_1^1 angehören.

An dieser Stelle soll nochmals betont werden, daß die Vektorzüge nur die mathematisch-geometrische Abstraktion der Begriffe darstellen, obwohl die Semantik der hierarchischen Vektorzüge mit der Hierarchieebene immer komplexer wird. Sie müssen daher als mehr oder weniger umfangreiche geometrische Punkt-mengen betrachtet werden. Deswegen beinhalten die Verwendungscalls in der Vektorzugmenge gewöhnlich nur mathematisch-geometrische Attribute A_{MV} , die die einzige Verwendungsart mathematisch-geometrischer Punkt-mengen vollständig beschrei-

ben. Bild 3.1 stellt nur ein statisches Bild der Datenstruktur der Vektorzüge dar. Die Abwicklung der Datenstruktur entsprechend deren Komponentenattributen und Verwendungscalls, die der Linksableitung eines logischen Baumes entspricht, findet durch die sequentielle Reduzierung von hierarchischen bis zu den elementaren Vektorzügen statt. Für jede Komponente eines hierarchischen Vektorzuges entsteht eine Sequenz von Verwendungscalls, die einerseits die elementaren Vektorzüge und andererseits deren resultierende Attribute für die Ausführung bestimmt. Diese Abwicklungsform ist allen Datenobjekttypen gemeinsam. Unterschiedlich sind nur die Gesetzmäßigkeiten für die Auswertung verschiedenartiger Attribute in den Verwendungscalls. Sie werden in Kapitel 4 ausführlich erläutert.

3.2.1.2. Datenstruktur der Flächen

Die Modellvorstellung für die Datenstruktur der Flächen F ist für das gleiche Beispiel eines Hauses in Bild 3.2 gezeigt.

Die Menge F^0 der elementaren Flächen entsteht durch die Anwendung der polygonalen Flächendefinition auf die Menge hierarchischer Vektorzüge V . Diese Vektorzüge müssen jedoch die in Abschnitt 2.2 formulierten Bedingungen für die polygonalen Flächen erfüllen. Die elementaren Flächen stellen wie die elementaren Vektorzüge auch das grundsätzliche Repertoire für die Erzeugung weiterer, komplexerer Flächen dar.

Die Menge F^n der hierarchischen Flächen entsteht durch Bildung von Hierarchierelationen über die Menge untergeordneter Flächen, wobei mindestens eine Fläche der Menge F^{n-1} angehören muß. Um bei der Bildung hierarchischer Flächen den erweiterten Satz logischer Flächenoperatoren ausnutzen zu können, ist es sinnvoll, die Hierarchierelationen über die Flächengruppen herzustellen. In der Datenstruktur der Flächen entsteht dadurch für jede Hierarchieebene der Flächen F^i auch eine Hierarchieebene der Flächengruppen G^i . Die Flächengruppe braucht zwei strukturierte Attribute, um vollständig beschrieben zu werden:

- (1) Die an der Flächengruppe beteiligten Flächen sind wie bei den Vektorzügen durch das Komponentenattribut \textcircled{K} spezifiziert. Die Verwendungscalls beinhalten in diesem Falle, wie bei den Vektorzügen, auch nur die mathematisch-geometrischen Transformationsattribute A_{MF} .

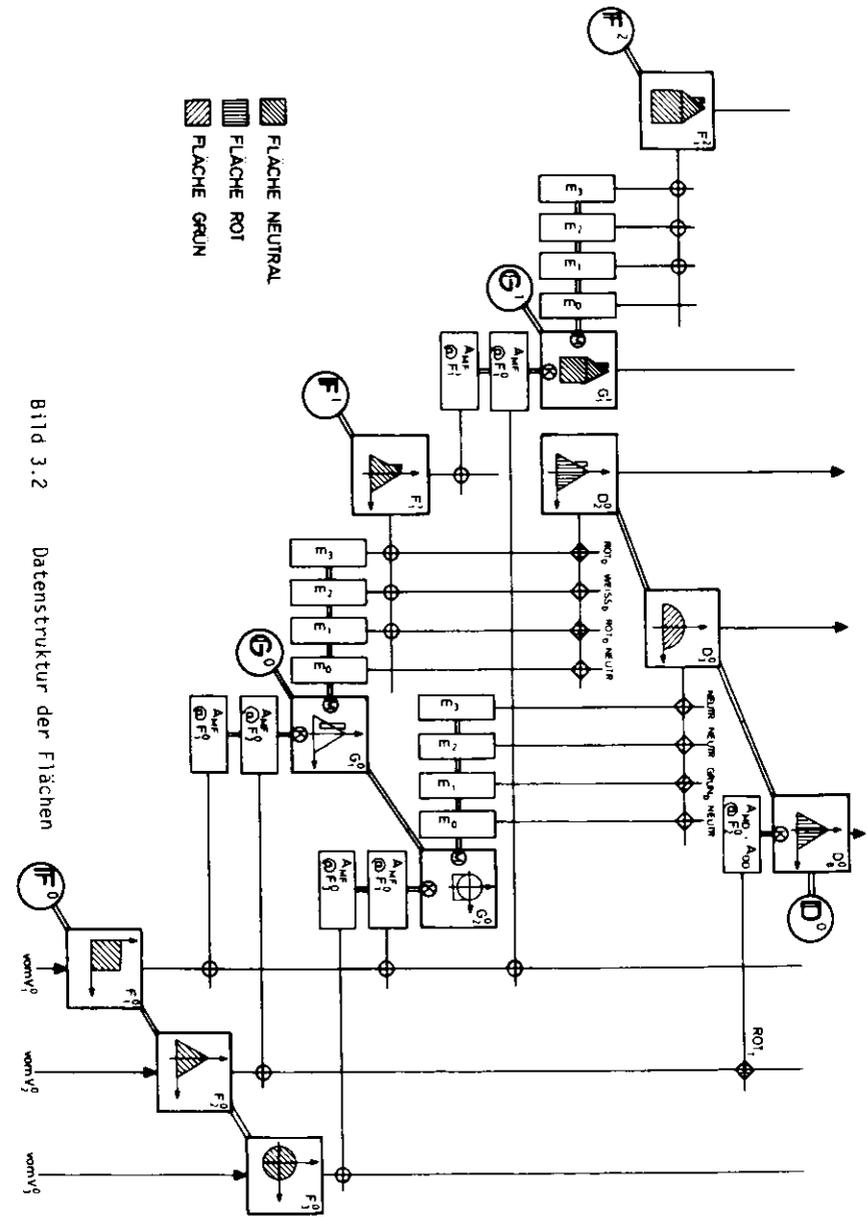


Bild 3.2 Datenstruktur der Flächen

(2) Die logische Operation, die mit der Flächengruppe realisiert werden soll, ist mit dem zweiten strukturierten Attribut, dem sogenannten Mintermattribut (M) , spezifiziert. Aus der Flächengruppe entsteht die neue Fläche durch Vereinigung der Minterme der Flächengruppe, wobei die ausgewählten Minterme mit logischen Verknüpfungskreisen graphisch gekennzeichnet werden (s. auch Abschnitt 2.2.3).

Beispielsweise entsteht die hierarchische Fläche F_1^1 , die das Dach eines Hauses darstellen soll, mit Hilfe der Flächengruppe G_1^0 . Das Komponentenattribut spezifiziert eine quadratische (F_1^0) und eine dreieckige (F_2^0) Fläche als Komponenten der Flächengruppe. Die resultierende hierarchische Fläche F_1^1 kann erneut in gleicher Weise zur Erzeugung anderer hierarchisch höherstehender Flächen (beispielsweise für das "Haus" F_1^2) verwendet werden.

Mit Hilfe der Komponenten- und Mintermattribute gelingt es, die Erzeugung hierarchischer Flächen einfach, flexibel und gleichzeitig transparent zu gestalten.

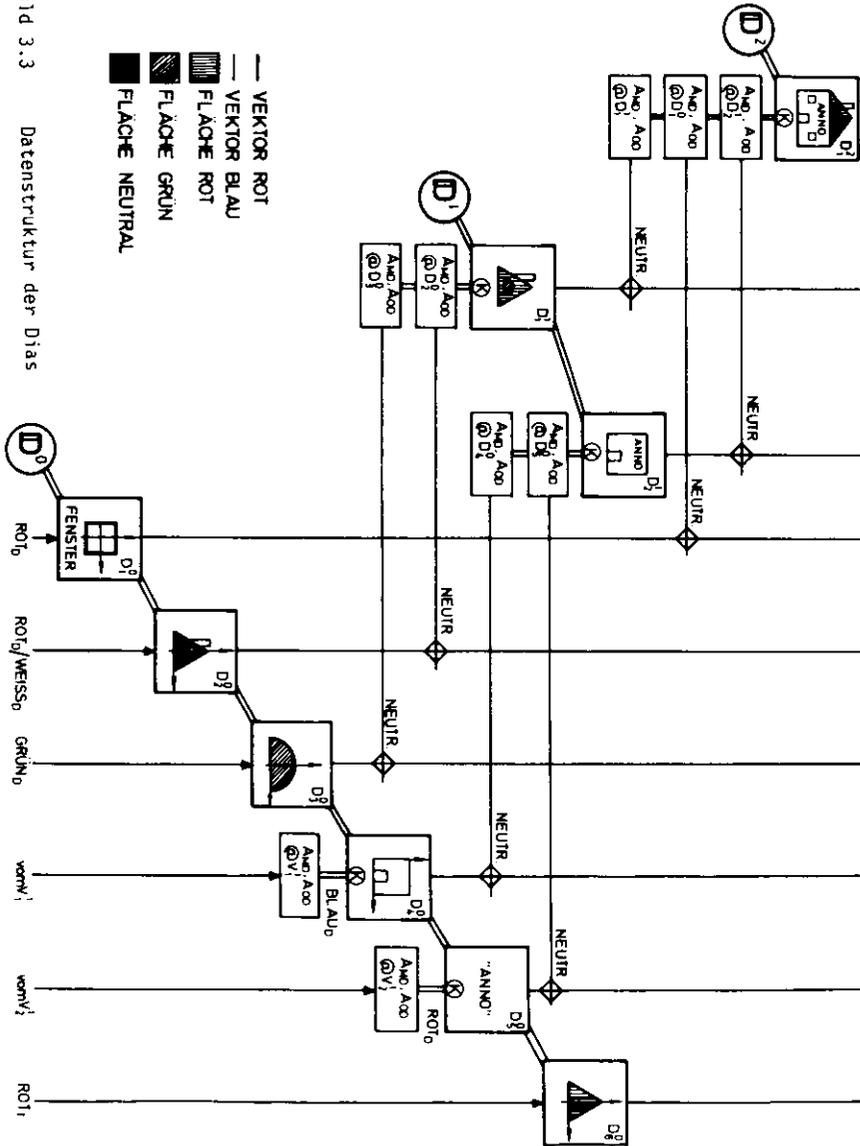
Die Erzeugung von Dias aus der Flächenmenge kann auf zwei Arten geschehen: Einerseits können durch die Spezifizierung eines optischen Attributes A_{OF} in einem Verwendungscall ähnlich wie bei den Vektorzügen homogene optische Attribute für die Punktmenge einer Fläche zugeordnet werden, beispielsweise D_6^0 in Bild 3.2. Andererseits eröffnet die Flächengruppe die Möglichkeit, heterogene optische Attribute zuzuordnen. Da die Flächengruppe ohnehin alle Minterme, die mit den Flächenkomponenten gebildet werden können, beinhaltet, kann man jedem Minterm, d. h. jeder zu einem Minterm gehörenden Punktmenge, ein optisches Attribut zuordnen und dadurch direkt ein optisch gesehen heterogenes Dia erzeugen. In Bild 3.2 wird zum Beispiel der Flächengruppe G_1^0 der optische Attributenvektor (ROT_D, WEISS_D, ROT_D, NEUTR) zugewiesen und somit direkt ein rot/weißes Dia erzeugt. Auf diese Weise ist es möglich, aus einer Flächengruppe mit N Komponenten ein Dia mit maximal 2^N verschiedenen optischen Attributen zu erzeugen. Die Dias als optische Erscheinung der Begriffe entstehen somit entweder aus der Menge F hierarchischer Flächen oder aus der Menge G hierarchischer Flächengruppen. Die Abwicklung der Datenstruktur der Flächen findet wegen ihrer hierarchischen Struktur in gleicher Form wie bei den Vektorzügen statt.

3.2.1.3. Datenstruktur der Dias

Die Modellvorstellung für die Datenstruktur der Dias ist in Bild 3.3 verdeutlicht. Hier wird das bei der Datenstruktur der Vektorzüge und der Flächen verwendete Beispiel weiterverfolgt.

Die Menge D^0 der elementaren Dias entsteht, wie schon angedeutet, aus der Menge der Vektorzüge, Flächen und Flächengruppen durch die Zuordnung optischer Attribute als Parameter in den Verwendungscalls. Die elementaren Dias stellen auch hier das grundsätzliche Repertoire dar, das für die Erzeugung aller weiteren Dias verwendet werden kann. Es soll aus Gründen der Übersichtlichkeit und des Aufwandes für den benötigten Speicher möglichst klein gehalten werden.

Die Menge D^n der hierarchischen Dias wird ähnlich wie bei den Vektorzügen durch die iterative Bildung von Hierarchierelationen über ausgewählte untergeordnete Dias gebildet, wobei mindestens ein Dia aus der Menge D^{n-1} stammen muß. Um die größtmögliche Flexibilität bei der Erzeugung hierarchischer Dias zu gewährleisten, dürfen die Verwendungscalls der Dias neben den mathematisch-geometrischen Transformationsattributen A_{MD} auch die optischen Attribute A_{OD} beinhalten. Die mathematisch-geometrischen Transformationsattribute haben die gleiche Wirkung wie bei den Vektorzügen. Daher wird hier darauf nicht näher eingegangen. Die besondere Problematik bei der Datenstruktur der Dias ergibt sich aus der Interpretation und Auswertung der optischen Attribute. Wegen der neu zugeführten Farbtransparenz-Eigenschaft sind die additiven Mischungsoperationen mit den optischen Attributen zeitsequenzabhängig geworden. Entsprechend den Gleichungen 2.5 haben die auf der Rückseite der gedachten Glasscheibe früher aufzutragenden Farben ein höheres Gewicht für die Bestimmung der resultierenden Farben. Bei der Abwicklung der Datenstruktur der Dias bis zu den elementaren Dias entsteht eine Sequenz von optischen Attributen, die auf verschiedenen Hierarchieebenen spezifiziert worden sind. Eine eindeutige Auswertung dieser Konkatenation von optischen Attributen kann nur dann stattfinden, wenn vorher eine Korrelation zwischen der hierarchischen Spezifizierung des optischen Attributes und seiner (gedachten) Auftragszeit vereinbart wurde. Im Rahmen dieser Arbeit wird festgelegt, daß die hierarchisch höher angegebenen optischen Attribute als die auf der Rückseite der gedachten Glasscheibe früher aufzutragenden Farben betrachtet werden sollen. Dadurch bekommen die hierarchisch höheren optischen Attribute einen größeren Einfluß auf die resultierenden optischen Attribute. Dies bedeutet, daß es durch Spezifizierung des einzigen, aber hierarchisch höher angesiedelten optischen Attributes möglich ist, die optische Erscheinungsform eines ganzen



optischen Begriffes entscheidend zu bestimmen bzw. zu verändern. In Bild 3.3 könnte man durch die Angabe eines nicht neutralen optischen Attributes, beispielsweise $A_0 = \text{GELB}_0$, in dem Verwendungscall für das Dia D_2^1 seine bisherige "BLAU₀/ROT₀"-Optik unmittelbar in GELB_0 verändern.

In Bild 3.3 sind wegen der beabsichtigten Überschaubarkeit der aus technischen Gründen nur einfarbig erfolgten graphischen Darstellung alle hierarchischen optischen Attribute neutral (NEUTR.) spezifiziert worden. Eine Veränderung der hier schon auf der D^0 -Ebene festgelegten optischen Attribute kann in diesem Falle nur durch die geometrische Oberlagerung von Komponenten in den hierarchischen Dias stattfinden. Da bei der Oberlagerung die Veränderung der Optik nicht durch die Spezifizierung in der Datenstruktur, sondern durch eine additive Mischung mit den optischen Attributen der später abgewickelten Komponenten stattfindet, haben die optischen Attribute dieser zuletzt abgewickelten Komponenten den überwiegenden Einfluß auf die resultierende Optik. Bei der geometrischen Oberlagerung von Dias geht man also davon aus, daß die optischen Attribute der Dias auf der Vorderseite der gedachten Glasscheibe aufgetragen werden. In Bild 3.3, wo die Abwicklung der Komponenten durch ihre Reihenfolge in dem Komponentenattribut (K) vorgegeben ist, überlagert zum Beispiel in dem Dia D_1^1 das "GRÜN₀"-Dachfenster D_3^0 das "ROT₀/WEISS₀"-Dach D_2^0 . Die Optik des Dachfensters als zuletzt abgewickelte Komponente "überschreibt" einfach die bisher vorhandene Optik des Daches, so daß die resultierende Farbe "GRÜN₀" ist. Solche Veränderungen der Optik durch Oberlagerung werden in der Datenstruktur nur implizit durch die mathematisch-geometrischen Attribute A_{MD} spezifiziert. Dadurch stellen diese Optikveränderungen, je nachdem ob die Oberlagerung beabsichtigt ist oder nicht, eine zusätzliche Optikgestaltungsmöglichkeit bzw. einen nicht erwünschten störenden Effekt dar. Da in den meisten Fällen eine Oberlagerung beabsichtigt wird, kann man die dadurch hervorgerufene Optikveränderung als zusätzliche Gestaltungsmöglichkeit betrachten, wobei dann die Reihenfolge bei der Abwicklung der Komponenten für die resultierende Optik ausschlaggebend ist.

3.2.1.4. Datenstruktur der Kacheln

Mit der bisher beschriebenen Datenstruktur der Vektorzüge, Flächen und Dias können beliebig komplexe logische Bilder modelliert werden. Deren Abbildung in dem Weltbild soll jetzt näher betrachtet werden.

Entsprechend dem in Abschnitt 2.3 eingeführten Konzept der Kachelstruktur des Weltbildes kann diese Abbildung nur kachelweise geschehen. Dies bedeutet,

daß eine beliebige Teilmenge von Dias nur innerhalb einer Kachel des Weltbildes projiziert werden darf. Dafür wird das Weltkoordinatensystem KS_W , wie in Bild 3.4 veranschaulicht, matrixförmig in Kacheln eingeteilt, wobei hier der Einfachheit halber nur vier Kacheln für jede Achse dargestellt sind. Die Anzahl der Kacheln wird so gewählt, daß die Auflösung des Kachelkoordinatensystems (KS_K) identisch mit der des Bildschirmkoordinatensystems (KS_B) ist. Das Weltbild wird nicht mehr integral im KS_W , sondern kachelweise im KS_K codiert. Damit erhalten das logische und das optische Bild Koordinatensysteme mit gleicher Auflösung, wodurch die Domäne-Transformationen prinzipiell überflüssig werden. Da eine Kachel gewöhnlich ein komplexes Dia mit vorgegebener Position im Weltbild repräsentiert, kann die Datenstruktur der Kachel als die höchste Hierarchieebene von Dias aufgefaßt werden. Genau wie bei den Dias beinhalten die Ver-

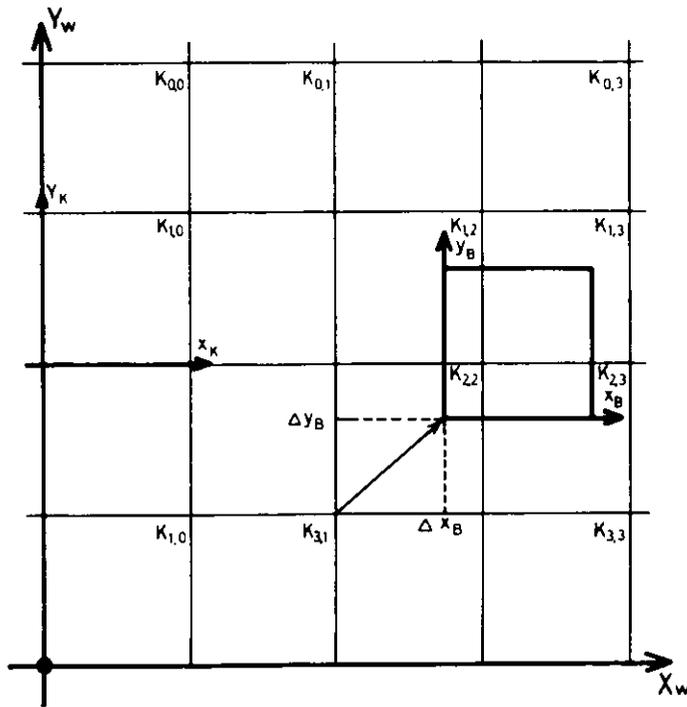


Bild 3.4 Kachelstruktur des Weltbildes

wendungscalls sowohl mathematisch-geometrische Transformationsattribute A_{MK} als auch optische Attribute A_{OK} , die die vollständige Gestaltung der Kachel bei der Modellierung logischer Bilder ermöglichen.

Durch die Erteilung eines Bildauftrags muß, wie in Bild 3.4 verdeutlicht, ein Ausschnitt der Bildschirmgröße aus dem Weltbild dargestellt werden. Diese Abbildung stellt eine Window-Transformation

$$W : (X_W \times Y_W) \rightarrow (X_B \times Y_B) \quad (3.5)$$

dar, die wegen der konstanten Fenstergröße nur Teile von maximal vier benachbarten Kacheln abzubilden vermag. Da bei dieser Window-Transformation der abzubildende Bereich den doppelten Koordinatenwertebereich des Bildschirmfensters enthält, kann die Überprüfung der Zugehörigkeit einzelner Bildpunkte zum Bildschirmfenster im Gegensatz zu den in Abschnitt 1.2.1 beschriebenen Windowing-Algorithmen erheblich vereinfacht werden. Dies wird mit einem einfachen Beispiel verdeutlicht. In Bild 3.5 wird gezeigt, daß für die gewünschte Abbildung der

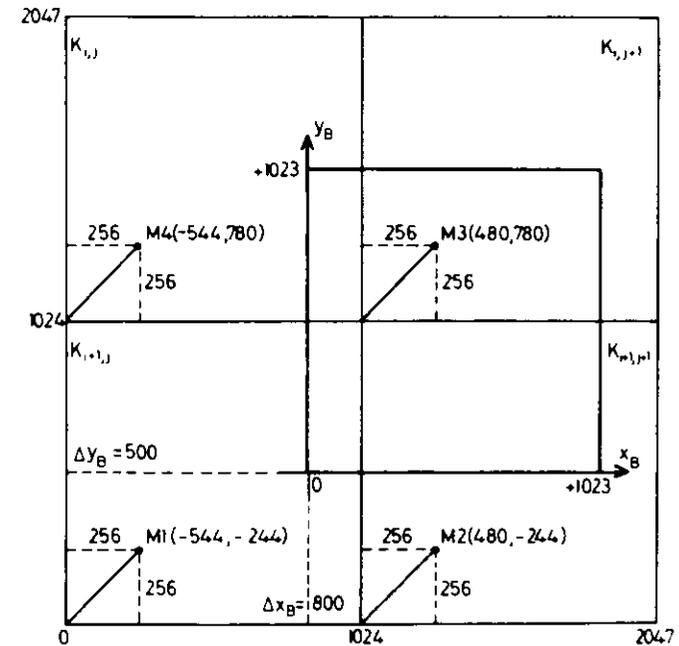


Bild 3.5 Window-Transformation der Kacheln

Kacheln auf dem Bildschirm ausschließlich Translationstransformationen dieser Kacheln erforderlich sind. Die jeweiligen Translationsbeträge können wegen der vorgegebenen relativen Position der Kacheln zueinander aus der durch $(\Delta X_B, \Delta Y_B)$ definierten Position des Bildschirmkoordinatensystems KS_B relativ zu einer der Kacheln abgeleitet werden. Nach der Ausführung dieser Translationstransformationen können nur die Bildpunkte als zum Bildschirmfenster gehörend betrachtet werden die für beide Koordinaten X und Y positive Werte haben:

$$0 \leq X, Y \leq 1023 \quad (3.6)$$

Dies wird in Bild 3.5 am Beispiel der vier Bildpunkte M1 bis M4 gezeigt. Das Vorzeichen kann durch die Verdoppelung der Rechengenauigkeit für die Koordinatenwerte erreicht werden, was lediglich die Einführung jeweils eines zusätzlichen Bit - x_{10} und y_{10} - für die beiden Koordinatenwerte X und Y bedeutet. Die Zugehörigkeit zum Bildschirmfenster ist damit durch ein binäres Kriterium

$$\begin{aligned} x_{10} \cdot y_{10} &= 0 \rightarrow \text{Bildpunkt im Bildschirmfenster} \\ x_{10} \vee y_{10} &= 1 \rightarrow \text{Bildpunkt außerhalb des Bildschirmfensters} \end{aligned} \quad (3.7)$$

entscheidbar. Die Kachelstruktur des Weltbildes hat also die Domain-Transformationen nicht vollständig eliminieren können, sie aber so erheblich vereinfacht, daß sie leicht realisierbar sind.

Die Datenstruktur der Bildmenge B ist in Bild 3.6 gezeigt. Das Komponentenattribut eines Bildes kann Verwendungscalls auf maximal vier benachbarten Kacheln spezifizieren. Die Translationsbeträge für die Window-Transformation sind in Form eines 4-Tupels in den Verwendungscalls angegeben und können als absolute mathematisch-geometrische Transformationsattribute A_{MB}^{ABS} angesehen werden. An dieser Stelle muß noch bemerkt werden, daß bei der üblichen Window-Transformation (s. Abschnitt 1.2.1) durch die Wahl der Fenstergröße zusätzlich ein Skalierungseffekt des Bildes erzielt werden kann, wofür diese Transformation - unrechtmäßigerweise - oft zweckentfremdet wird. Da es sich in diesem Falle nicht um eine Veränderung der Bildauflösung, sondern um eine reine Bildskalierung handelt, kann dieser Effekt mit der hier dargestellten Datenstruktur der Kacheln nur durch zusätzliche Skalierungseinrichtungen (Zoom oder Pan) in der Bilddarstellungsinstantz realisiert werden.

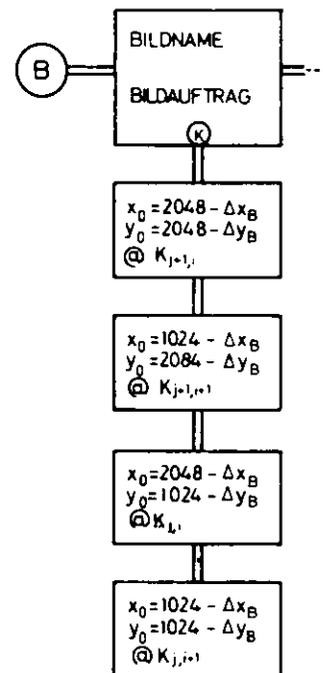


Bild 3.6 Datenstruktur des Bildes

Abschließend ist für die Beschreibung der Datenstruktur der Graphik in Bild 3.7 die gesamte Datenstruktur des verwendeten Beispiels gezeigt. Dieses Bild stellt die Gesamtheit aller verknüpften Hierarchierelationen dar und dokumentiert anschaulich die Effizienz und die Transparenz des gesamten Datenstrukturmodells der Graphik.

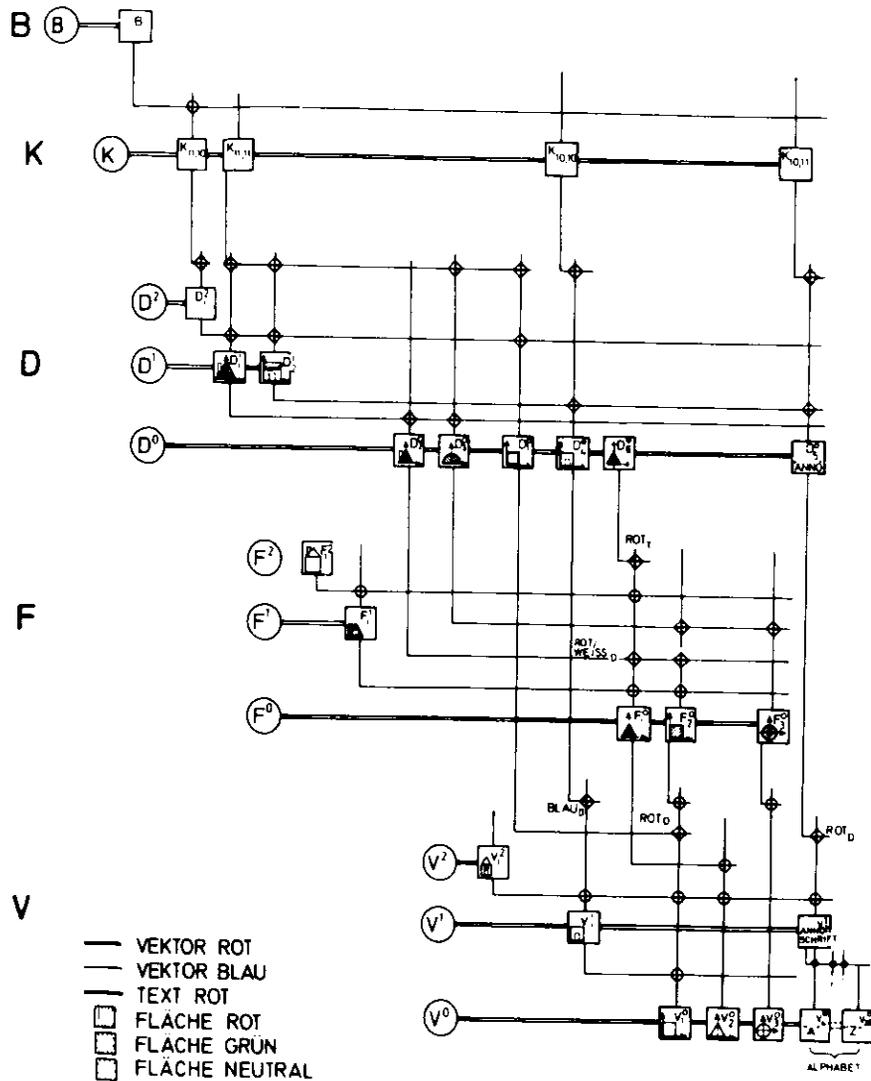


Bild 3.7 Gesamte Datenstruktur der Graphik

3.2.2. Datenstruktur der Bildinformation

Wie in Kapitel 1 schon angedeutet, ist ein Bildmodell für die logische Beschreibung der reellen Weltbilder wegen der verschiedenartigen Datenobjekte und deren extrem komplexen Strukturrelationen z. Zt. nicht vorhanden und in absehbarer Zukunft auch nicht zu erwarten /SLOAN-77, McKLOWN-77/.

Der heutige Zustand der logischen Abbildbeschreibung beschränkt sich vorwiegend auf das Gebiet der Bildsegmentierung, wobei bis jetzt nur einfache Objekte - beispielsweise Kanten und Texturen - aus einem Bild heraus erfolgreich extrahiert werden können /FU-76/. Obwohl in der letzten Zeit im Bereich der Mustererkennung und der künstlichen Intelligenz neue leistungsfähigere Werkzeuge - wie z. B. relationelle Grammatiken und symbolische Repräsentationsmethoden - eingeführt wurden, ist man von einer befriedigenden Problemlösung noch weit entfernt /ANDREWS-79/.

Da offensichtlich die logische Struktur der Bilder zu komplex ist, um mit einem Modell beschrieben werden zu können, wird im Rahmen dieser Arbeit nicht die logische, sondern die technische Struktur der Bilder für die Modellbildung betrachtet. Diese technische Struktur, die durch den Bildpunktgenerierungsprozeß vorgegeben wird, ist in der Regel einfach genug, um mit einem hierarchischen Datenstrukturmodell beschrieben zu werden.

Wegen ihrer großen Verbreitung sind in dieser Arbeit nur die Videobilder behandelt. Der zeilenweise Abtastungs- bzw. Generierungsvorgang der Videobilder ermöglicht die Betrachtung der Raster-Scan-Bilder als Datenobjekte mit zwei Hierarchieebenen: die Bildzeilen und auf nächster Ebene die Bildpunkte. Diese einfache Modellvorstellung für die Datenstruktur der Videobilder ermöglicht jedoch eine einheitliche Behandlung der Graphik und Abbildinformation innerhalb der Display-Systeme

3.3. Codierungsvorschlag für die graphischen Datenobjekte

Ein Codierungsvorschlag für die graphischen Datenobjekte muß die wesentlichen Aspekte der maschinenspezifischen Organisationsform für die in der vorangegangenen Diskussion beschriebene Datenstruktur beinhalten. Sicherlich muß die Codierung die vollständige Information bezüglich der Attribut- und Strukturrelationen der Datenstruktur enthalten. Bei der Codierung der Datenobjekte ist

man besonders darauf bedacht, für die verwendete Maschine eine optimale Organisationsform hinsichtlich des Speicheraufwandes, der Zugriffszeit und der notwendigen Abwicklungsalgorithmen zu finden.

Alle in dem Datengitter aus Bild 3.7 vorhandenen Datenobjekte (dort graphisch als Quadrate mit eingetragenen Symbolen dargestellt) müssen im Speicher der Maschine als ein zusammenhängender Block vertreten sein. Ein prinzipieller Codierungsvorschlag für die graphischen Datenobjekte aus den Mengen V, F, D, K, B , der nur die logische, nicht aber die physikalische Organisationsform des logischen Bildes darstellt, ist in den Tabellen 3.1 bis 3.10 veranschaulicht. Demnach beinhaltet der Codierungsblock Information bezüglich des Typs, des Namens, der Hierarchierelation und des zugelassenen Satzes von Attributen jedes Datenobjektes. Obwohl die Bedeutung der meisten in den Tabellen 3.1 bis 3.10 auftretenden Bezeichnungen durch die vorangegangene Modelldarstellung schon geklärt wurde, bedarf das bessere Verständnis der Codierungsaspekte folgender Bemerkungen:

Ein wichtiges Problem stellt die Codierung der Hierarchierelationen dar, die in erster Linie die Zugriffsreihenfolge auf die Datenobjekte der Netzhierarchie festlegen will und daher ohne die Datenobjektmenge selbst nur durch Verweise auf diese Datenobjekte verkettet werden kann. Für diesen Zweck eignet sich die explizite Doppelring-Verkettung, die auch weitgehend verwendet wird /GIL01-78/. Im vorliegenden Codierungsvorschlag wurde mit Rücksicht einerseits auf die umfangreichen, beispielsweise durch die Bildmanipulation auftretenden Veränderungen der Datenobjekte und andererseits auf eine adressunabhängige Gestaltung der Datenstruktur die assoziative Verkettung für die Verweiscodierung gewählt. Die als Subroutinencalls implementierten Verweise (CALL- und RETURN-Aufrufe) gewährleisten eine assoziative Doppelring-Verkettung der Datenobjekte, die sich für den Anwender einfach und gleichzeitig flexible erweist.

Die mathematisch-geometrischen Transformationsattribute A_M beinhalten zusätzlich zu den Angaben für die relative Position, Rotation und den Skalierungsfaktor auch die Angabe eines wählbaren Referenzpunktes für die spezifizierte Positionierung: Als Referenzpunkt kann entweder der Ursprung des aktuellen Koordinatensystems - hier "ORIGIN POSITION" genannt - oder die bei der Abwicklung des letzten Datenobjektes zuletzt erzeugte Position - hier "AKTUELLE POSITION" genannt - verwendet werden. Die Frage nach der zweckmäßigen Verwendung der beiden Positionierungsmöglichkeiten ist in Bild 3.8 verdeutlicht.

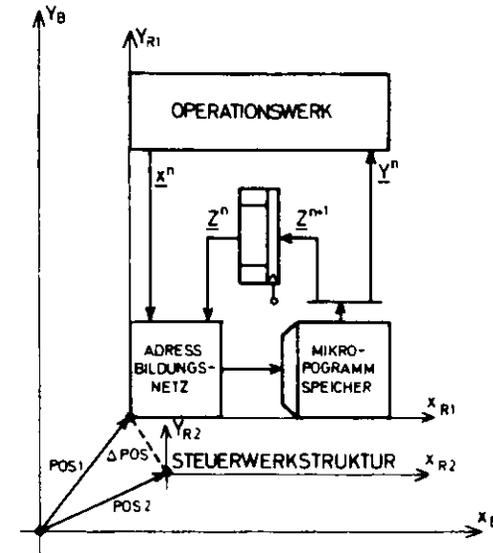


Bild 3.8 Zur Veranschaulichung der aktuellen Position als Referenzpunkt

Das hier dargestellte Bild setzt sich aus einem Bild- und einem Textteil zusammen, die sich jeweils auf die relativen Koordinatensysteme $(x_{R1} \times y_{R1})$ bzw. $(x_{R2} \times y_{R2})$ beziehen. Der Einfachheit halber sei angenommen, daß die letzte Zeichnungsposition der Teilbilder jeweils in dem Ursprung ihrer relativen Koordinatensysteme liegt. Bei der Spezifizierung der "origin Position" für die Bildunterschrift - als Textteil - ist es notwendig, aus dem Positionierungsbetrag $POS1$ des Bildteils den Positionierungsbetrag $POS2$ für die Bildunterschrift jedesmal zu berechnen, so daß die gewünschte relative Position der beiden Teile erreicht wird. Bei der Verwendung der "aktuellen Position" dagegen braucht man nur den relativen Positionierungsbetrag ΔPOS anzugeben, um die Bildunterschrift bei jeglichen Positionsveränderungen des Bildteils immer richtig positionieren zu können, ohne ihre tatsächliche Position jedesmal ausrechnen zu müssen.

TABELLE 3.1

DATENOBJEKT := V^0	V
TYP := ELEMENTARER VEKTORZUG/CHARACTER	
NAME := "POLYLINE"	
EXEC A_{GV}	
A_{GV} := GRAPHISCHES ATTRIBUT	
begin $\sum_{i=1}^n (\pm \Delta x_i, \pm \Delta y_i, h_i)$ end	
RETURN	

TABELLE 3.2

DATENOBJEKT := V^n	W			
TYP := HIERARCHISCHER VEKTORZUG				
NAME := "COMPLEX POLYLINE"				
CALL $V_k^{n-1} [A_{MV}, A_{FV}]$				
A_{MV}	POS _R	ROT _R	SF _R	ORIGIN POS AKTUELLE POS
A_{FV}	VEKTOR STIL	FREI	ZUGRIFFSRECHT CODE	MARKIERUNG EMPFINDL
ⓐ	V_k^{n-1}			
CALL $V_j^{n-3} [A_{MV}, A_{FV}]$				
A_{MV}	POS _R	ROT _R	SF _R	ORIGIN POS AKTUELLE POS
A_{FV}	VEKTOR STIL	FREI	ZUGRIFFSRECHT CODE	MARKIERUNG EMPFINDL
ⓐ	V_k^{n-3}			
RETURN				

TABELLE 3.5

DATENOBJEKT := F^n	F			
TYP := HIERARCHISCHE FLÄCHE				
NAME := "COMPLEX AREA"				
CALL $G_k^{n-1} [M_F, A_{FF}]$				
M_F	$F(G) := [0110...1]$			
A_{FF}	—	—	ZUGRIFFSRECHT CODE	MARKIERUNG EMPFINDL
ⓐ	G_k^{n-1}			
RETURN				

TABELLE 3.6

DATENOBJEKT := D^0	D			
TYP := ELEMENTARES DIA				
NAME := "PICTURE"				
CALL $V_k^0 / F^0 [A_{DD}, A_{FD}]$				
A_{DD}	ROT	GRÜN	BLAU	FT = D, T
A_{FD}	—	—	ZUGRIFFSRECHT CODE	MARKIERUNG EMPFINDL
ⓐ	V_k^0 / F^0			
RETURN				

TABELLE 3.3

DATENOBJEKT := F^0	IF			
TYP := ELEMENTARE FLÄCHE				
NAME := "AREA"				
CALL $V_k^0 [A_{MF}, A_{FF}]$				
A_{MF}	POS _R	ROT _R	SF _R	ORIGIN POS AKTUELLE POS
A_{FF}	FLÄCHE STIL	FREI	ZUGRIFFSRECHT CODE	MARKIERUNG EMPFINDL
ⓐ	V_k^0			
RETURN				

TABELLE 3.4

DATENOBJEKT := G^n	G			
TYP := FLÄCHENGRUPPE				
NAME := "AREAS GROUP"				
CALL $F_k^n [A_{MF}]$				
A_{MF}	POS _R	ROT _R	SF _R	ORIGIN POS AKTUELLE POS
ⓐ	F_k^n			
CALL $F_j^{n-2} [A_{MF}]$				
A_{MF}	POS _R	ROT _R	SF _R	ORIGIN POS AKTUELLE POS
ⓐ	F_j^{n-2}			
RETURN				

TABELLE 3.7

DATENOBJEKT := D^0	D			
TYP := ELEMENTARES DIA				
NAME := "PICTURE"				
CALL $G_k^0 [M_F, A_{DD}, A_{FD}]$				
M_F	$F(G) := [0110...1]$			
A_{DD}^0	ROT	GRÜN	BLAU	FT = D, T
	:	:	:	:
A_{DD}^{2k}	ROT	GRÜN	BLAU	FT = D, T
A_{FD}	—	—	ZUGRIFFSRECHT CODE	MARKIERUNG EMPFINDL
ⓐ	G_k^0			
RETURN				

TABELLE 3.8

DATENOBJEKT := D^n	D			
TYP := HIERARCHISCHES DIA				
NAME := "COMPLEX PICTURE"				
CALL $D_k^{n-1} [A_{DD}, A_{DD}, A_{FD}]$				
A_{DD}	POS _R	ROT _R	SF _R	ORIGIN POS AKTUELLE POS
A_{DD}	ROT	GRÜN	BLAU	FT = D, T
A_{FD}	—	—	ZUGRIFFSRECHT CODE	MARKIERUNG EMPFINDL
ⓐ	D_k^{n-1}			
RETURN				

TABELLE 3.9

DATENOBJEKT: = $K_{j,i}$					K
TYP: = KACHEL					
NAME: = KACHEL $\approx jk$					
CALL $D_j^n [A_{nk}, A_{ok}]$					
A_{nk}	POS_k	ROT_k	SF_k	ORIGIN POS	
A_{ok}	ROT	GRÜN	BLAU	FT: = 0,1	
Ⓞ	D_j^n				
RETURN					

TABELLE 3.10

DATENOBJEKT: = B					B
TYP: = BILD					
NAME: = "ACTUAL IMAGE"					
EXEC RESET					
EXEC START					
CALL $K_{j,i} [A_{MB}^{ABS}]$					
A_{MB}^{ABS}	POS_A^1	$ROT_A=0$	$SF_A=1$	ORIGIN POS	
Ⓞ	$K_{j,i}$				
CALL $K_{j,i-1} [A_{MB}^{ABS}]$					
A_{MB}^{ABS}	POS_A^2	$ROT_A=0$	$SF_A=1$	ORIGIN POS	
Ⓞ	$K_{j,i-1}$				
CALL $K_{j-1,i} [A_{MB}^{ABS}]$					
A_{MB}^{ABS}	POS_A^3	$ROT_A=0$	$SF_A=1$	ORIGIN POS	
Ⓞ	$K_{j-1,i}$				
CALL $K_{j-1,i-1} [A_{MB}^{ABS}]$					
A_{MB}^{ABS}	POS_A^4	$ROT_A=0$	$SF_A=1$	ORIGIN POS	
Ⓞ	$K_{j-1,i-1}$				
RETURN					
EXEC STOP					

Der Codierungsvorschlag beinhaltet auch die freien Attribute A_F , die bei der Modelldarstellung der Übersichtlichkeit wegen nicht behandelt worden sind. Obwohl im Zusammenhang mit der Semantik der dargestellten Begriffe den Datenobjekten beliebig viele verschiedene freie Attribute zugeordnet werden könnten, beschränkt sich der hier verwendete Satz freier Attribute auf drei, von der Semantik unabhängige Aspekte der graphischen Datenobjekte:

Das Stil-Attribut spezifiziert das Muster, nach dem das Datenobjekt erzeugt werden soll. Zum Beispiel gibt der Vektorstil das gewünschte Vektormuster (durchgezogen, gepunktet) an. Flächen als zweidimensionale Datenobjekte benötigen auch zweidimensionale Muster, beispielsweise ein quadratisches Schraffierungsmuster für eine Fläche.

Das Zugriffsrechtsattribut spezifiziert in Verbindung mit einem dem Anwender zugewiesenen Identifizierungscode, ob das Datenobjekt als auflösbarer optischer Begriff erscheinen darf oder nicht. Damit ist es möglich, das gleiche logische Bild unterschiedlichen Anwendern mit unterschiedlichen Begriffsaufösungen anzubieten und somit eine Art Bilddatenschutz zu gewährleisten.

Das Markierungs-Empfindlichkeits-Attribut spezifiziert, ob das Datenobjekt bei einer Markierung identifizierbar sein soll oder nicht. Diese binäre Eigenschaft bestimmt die unterste hierarchische Begriffsebene, auf der die Begriffe noch identifizierbar und somit manipulierbar sind.

Diese verschiedenen freien Attribute, die auf verschiedenen Hierarchieebenen angegeben werden dürfen, müssen - wie auch alle anderen - bei der Abwicklung der Datenstruktur zu einem resultierenden Attribut ausgewertet werden. Die Auswertungsregeln werden in Abschnitt 4.1.2.3 unter Berücksichtigung der benötigten informationellen Strukturen detailliert behandelt.

4 Informationelle Strukturen

Nach der vorangegangenen Festlegung der Datenstruktur kann nun mit der Beschreibung der informationellen Strukturen zur Erzeugung und Darstellung der logischen Bilder begonnen werden. Die im folgenden dargestellte Struktur eines komplexen Raster-Scan-Display-Systems sollte als Ergebnis eines in iterativen Schritten durchgeführten Entwurfsprozesses angesehen werden. Kennzeichnend für diesen Entwurfsprozeß ist seine Struktur-Transparenz, die nur durch die konsequente Verwendung anschaulicher, aber mächtiger Darstellungsmittel erzielt werden konnte. Die transparente Darstellung sowohl des Datenflusses /KELLER-78, WENDT-79-2/ als auch des Aktionsablaufes /PETRI-62, WENDT-74-2/ hat dem Entwurfenden die Möglichkeit eröffnet, einen ursprünglichen Entwurfsvorschlag objektiv zu analysieren und infolgedessen mit formalen oder heuristischen Mitteln optimieren zu können.

Gegenstand dieses Kapitels sind die informationellen Strukturen der Bilddarstellungs- und Bilderfassungsinstanz. Die informationelle Struktur der Bilddarstellungsinstanz ist so detailliert beschrieben, daß sie unmittelbar als Modell für die anschließende Implementierung verwendet werden kann (siehe Kapitel 5). Die Präsentation der informationellen Strukturen, die gleichzeitig den durchgeführten Entwurfsprozeß widerspiegelt, findet grundsätzlich "top-down" statt, d. h. von hierarchisch höheren zu niedrigeren Abstraktionsebenen. Die wesentlichen Entwurfsentscheidungen können somit auf der jeweiligen Betrachtungsebene klargemacht werden.

Den Ausgangspunkt dieser Beschreibung stellt die in Bild 4.1 gezeigte Grundstruktur des Display-Systems dar. Es ist eine verfeinerte Darstellung der Struktur in Bild 1.2 und wird dem Leser auf Grund der bisherigen Diskussion im wesentlichen verständlich sein.

4.1. Bilddarstellungsinstanz

Die Bilddarstellungsinstanz ist zweckmäßigerweise in mehrere spezialisierte Instanzen zerlegt, wodurch die wesentlichen Prozesse bei der Bilddarstellung in dedizierter und daher effizienter Weise unterstützt werden können. Diese Prozesse, die prinzipiell schon in Abschnitt 1.2.1. beschrieben wurden, sind: Datenstrukturabwicklung, Bildtransformation, Bildpunktgenerierung und Bildpunktdarstellung. Der Datenfluß in Bild 4.1 zeigt die vorgeschlagene Struktur der Bild-

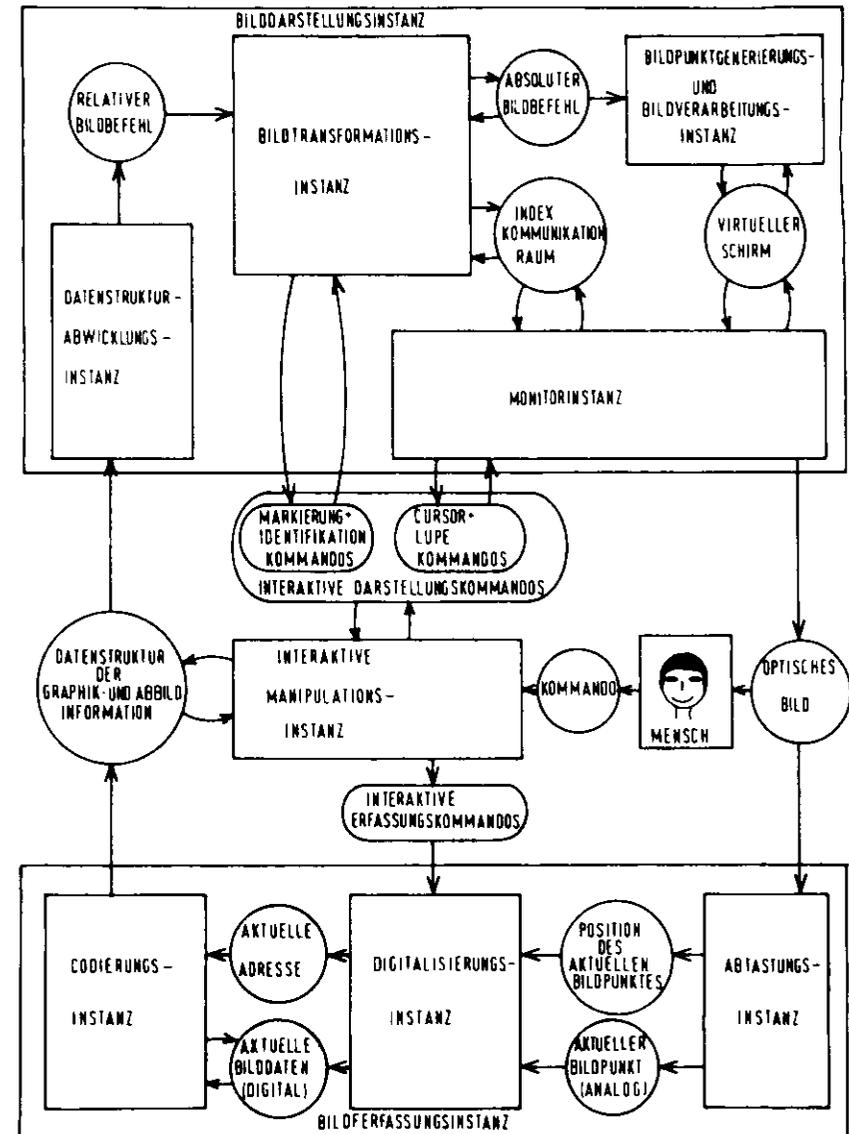


Bild 4.1 Globale informationelle Struktur des Raster-Scan-Display-Systems

darstellungsinstanz, wobei hier weniger die interne Struktur der verschiedenen Instanzen, sondern vielmehr deren Schnittstellen mit der Umwelt und deren Positionierung im Datenfluß veranschaulicht werden sollen. Von wesentlicher Bedeutung für die Transparenz der Bildtransformationen ist - im Gegensatz zu Bild 1.4 - die Anordnung der Bildtransformationsinstanz vor der Bildpunktgenerierungsinstanz. Dies bedeutet, daß die mathematisch-geometrischen Bildtransformationen nicht mehr für einzelne Bildpunkte, sondern für ganze Bildelemente gleichzeitig ausgeführt werden. Damit können die gewöhnlich hohen Durchsatzanforderungen für die gesamte Bildtransformationsinstanz erheblich reduziert werden.

Da die Wirkungsweise einer Instanz am besten durch die Bedeutung und den Inhalt der benachbarten Zustandsräume beschreibbar ist, wird hier zunächst die funktionale Bedeutung der verschiedenen Zustandsräume innerhalb der Bilddarstellungsinstanz erläutert.

Aus der Datenstruktur der Graphik- und Abbildinformation wird durch die Datenstrukturabwicklungsinstanz die Folge der sogenannten relativen Bildbefehle erzeugt. Diese Bildbefehle werden deshalb "relative Bildbefehle" genannt, weil sie nur mittelbar auf das optische Bild wirken und ihre Wirkung nur im Zusammenhang mit dem aktuellen Bildstatus betrachtet werden kann. Die Tabelle 4.1 zeigt, daß die relativen Bildbefehle entsprechend ihrer Funktion in drei Klassen eingeteilt werden können:

- Die Kontroll-Befehle (EXEC CNTR) beinhalten Anweisungen, die entweder zur Initialisierung des Darstellungsprozesses (RESET) oder zur Manipulation des aktuellen Bildstatus (CALL, RETURN) entsprechend der gültigen Hierarchieebene in der Datenstruktur dienen.
- Die Attribut-Befehle (EXEC ATTR) beinhalten Anweisungen, die die Veränderung des Bildstatus entsprechend den verschiedenen Attributsorten bewirken.
- Die Bildelement-Befehle (EXEC MBEF) beinhalten Anweisungen, die die mathematisch-geometrische Transformation für ganze Bildelemente spezifizieren. Als Bildelemente werden hier zwecks einheitlicher Durchführung der mathematisch-geometrischen Transformationen nur elementare Vektoren zugelassen. Das setzt jedoch voraus, daß der Character-Generator, der das Vektorzugmuster der alphanumerischen Symbole tabellarisch erzeugt, schon in der Datenstrukturabwicklungsinstanz vorhanden sein muß.

Tabelle 4.1 Relative Bildbefehle

Befehlsklasse	Befehl	Bedeutung
Kontroll-Befehle (EXEC CNTR)	RESET	Allgemeine Grundstellung des Systems
	CALL	Wechsel auf eine tiefere Hierarchieebene
	RETURN	Wechsel auf eine höhere Hierarchieebene
Attribut-Befehle (EXEC ATTR)	EXEC A _M	Berechne neuen mathematisch-geometrischen Status (STATUS _M)
	EXEC A _O	Berechne neuen optischen Status (STATUS _O)
	EXEC A _F	Berechne neuen freien Attributen-Status (STATUS _F)
	EXEC BIT	Berechne die Begriff/Index-Transformation (BIT)
Bildelement-Transformations-Befehle (EXEC MBEF)	EXEC M	Transformiere mathematisch-geometrisch des Vektor (Δx, Δy) eines Vektorzuges

Die Bildtransformationsinstanz verwendet die relativen Bildbefehle, um zwei neue, semantisch verschiedene Zustandsräume inhaltlich zu erzeugen.

Der erste Zustandsraum enthält die sogenannten absoluten Bildbefehle. Sie stellen im Gegensatz zu den relativen Bildbefehlen unmittelbare Anweisungen zur Erzeugung oder Verarbeitung des logischen Indexbildes im virtuellen Schirm dar. Die absoluten Bildbefehle werden von der Bildpunktgenerierungs- und Bildverarbeitungsinstanz ausgeführt, die als "virtueller Zeichner" den virtuellen

Schirm in ausgewählten Positionen über eine in Bild 4.1 nicht gezeigte Zugriffsinstanz beschreibt. Die Tabelle 4.2 zeigt, daß die absoluten Bildbefehle entsprechend ihrer Funktion drei Klassen angehören können:

Tabelle 4.2 Absolute Bildbefehle

Befehlsklasse	Befehl	Bedeutung
Kontroll-Befehle (EXEC CNTA)	RESET	Allgemeine Grundstellung des Systems
	RESET FLÄCHEN- FENSTER	Grundstellung des Flächenarbeitsfensters beim Übergang F → V in der Datenstruktur
Bildelement-Befehle (EXEC OP)	EXEC VEC	Zeichne Vektor (Δx,Δy) ausgehend von der aktuellen Zeichnungsposition
	EXEC AREA	Fülle die Fläche des letzten Vektorzuges beim Übergang V → F in der Datenstruktur
Bildverarbeitungs-Befehle (EXEC VER)	EXEC BEF	Führe einen beliebigen Bildverarbeitungsbehl aus (z. B. Bild glätten)

- Die Kontroll-Befehle (EXEC CNTA) stellen lediglich Initialisierungsanweisungen bei der Generierung des logischen Indexbildes dar.
- Die Bildelement-Befehle (EXEC OP) beinhalten Anweisungen für die Bildpunktgenerierung von Bildelementen, die hier nur Vektoren (EXEC VEC) oder Flächen (EXEC AREA) sein dürfen. Diese Bildpunktgenerierung bedeutet entweder das Setzen oder das Modifizieren von logischen Indexwerten im virtuellen Schirm für

die Punktmenge, die aus der Definition des Bildelementes resultiert. Formell ist die Bildpunktgenerierung durch die Gleichung

$$p^{n+1} = f(\text{BEF}, P)^n \quad (4.1)$$

mit P := Punktmenge des Bildelements,

BEF := Bildelementbefehl,

f := modifizierende Funktion für die logischen Indexwerte,

n := diskrete Zeit

beschrieben, wobei f entweder "SET INDEX" oder "ADD INDEX" sein kann.

- Die Bildverarbeitungsbehle (EXEC VER) beinhalten Anweisungen zur allgemeinen Verarbeitung des Bildes im virtuellen Schirm. Diese Befehle schreiben die Transformationsregel für jeden Bildpunkt einer ausgewählten Bildpunktmenge in Abhängigkeit von seiner Nachbarschaft, die i. a. der ganze virtuelle Schirm ist, vor. Formell ist die Bildverarbeitung durch die Gleichung

$$p^{n+1} = f(\text{BEF}, \text{VS})^n \quad (4.2)$$

mit P := ausgewählte Menge des virtuellen Schirms (P ∈ VS),

BEF := Bildverarbeitungsbehl,

VS := virtueller Schirm,

f := modifizierende Funktion für die Bildpunkte im virtuellen Schirm,

n := diskrete Zeit

beschrieben, wobei hier f eine freiprogrammierbare Funktion sein darf.

Ein absoluter Bildbefehl muß also folgende Informationsanteile beinhalten:

- Operationscode des Befehls (z. B. EXEC VEC),
- Spezifikation der ausgewählten Punktmenge (z. B. den Anfangspunkt (x_0, y_0) und den Vektor $(\Delta x, \Delta y)$),

- Index-Operand als modifizierende Funktion (z. B. SET INDEX),
- sonstige Verarbeitungsangaben (z. B. vektor Stil).

Da die einzelnen Komponenten des aktuellen Bildbefehls von verschiedenen Instanzen in der Bildtransformationsinstanz geliefert werden, bedeutet das jeweils einen modifizierenden Zugriff auf den Zustandsraum "absoluter Bildbefehl". (siehe Bild 4.1).

Der zweite Zustandsraum, dessen Inhalt hauptsächlich von der Bildtransformationsinstanz erzeugt wird, ist der Index-Kommunikationsraum. Er beinhaltet die gesamte Information bezüglich der optischen Erscheinungsform des logischen Indexbildes, die sich nach der Durchführung der Begriff-Index-Transformation eindeutig ergibt (siehe Abschnitt 2.1). Die gesamte Index-Optik-Tabelle, die die optische Erscheinung jedes logischen Indexwertes beinhaltet, stellt beispielsweise einen wesentlichen Teil des Index-Kommunikationsraumes dar. Die vollständige Struktur des Index-Kommunikationsraumes, die auch im Zusammenhang mit den Indexkompressionsalgorithmen steht, ist in Abschnitt 4.2.2. beschrieben.

Die Monitorinstanz verwendet einerseits das logische Indexbild aus dem virtuellen Schirm und andererseits die Information aus dem Index-Kommunikationsraum, um nach der dynamischen Index-Optik-Zuordnung das optische Bild darzustellen. Darüber hinaus wird die Monitorinstanz wegen ihres parallelen Zugriffes zum virtuellen Schirm eingesetzt, um die notwendige Index-Kompression im virtuellen Schirm zeitsparend durchzuführen.

Der bisher nicht erwähnte Zustandsraum für die interaktiven Darstellungs-kommandos in Bild 4.1 dient ausschließlich der Kommunikation zwischen der Bilddarstellungs- und der Bildmanipulationsinstanz. Die interaktiven Darstellungs-kommandos, mit Hilfe derer eine anwendungsunabhängige Bildmanipulation durchgeführt werden soll, können entsprechend ihrer Funktion in zwei Kategorien unterteilt werden, die in Bild 4.1 durch die Einteilung des gesamten Kommunikationsraumes in zwei Teilräume angedeutet wurde:

Die Cursor- und Lupe-Kommandos sollen sämtliche Manipulationen des Cursors und des Bildformates in einer interaktiven Weise ermöglichen. Sie werden zweckmäßigerweise direkt an die Monitorinstanz weitergeleitet, die sowohl den Cursor als auch den Lupe-Status unmittelbar kontrollieren kann.

Die Markierungs- und Identifikations-Kommandos müssen sämtliche Wünsche bezüglich der Markierung und/oder der Identifikation aller unmittelbaren oder mittelbaren Begriffe aus der Datenstruktur ermöglichen. Diese Kommandos können sinnvollerweise nur an die Bildtransformationsinstanz weitergeleitet werden, die während der Durchführung der Begriff-Index-Transformation über sämtliche Informationen bezüglich der Begriffe und der logischen Indexwerte verfügt.

Da die interaktive Kommunikation in Form eines Dialogs zwischen den Instanzen stattfindet, muß der Zugriff der beteiligten Instanzen auf die entsprechenden (Teil-)Räume immer modifizierend sein.

Bezugnehmend auf diese globale Strukturdarstellung wird nun im folgenden die detaillierte Struktur der vier dedizierten Instanzen der Bilddarstellungsinstanz beschrieben.

4.1.1. Datenstrukturabwicklungsinstanz

Die Abwicklung der Datenstruktur zu einer Folge relativer Bildbefehle kann wie in Bild 4.2 in drei Subprozesse strukturiert werden:

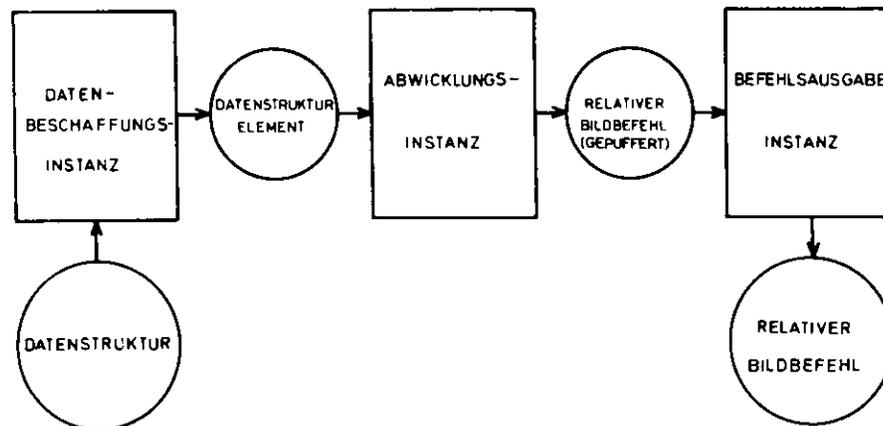


Bild 4.2 Datenflußgraph der Datenstrukturabwicklungsinstanz

- Datenbeschaffung aus der Datenstruktur, wobei einzelne Datenstrukturelemente gelesen und dann der nachfolgenden Instanz sequentiell angeboten werden. Wesentlich bei der Datenbeschaffung aus der Datenstruktur, die gewöhnlich in einem Massenspeicher mit direktem Zugriff gespeichert ist, ist die Berücksichtigung der physikalischen und logischen Zugriffswege, so daß die mittleren Zugriffszeiten klein gehalten werden können.
- Datenabwicklung, bei der aus der Sequenz der Datenstrukturelemente die relativen Bildbefehle abgeleitet werden. Voraussetzung für diesen Prozeß ist das Befehlsrepertoire in der Tabelle 4.1.
- Befehlsausgabe, bei der die relativen Bildbefehle an die Bildtransformationsinstanz weitergeleitet werden. Es muß jedoch angemerkt werden, daß die Befehlsausgabe keinen Bearbeitungsprozeß darstellt, sondern lediglich der Kommunikation mit der nachfolgenden Instanz dient.

Der Aktionsablauf der Datenstrukturabwicklungsinstanz kann hinsichtlich des angestrebten hohen Durchsatzes für die gesamte Bilddarstellungsinstanz wie in Bild 4.3 als Pipeline-Ablauf realisiert werden.

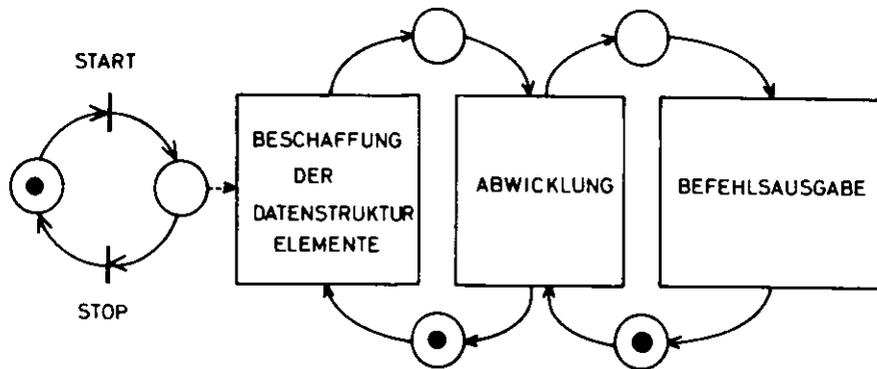


Bild 4.3 Aktionsablauf in der Datenstrukturabwicklungsinstanz

4.1.2. Bildtransformationsinstanz

Die Transformation der relativen Bildbefehle in absolute Bildbefehle und in Information für die Index-Kommunikation stellt durch ihre Vielseitigkeit und Komplexität eine so umfangreiche Aufgabe dar, daß sie nur mit Hilfe einer besonders leistungsfähigen Bildtransformationsinstanz zweckmäßig realisiert werden kann. Das Bild 4.4 zeigt die vorgeschlagene Struktur der Bildtransformationsinstanz. Da die Ausführung verschiedener Klassen relativer Bildbefehle im allgemeinen auch verschiedene Algorithmen benötigt, wurde die Bildtransformationsinstanz nach den auftretenden Befehlsklassen strukturiert. In dieser Betrachtungsebene beinhaltet sie daher drei spezialisierte Instanzen, die die Ausführung der relativen Bildbefehle wie folgt unterstützen:

- Die Attributauswertungsinstanz ist zuständig für die Durchführung der Attributbefehle (EXEC ATTR) und erzeugt einerseits den geltenden Bildstatus und andererseits unterschiedliche Informationsanteile für den absoluten Bildbefehl. Für jedes Attribut sind innerhalb der Attributauswertungsinstanz dedizierte Instanzen vorgesehen, die nur den dazugehörigen Aspekt des Bildstatus berechnen können. Da der geltende Bildstatus von der Abwicklungssequenz der Datenstruktur durch die verschiedenen Hierarchieebenen abhängt, muß der jeweils berechnete Abwicklungsstatus für jede Hierarchieebene in der Datenstruktur in einem Bildabwicklungsstack protokolliert werden. Bei jedem Wechsel der Hierarchieebene, der durch die Kontroll-Befehle CALL und RETURN mitgeteilt wird, muß der zuletzt berechnete Bildstatus auf den Stack "gepusht" bzw. vom Stack "gepoppt" werden. Der Top des Stacks beinhaltet dann bei einer ausgewogenen Gesamtanzahl von CALL- und RETURN-Befehlen immer den geltenden Bildstatus für die aktuelle Hierarchieebene. Der Bildabwicklungsstack ist entsprechend den behandelten mathematisch-geometrischen, optischen und freien Attributen in drei semantisch verschiedene Felder eingeteilt. Dies ist in Bild 4.4 durch die Zerlegung des Raumes "Bildabwicklungsstatus", der hier nur den Top des Stacks darstellt, in drei nur von den zugehörigen Instanzen einzeln zugreifbare Teilräume angedeutet. Die Aktualisierung des Bildabwicklungsstacks durch PUSH- und POP-Befehle findet jedoch für alle Felder des Stacks gleichzeitig statt, da bei einem Wechsel der Hierarchieebene sich alle Aspekte des Bildstatus verändern. Der modifizierende Zugriff auf den gesamten Bildabwicklungsstack kann daher sinnvollerweise (siehe Bild 4.4) von der Kontroll-Instanz vorgenommen werden.

mation gemeinsam behandelt.

Unter Berücksichtigung der dargestellten Problematik der mathematisch-geometrischen Transformationen hinsichtlich ihrer mangelnden Transparenz soll hier zunächst ihre Durchführungsart neu diskutiert werden. Dabei brauchen nur die Euklidischen Transformationen betrachtet zu werden, weil durch die Einführung der Kachelstruktur des Weltbildes die Domaine-Transformationen als explizit durchzuführende Transformationen überflüssig geworden sind. Die zweckmäßige Gestaltung der Euklidischen Transformationen erfordert eine nähere Untersuchung der zwei im Display-System vorkommenden Transformationsketten: die Bilddefinitions-kette als Teil der Datenstrukturgenerierung, und die Bildabwicklungskette als Teil der Bilddarstellung. Diese beiden gegeneinander laufenden Transformationsketten sind im Bild 4.5 veranschaulicht.

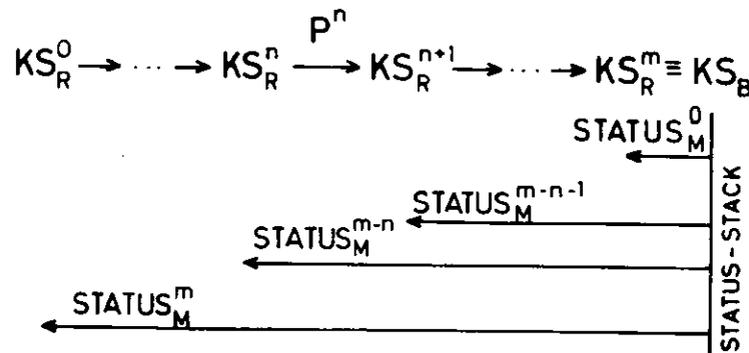


Bild 4.5 Zur Verdeutlichung der Bilddefinitions- und Bildabwicklungskette

Die Bilddefinitions-kette erzeugt durch sukzessive mathematisch-geometrische Transformationen eines Repertoireelements neue Repertoireelemente, die hier auch Figuren genannt werden. Entsprechend der Hierarchierelation in der Datenstruktur findet die Bilddefinition bottom - up statt. Das ursprüngliche Repertoireelement ist in einem eigenen Koordinatensystem KS_R^0 definiert, wobei man sich dieses Definitionskoordinatensystem rechtwinklig, normalliegend und mit gleichen Einheitslängen auf beiden Achsen vorstellt. Im Gegensatz zur üblichen Technik, wo einzelne Bildpunkte in dem starr bleibenden Bildschirmkoordinatensystem (KS_B) transformiert werden, wird in dieser Arbeit die Bilddefinitions-kette als eine Transformationssequenz des Definitionskoordinatensystems KS_R^0 betrachtet, wobei jeder Transformationsschritt durch Trans-

formationsparameter (P) angegeben wird. Für die zweckmäßige Gestaltung der Bilddefinitions-kette spielen zwei Kriterien eine ausschlaggebende Rolle: Erstens müssen die Transformationen repertoirebildend sein, d.h. die Transformationsparameter dürfen nur relativ zum aktuellen Koordinatensystem aus der Bilddefinitions-kette definiert werden. Zweitens müssen die Transformationen transparent sein, d.h. die Angabe der Transformationsparameter für eine gewünschte Transformation eines Repertoireelements muß für den Anwender einfach sein. Die Erfüllung dieser beiden Kriterien kann erreicht werden, wenn ein Transformationsschritt

$$KS_R^n \xrightarrow{P^n} KS_R^{n+1} \quad (4.3)$$

in der Bilddefinitions-kette wie folgt definiert wird: ein Repertoireelement wurde durch Transformation in KS_R^n gebracht, wobei KS_R^n rechtwinklig, normalliegend und mit gleichen Einheitslängen auf den beiden Achsen vorzustellen ist. Dieses Repertoireelement wird im KS_R^{n+1} transformiert verwendet, wobei man sich KS_R^{n+1} auch rechtwinklig, normalliegend und mit gleichen Einheitslängen auf den beiden Achsen vorstellt. Die Transformationsparameter

$$P^n := [(X_{op}, Y_{op}), \varphi_p, (S_{xp}, S_{yp})] \quad (4.4)$$

mit (X_{op}, Y_{op}) : Lage des Ursprungs des KS_R^n im KS_R^{n+1} ,
 φ : Winkel der x-Achse von KS_R^n gegenüber der x-Achse von KS_R^{n+1} ,
 (S_{xp}, S_{yp}) : achsenspezifische Einheitslängen in KS_R^n :
 Einheitslängen in KS_R^{n+1}

geben die Lage des KS_R^n im normalliegenden KS_R^{n+1} an. Sie werden in der Datenstruktur durch den mathematisch-geometrischen Attributbefehl A_M spezifiziert. Die Reihenfolge der elementaren Rotations-, Skalierungs- und Translationstransformationen innerhalb eines Bilddefinitionsschrittes kann unter Berücksichtigung folgender zwei Kriterien gewählt werden:

- Für die Lage der Achsenrichtung ist es unerheblich, wo die Rotation (R) in der Reihenfolge liegt.
- Für die Lage des Ursprungs ist es wesentlich, wo die Translation (T) in Relation sowohl zur Rotation (R) als auch zur Skalierung (S) liegt.

In dieser Arbeit soll eine Transformationsreihenfolge gewählt werden, die durch Angabe eines einzigen Attributbefehls die Rotation eines Repertoireelements um einen durch die Translation angegebenen Punkt erlaubt. Dies bedeutet, daß die Translation (T) vor der Rotation (R) und vor der Skalierung (S)

liegen muß. Aus den zwei möglichen und auch gleichwertigen Sequenzen ToRoS und ToSoR wird hier willkürlich die ToRoS - Konkatenation gewählt.

Die Bildabwicklungskette wird dadurch gestartet, daß ein Repertoireelement, das am Ende der Bilddefinitions-kette im KS_R^m definiert ist, auf dem Bildschirmkoordinatensystem KS_B dargestellt wird. Dies geschieht, indem das KS_R^m gleich dem KS_B gesetzt wird. Diese Gleichsetzung ist immer möglich, weil das Bildschirmkoordinatensystem KS_B durch Hardware vorgegeben ist und daher i.a. rechtwinklig, normalliegend und mit gleichen Einheitslängen auf den beiden Achsen definiert ist. Da die Abbildung eines Repertoireelements im KS_B die Berechnung der relativen Lage des Definitionskoordinatensystems KS_R^0 im KS_B erfordert, muß die Bildabwicklungskette die Transformationen aus der Bilddefinitionskette schrittweise zurückverfolgen. Die relative Lage eines Koordinatensystems KS_R aus der Bilddefinitionskette im normalliegenden Bildschirmkoordinatensystem KS_B wird hier als mathematisch-geometrischer Status bezeichnet. Der mathematisch-geometrische Status ist formell als ein Vektor

$$\underline{STATUS}_M^j := [(X_{os}, Y_{os}), \vartheta_s, \phi_s, (S_{xs}, S_{ys})]^j \quad (4.5)$$

mit (X_{os}, Y_{os}) : Position des Ursprungs des KS_R^{m-j} im KS_B ,
 ϑ_s : Rotationswinkel der x-Achse von KS_R^{m-j} gegenüber der x-Achse von KS_B ,
 ϕ_s : Orthogonalitätsabweichung des KS_R^{m-j} , d.h. Winkel zwischen der y-Achse von KS_R^{m-j} gegenüber der Senkrechten auf der x-Achse von KS_R^{m-j} ,
 (S_{xs}, S_{ys}) : achsenspezifische Einheitslängen des KS_R^{m-j} gegenüber des KS_B

definiert, der die Lage des KS_R^{m-j} im KS_B eindeutig beschreibt. Die schrittweise Berechnung des mathematisch-geometrischen Status für jedes Koordinatensystem KS_R aus der Bilddefinitionskette findet wie die Abwicklung der Datenstruktur top - down statt. Die Abwicklungsschritte sind durch die Gleichung

$$\underline{STATUS}_M^{j+1} = f(\underline{STATUS}_M^j, p^{m-j-1}) \quad (4.6)$$

beschrieben und müssen bis zur Berechnung des $STATUS_M^m$ iteriert werden. Aus dem $STATUS_M^m$, der die Lage des ursprünglichen Definitionskoordinatensystems KS_R^0 im Bildschirmkoordinatensystem KS_B beschreibt, kann die Abbildung des Repertoireelements auf dem Bildschirm nach der Gleichung

$$(X, Y)_{KS_B} = g \left[(X, Y)_{KS_R^0}, \underline{STATUS}_M^m \right] \quad (4.7)$$

direkt abgeleitet werden.

Für die gewählte ToRoS Konkatenation ergeben sich für die Veränderung des mathematisch-geometrischen Status durch einen Attributbefehl folgende Beziehungen:

$$X_{os}^{n+1} = [X_{os} + X_{op} S_{xs} \cos \vartheta_s - Y_{op} S_{ys} \sin(\vartheta_s + \phi_s)]^n \quad (4.8)$$

$$Y_{os}^{n+1} = [Y_{os} + X_{op} S_{xs} \sin \vartheta_s + Y_{op} S_{ys} \cos(\vartheta_s + \phi_s)]^n$$

$$S_{xs}^{n+1} = (E^2 + F^2)^{1/2} \quad ; \quad S_{ys}^{n+1} = (G^2 + H^2)^{1/2} \quad (4.9)$$

$$\vartheta_s^{n+1} = \begin{cases} \arcsin(F/S_{xs}^{n+1}) & \text{bei } E \geq 0 \\ - \arcsin(F/S_{xs}^{n+1}) & \text{bei } E < 0 \end{cases} \quad (4.10)$$

$$\phi_s^{n+1} = \begin{cases} \arcsin(H/S_{ys}^{n+1}) - \pi/2 - \phi_s^{n+1} & \text{bei } G \geq 0 \\ \pi/2 - \arcsin(H/S_{ys}^{n+1}) - \phi_s^{n+1} & \text{bei } G < 0 \end{cases} \quad (4.11)$$

wobei $E = [S_{xp} \sin \vartheta_p S_{xs} \cos \vartheta_s - S_{xp} \cos \vartheta_p S_{ys} \sin(\vartheta_s + \phi_s)]^n$

$$F = [S_{xp} \sin \vartheta_p S_{xs} \sin \vartheta_s + S_{xp} \cos \vartheta_p S_{ys} \cos(\vartheta_s + \phi_s)]^n$$

$$G = [-S_{yp} \sin \vartheta_p S_{xs} \cos \vartheta_s - S_{yp} \cos \vartheta_p S_{ys} \sin(\vartheta_s + \phi_s)]^n$$

$$H = [-S_{yp} \sin \vartheta_p S_{xs} \sin \vartheta_s + S_{yp} \cos \vartheta_p S_{ys} \cos(\vartheta_s + \phi_s)]^n$$

bedeuten.

Die Verwendung von Transformationsparametern mit Rotationswinkel $\theta_p \neq n \cdot \pi/2$ und unterschiedlichen Skalierungsfaktoren $S_{xp} \neq S_{yp}$ führt zur Orthogonalitätsabweichung $\phi_s \neq 0$ und somit zu relativ komplexen Transformationsregeln für die Statusparameter θ_s, ϕ_s und (S_{xs}, S_{ys}) . In einem System, wo nur Transformationen mit $S_{xp} = S_{yp}$ oder mit $\theta_p = n \cdot \pi/2$ zugelassen werden, gelten dann folgende einfache Transformationsregeln:

$$S_{xs}^{n+1} = [S_{xs} \cdot S_{xp}]^n \quad ; \quad S_{ys}^{n+1} = [S_{ys} \cdot S_{yp}]^n \quad (4.9.1)$$

$$\theta_s^{n+1} = [\theta_s + \theta_p]^n \quad (4.10.1)$$

$$\phi_s^{n+1} = \phi_s^n = 0 \quad (4.11.1)$$

Die Realisierung der Euklidischen Transformationen läßt sich dadurch hinsichtlich des operationellen Aufwandes und der erforderlichen Durchsatzraten erheblich vereinfachen.

Wie bereits dargestellt, werden die Euklidischen Transformationen nicht auf einzelne Bildpunkte, sondern auf ganze Bildelemente, die hier ausschließlich Vektoren sind, angewandt. Ein Vektor ist in der Datenstruktur in Form seiner beiden Komponenten gegeben:

$$\text{VEKTOR} = (\Delta x, \Delta y) \text{KS}_R^0 \quad (4.12)$$

Zur Darstellung des Vektors auf dem Schirm wird vorausgesetzt, daß er in der aktuellen Zeichnungsposition beginnend, im Koordinatensystem KS_R^m gezeichnet wird, wodurch sich die aktuelle Zeichnungsposition wieder verändert. Zu Beginn der Darstellung eines Vektorzuges aus V^0 ist die aktuelle Zeichnungsposition identisch mit dem Ursprung (X_0, Y_0) des Koordinatensystems KS_R^0 im normalliegenden KS_R^m . Das Zeichnen des Vektors im Koordinatensystem KS_R^m bedeutet die Transformation der beiden Vektorkomponenten entsprechend dem mathematisch-geometrischen Status STATUS^m . Diese Transformation ist durch folgende Gleichungen beschrieben:

$$\Delta x_{\text{KS}_B} = [\Delta x_{\text{KS}_R^0} S_{xs} \cos \theta_s - \Delta y_{\text{KS}_R^0} S_{ys} \sin(\theta_s + \phi_s)]^m \quad (4.13)$$

$$\Delta y_{\text{KS}_B} = [\Delta x_{\text{KS}_R^0} S_{xs} \sin \theta_s + \Delta y_{\text{KS}_R^0} S_{ys} \cos(\theta_s + \phi_s)]^m$$

Somit ergibt sich der gesuchte Vektorendpunkt als:

$$\begin{aligned} X_e &= [X_0 + \Delta x S_{xs} \cos \theta_s - \Delta y S_{ys} \sin(\theta_s + \phi_s)]^m \\ Y_e &= [Y_0 + \Delta x S_{xs} \sin \theta_s + \Delta y S_{ys} \cos(\theta_s + \phi_s)]^m \end{aligned} \quad (4.14)$$

Die Strukturgleichheit der Gleichungen 4.8 und 4.14 zeigt, daß für die Auswertung der mathematisch-geometrischen Attribute (A_M) und für die Transformation der elementaren Vektoren teilweise gleiche Verarbeitungsvorgänge notwendig sind. Sie können daher sinnvollerweise von der gleichen Instanz ausgeführt werden. In Bild 4.6 ist die informationelle Struktur der beiden hier behandelten Instanzen als Datenflußgraph veranschaulicht. Die Instanzen erzeugen im wesentlichen aus einer Teilmenge relativer Bildbefehle die Teilräume "aktueller Vektor" und "aktuelle Zeichnungsposition" für den absoluten Bildbefehl, wobei dafür die vorherige Berechnung des mathematisch-geometrischen Status (STATUS_M) notwendig ist. Die Verwendung der mathematisch-geometrischen Transformationsinstanz sowohl für die Transformation der elementaren Vektoren ($\Delta x, \Delta y$) als auch für die der relativen Translationsbeträge (x_{op}, y_{op}) ist in Bild 4.6 durch den zusätzlichen lesenden Zugriff auf den Zustandsraum "aktueller Vektor" angedeutet.

Die A_M -Auswertungsinstanz setzt sich aus drei spezialisierten Instanzen zusammen, die für die parallele Berechnung des Positions-, Skalierungs- und Rotationsstatus nach den Gleichungen 4.8 bis 4.11 vorgesehen sind. Infolgedessen ist auch das mathematisch-geometrische Feld des Bildabwicklungsstacks in drei weitere (Unter-)Felder eingeteilt, auf die nur von der zugeordneten dedizierten Instanz modifizierend zugegriffen werden kann.

Die mathematisch-geometrische Transformationsinstanz kann den aktuellen mathematisch-geometrischen Bildstatus stets aus dem Top des Stacks lesen, wobei entsprechend der Gleichung 4.13 nur der aktuelle Rotations- und Skalierungsstatus benötigt wird. Die mathematisch-geometrische Transformationsinstanz kann daher in eine Rotations- und eine Skalierungsinstanz unterteilt werden, die in dieser Reihenfolge die Transformationsanteile für die Rotation bzw. für die Skalierung ausführen.

Das Petri-Netz in Bild 4.7 zeigt den Aktionsablauf innerhalb des Datenflußgraphen in Bild 4.6. Der Aktionsablauf ist hinsichtlich der notwendigen hohen Datenraten in der gesamten Bildtransformationsinstanz als eine zweistufige Pipe-

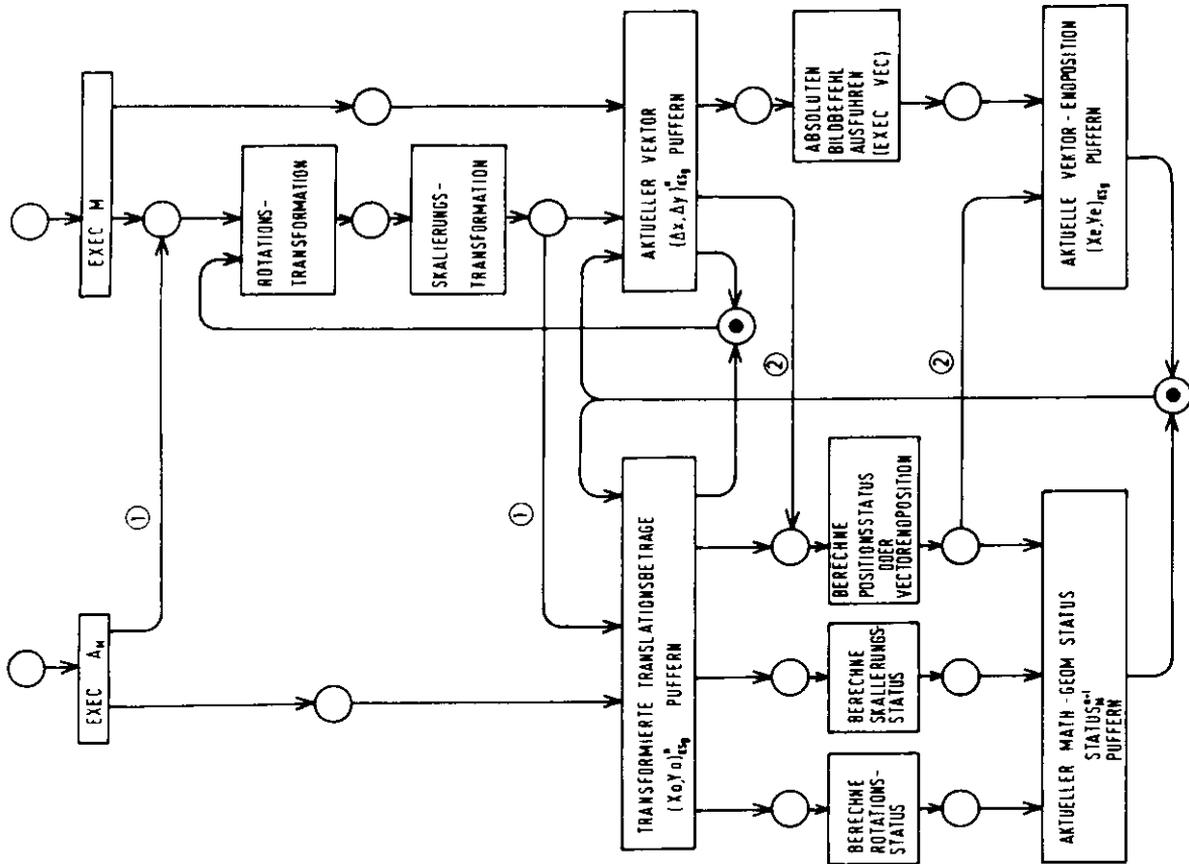


Bild 4.7 Aktionsablauf bei der mathematisch-geometrischen Transformation

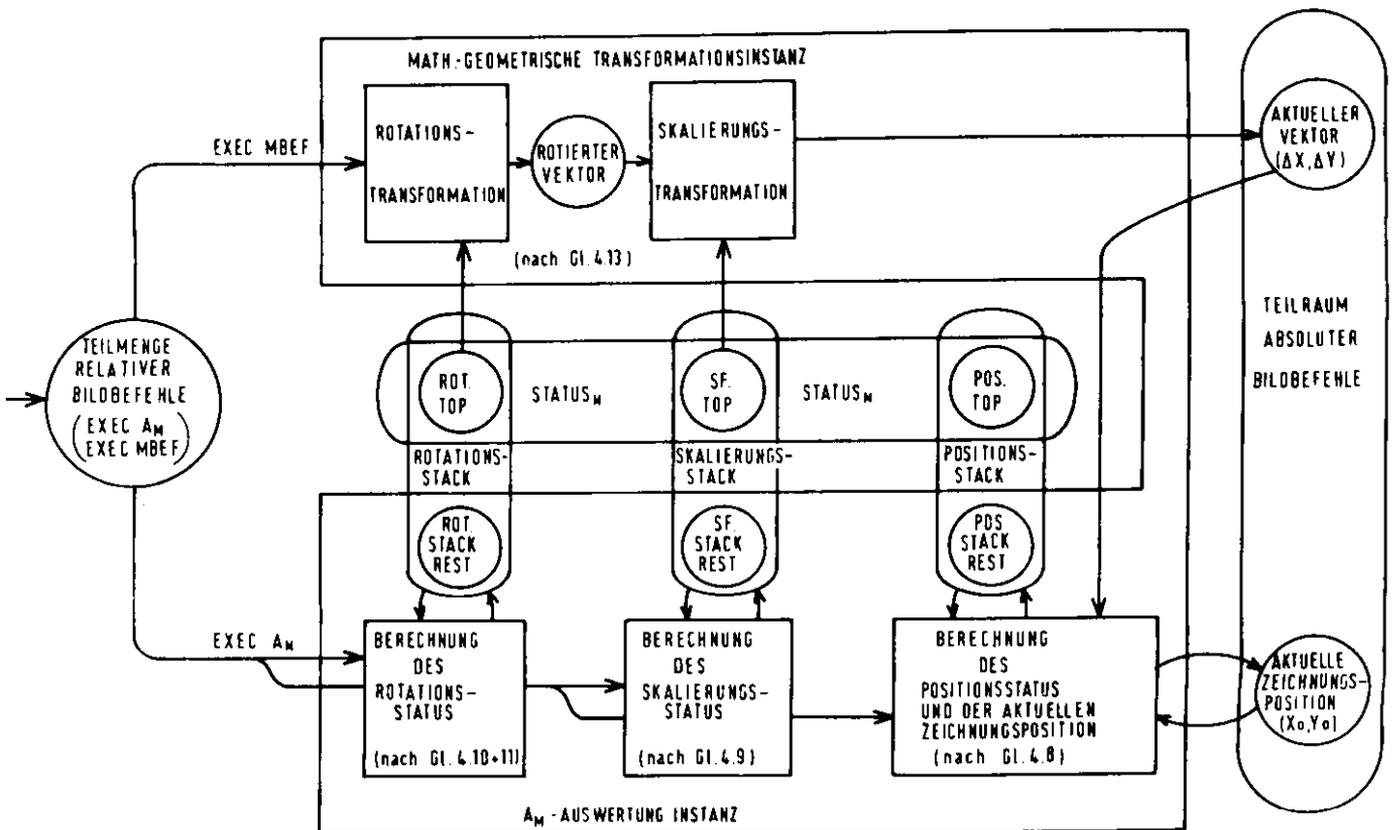


Bild 4.6 Informationelle Struktur der mathematisch-geometrischen Transformation

line-Struktur realisiert. Dadurch kann die mathematisch-geometrische Transformation der elementaren Vektoren oder der relativen Translationsbeträge gleichzeitig mit der Ausführung eines absoluten Bildbefehls bzw. mit der Berechnung des mathematisch-geometrischen Status stattfinden. Für die Durchsatzserhöhung ist insbesondere die zeitlich überlappte Durchführung der Transformation der elementaren Vektoren und der Ausführung der absoluten Bildbefehle von Bedeutung, da die Häufigkeit solcher Bildelementbefehle erheblich höher ist als die der Attributbefehle. Eine weitere Pipeline-Stufe könnte bei Bedarf zwischen den zeitaufwendigen Rotations- und Skalierungstransformationen eingebaut werden, ohne daß sich der hier vorgeschlagene Aktionsablauf wesentlich verändern würde. Die handshakeartigen Markenübergänge zwischen der A_M -Auswertungsinstanz und der Transformationsinstanz (siehe ① und ② in Bild 4.7) deuten auf die mehrfache Verwendung verschiedener Instanzen in Zeitmultiplex hin: Einerseits wird, wie schon erwähnt, die mathematisch-geometrische Transformationsinstanz benutzt, um die Rotations- und die Skalierungstransformationen für die elementaren Vektoren $(\Delta x, \Delta y)$ und für die relativen Translationsbeträge (x_0, y_0) auszuführen. Damit wird es möglich, die komplexen Operationen für die Matrizenmultiplikation (siehe Gleichung 4.13) kostensparend in nur einer physikalischen Instanz zu realisieren. Andererseits wird die Positionierungsinstanz für die Berechnung sowohl des neuen Positionstatus als auch der aktuellen Zeichnungsposition verwendet.

Das Ablaufdiagramm in Bild 4.8 zeigt, daß die aktuelle Zeichnungsposition in zwei Situationen benötigt wird:

Erstens bei der Generierung von Vektorzügen aus elementaren Vektoren, wo der Endpunkt eines elementaren Vektors dem Anfangspunkt für den nachfolgenden Vektor entspricht.

Zweitens bei der Berechnung des neuen Bildstatus, wo als Referenzpunkt für die relativen Translationsbeträge die "aktuelle Position" anstelle der "Origin-Position" angegeben wird.

Um die aktuelle Zeichnungsposition stets verfügbar zu haben, wird die Vektorposition ständig parallel zur Ausführung des absoluten Bildbefehls "EXEC VEC" berechnet und in dem Teilraum "aktuelle Zeichnungsposition" gespeichert. Bei der Berechnung eines neuen Bildstatus mit spezifizierter "aktueller Position" muß dann, wie im Ablaufdiagramm gezeigt, die "aktuelle Zeichnungsposition" auf den Top des Positionsstacks transferiert werden. Außerdem zeigt das Ablaufdiagramm alle

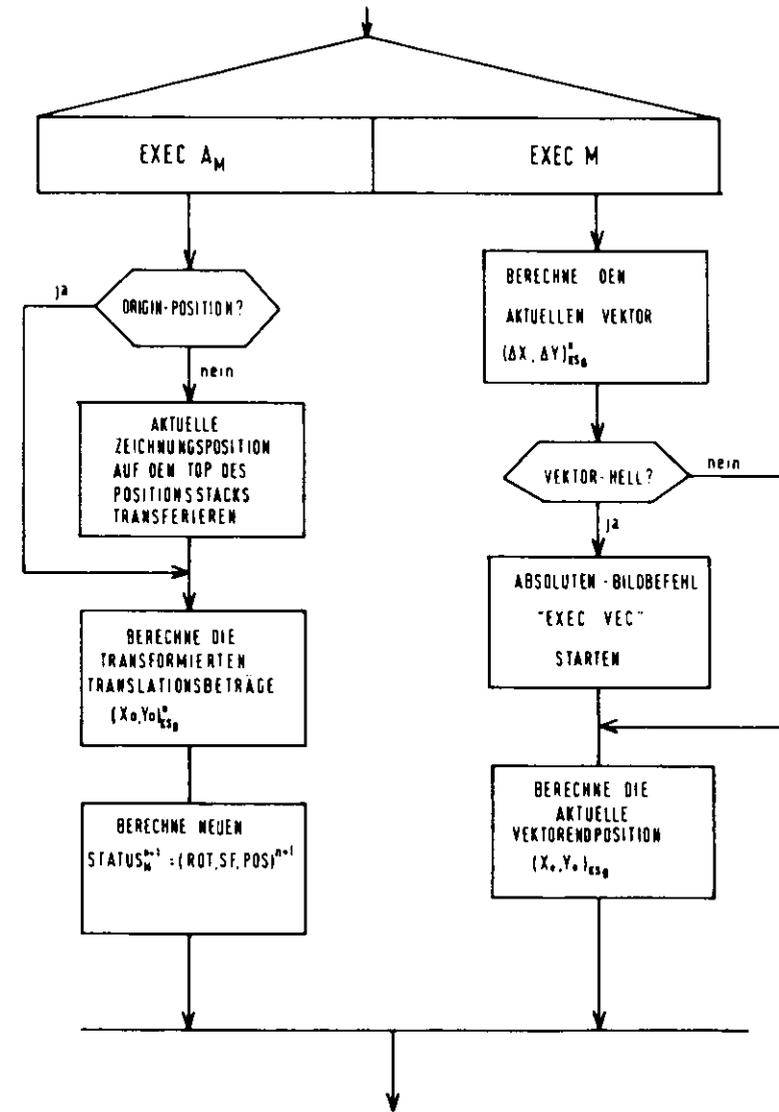


Bild 4.8 Ablaufdiagramm der mathematisch-geometrischen Transformationsinstanz

Verarbeitungsschritte, die in der mathematisch-geometrischen Transformations- und der A_M -Auswertungsinstanz durchgeführt werden.

4.1.2.2. Auswertung optischer Attribute

Die Diskussion bezüglich der Relation Begriff/Optik hat im Zusammenhang mit dem Konzept des logischen Indexbildes (siehe Abschnitt 2.1) deutlich gemacht, daß die Auswertung der optischen Attribute nur im Hinblick auf die nachfolgende Begriff/Index Transformation betrachtet werden muß. Das Bild 4.9 zeigt die grobe informationelle Struktur der Auswertungsinstanz für die optischen Attribute (A_0).

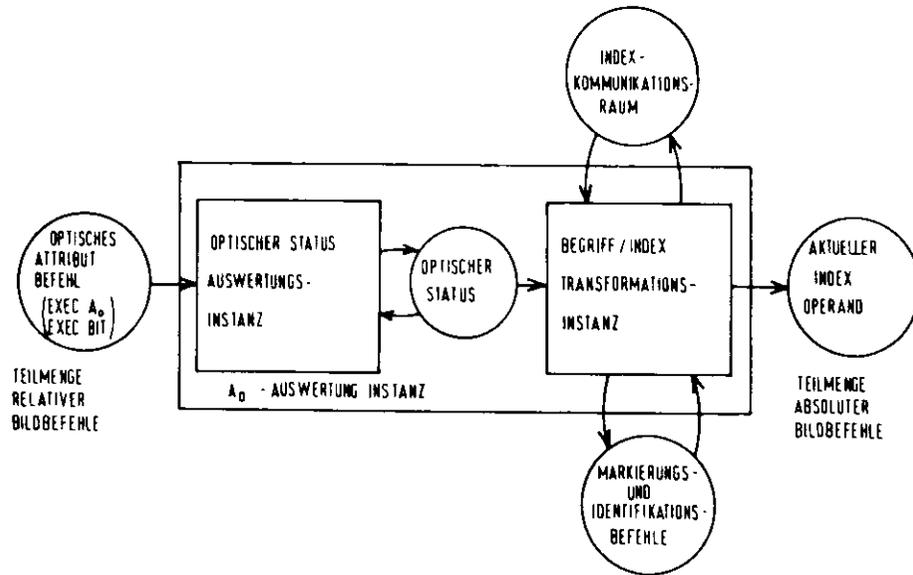


Bild 4.9 Grobe Informationelle Struktur der A_0 -Auswertungsinstanz

Sie muß aus einer Teilmenge der relativen Bildbefehle einerseits den "Index-Operand" als Teilraum des absoluten Bildbefehls und andererseits den Inhalt des Index-Kommunikationsraumes und des Markierungs- und Identifikationsbefehl-Raumes erzeugen. Aus Bild 4.9 ist leicht zu entnehmen, daß die Auswertung der optischen

Attribute A_0 in einen optischen Status die Voraussetzung für die anschließende Begriffsableitung und die Begriff/Index-Transformation ist. Bevor die informationelle Struktur in Bild 4.9 feiner aufgelöst wird, soll hier zunächst auf die schon in Abschnitt 2.1.3 prinzipiell angedeutete Problematik der Begriff/Index-Transformation näher eingegangen werden. Die Hauptschwierigkeit bei der informationellen Modellierung der Begriff/Index-Transformation, die die getrennte Repräsentation der Begriffe als logische Indexwerte und deren optischer Erscheinung als Index/Optik-Tabelle ermöglichen soll, besteht darin, daß die Abbildung der Begriffsmenge von der Datenstruktur in die Indexmenge im virtuellen Schirm nicht in beiden Richtungen eindeutig ist. Diese Abbildungsfunktion ist von der Abwicklungssequenz der Datenstruktur abhängig, d. h. die resultierende Menge der logischen Indizes im virtuellen Schirm hängt wesentlich von der Sequenz der Begriffsgenerierung aus der Datenstruktur ab. Dies kann mit einem einfachen Beispiel veranschaulicht werden. Das Bild 4.10 zeigt die Datenstruktur und die resultierende Belegung des virtuellen Schirmes für drei einfache Dias - ein Quadrat (Q), einen Kreis (K) und ein Dreieck (Z) - die sich auf Grund der in den Verwendungscalls spezifizierten mathematisch-geometrischen Attributen (A_M) geometrisch überlagern. Bei der Abwicklung dieser so spezifizierten Datenstruktur resultieren aus den ursprünglichen drei Datenbegriffen jedoch vier optisch unterscheidbare Bildbereiche, die zur Darstellung auch vier logische Indexwerte entsprechend der beigefügten Tabelle in Bild 4.10 benötigen. Im Zusammenhang mit der hier resultierenden Optik muß angemerkt werden, daß bei der geometrischen Überlagerung von Dias die optischen Attribute der Dias in ihrer Abwicklungsreihenfolge auf die Vorderseite der gedachten Glasscheibe aufgetragen werden. Das gilt im Gegensatz zur Auswertung optischer Attribute, wo für die Auftragung einzelner Attribute die Rückseite der gedachten Glasscheibe verwendet wird (siehe Kapitel 3). Es gibt zwei Fälle, die die Mehrdeutigkeit der Begriff/Index-Transformation hervorrufen:

- Erstens können durch die Generierung eines Begriffes mit einem deckenden optischen Attribut andere früher generierte Begriffe durch die geometrische Überlagerung teilweise oder vollständig überschrieben werden. Beispielsweise überschreibt in Bild 4.10 der Kreis D_2^0 das Quadrat D_3^0 teilweise. Die Überschreibung von Datenbegriffen führt dazu, daß möglicherweise einem Begriff aus der Datenstruktur kein logischer Index im virtuellen Schirm (mehr) assoziiert werden kann.
- Zweitens können durch die Generierung eines Begriffes mit einem transparenten

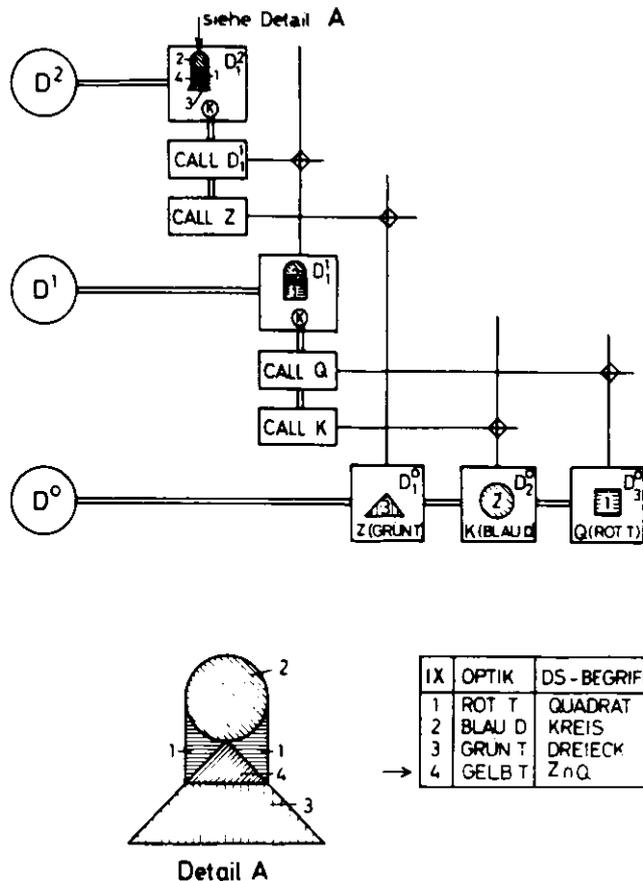


Bild 4.10 Zur Eindeutigkeit der Begriff/Index-Transformation

optischen Attribut andere schon früher generierte Begriffe durch die geometrische Überlagerung ihre eigene Optik bereichsweise so verändern, daß jetzt mehrere

logische Indizes für ihre Darstellung benötigt werden. Beispielsweise wird in Bild 4.10 das Quadrat D_3^0 von dem Dreieck D_1^0 so überlappt, daß der Intersektionsbereich $Z = Q$ eine veränderte Optik (nämlich "gelb") aufweist, die nur mit einem zusätzlichen Indexwert (4) darstellbar ist. Auf diese Weise kann einem Begriff aus der Datenstruktur eine Menge logischer Indizes im virtuellen Schirm assoziiert werden. Diese Menge logischer Indizes wird im folgenden als Teilindexmenge (TIM) bezeichnet. Jeder Indexwert einer Teilindexmenge stellt jedoch nicht mehr die codierte Repräsentation eines Begriffes dar, sondern kennzeichnet nur eine bestimmte topologische Relation - nämlich die Intersektion - zwischen den Datenbegriffen. Dies bedeutet, daß ein Index einer Teilindexmenge logisch einer ganzen Menge von Begriffen zugeordnet werden muß, was die Mehrdeutigkeit der Begriff/Index-Transformation klar zum Ausdruck bringt. Für die Behandlung der Teilindexmenge als logische Repräsentation eines Begriffes ist von wesentlicher Bedeutung, daß die theoretische Größe einer Teilindexmenge auch von der Abwicklungssequenz der Datenstruktur abhängig ist: Ein Begriff, der dann generiert wird, wenn bereits N Indizes vergeben wurden, kann eine Teilindexmenge mit maximal N neuen Indexwerten erzeugen.

Die Begriff/Index-Transformation kann trotz ihrer prinzipiellen Abhängigkeit von der Abwicklungssequenz der Datenstruktur und ihres erhöhten Bedarfs für Indexwerte als eine eindeutige und technisch implementierbare Abbildung modelliert werden, wenn folgende Maßnahmen getroffen werden:

Die erste Maßnahme betrifft die Mehrdeutigkeit der Begriff/Index-Transformation, die aus der prinzipiellen Abhängigkeit des logischen Indexbildes von der Generierungssequenz der Datenbegriffe folgt. Sie ist daher logisch zwingend. Obwohl diese störende Abhängigkeit prinzipiell nicht beseitigt werden kann, wäre die Begriff/Index-Transformation in beiden Richtungen eindeutig, wenn die Generierungssequenz der Begriffe bekannt wäre. Daher muß bei der Abwicklung der Datenstruktur die resultierende Generierungssequenz der Begriffe in Form einer Begriff-Sequenz-Tabelle festgehalten werden, die dann später zur eindeutigen Identifizierung eines Begriffes oder eines Indexwertes verwendet werden kann.

Die zweite Maßnahme betrifft den erhöhten Bedarf an Indexwerten, der sowohl durch die Konstruktion der Flächengruppen als auch durch die Generierung der Teilindexmengen verursacht wird. Sie ist im Hinblick auf die begrenzte Anzahl der Indexwerte im virtuellen Schirm technisch zwingend. Der erhöhte Bedarf an Indexwerten kann nur durch eine effiziente Index-Kompression radikal vermin-

dert werden. Die Algorithmen für die Index-Kompression müssen bei einer gegebenen Pixelbreite des virtuellen Schirmes eine maximale Begriffsauflösung des logischen Indexbildes gewährleisten.

Eine informationelle Struktur für die Auswertung optischer Attribute (EXEC A₀) und für die Durchführung der komplexen Begriff/Index-Transformation (EXEC BIT), die die beiden obigen Maßnahmen berücksichtigt, ist als Datenflußgraph in Bild 4.11 und als Aktionsablauf in Bild 4.12 vorgeschlagen.

Der relative Bildbefehl EXEC A₀ bewirkt zunächst die Berechnung eines neuen optischen Bildstatus und anschließend die Ableitung der Begriffe aus der Datenstruktur. Wegen ihrer Abhängigkeit von der jeweiligen Hierarchieebene in der Datenstruktur wird die Information für den Optik- und Begriffstatus in separaten Stacks geführt, die ähnlich wie beim mathematisch-geometrischen Status als Felder des gesamten Bildabwicklungsstacks zu betrachten sind.

Der relative Bildbefehl EXEC BIT verwendet die gesamte Optik- und Begriffsinformation aus den Stacks einerseits, um den aktuellen Index-Operanden als Generierungsanweisung des logischen Indexbildes zu erzeugen, und andererseits, um die gesamte Index/Optik-Tabelle als codierte optische Erscheinungsform des logischen Indexbildes aufzufüllen.

Die detaillierte Struktur und Funktion der verschiedenen Instanzen und Zustandsräume in Bild 4.11 werden im folgenden näher beschrieben.

Die Berechnung des optischen Bildstatus findet entsprechend dem in Abschnitt 2.1.1 beschriebenen Modell der sequentiellen Farbauftragung auf der Rückseite der virtuellen Glasscheibe statt. Der optische Bildstatus

$$\text{STATUS}_0^n = (R_S, G_S, B_S, FT_S)^n \quad (4.15)$$

verändert sich auf Grund des optischen Attributbefehls

$$A_0^n = (R_A, G_A, B_A, FT_A)^n \quad (4.16)$$

entsprechend der Übergangsfunktion

$$\text{STATUS}_0^{n+1} = f(A_0, \text{STATUS}_0^n), \quad (4.17)$$

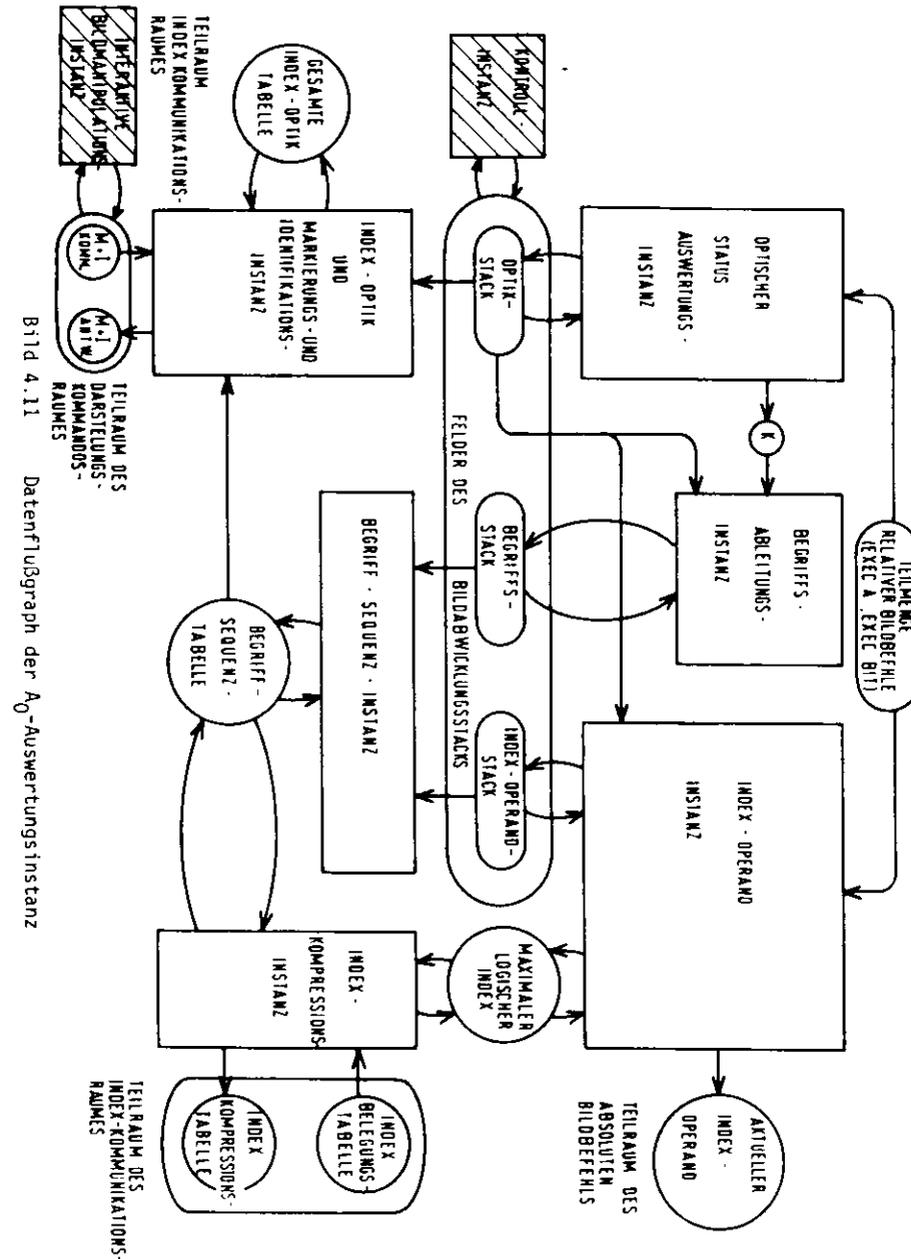


Bild 4.11 Datenflußgraph der A₀-Auswertungsinstanz

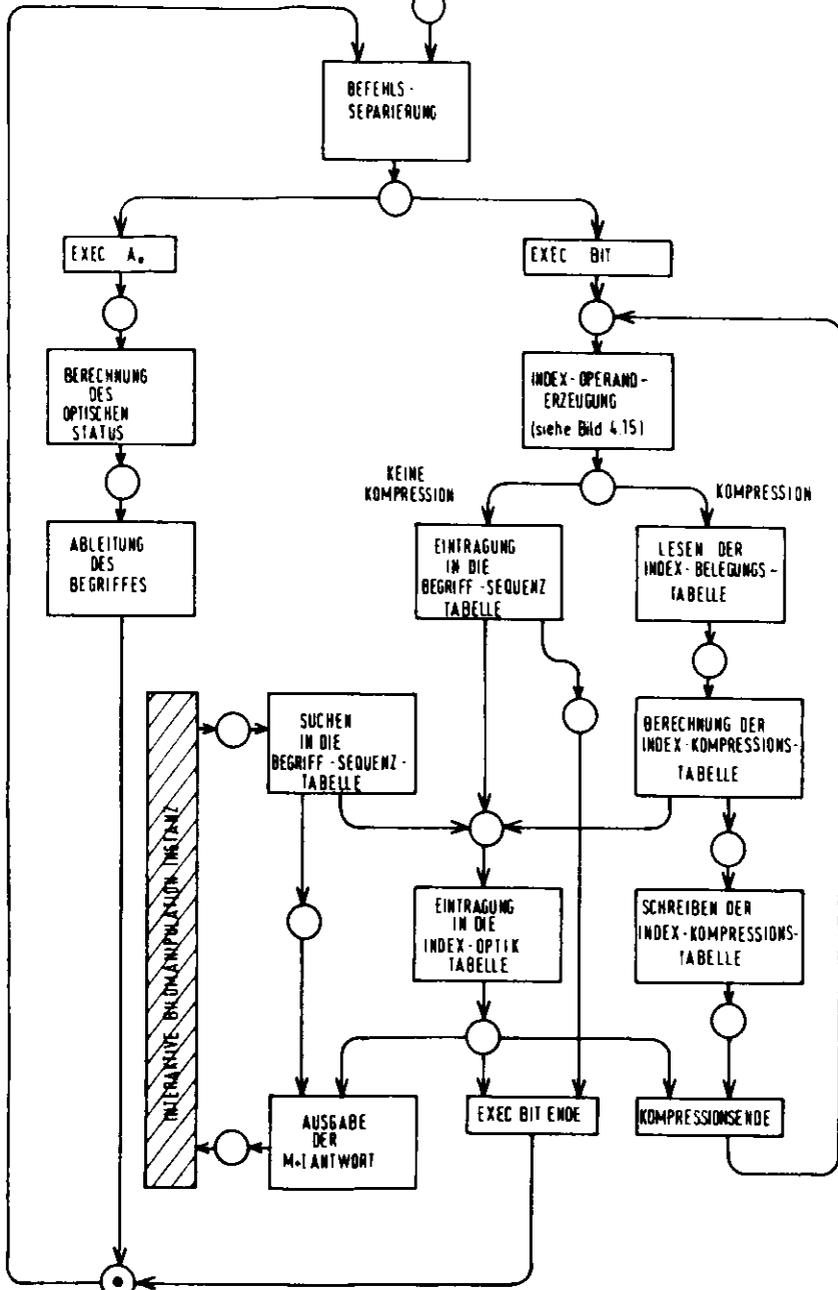
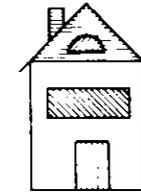


Bild 4.12 Aktionsablauf in der A₀-Auswertungsinstanz

wobei bei der Annahme, daß A_0^n immer die später aufgetragene Farbe darstellt, die Regeln für die additiven Mischoperationen aus den Gleichungen 2.5 gelten. Der so berechnete optische Bildstatus ist in der Reihenfolge der Abwicklung der Datenstruktur in dem Optik-Stack protokolliert.

Die Information in dem Optik-Stack wird unmittelbar nach der Berechnung des optischen Status für die Ableitung der optischen Begriffe verwendet. Voraussetzung für die Ableitung der Begriffe ist, daß die Reihenfolge ihrer Entstehung korrekt erkannt und festgehalten wird. Dafür muß zunächst der "optische Begriff" im Sinne des Abschnitts 2.1.2 näher definiert werden. Bild 4.13 zeigt mit Hilfe eines einfachen Beispiels die Begriffsentstehung auf verschiedenen Ebenen der Datenstruktur, die hier als logischer Baum mit Linksabwicklung dargestellt ist.



- = GRÜN
- = BLAU
- = ROT
- = GELB

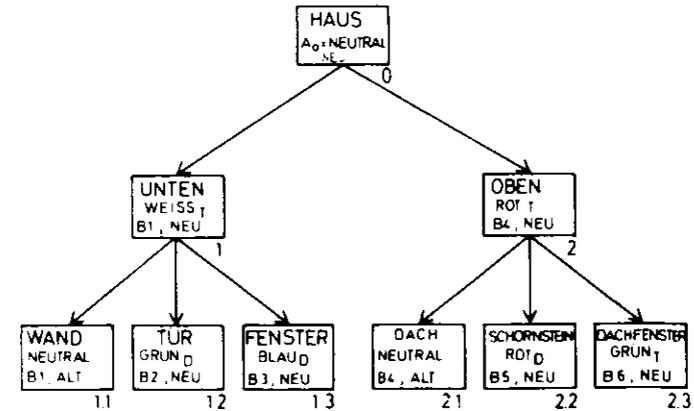


Bild 4.13 Zur Definition der optischen Begriffe und deren Entstehung

Als optischer Begriff gelten die Wurzel des logischen Baumes und alle Datenobjekte, deren optischer Bildstatus durch die Auswertung eines nicht neutralen optischen Attributbefehls EXEC $A_0 \neq \text{Neutral}$ entstanden ist. Jeder Begriff ist in seinem Entstehungsknoten als "NEU" und in allen anderen Knoten als "ALT" bezeichnet. Für jede Hierarchieebene in der Datenstruktur wird der dort vorhandene Begriff mit seinem Namen und seinem Entstehungsattribut (NEU oder ALT) durch die Begriffsableitungsinstanz in dem Begriffsstack geführt.

Die Index-Operand-Instanz verwendet die Optik- und Begriffsinformation für die Erzeugung des aktuellen Index-Operanden, was in Bild 4.11 durch ihre lesenden Zugriffe zu dem Optik- und Begriff-Stack angedeutet ist. Der aktuelle Index-Operand als Teil des absoluten Bildbefehls spezifiziert die durchzuführende Indexoperation und den Indexwert selbst mit der folgenden Syntax:

$$\text{Index-Operand} := \langle \text{Indexoperation, Indexwert} \rangle \quad (4.18)$$

Die Indexoperation wird nur aus der Optikinformation abgeleitet. Hier ist allein das Farbtransparenzattribut (FT) von Bedeutung: die Begriffe mit deckenden Farben (D) werden mit "SET INDEX", die mit transparenten Farben (T) mit "ADD INDEX" generiert, so daß

$$\begin{aligned} \text{für } FT := D \rightarrow \text{Indexoperation} &:= \text{SET} \\ \text{und für } FT := T \rightarrow \text{Indexoperation} &:= \text{ADD} \end{aligned} \quad (4.19)$$

gilt. Der Indexwert dagegen wird aus der Begriffsinformation abgeleitet. Um die gleiche Begriffsauflösung in der Datenstruktur und im virtuellen Schirm zu erreichen, muß jedem "neuen" Begriff ein Indexwert oder auch eine Indexwertmenge eindeutig zugeordnet werden. Dafür ist es notwendig, daß die Index-Operand-Information ebenso wie die Begriffsinformation in einem weiteren Feld des Bildabwicklungsstacks - hier Index-Operand-Stack bezeichnet - geführt wird (siehe Bild 4.11). Die Problematik der Zuweisung der Indexwerte besteht hauptsächlich darin, daß die Begriffe in einem beliebigen Knoten des logischen Baumes entstehen können, während die Zuweisung der Indexwerte nur an den Blättern des Baumes, d. h. unmittelbar vor der Ausführung eines absoluten Bildbefehls, vorgenommen werden kann. Dies ist in Bild 4.13 leicht erkennbar: Beispielsweise entsteht der Begriff B1 im Knoten 1. Die Zuweisung eines Indexwertes kann aber erst später im Knoten 1.1 stattfinden. Aus diesem Grunde kann sich die Sequenz der Begriffsgenerierung im Begriffsstack von der der Zuweisung von Index-Operanden im Index-

Operand-Stack stark unterscheiden. Dies bedeutet, daß bei einer PUSH-Operation auf dem gesamten Bildabwicklungsstack in Bild 4.11 ein Element auf den Stack gelegt werden kann, bei dem zwar die Optik- und Begriffsinformation, nicht aber der zugehörige Index-Operand definiert ist. Letzterer muß dann irgendwann später ausgefüllt werden, wobei das Element durchaus unterhalb des Stack-Tops gelegt werden darf. Dies soll mit einem einfachen Beispiel veranschaulicht werden. Bild 4.14 zeigt die Belegung des Bildabwicklungsstacks im Knoten 1.1 während der Abwicklung der Datenstruktur aus dem Bild 4.13. Die oberen zwei Ebenen

TOP →	2	WEISS	T	B1	WAND	ALT	ADD	1
	1	WEISS	T	B1	UNTEN	NEU	ADD	1
	0	NEUTRAL	T	B0	HAUS	NEU	-	0
STACK-	CHROMINANZ	FT	BEGRIFF	NAME	NEU/ALT	INDEX-OPERATION	INDEX-WERT	
EBENE	OPTIK - STACK		BEGRIFF - STACK			INDEX-OPERAND STACK		

Bild 4.14 Zur Veranschaulichung der Indexwert-Zuordnung

des Index-Operand-Stacks (siehe schraffierte Fläche in Bild 4.14) beinhalten beim Erreichen des Knotens 1.1 noch keine Information, weil bisher eine Zuweisung eines Index-Operanden nicht erforderlich war. Bevor jedoch der Begriff B1 (WAND) im virtuellen Schirm generiert werden kann, muß mit Hilfe eines zusätzlichen Suchpointers der Entstehungsknoten dieses Begriffes gesucht (Stackebene 1), aus der Optik- und Begriffsinformation (FT := T, NEU) der aktuelle Index-Operand (ADD 1) abgeleitet und anschließend der Index-Operand-Stack mit diesem Index-Operanden bis zur aktuellen Top-Ebene gefüllt werden. Die Index-Operand-Instanz muß zusätzlich Protokoll über die schon verwendeten Indexwerte führen, um den nächsten verfügbaren Indexwert - hier maximaler logischer Index genannt - zwecks Zuweisung zu ermitteln. Dabei muß bedacht werden, daß der Index-Operand "SET-INDEX" nur einen Indexwert, der "ADD-INDEX" aber eine ganze Teilindexmenge

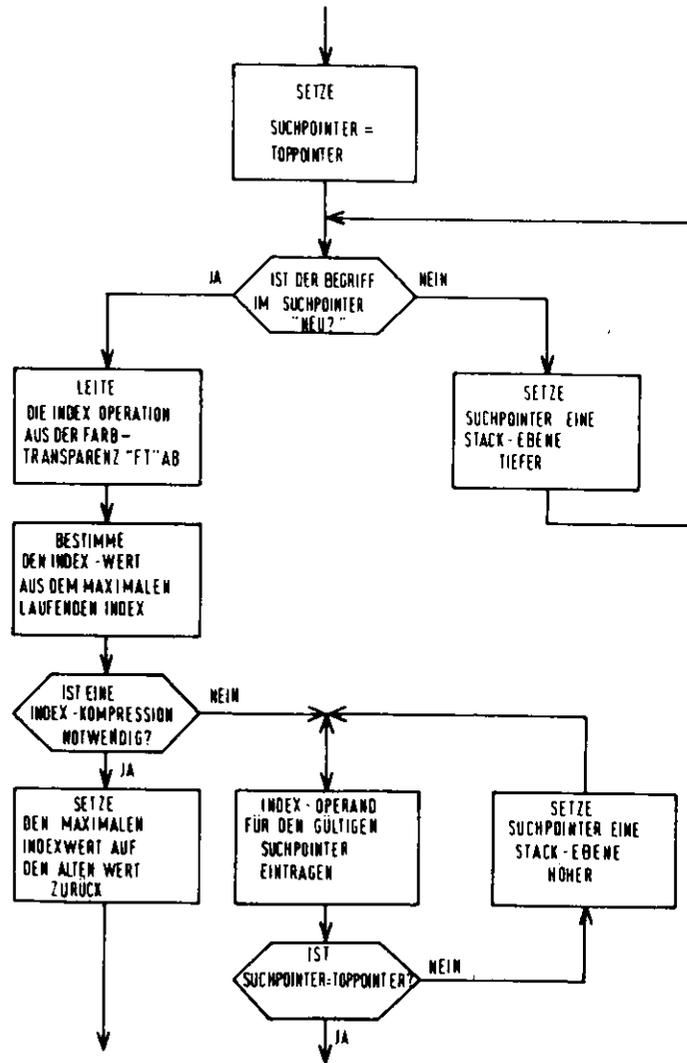


Bild 4.15 Ablaufdiagramm für die Index-Operand-Instanz

benötigt. Das Ablaufdiagramm der Index-Operand-Instanz, aus dem der interne Operationsablauf ersichtlich wird, ist in Bild 4.15 gezeigt.

Eine eindeutige Rücktransformation vom logischen Index zum Begriff wird mit Hilfe der schon erwähnten Begriff-Sequenz-Tabelle erreicht. Die Tabelle 4.3 zeigt die Organisationsform der Begriff-Sequenz-Tabelle für das in Bild 4.13 gegebene Beispiel. Die Tabelle beinhaltet die Sequenz der Index-Operanden, korreliert mit der entsprechenden Begriffsinformation, die aus direktiven und assoziativen Verweisen auf die Datenstruktur besteht. Zusätzlich wird in der Begriff-Sequenz-Tabelle die Konstruktion der Flächengruppen und die Index-Kompression protokolliert. In der Tabelle 4.3 ist beispielsweise der Begriff "Dachfenster" als Intersektionsminterm (m_3) einer Flächengruppe realisiert, die mit einem Quadrat und einem Kreis gebildet wurde. Mit Hilfe dieser Tabelle kann die Rücktransformation vom Indexwert zum Begriff, so wie sie bei den interaktiven Markierungs- und Identifikationskommandos gewünscht ist, immer eindeutig bestimmt werden: z. B. kann der Indexwert 31 (siehe Verteilung der Indexwerte im virtuellen Schirm in Bild 4.16) entsprechend der Begriff-Sequenz-Tabelle nur durch die Konkatenation $ADD18 \circ ADD9 \circ ADD4$ von Indexoperanden entstanden sein. Das zeigt seine Zugehörigkeit entweder zum Begriff "DACHFENSTER", der als Minterm m_3 der Flächengruppe durch $ADD18 \circ ADD9$ repräsentiert ist, oder zum Begriff "DACH", der durch den Indexoperanden $ADD4$ repräsentiert ist. Bei der Betrachtung der Begriff-Sequenz-Tabelle fällt auf, daß die direktiven Verweise nicht den absoluten Astnummern entsprechen, sondern nur die letzte Astnummer relativ zu einem assoziativen Verweis beinhalten. Diese hybride Verweisform ermöglicht die Codierung beliebig langer Astnummern mit einem konstanten Speicheraufwand in der Tabelle, so daß keine Einschränkungen hinsichtlich der Anzahl der Hierarchieebenen in der Datenstruktur notwendig sind. Für die Erzeugung der Begriff-Sequenz-Tabelle muß die Begriff-Sequenz-Instanz, wie in Bild 4.11 gezeigt, jeweils einen lesenden Zugriff zum Begriff- und Index-Operand-Stack durchführen.

Der festgestellte erhöhte Bedarf an Indexwerten kann mit Hilfe der speziell dafür vorgesehenen Indexkompressionsinstanz wie folgt minimiert werden:

1. Bei der Konstruktion einer Flächengruppe entsteht entsprechend der Gleichung 2.14 die theoretische "konstruktive" Teilindexmenge

$$TIM_k = N \cdot 2^k,$$

(4.20)

Tabelle 4.3 Begriff-Sequenz-Tabelle

Index-Operand	Optischer Status	Verweise zur Datenstruktur		Bemerkungen
		direktiv	assoziativ	
-	-	.0	HAUS	
-	-	HAUS.1	UNTEN	
ADD1	WEISS _T	UNTEN.1	WAND	
SET2	GRÜN _D	UNTEN.2	TÜR	
SET3	BLAU _D	UNTEN.3	FENSTER	
-	-	HAUS.2	OBEN	
ADD4	ROT _T	OBEN.1	DACH	
SET8	ROT _D	OBEN.2	SCHORNSTEIN	
-	(GRÜN _D ,N,N)	OBEN.3	DACHFENSTER	Flächengruppe
ADD9	-	-	-	Quadrat
ADD18	-	-	-	Kreis
Index-Belegungstabelle		0, 1, 2, 3, 4, 8, 13, 22, 31		
Index-Kompressionstabelle		0, 1, 2, 3, 4, 5, 4, 4, 6		
SET7				
Index-Belegungstabelle				
Index-Kompressionstabelle				

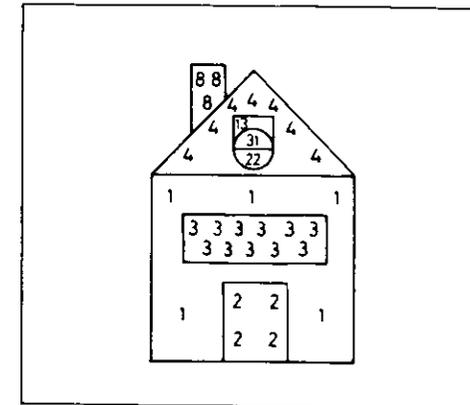


Bild 4.16 Verteilung der Indexwerte im virtuellen Schirm für das Bild 4.13 vor der Indexkompression

wobei ℓ := Anzahl der zu verknüpfenden Flächen,
 N := belegter Indexbereich

bedeuten. Falls für die logisch ausgewählte Menge von Mintermen das gleiche optische Attribut A_0 zugeordnet wird, kann die ganze konstruktive Teilindexmenge auf einen einzigen Indexwert reduziert werden. Grundsätzlich braucht man nur so viele Indexwerte, wie heterogene optische Attribute für die Flächengruppe vorkommen (siehe auch Abschnitt 3.2.1.2). Da die bei der Konstruktion der Flächengruppe verwendeten Indizes nicht logische, sondern konstruktiv bedingte Indizes sind, muß nach jeder Generierung einer Flächengruppe eine Index-Kompression durchgeführt werden, woraus sich die minimale Anzahl der logischen Indizes, die für die Darstellung der unmittelbaren Begriffe notwendig sind, ergibt.

2. Bei der Generierung eines Begriffs mit transparentem optischen Attribut entsteht durch die geometrische Überlagerung die theoretische "optische" Teilindex-

menge

$$TIM_0 = N \cdot 2^1, \quad (4.21)$$

wobei $N :=$ belegter Indexbereich

bedeutet. Topologisch bedingt, überlappt ein Begriff in der Regel nicht alle schon vorher generierten Begriffe im virtuellen Schirm, so daß die reelle optische Teilindexmenge TIM_0 viel kleiner als die theoretische ist. Grundsätzlich braucht man nur so viele logische Indexwerte, wie geometrisch überlappte Begriffe vorkommen. Eine vollständige Indexüberschreibung, die diese Index-Kompressionsmöglichkeit wirkungslos machen würde, kann möglicherweise auf Grund bewußter Manipulationswünsche, beispielsweise für die Überlagerung des gesamten Bildes mit einem Schraffierungsmuster, vorkommen. Da bei dieser Kompressionsmöglichkeit einerseits die zu reduzierenden Indizes schon logische Indizes sind und andererseits das Ergebnis der Kompression maximal die Halbierung des Indexbereiches ergeben kann, wird diese Index-Kompression zweckmäßigerweise nur bei Bedarf, d. h. nur bei einer möglichen Überschreitung des gesamten Indexwertbereiches, ausgeführt.

Diese beiden Index-Kompressionsmöglichkeiten können nur ausgeführt werden, wenn die reelle Belegung des virtuellen Schirmes mit Indexwerten bekannt ist. Aus der reellen Belegung des virtuellen Schirmes und aus dem vorhandenen Abschnitt der Begriff-Sequenz-Tabelle kann durch den sequentiellen Vergleich der theoretischen und reellen Teilindexmengen eine Index-Kompressionstabelle erstellt werden. In der Tabelle 4.3 ist gezeigt, wie die reelle Belegung des virtuellen Schirms (siehe auch Bild 4.16) zur Index-Kompressionstabelle reduziert werden kann. Dabei wird die Index-Belegungstabelle direkt in die Index-Kompressionstabelle abgebildet, d. h. für die Transformation ist nur die Position eines Indexwertes in der Reihenfolge maßgebend. Solche Kompressionsabschnitte, bestehend aus der Index-Belegungs- und Index-Kompressionstabelle, können beliebig oft in der Begriff-Sequenz-Tabelle auftreten, ohne daß die Rücktransformation vom logischen Index zum Begriff fehlerhaft wird. Dabei muß man beachten, daß bei jedem Übergang eines Kompressionsabschnittes die dort vorgenommene Index-Kompression rückgängig gemacht werden muß, was die umgekehrte Abbildung der Index-Kompressionstabelle in die (alte) Index-Belegungstabelle bedeutet. Das Lesen der Index-Belegungstabelle aus dem virtuellen Schirm und das Schreiben der Index-Kompressionstabelle im virtuellen Schirm geschieht, wie in Bild 4.11 veranschau-

licht, über den Index-Kommunikationsraum mit Hilfe der Monitorinstanz, die den unmittelbaren und daher schnellen Zugriff zum Speicher hat. Auf diese Weise kann die Index-Kompression während der Darstellungszeit eines Raster-Scanbildes, also in etwa 20 msek, durchgeführt werden.

Die gesamte Index/Optik-Tabelle als zweite wesentliche Komponente der Begriff/Index-Transformation muß in codierter Form die optische Erscheinungsform jedes logischen Indexwertes beinhalten. Der Ausschnitt aus der gesamten Index/Optik-Tabelle in Tabelle 4.4 zeigt, daß für jeden verwendeten logischen Indexwert die optischen Normalattribute und die Markierungsattribute eingetragen werden müssen. Die optischen Normalattribute resultieren für jeden unmittelbaren Begriff bei der Abwicklung der Datenstruktur. Sie werden, wie in Bild 4.11 gezeigt, von der speziell dafür vorgesehenen Index/Optik-Instanz ausschließlich aus dem Optik-Stack und aus der Begriff-Sequenz-Tabelle abgeleitet. Die optischen Normalattribute werden zweckmäßigerweise nur für die reellen Teilindexmengen eingetragen, was bedeutet, daß die tatsächliche Berechnung der optischen Normalattribute für die konstruktiven oder optischen Teilindexmengen nur nach dem Lesen der Index-Belegungstabelle erfolgen kann. Die Index/Optik-Instanz muß in zwei weiteren Fällen auf die Begriff-Sequenz-Tabelle lesend zugreifen: Einmal nach jeder Index-Kompression, um die gesamte Index/Optik-Tabelle zu komprimieren, und weiter für die Berechnung von optischen Markierungsattributen, die als Folge interaktiver Markierungs- und Identifikationskommandos von der Bildmanipulationsinstanz notwendig ist. Dabei können die Markierungskommandos entweder eine direktive Markierung, d. h. nach Astnummern geordnet, oder eine assoziative

Tabelle 4.4 Ausschnitt der gesamten Index/Optik-Tabelle

Logischer Indexwert	Optisches Attribut	
	Normalattribut	Markierungsattribut
0	SCHWARZ	-
1	WEISS _T	-
2	GRÜN _D	-
3	BLAU _D	-
4	ROT _T	-

Markierung, d. h. nach Begriffsnamen geordnet, spezifizieren. Dies ist, ohne die Datenstruktur neu abwickeln zu müssen, möglich, weil die Begriff-Sequenz-Tabelle sowohl direktive als auch assoziative Verweise zur Datenstruktur beinhaltet. Falls vorhanden, haben die Markierungsattribute jedoch eine höhere Priorität als die Normalattribute, d. h. sie werden anstelle der Normalattribute für die optische Gestaltung des logischen Indexbildes verwendet.

Wegen ihres günstigen Zugriffes sowohl zur gesamten Index/Optik-Tabelle als auch zur Begriff-Sequenz-Tabelle wird die Index/Optik-Instanz gleichzeitig für die bidirektionale Kommunikation mit der interaktiven Bildmanipulationsinstanz benutzt. Markierungs- und Identifikationskommandos können in diesem Falle nicht nur Veränderungen des optischen Bildes hervorrufen, sondern zu einem direkten Dialog mit der Bildmanipulationsinstanz führen.

4.1.2.3. Auswertung freier Attribute

Die informationelle Struktur für die hier ausschließlich behandelte Auswertung der Stil- und Zugriffsrechtattribute, deren Bedeutung und Funktion bei der Beschreibung der Datenstruktur eingeführt wurden, ist in Bild 4.17 veranschaulicht. Dies ist eine Verfeinerung der Struktur in Bild 4.4 und zeigt eine

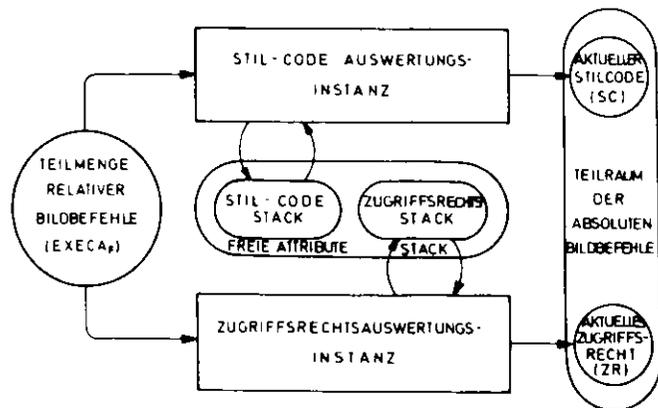


Bild 4.17 Informationelle Struktur für die Auswertung freier Attribute

typische Attributauswertungsinstanz, die einerseits den aktuellen Attributenstatus in einem Stack protokolliert und andererseits Teilräume des absoluten Bildbefehls erzeugt. Maßgebend für die interne Struktur dieser Instanzen ist die Festlegung geeigneter Auswertungsalgorithmen für die Stil- und Zugriffsrechtsattribute.

Durch die Angabe der Stil-Attribute in der Datenstruktur ist man, ähnlich wie bei den optischen Attributen, sinnvollerweise darauf bedacht, den Einfluß des einzelnen Stil-Attributes für die Bestimmung des Stil-Musters mit der Höhe der Hierarchieebene zu vergrößern. Auf diese Weise ist es möglich, durch die Spezifizierung eines einzigen Stil-Attributes das Stil-Muster aller hierarchisch untergeordneter Datenobjekte entscheidend zu bestimmen. Der aktuelle Stil-Status

$$\text{STATUS}_{SC}^n := (\text{SC}_S) \quad (4.22)$$

mit $\text{SC}_S :=$ aktueller Stil-Code des Status

verändert sich auf Grund eines Stil-Attributbefehls

$$A_{SC}^n := (\text{SC}_A) \quad (4.23)$$

mit $\text{SC}_A :=$ Stil-Code des Attributes

entsprechend der Übergangsfunktion

$$\text{STATUS}_{SC}^{n+1} = f(\text{STATUS}_{SC}, A_{SC}^n). \quad (4.24)$$

wobei folgende Beziehungen gelten:

$$\text{STATUS}_{SC}^{n+1} := \begin{cases} \text{für } A_{SC}^n := \text{Stil-Neutral} : \text{SC}_S^n \\ \text{für } A_{SC}^n \neq \text{Stil-Neutral} : \text{SC}_A^n \end{cases} \quad (4.25)$$

Die Gleichungen (4.25) besagen, daß die stilneutral spezifizierten Datenobjekte das Stilmuster der ihnen direkt übergeordneten Datenobjekte übernehmen, während alle anderen Datenobjekte das angegebene Stilmuster bekommen. Dem so resultierenden aktuellen Stil-Code wird in der Bildpunktinstanz das tatsächliche

Stil-Muster zugeordnet, das das ein- oder zweidimensionale Bitmuster als Kriterium für die Generierung der Bildpunkte im virtuellen Schirm beinhaltet. Beispielsweise spezifiziert ein Vektor-Stil-Muster $SM := (10101010)$ einen gepunkteten Vektor, wobei der erste Punkt sichtbar ist. Die Verwendung des Stil-Musters (SM) als direktes binäres Kriterium zur Bildpunktgenerierung unterliegt zwei wesentlichen Einschränkungen:

Die erste Einschränkung betrifft die von dem Anwender gewünschte Veränderung des Stil-Musters im Zusammenhang mit dem mathematisch-geometrischen Bildstatus. Man möchte beispielsweise, daß bei einer Bildskalierung auch das Stil-Muster entsprechend skaliert wird. Damit ist ein eindimensionales Stil-Muster vom Skalierungsstatus, ein zweidimensionales Stil-Muster sogar vom Skalierungs- und Rotationsstatus abhängig. Eine in diesem Sinn korrekte Generierung des Stil-Musters ist nur dann möglich, wenn in der Bildpunktinstanz ein transformiertes Stil-Muster

$$SM_T := f(SM, ROT, SF) \quad (4.26)$$

als Funktion des tabellarisch angegebenen Stil-Musters (SM) und des mathematisch-geometrischen Bildstatus berechnet wird. Das bedeutet, daß die Bildpunktinstanz zusätzlich einen lesenden Zugriff auf den mathematisch-geometrischen Bildstatus haben muß. In dieser Arbeit wird jedoch diese Forderung nicht weiter betrachtet.

Die zweite Einschränkung betrifft die Interpretation des Stil-Codes (SC) während der Generierung von Konturen für die polygonalen Flächen (F^0) oder von Flächengruppen (G^n). In diesen Fällen müssen die Bildpunkte im virtuellen Schirm ungeachtet des geltenden Stil-Codes mit einem kontinuierlichen Stil-Muster $SM := (11111111)$ generiert werden. Das bedeutet, daß die Generierung des Stil-Musters und des logischen Indexbildes nicht immer gleichzeitig stattfinden kann. Dadurch wäre die Effizienz der Bildpunktgenerierung empfindlich vermindert. In der vorliegenden Arbeit wird daher nur der Kompromißfall betrachtet, bei dem die elementaren Flächen F^0 , nicht jedoch die hierarchischen Flächen (F^n), ein beliebiges Stil-Muster haben dürfen.

Diese beiden wesentlichen Einschränkungen sind grundsätzlich darauf zurückzuführen, daß der Stil keine logische Eigenschaft der Bilder ist und daher auch nicht in der Datenstruktur vollständig spezifiziert werden kann. Da das Stil-Muster ein Sonderfall eines Beleuchtungsmodells ist, kann die hier dargestellte

Problematik der Auswertung von Stil-Attributen nur durch Einführung eines Beleuchtungsmodells zufriedenstellend gelöst werden (siehe auch Abschnitt 2.1.2.). Dafür wäre es aber notwendig, für jeden logischen Indexwert eine ganze Teilmenge optischer Indexwerte vorzusehen, die die Chrominanzverteilung des gewünschten Stil-Musters eines Begriffes darstellt. Das wird jedoch im Rahmen dieser Arbeit nicht weiter betrachtet.

Für die Auswertung der Zugriffsrechtsattribute wird vorausgesetzt, daß einerseits für jedes Datenobjekt in der Datenstruktur und andererseits für jeden Anwender ein relatives Zugriffsrechtsattribut spezifiziert wird. Aus diesen beiden Angaben muß die Zugriffsrechtsauswertungsinstanz ein absolutes Zugriffsrecht ableiten, das als binäres Kriterium für die Sichtbarkeit des Datenobjektes im optischen Bild gilt. Das relative Zugriffsrecht eines Datenobjektes ist durch das Zugriffsrechtsattribut

$$A_{DIC}^n := (DIC_A) \quad (4.27)$$

mit $DIC := \underline{\text{Datenobjekt-Identifikations-Code}}$

spezifiziert. Die Veränderung des Zugriffsrechtsstatus

$$STATUS_{DIC}^n := (DIC_S) \quad (4.28)$$

durch das Zugriffsrechtsattribut A_{DIC}^n hat die Übergangsfunktion

$$STATUS_{DIC}^{n+1} = f(STATUS_{DIC}, A_{DIC}^n), \quad (4.29)$$

wobei für die Funktion f der Ausdruck

$$f = \max(DIC_S, DIC_A)^n \quad (4.30)$$

gilt. Das relative Zugriffsrecht eines Anwenders zur Datenstruktur ist durch einen Deskriptor, hier Anwender-Identifikations-Code (AIC) genannt, angegeben. Das absolute Zugriffsrecht des Anwenders auf ein Datenobjekt wird durch einen frei definierbaren logischen Ausdruck

$$G_{ZR} = g(STATUS_{DIC}^n, AIC) \quad (4.31)$$

berechnet, wobei folgendes binäres Kriterium gilt:

- für $G_{ZR} = \text{"WAHR"}$ wird das Datenobjekt mit seinen optischen Normalattributen dargestellt,
- für $G_{ZR} = \text{"UNWAHR"}$ wird das Datenobjekt mit den optischen Normalattributen des ihm unmittelbar übergeordneten Datenobjektes dargestellt und ist somit optisch kein auflösbarer Begriff.

4.1.2.4. Kontrollinstanz

Die Kontrollinstanz ist neben der schon beschriebenen Aktualisierung des gesamten Bildabwicklungsstacks durch PUSH- und POP-Befehle hauptsächlich für die Erzeugung des Operations-Codes für den absoluten Bildbefehl zuständig. Sie hat daher einen modifizierenden Zugriff zum Teilraum EXEC OP des absoluten Bildbefehls (siehe Bild 4.4). Da auf Grund eines relativen Bildbefehls kein, ein oder aber eine ganze Reihe von absoluten Bildbefehlen erzeugt werden müssen, kann die Kontrollinstanz in diesem Sinne als ein einfacher Interpreter mit vorgegebenem Repertoire betrachtet werden. Jeder relative Bildbefehl wird dabei zu einer programmartigen Sequenz von aktuellen Bildbefehlen expandiert, wobei manche Sequenzen nur einen einzigen absoluten Bildbefehl beinhalten.

Wie schon angedeutet, können die aktuellen Bildbefehle Bildpunktgenerierungs- oder Bildverarbeitungsvorgänge im virtuellen Schirm bewirken. Die interne Struktur der Kontrollinstanz wird hier nicht weiter beschrieben.

4.1.3. Bildpunktinstanz

Die Bildpunktinstanz hat die Aufgabe, aus den absoluten Bildbefehlen die einzelnen Bildpunkte als logische Indexwerte in dem virtuellen Schirm korrekt und möglichst schnell zu erzeugen (siehe Bild 4.1). Da die Bildpunktinstanz die einzige Instanz der Bilddarstellung ist, die einen direkten Zugriff zu dem virtuellen Schirm hat, ist es zweckmäßig, sie zusätzlich für die Ausführung allgemeiner Bildverarbeitungsoperationen einzusetzen. Die resultierende informationelle Struktur ist in Bild 4.18 gezeigt. Wegen der sich stark unterscheidenden Algorithmen für die Bildpunktgenerierung und für die Bildpunktverarbeitung wurden dedizierte Instanzen jeweils für die Vektor- und Flächengenerierung sowie

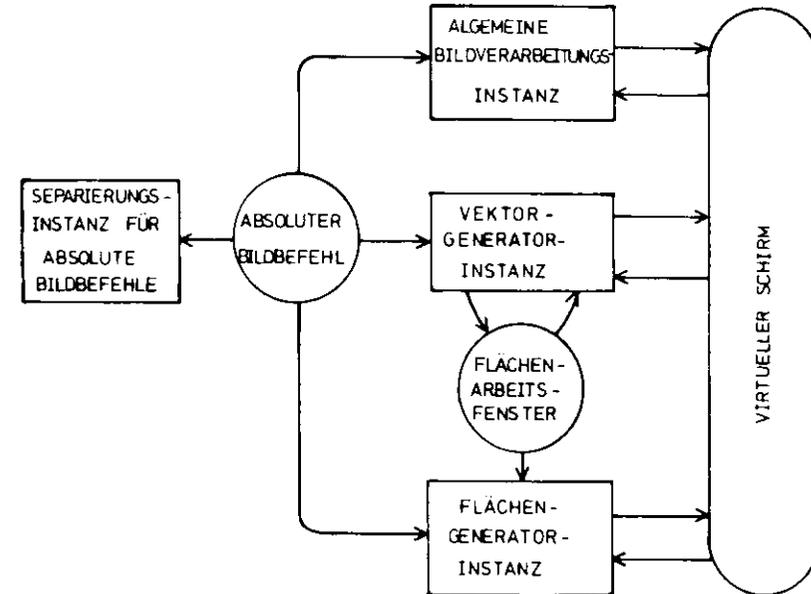


Bild 4.18 Datenflußgraph der Bildpunktinstanz

für die Bildverarbeitung vorgesehen. Aus dem Aktionsablauf in Bild 4.19 geht hervor, daß, obwohl die drei spezialisierten Instanzen schon allein aus logischen Gründen nicht parallel ablaufen dürfen, eine beachtliche Steigerung des Durchsatzes durch die Pipeline-Struktur zwischen den Bildpunktberechnungsoperationen und dem Zugriff zum virtuellen Schirm erreicht werden kann.

Die informationellen Strukturen der einzelnen dedizierten Instanzen, die allein durch die durchzuführenden Algorithmen bestimmt sind, werden im folgenden näher erläutert.

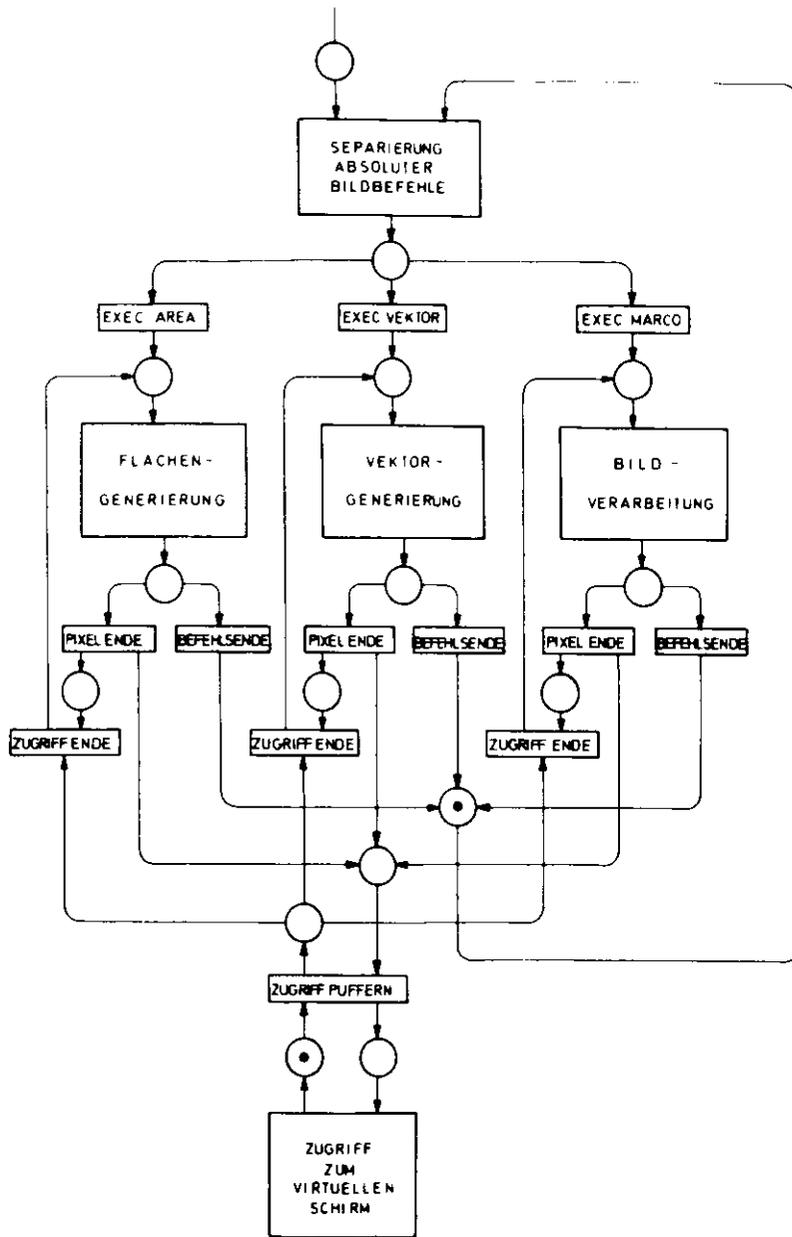


Bild 4.19 Aktionsablauf in der Bildpunktinstanz

4.1.3.1. Vektorgenerator

Algorithmen für die Vektorgenerierung, die im allgemeinen anstreben, mit einfachen operationellen Mitteln und möglichst schnell ausreichend "gerade" Vektoren für die gegebene Schirmauflösung zu generieren, sind in der Literatur erschöpfend behandelt worden /THOMSON-64, BRESHENHAM-65, NATHERAT-78, EARNSHAW-79/ und brauchen daher hier nicht beschrieben zu werden. Bei der Vektorgenerierung in einem virtuellen Schirm mit endlicher Auflösung und vorgegebener Zugriffszeit muß man besonders auf zwei Eigenschaften des verwendeten Vektoralgorithmus achten: Erstens muß der Algorithmus möglichst einfach und damit vergleichbar schnell mit der Zugriffszeit des virtuellen Schirmes sein, und zweitens darf die Güte des resultierenden Vektors die räumliche Auflösung des virtuellen Schirmes nicht unterschreiten. Ein von /NATHERAT-78/ inspirierter Vektoralgorithmus, der aus einem idealen Vektor

$$y = \frac{\Delta y}{\Delta x} \cdot x \quad (4.32)$$

einen realen Vektor als Punktmenge

$$\{x_i, y_i \mid i = 1, 2, \dots, n \text{ und } y_i \leq \frac{\Delta y}{\Delta x} \cdot x_i + \frac{1}{2}\} \quad (4.33)$$

im Raster des virtuellen Schirmes erzeugt, ist im Anhang 6.1 im Pascal angegeben. Mit diesem Algorithmus werden nur die Adressen der jeweiligen Vektorpunkte im virtuellen Schirm berechnet, die die geometrische Position auf dem Bildschirm bestimmen. Da die in dieser Arbeit definierten Vektoren zusätzliche Eigenschaften, wie Index-Operand und Stil-Muster, besitzen, müssen diese Eigenschaften vor der Ausgabe des einzelnen Vektorpunktes auch berücksichtigt werden. Der Datenflußgraph für den Vektorgenerator ist dem Bild 4.20 zu entnehmen. Der Vektorgenerator wird gleichzeitig für die Berechnung des Flächenarbeitsfensters eingesetzt, da sich das Arbeitsfenster ohnehin bei der Generierung von polygonalen Flächenkonturen ergibt.

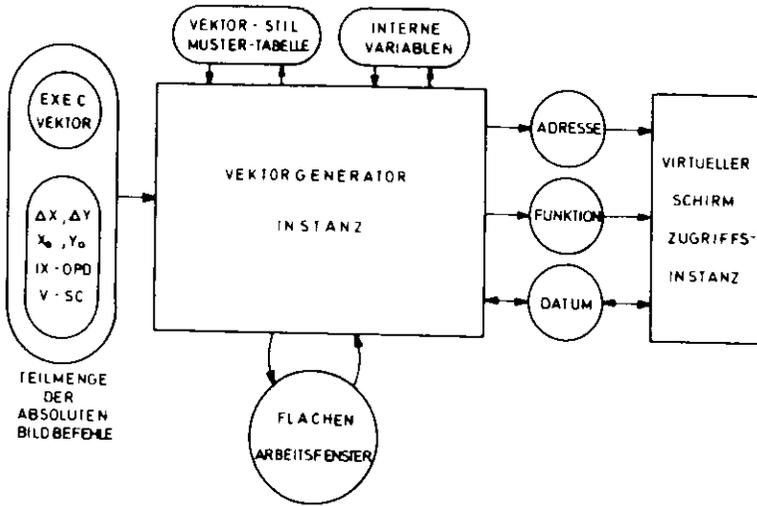


Bild 4.20 Informationelle Struktur des Vektorgenerators

4.1.3.2. Flächengenerator

Flächengenerierungsalgorithmen, die die korrekte Füllung von polygonalen Vektorzügen durch die Identifizierung der Konturpunkte ermöglichen, operieren zeilenweise auf das Arbeitsfenster der Fläche. Dies ist schematisch in Bild 4.21 veranschaulicht. Durch die Identifizierung der Konturpunkte wird das Arbeitsfenster in zwei Punktmengen eingeteilt: eine Punktmenge mit geraden Abtast-Zustandswerten, die nicht zu der Fläche gehörende Punkte kennzeichnet, und eine zweite Punktmenge mit ungeraden Abtast-Zustandswerten, die die Fläche selbst darstellt. Um die in Abschnitt 2.2.1 erläuterte Problematik bezüglich der korrekten Flächengenerierung mit einem vertretbaren Aufwand lösen zu können ist im Anhang 6.2 ein Flächengenerierungsalgorithmus vorgeschlagen, der für die korrekte Identifizierung der Konturpunkte gleichzeitig drei Zeilen im Arbeitsfenster untersucht. Es handelt sich hier um eine Modifikation des Algorithmus in /Pavlidis-78/. Zwei interne Variablen kennzeichnen, ob es sich bei einem Konturpunkt um einen normalen oder singulären Pixel handelt.

Als singuläre Pixel gelten Konturmaxima, Konturminima oder benachbarte Konturpixel. Da sich jede beliebige Struktur, die den vorgeschlagenen Flächengenerierungsalgorithmus durchführen kann, theoretisch als Flächengenerator eignet, braucht hier auf die informationelle Struktur des Flächengenerators nicht weiter eingegangen zu werden.

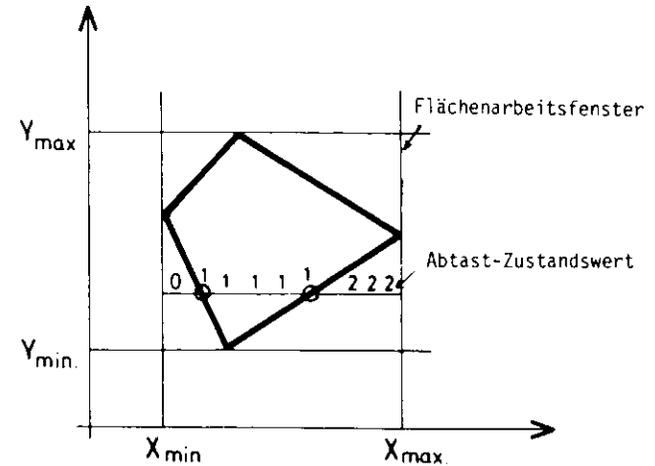


Bild 4.21 Zur Veranschaulichung der Flächengenerierung auf der Basis der Abtast-Zustandswerte

4.1.3.3. Bildverarbeitungsinstanz

Die Bildverarbeitungsinstanz sind im allgemeinen so anwendungsspezifisch und daher so vielfältig, daß ihre nähere Beschreibung hier nicht gegeben werden kann. Typisch für die Bildverarbeitungsinstanz ist jedoch die Tatsache, daß bei der Bildverarbeitung arithmetisch-logische Operationen auf eine ganze Nachbarschaft eines jeden Bildpunktes durchgeführt werden (siehe Gleichung 4.2). Das bedeutet, daß einerseits das Anweisungsrepertoire der Bildverarbeitungsinstanz allgemein und reichhaltig gestaltet und andererseits die Operandenbeschaffung aus dem virtuellen Schirm schnell durchgeführt werden muß. Die in Bild 4.22 vorgeschlagene informationelle Struktur benutzt zu diesem Zwecke eine programmierbare Instanz mit festem Befehlsrepertoire und einen eigenen Datenspeicher, mit Hilfe dessen auf einen für die laufende Bildverarbeitung aktuellen Bildausschnitt schnell zugegriffen werden kann.

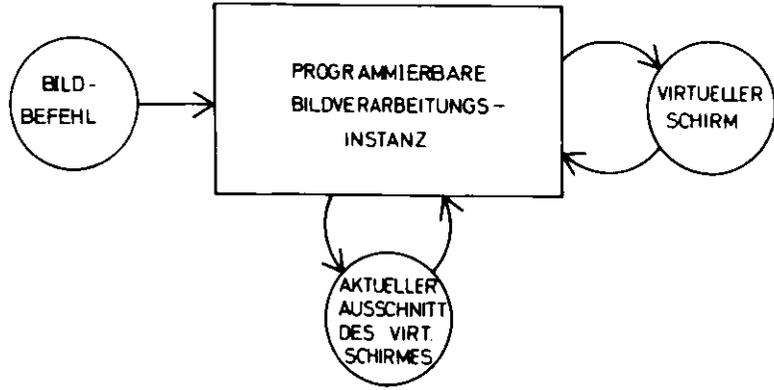


Bild 4.22 Struktur der Bildverarbeitungsinstanz

4.1.4. Monitorinstanz

Die Monitorinstanz hat eine doppelte Aufgabe durchzuführen, die schon bei der Beschreibung der groben informationellen Struktur in Bild 4.1 angedeutet wurde: Die Monitorinstanz muß das logische Indexbild vom virtuellen Schirm auf den reellen Raster-scan-Bildschirm geometrisch und optisch abbilden, wobei die Zuweisung der optischen Attribute für die logischen Indizes während des Darstellungsprozesses stattfindet (siehe Abschnitt 2.1.4). Außerdem muß die Monitorinstanz mit den benachbarten Instanzen kommunizieren, wodurch entweder verschiedene Tabellen aus dem Index-Kommunikationsraum ausgetauscht werden oder auf Grund interaktiver Darstellungskommandos der Dialog mit der Bildmanipulationsinstanz durchgeführt wird. Die informationelle Struktur der Monitorinstanz ist in Bild 4.23 vorgeschlagen. Sie besteht entsprechend den beiden semantisch unterschiedlichen Aufgaben aus zwei miteinander kommunizierenden (Unter-)Instanzen: dem Bildrefresher und der Optik-Transformationsinstanz. Bei der Modellierung der informationellen Strukturen dieser Instanzen muß man stets beachten, daß die Abläufe in den beiden Instanzen völlig unterschiedlichen Zeitbedingungen unterliegen: Während der Darstellungsprozeß der logischen Indexwerte im Bildrefresher

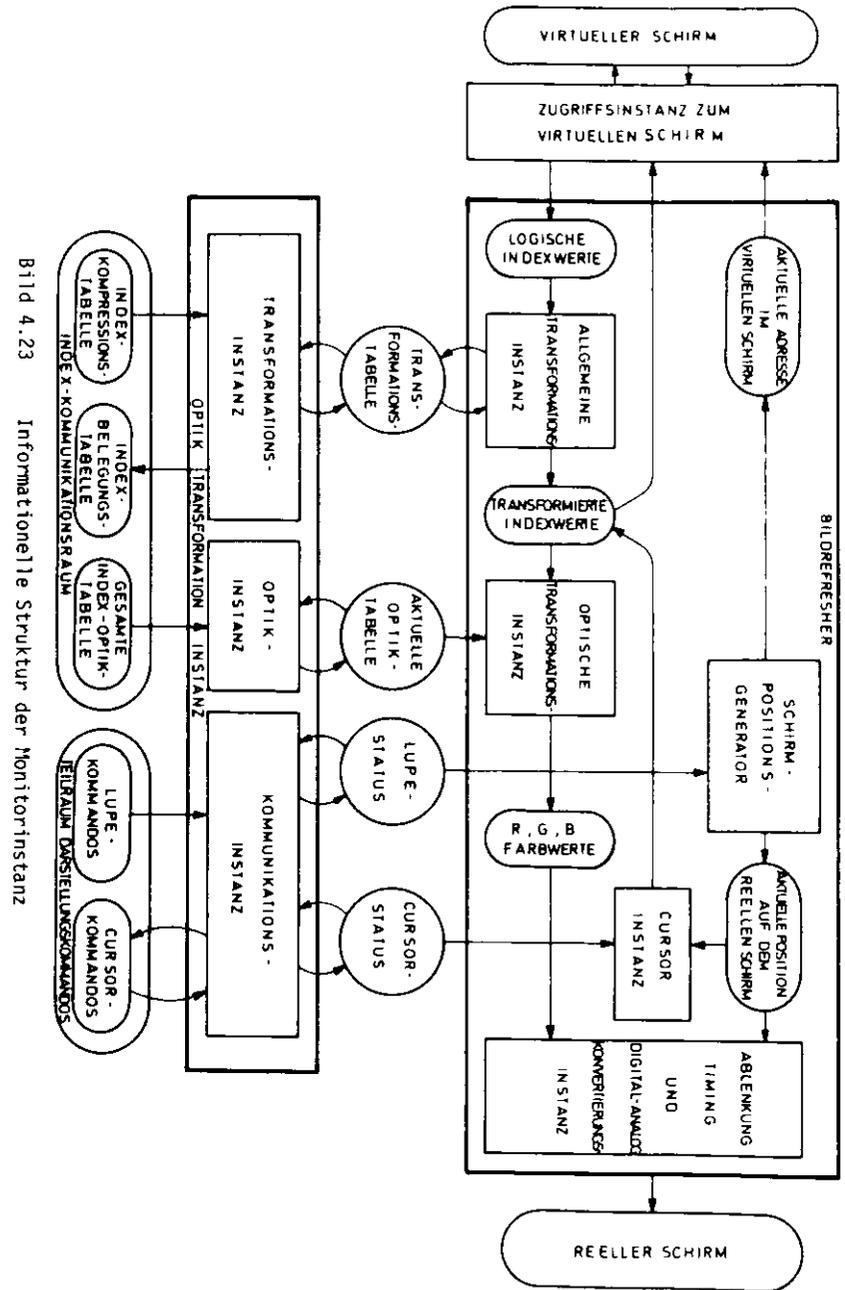


Bild 4.23 Informationelle Struktur der Monitorinstanz

mit der Pixeldarstellungsrate von 10 Megapixeln/sek ablaufen muß, brauchen die Kommunikationsprozesse in der Optik-Transformationsinstanz nur die Videobildfrequenz von 50 Bildern/sek zu berücksichtigen. Dies bedeutet, daß der Bildrefresher höchst parallel, die Optik-Transformationsinstanz dagegen beliebig sequentiell strukturiert werden kann. Die interne Struktur des Bildrefreshers und der Optik-Transformationsinstanz wird im folgenden näher beschrieben.

4.1.4.1. Bildrefresher

Entsprechend seiner Aufgabe, den virtuellen Schirm geometrisch und optisch auf den reellen Bildschirm abzubilden, kann der Bildrefresher in zwei funktionelle Teile zerlegt werden:

Für die geometrische Abbildung des virtuellen Schirmes beinhaltet der Bildrefresher einen Schirmpositionsgenerator, der für jede aktuelle Position auf dem reellen Bildschirm die zugehörige Adresse im virtuellen Schirm ableitet. Die parallele Berechnung der reellen und virtuellen Schirmposition erlaubt, daß die geometrische Abbildung des virtuellen Schirmes auf den reellen Bildschirm entsprechend dem Lupe-Status beliebig verändert werden kann. Dadurch können spezielle Abbildungseffekte wie Window, Zoom oder Pan in Echtzeit erzielt werden, ohne das logische Bild im virtuellen Schirm verändern zu müssen. Solche Skalierungsoperationen auf das optische Bild, die keine logische Eigenschaft der Bilder darstellen und daher in der Datenstruktur nicht spezifiziert werden können, haben für die interaktive Bildmanipulation große Bedeutung.

Für die optische Abbildung des virtuellen Schirmes müssen die logischen Indexwerte aus dem virtuellen Schirm in optische Indexwerte transformiert werden, die lediglich Chrominanzvektoren darstellen. Diese Transformation findet - vorwiegend aus semantischen Gründen - in zwei Phasen statt:

- Zuerst wird die allgemeine Transformation der logischen Indexwerte durchgeführt. Dadurch werden die logischen Indexwerte aus dem virtuellen Schirm entsprechend einer vorgegebenen Transformationstabelle wieder in logische Indexwerte transformiert. Bei dieser Transformation werden die logischen Indexwerte aus dem virtuellen Schirm - unabhängig von ihrer ursprünglichen Bedeutung als Begriffscodes - als abstrakte mathematische Operanden betrachtet, die nach irgendwelchen Gesetzmäßigkeiten transformiert werden müssen. Der Rückkopplungspfad über die allgemeine Transformationsinstanz in Bild 4.23 deutet darauf

hin, daß die transformierten Indexwerte gleichzeitig auch zurück in den virtuellen Schirm geschrieben werden können. Mit Hilfe der allgemeinen Transformation können beispielsweise die Index-Kompression im virtuellen Schirm oder beliebige monadische Bildtransformationen - wie logarithmische oder exponentielle Transformationen - während der Darstellungszeit eines einzigen Videobildes durchgeführt werden.

- Nach der allgemeinen Transformation findet die optische Transformation statt, die aus den transformierten logischen Indexwerten optische Indexwerte erzeugt, die zweckmäßigerweise direkt als R,G,B-Farbwerte codiert sind. Bei dieser Transformation werden den ankommenden logischen Indexwerten, die ursprünglich als Begriffscodes betrachtet wurden, die zugehörigen optischen Attribute entsprechend der aktuellen Optik-Tabelle dynamisch zugewiesen.

Die allgemeine und die optische Transformation müssen jeweils innerhalb der Darstellungszeit eines Bildpunktes stattfinden und können daher nur mit Look-up-Tabellen implementiert werden, die zusammen eine Pipeline-Struktur bilden.

Die in Bild 4.23 gezeigte Cursor-Instanz dient der Einblendung des Cursors in das optische Bild, wobei die geometrische Position, die Form und die Optik des Cursors vom Cursor-Status angegeben wird. Um die Optik des Cursors dynamisch verändern zu können, muß seine Einblendung vor der optischen Transformation geschehen. Das ist in Bild 4.23 durch den schreibenden Zugriff der Cursor-Instanz auf dem Raum "transformierte Indexwerte" angedeutet. Die triviale Instanz für Ablenkung und Timing des Rasterscanbildes sowie für die Digital-Analog-Konvertierung der RGB-Farbwerte ist im allgemeinen bekannt und braucht hier nicht beschrieben zu werden.

4.1.4.2. Optik-Transformationsinstanz

Die Aufgabe der Optik-Transformationsinstanz besteht darin, im Rahmen der Kommunikation mit der Bildtransformations- und Bildmanipulationsinstanz zusätzliche Information, die bei der Darstellung des optischen Bildes notwendig ist, für den Bildrefresher aufzubereiten. Diese Information umfaßt die allgemeine und die optische Transformationstabelle und den Lupe- und Cursor-Status. Diese Information befindet sich in einzelnen Kommunikationsräumen zwischen dem Bildrefresher und der Optik-Transformationsinstanz, zu denen die Optik-Transformationsinstanz stets einen modifizierenden, der Bildrefresher aber nur einen le-

senden Zugriff haben. Funktionell kann die Optik-Transformationsinstanz in drei Unterinstanzen zerlegt werden:

- Die Transformationsinstanz ist für die Verwaltung der allgemeinen Transformationstabelle zuständig. In Bild 4.23 ist die Belegung der allgemeinen Transformationstabelle für zwei Sonderfälle gezeigt. In einem wird die Index-Kompressionstabelle aus dem Index-Kommunikationsraum in die allgemeine Transformationstabelle transferiert, um die Index-Kompression im virtuellen Schirm in Echtzeit durchzuführen. In dem anderen erzeugt der Bildrefresher in einem Ausschnitt der allgemeinen Transformationstabelle eine binäre Aufstellung der Belegung des virtuellen Schirmes mit Indexwerten, die dann der Bildtransformationsinstanz als Index-Belegungstabelle angeboten wird (siehe Abschnitt 4.1.2.2). Daß der Bildrefresher einen bestimmten Ausschnitt der allgemeinen Transformationstabelle beschreiben darf, ist in Bild 4.23 durch seinen modifizierenden Zugriff zu diesem Zustandsraum zum Ausdruck gebracht. Darüber hinaus kann die allgemeine Transformationstabelle mit beliebigen Funktionen belegt werden, z. B. mit den schon erwähnten monadischen Bildpunkttransformationen.
- Die Optik-Instanz ist für die Belegung der aktuellen Optik-Tabelle zuständig, die die momentan geltende Zuordnung optischer Attribute zu den logischen Indizes bestimmt. Für die Erzeugung der aktuellen Optik-Tabelle muß die Optik-Instanz die gesamte Index/Optik-Tabelle lesen und aus den Normal- und Markierungsattributen direkt die R,G,B-Farbwerte für jeden logischen Index berechnen.

- Die Kommunikationsinstanz ist für die Erzeugung des Lupe- und Cursor-Status zuständig. Sie kommuniziert über einen Teilraum der Darstellungskommandos direkt mit der Bildmanipulationsinstanz (siehe Bild 4.1). Da die Lupe-Kommandos nur Anweisungen zur Modifikation des aktuellen Lupe-Status beinhalten, findet der Dialog über diesen Teil nur in einer Richtung statt, und zwar zur Kommunikationsinstanz hin. Die Cursor-Kommandos können neben den Anweisungen für die Cursor-Manipulation auch Abfragen bezüglich der aktuellen Position des Cursors oder des getroffenen logischen Index beinhalten, so daß für deren Beantwortung ein bidirektionaler Dialog erforderlich ist.

4.2. Bilderfassungsinstanz

Die informationelle Struktur der Bilderfassungsinstanz ist als Datenflußgraph in Bild 4.24 dargestellt. Sie beinhaltet wie die im Abschnitt 1.2.2 einge-

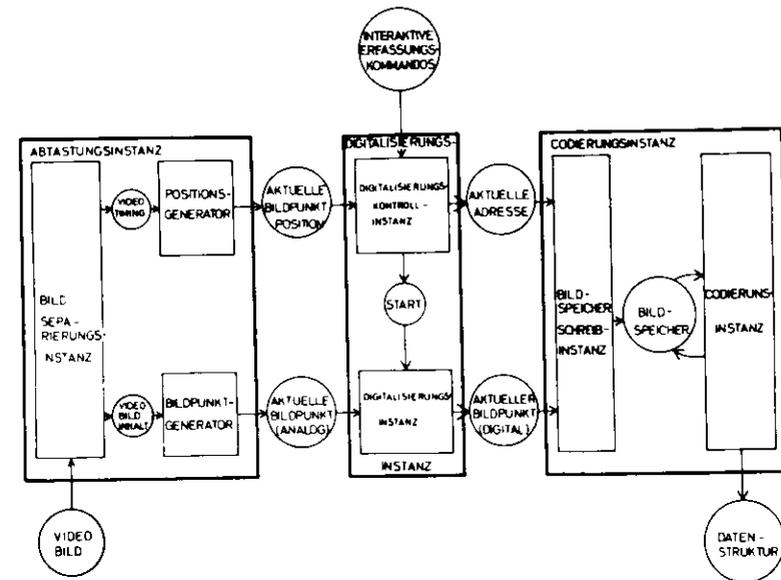


Bild 4.24 Datenflußgraph der Bilderfassungsinstanz

führte Grundstruktur drei spezialisierte Instanzen für die Abtastung, Digitalisierung und eventuelle Codierung der Videobilder. Bei dieser Struktur fällt abweichend von der in Bild 1.8 zweierlei auf: Erstens ist die Digitalisierungsinstanz mit einer Kontroll-Instanz angereichert, die es bei Bedarf ermöglicht, nur bestimmte Bildbereiche zu digitalisieren. Diese gewöhnlich als Rechteck definierbaren Bildbereiche werden mit Hilfe interaktiver Erfassungskommandos von der Bildmanipulationsinstanz vorgegeben. Diese Programmierbarkeit des erfaßten Bildbereiches ermöglicht eine Datenreduktion gleich bei der Erfassung und führt somit zu einer besseren Ausnutzung der Bandbreite des Analog-Digital-Konverters und der vorhandenen Bildspeicherkapazität. Zweitens werden die digitalisierten

5. Implementierung

Dieses Kapitel befaßt sich mit den wesentlichen Implementierungsaspekten des hier entwickelten Raster-Scan-Display-Systems. Als Grundlage für die Implementierung dienen die in dem vorangegangenen Kapitel modellierten informationellen Strukturen, die die Struktur und Funktion des Systems festlegen. Bei der Implementierung werden vornehmlich die Realisierungsaspekte der einzelnen Instanzen behandelt, wobei Fragestellungen bezüglich der Architektur, Spezifikation, Konstruktion, Verifikation und nicht zuletzt der Dokumentation Vordergrund sind. Die transparente Darstellung der informationellen Strukturen hat dazu geführt, daß wesentliche Entwurfsentscheidungen objektiv und daher effizient getroffen werden konnten.

Den praktischen Hintergrund für die nachfolgend behandelten Implementierungsaspekte bildet ein im Rahmen eines Projektes für die Digitale Echtzeit-Verarbeitung medizinischer Bildserien realisiertes System /HÖHNE-78, HÖHNE-79/, mit Hilfe dessen die Wirkungsweise der hier dargestellten Strukturen in der Praxis erprobt werden konnte /NICOLAE-79-1/.

5.1. Multiprozessor-Systemarchitektur

Das Display-System wurde wegen der erforderlichen Parallelität und Flexibilität seiner informationellen Struktur als Multiprozessorsystem realisiert /NICOLAE-79-2/. Die Architektur des Display-Systems ist in Bild 5.1 dargestellt. Das System besteht aus mehreren dedizierten Prozessoren, die über einen schnellen asynchronen Bus miteinander verbunden sind. Die verschiedenen Prozessoren sind funktionell spezialisiert, so daß die zeitkritischen Prozesse in dem Display-System gezielt unterstützt werden können. Die notwendigen Datenwege werden durch den Bus zwischen beliebigen Prozessorenpaaren im Zeitmultiplexverfahren umgeschaltet, so daß mehrere Prozesse im System "gleichzeitig" ablaufen können. Der asynchrone Bus unterstützt das Ablaufen von Prozessen unter Echtzeit-Bedingungen, d. h. er garantiert minimale Vergabe- und Abgabezeiten für die Datenwege und wird daher Real-Time-Asynchronous-Bus (RTA-Bus) bezeichnet. Die Abbildung der informationellen Struktur des Display-Systems auf die physikalische Konfiguration in Bild 5.1 ist aus der Tabelle 5.1 zu entnehmen. Bei der Betrachtung dieser Tabelle fällt auf, daß für die zweckmäßige Implementierung des Display-Systems die Aufgaben mehrerer informationeller Instanzen durch einen einzigen

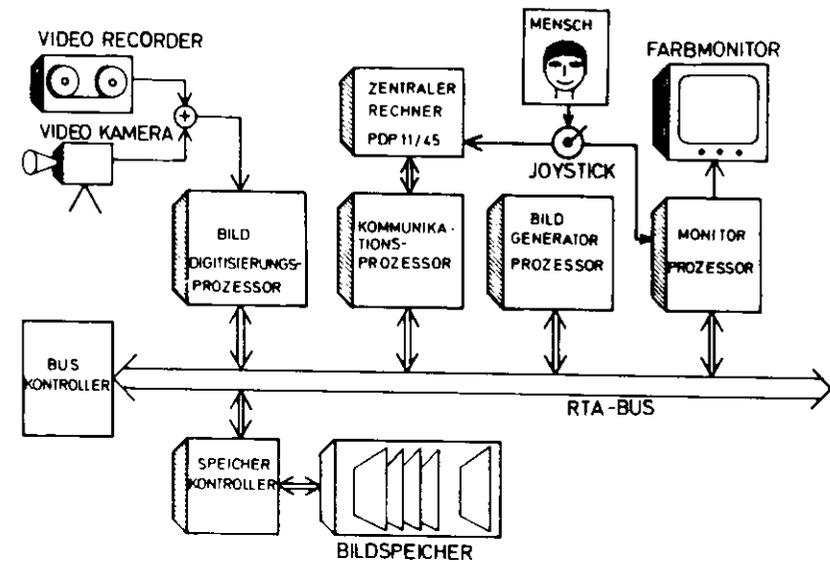


Bild 5.1 Architektur des Raster-Scan-Display-Systems

physikalischen Prozessor in Zeitmultiplex (Instanzenmultiplex) realisiert werden können. Dies schließt jedoch nicht aus, daß Teilaufgaben einer informationellen Instanz auf mehrere physikalische Prozessoren (Abwicklungsmultiplex) verteilt werden dürfen.

5.1.1. Funktionelle Systembeschreibung

Die Erfassung der Abbildinformation findet mit Hilfe des Bild-Digitalisierungsprozessors statt, der Videobilder aus beliebigen Quellen - z. B. Videokameras oder Videorecordern - in Echtzeit erfassen und digitalisieren kann. Das Bild 5.2 zeigt ein digitalisiertes Videobild in einer Auflösung von 256 x 256 Pixeln mit jeweils 8 Bits/Pixel. Um eine Datenreduktion schon während der Bilderfassung zu realisieren, können sowohl die räumliche Auflösung als auch das Aufnahmeformat der Bilder programmiert vorgegeben werden.

Tabelle 5.1 System-Konfiguration

SYSTEM KOMPONENTEN		ZENTRALER RECHNER	KOMMUNIKAT. PROZESSOR	BILD- SPEICHER	BILD- GENERATOR- PROZESSOR	MONITOR- PROZESSOR	BILD- DIGITALISIERUNG PROZESSOR	INTERAKTIVES EINGABEGERÄT	RTA-BUS
BILDERFASSUNG	ABTASTUNG						X		
	DIGITALISIERUNG			X			X		X
	CODIERUNG	X	X	X					X
BILDDARSTELLUNG	DATEN- STRUKTUR ABWICKLUNG	X	X						
	BILD- TRANSFORMATION	X	X		(X)				
	BILDPUNKT- GENERIERUNG			X	X				X
	ZUGRIFF ZUM VIRTUELLEN SCHIRM			X					X
	MONITOR- INSTANZ			X		X			X
BILD- MANIPULATION	MARKIERUNG und IDENTIFIKATION	X	X					X	X
	CURSOR und LUPE					X		X	X

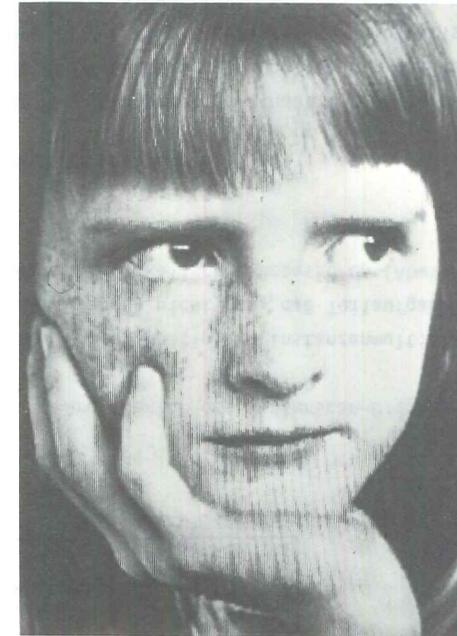


Bild 5.2 Beispiel eines digitalisierten Bildes

Die Bilddaten werden in dem Bildspeicher gespeichert, der einen virtuellen Schirm mit erweiterter Speicherkapazität (max. 8 Mbytes) darstellt. Obwohl der Bildspeicher physikalisch aus 8 Blöcken von jeweils 128 KWörter x 64 Bits/Wort aufgebaut ist, kann er mit Hilfe des vorgeschalteten Speicher-Kontrollers zu einem Stapel von Bildern beliebiger Größe logisch reorganisiert werden. Damit kann die vorhandene Speicherkapazität optimal ausgenutzt werden.

Die Abwicklung der Datenstruktur und die anschließende Bildtransformation finden in dem zentralen Rechner - hier eine PDP11/45 - statt. Die resultierenden absoluten Bildbefehle werden über einen Kommunikationsprozessor an den Bild-generatorprozessor weitergeleitet. Über den Kommunikationsprozessor werden i. a. sowohl Daten als auch Anweisungen für verschiedene Prozessoren transferiert, so

daß das gesamte System nur eine einzige Schnittstelle mit der Umwelt benötigt. Der Monitorprozessor führt schließlich die optische Darstellung der Graphik- und Abbildinformation auf einem Farbvideomonitor durch. Die Bildmanipulation wird entsprechend den Anweisungen interaktiver Eingabegerät - in Bild 5.1 nur als Joystick dargestellt - teilweise im Monitorprozessor und teilweise im zentralen Rechner ausgeführt.

Die Multiprozessor-Architektur in Bild 5.1 stellt ein System mit verteilter Verarbeitungsleistung dar, das auf Grund seiner Modularität jederzeit durch andere spezielle Prozessoren erweitert werden kann.

5.1.2. Implementierungstechniken

Bei der Implementierung verschiedener Prozessoren des Display-Systems wird der Entwerfer mit der Problematik des strukturierten Entwurfs komplexer asynchroner und synchroner Schaltwerke konfrontiert /WENDT-74, HARTENSTEIN-77/. Die Komplexität der Aufgabenstellung einerseits und die hohen Durchsatzwerte andererseits haben den Einsatz moderner Implementierungstechniken notwendig gemacht:

- Die komplexen Kommunikationsregeln zwischen den parallel ablaufenden Prozessen wurden formell mit Petri-Netzen modelliert /PETRI-62, PETERSON-77/. Die Netze werden als konsequente Weiterentwicklung von Zustandsgraphen dargestellt und die Netzereignisse durch Signalflanken physikalisch repräsentiert /WENDT-74-2, WENDT-77/. Auf diese Weise gelingt es, die Kommunikation zwischen verkoppelten Schaltwerken geschlossen und übersichtlich zu beschreiben /ULLRICH-76/.
- Die Steuerwerke wurden bei den synchronen Schaltwerken ausnahmslos als Mikroprogrammwerke realisiert /HUSSON-70/. Das verwendete Konzept der Mikroprogrammierung, das ein begrenztes, aber leistungsfähiges Repertoire von Mikrobefehlen verwendet, hat die systematische Realisierung schneller und gleichzeitig flexibler Steuerwerke ermöglicht /WENDT-71, HARTENSTEIN-74/.
- Die Operationswerke der synchronen Schaltwerke wurden durch Verwendung anschaulicher Symbole für die operationellen Blöcke heuristisch modelliert. Der angestrebte strukturierte Entwurf der Operationswerke könnte nur durch den Einsatz neuartiger Hardware-Beschreibungssprachen erreicht werden, die eine algorithmische Beschreibung des Datenflusses auf der Registertransfer-Ebene ermöglichen /HARTENSTEIN-77/.

- Die Steuer- und Operationswerke bilden zusammen eine Pipeline-Struktur. Dies bedeutet, daß die Interpretation und die Ausführung der Befehle zeitlich überlappt stattfindet, was zu einer wesentlichen Durchsatzserhöhung für das gesamte Schaltwerk führt. Dafür müssen sowohl das Steuerwerk als auch das Operationswerk als Moore-Automaten implementiert werden.

Die konsequente Verwendung dieser Implementierungswerkzeuge hat die Möglichkeit eröffnet, den Entwurf weitgehend methodisch durchzuführen. Dies führt zwangsläufig zu einer transparenten Implementierung, zu kürzeren Implementierungszeiten, zu einer besseren Dokumentation und damit zu besseren Endprodukten.

5.2. Struktur der Systemkomponenten

Die Strukturen der Systemkomponenten, die bezüglich der Hardware-Implementierungsaspekte besonders relevant sind, werden im folgenden kurz diskutiert.

5.2.1. Asynchroner Bus

Der RTA-Bus muß zwei Aufgaben durchführen: Er muß die asynchrone Kommunikation zwischen beliebigen Prozessorenpaaren im Zeitmultiplexverfahren ermöglichen, und außerdem muß er die Synchronisation der verschiedenen Bus-Transferprozesse entsprechend einem vom Anwender zu bestimmenden Prioritätenschema regeln.

5.2.1.1. Bus-Struktur

Die Struktur des RTA-Busses ist in Bild 5.3 veranschaulicht /WENDT-74-2, WENDT-77/. Der Bus besteht aus gleichwertigen Kommunikationsblöcken, die entweder "Master", d. h. aktive Kommunikationspartner, oder "Slave", d. h. passive Kommunikationspartner sein können. Jeder potentielle Master muß sich, bevor er den Bus benutzen darf, durch einen asynchronen Bus-Request (BR) beim zentralen Bus-Kontroller um die Bus-Benutzungsrechte bewerben. Das Bus-Benutzungsrecht - Bus-Grant (BG) - wird vom Bus-Kontroller durch eine "daisy-chain"-Kette ausgestellt, so daß dem dem Kontroller nächstliegenden potentiellen Master der Bus zugeteilt wird. Ober eine zusätzliche Prioritätenleitung (P_N), die ebenfalls als "daisy-chain" gekettet ist, kann ein aktiver Master zur Freigabe des Busses (frühestens am Ende des laufenden Bustransfers) gezwungen werden, wenn sich ein anderer potentieller Master höherer Priorität um die Kontrolle über den Bus bewirbt. Damit können garantierte Vergabe- und Abgabezeiten für den Bus gewährlei-

stet werden, die für die Echtzeit-Verarbeitung der Graphik- und Abbildinformation unerlässlich sind.

Jeder Kommunikationsblock ist in folgende Funktionseinheiten unterteilt:

- Anwendersystem, das aus einem beliebigen Prozessor in Bild 5.1 bestehen kann und direkt mit dem aktiven oder passiven Transferwerk kommuniziert.
- Bus-Anforderungswerk, das den Dialog mit dem Bus-Kontroller für die Erlangung der Bus-Benutzungsrechte führt.
- Aktives Transferwerk, das den Handshake-Dialog mit dem Anwendersystem führt und den Bustransfer als Master veranlaßt.
- Passives Transferwerk, das den Handshake-Dialog mit dem Anwendersystem führt und am Bustransfer als Slave teilnimmt.

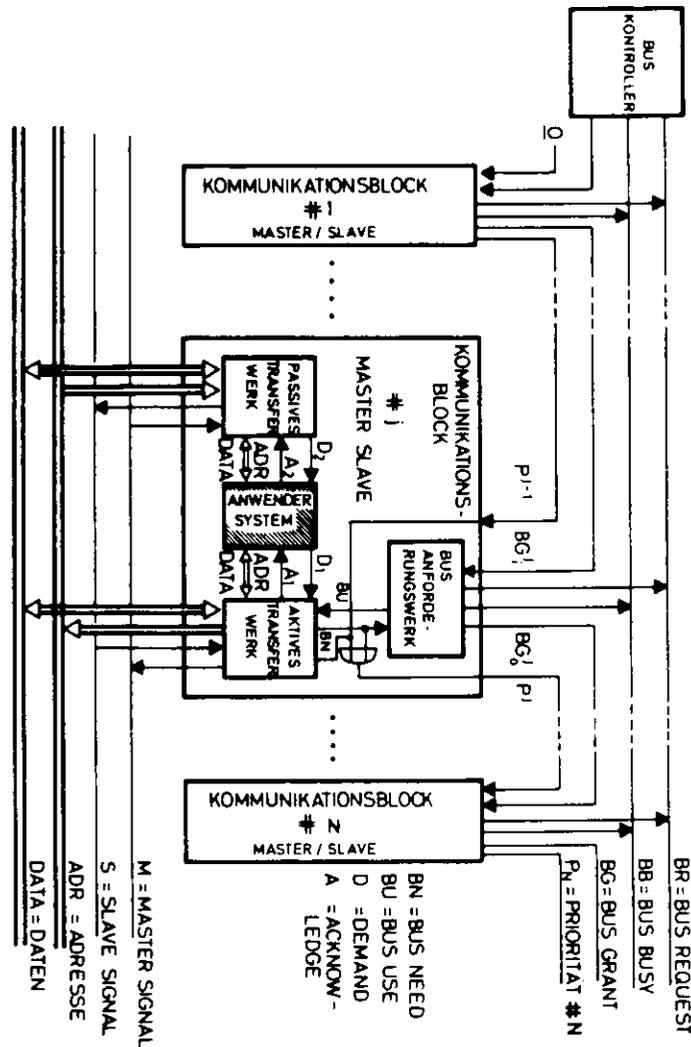
Der Datentransfer über den Bus findet asynchron mit einem Handshake-Protokoll statt und ist daher von der Geschwindigkeit verschiedener Prozessoren vollkommen unabhängig.

Physikalisch sind das passive Transferwerk zusammen mit dem Bus-Kontroller und das aktive Transferwerk zusammen mit dem Busanforderungswerk jeweils als getrennte Module realisiert. Sie bilden ein modulares Interface zwischen dem Anwenderprozessor und dem RTA-Bus, so daß das Anwendersystem stets eine einfache Schnittstelle zum RTA-Bus hat.

5.2.2. Bildspeicher

Die implementierte Architektur des Bildspeichers ist in Bild 5.4 dargestellt. Der Bildspeicher besteht aus Speicher-Modulen und aus einem Speicher-Kontroller, der maximal 64 Speicher-Module ansteuern kann. Der gesamte Bildspeicher ist an den RTA-Bus über ein passives Transferwerk angeschlossen. Jeder Bildspeicher-Modul hat eine Speicherkapazität von 128 KWörtern x 8 Bits und eine Zugriffszeit von 270 nsek. Die Bildspeicher-Module können für die Realisierung verschiedener physikalischer Konfigurationen des Bildspeichers sowohl in Adressen- als auch in Datenrichtung kaskadiert werden. Im folgenden wird die Struktur des Bildspeicher-Moduls näher erläutert.

Bild 5.3 Architektur des RTA-Busses



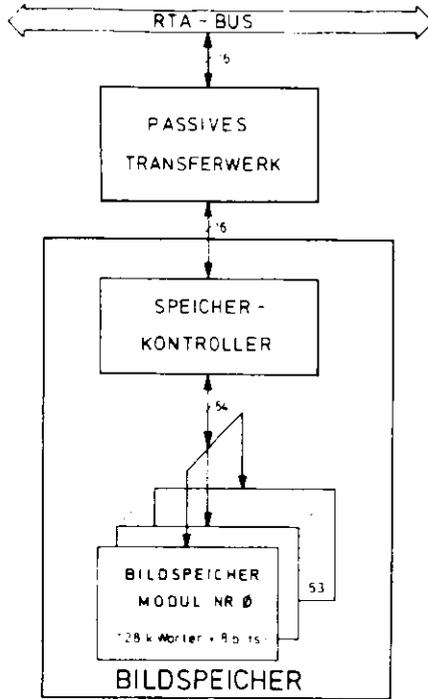


Bild 5.4 Struktur des Bildspeichers

Der realisierte Bildspeicher-Modul ist in Bild 5.5 veranschaulicht. Für die Schnittstelle zum Kontroller wurde eine einfache parallele Handshake-Schnittstelle gewählt, so daß der Bildspeicher-Modul einfach anzusteuern ist. Die Speichermatrix enthält 64 dynamische Speicher-Bausteine mit einer Speicherkapazität von jeweils 16 KWörtern x 1 Bit. Da diese Bausteine nur im Adreß-Multiplex bezüglich der Taktsignale "RAS" und "CAS" adressiert werden können /MOSTEK-76/, muß man sowohl die Taktsignale selbst als auch den Adreß-Multiplex mit Hilfe eines vorgeschalteten Steuerwerkes erzeugen. Der Zustandsgraph in Bild 5.6 beschreibt das realisierte Verhalten des Steuerwerkes, wobei die ange-

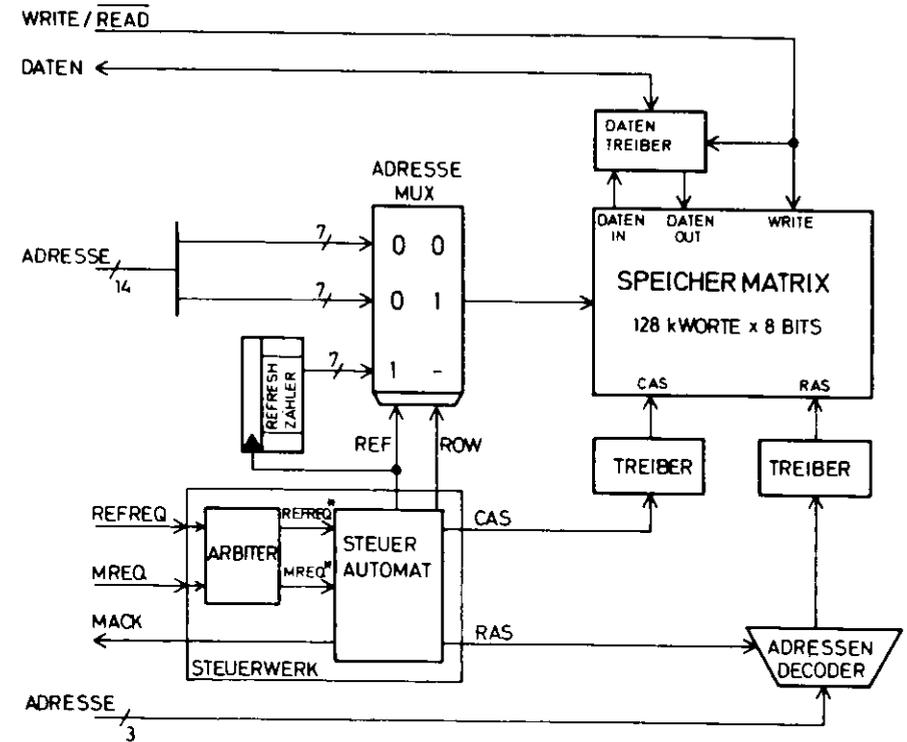


Bild 5.5 Struktur des Bildspeicher-Moduls

gebenen Verzögerungszeiten T1 bis T6 durch die dynamischen Speicherbausteine technisch bedingt sind. Der wegen der dynamischen Speicherbausteine notwendige Refresh (REFREQ) der Speichermatrix kann sowohl synchron als auch asynchron in bezug auf den Speicherzyklus (MREQ) stattfinden. Im letzteren Fall ist die Entscheidung zwischen einem Speicherzyklus und einem Refreshzyklus dem vorgeschalteten Arbitrer überlassen, dessen Ausgangssignale MREQ* und REFREQ* sich gegenseitig ausschließen.

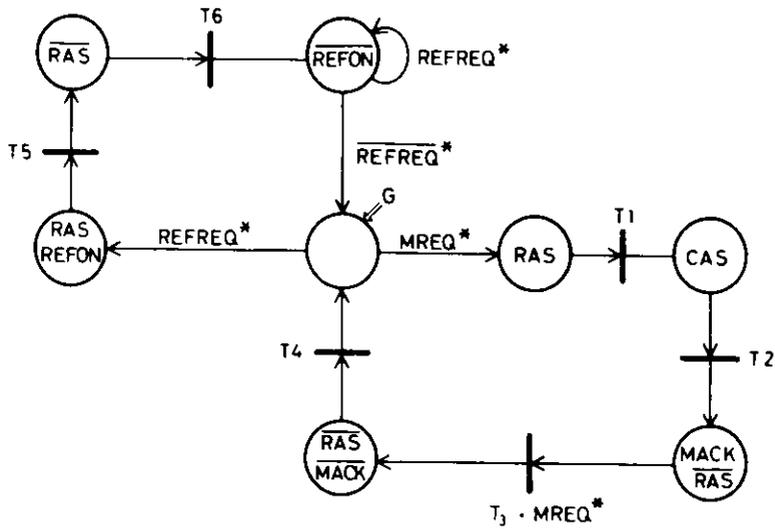


Bild 5.6 Zustandsgraph des Steuerwerkes für den Bildspeicher-Modul

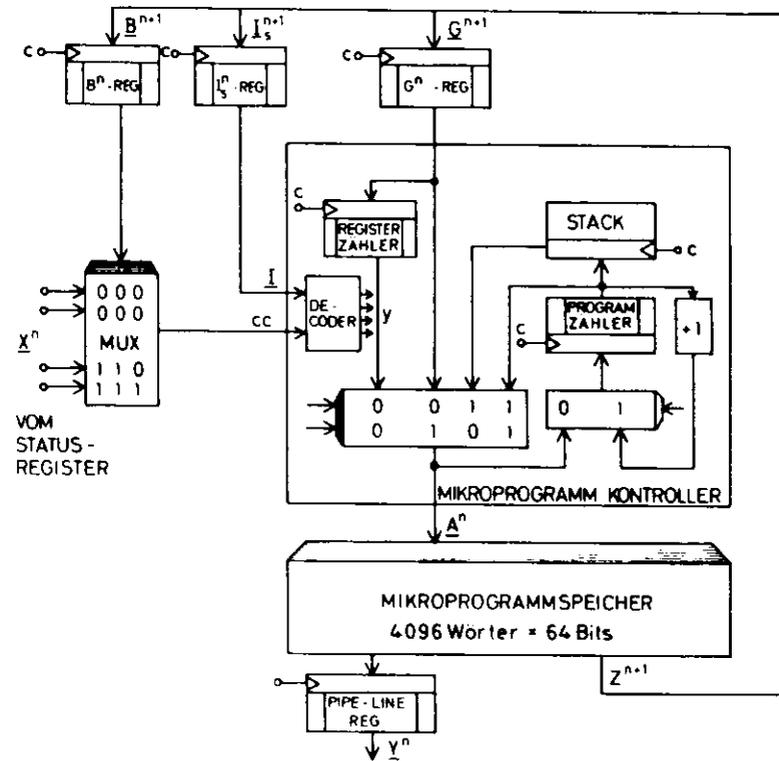


Bild 5.7 Struktur des Mikroprogrammwerkes

5.2.3. Bildgeneratorprozessor

Bei der Implementierung des Bildgeneratorprozessors spielen zwei Bedingungen eine maßgebliche Rolle: Einerseits muß der Bildpunktgenerator so "intelligent" sein, daß sowohl die Bildpunktgenerierung der graphischen Bildelemente als auch die allgemeinen Bildverarbeitungsoperationen durchgeführt werden können. Andererseits müssen diese Aufgaben so schnell ausgeführt werden, daß die Bandbreite des Bildspeichers vollständig ausgenutzt wird. Aus diesen Gründen wurde der Bildpunktgenerator als eine schnelle mikroprogrammierte Maschine mit einem erweiterten Mikrobefehlsrepertoire entworfen. Um die geforderte Schnelligkeit von 125 nsek pro Mikrobefehlszyklus erreichen zu können, müssen das Mikroprogrammwerk und das Operationswerk als Pipeline-Struktur implementiert werden. Sie kommunizieren miteinander ausschließlich über ein Mikrobefehlsregister bzw. über ein Statusregister. Das in Bild 5.7 vorgeschlagene Mikroprogrammwerk ist ein

Moore-Automat, dessen Architektur ein Grundwertadreßregister (G^n), ein Branch-Informationsregister (B^n), ein Befehlsregister (I^n) sowie einen Mikroprogramm-speicher mit 4096 Wörtern x 64 Bits enthält. Das aus zwei kaskadierten Multiplexern bestehende Adreßbildungsnetz kann während einer Taktperiode nur binäre Abfragen des Statusregisters durchführen und wählt die nächste Mikroprogramm-adresse aus vier möglichen Quellen aus: Programm-Zähler, Register-Zähler, Grundwertadreßregister oder Mikroprogrammstack. Das Mikrobefehlsrepertoire enthält

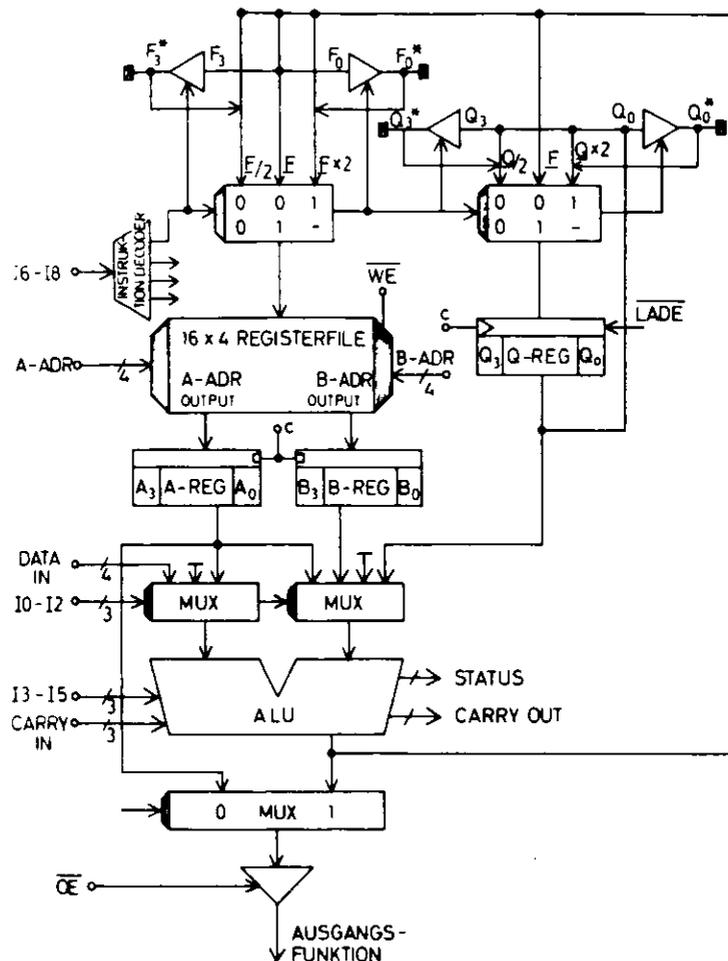


Bild 5.8 Ersatzschaltbild des AM2901-Bitslice-Bausteines

12 Mikrobefehle: RESET, CONTINUE, JUMP TO START, CONDITIONAL BRANCH, CONDITIONAL JUMP, CONDITIONAL JUMP TO SUBROUTINE, CONDITIONAL RETURN FROM SUBROUTINE, REPEAT CONDITIONAL, LOAD LOOP COUNTER, TEST END LOOP, PUSH und POP. Mit Hilfe dieser leistungsvollen Befehle kann ein reichhaltiger Makrobefehlssatz für den Bildgenerator erzeugt werden. Das Operationswerk muß ebenfalls ein Moore-Automat sein, der wegen seiner Vielseitigkeit und Schnelligkeit zweckmäßigerweise mit Bitslice-Bausteinen realisiert werden muß. Besonders geeignet sind Bitslice-Bausteine mit einer Zwei-Adreß-Architektur, bei denen die Operandenregister innerhalb eines einzigen Maschinenzyklus sowohl gelesen als auch geschrieben werden können. Das Ersatzschaltbild eines solchen Bitslice-Bausteines (AM2901) ist in Bild 5.8 veranschaulicht.

Die Operanden und die Ergebnisse einer Rechenoperation werden in einem 16 Wörter langen Register-File aufbewahrt. Zwei 4-Weg-Multiplexer erlauben eine komfortable Operandenselektierung für die Arithmetische Logische Einheit (ALU). Die ALU kann insgesamt 16 Operationen durchführen und das Ergebnis zusätzlich mit Hilfe eines 4 Bit langen Statusregisters anzeigen. Die Ergebnisse der ALU-Operationen können bei Bedarf um eine Stelle nach links oder rechts verschoben werden und dann entweder in das Akkumulator-Register (Q-REG) oder in das Register-File gespeichert werden.

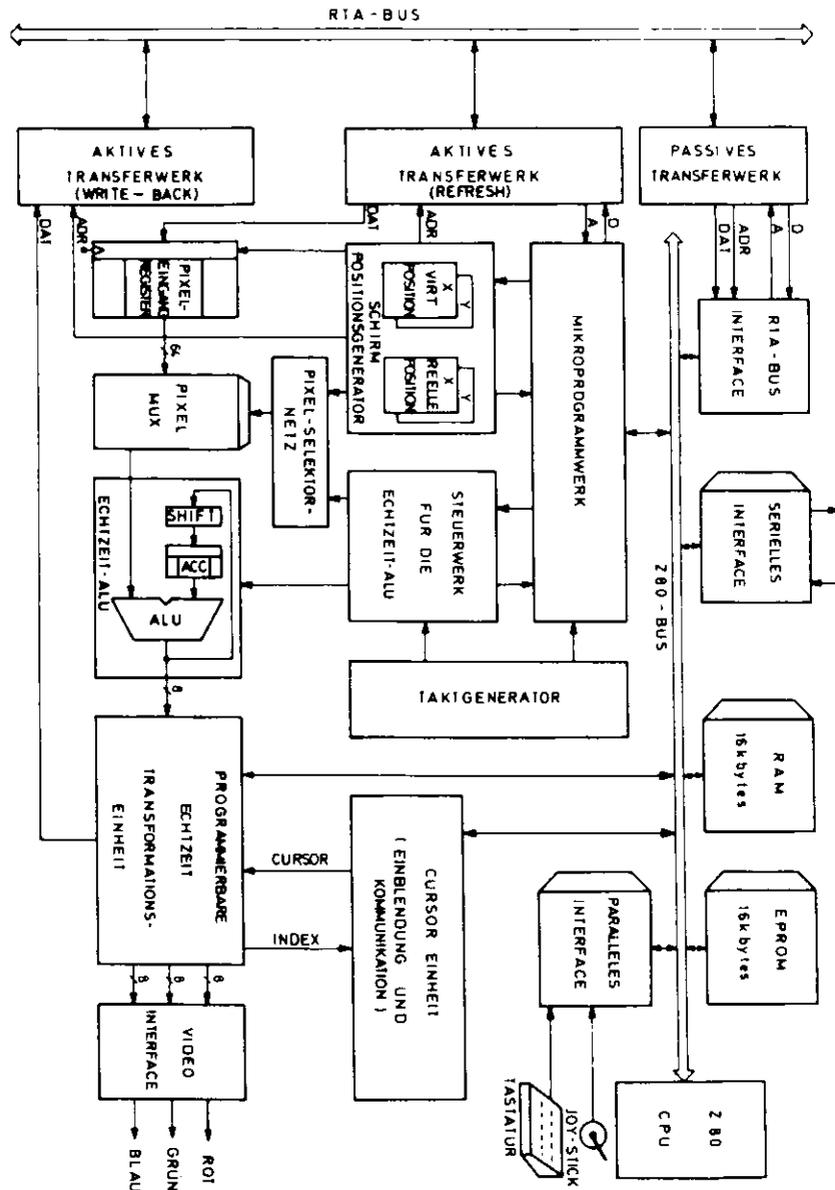
5.2.4. Monitorprozessor

Bei der Implementierung des Monitorprozessors spielt die Forderung nach Echtzeit-Zuweisung optischer Attribute zu den logischen Indizes eine maßgebende Rolle. Das bedeutet, daß die allgemeine und die optische Transformation der Indexwerte mit der Pixeldarstellungsfrequenz von 10 Megapixeln/sek durchgeführt werden muß. Diese aus der informationellen Struktur abgeleitete Forderung führt zur Implementierung einer parallelen Struktur für den Monitorprozessor, die zweckmäßigerweise durch eine mehrstufige Pipeline realisiert wird. Im folgenden werden hauptsächlich die Implementierungsaspekte des Bildrefreshers, der die zeitkritischen Prozesse realisieren muß, beschrieben.

5.2.4.1. Struktur des Bildrefreshwerkes

Das Bildrefreshwerk ist in Bild 5.9 schematisch dargestellt. Es ist ein mikroprogrammiertes Schaltwerk, das in Verbindung mit einem Mikroprozessorsystem auf Z80-Basis aufgebaut ist. Das Mikroprozessorsystem übernimmt sämtliche Auf-

Bild 5.9 Struktur des Monitorprozessors



gaben der Optik-Transformationsinstanz, die im wesentlichen die Kommunikation mit den benachbarten Prozessoren und die Aufbereitung der Transformationstabellen für das Bildrefreshwerk ausführt. Dadurch können alle zeitunkritischen Prozesse in dem Monitorprozessor mit Firmware- und Softwaremitteln realisiert werden, was eine flexible und gleichzeitig komfortable Lösung darstellt. Das Bildrefreshwerk besteht aus einem mikroprogrammierten Steuerwerk und aus einem umfangreichen Operationswerk, die wegen der zeitlich überlappten Interpretation und Ausführung der Mikrobefehle eine Pipeline-Struktur bilden. Das Mikroprogrammwerk ist direkt an den Z80-Bus angeschlossen und kann durch Makrobefehle unmittelbar programmiert werden. Das komplexe Operationswerk enthält entsprechend der informationellen Struktur in Bild 4.23 mehrere funktionell spezialisierte Operationsblöcke /NICOLAE-79-3/:

- Der Taktgenerator erzeugt sämtliche digitalen Taktsignale und Videosynchronsignale für das Bildrefreshwerk.
- Der Schirm-Positionsgenerator berechnet in Echtzeit sowohl die laufende Adresse im virtuellen Schirm als auch die entsprechende Position auf dem realen Bildschirm. Entsprechend dem vorhandenen Lupe-Status, der in Form von Makrobefehlen dem Mikroprogrammwerk mitgeteilt wird, können verschiedene Lupe-Effekte, wie Zoom und Pan, realisiert werden. Der Schirm-Positionsgenerator ist mit Bit-slice-Bausteinen realisiert und ermöglicht die Darstellung eines optischen Rasterscanbildes mit einer Auflösung von 512 x 512 Pixeln.
- Der Operationsblock für die Transformation und Darstellung der logischen Indexwerte besteht aus mehreren Einheiten, die eine mehrstufige Pipeline bilden. Die logischen Indizes werden über den RTA-Bus in Gruppen von 8 Pixeln zunächst in ein 64 Bits breites Eingangsregister gelesen. Mit Hilfe eines speziell gesteuerten Pixelmultiplexers wird jeder einzelne Pixel in die sogenannte Echtzeit-ALU transferiert. Die Echtzeit-ALU kann einfache Nachbarschaftsoperationen in Echtzeit durchführen. Sie benutzt ein Akkumulatorregister für die Pufferung von Zwischenergebnissen und wird durch ein speziell dafür vorgesehenes Steuerwerk kontrolliert, das synchron mit dem zentralen Mikroprogrammwerk arbeitet. Die nachfolgende programmierbare Echtzeit-Transformationseinheit führt die allgemeine und die optische Transformation der logischen Indizes durch und erzeugt schließlich die R,G,B-Farbwerte, die die unmittelbare, aber redundante Codierungsform des optischen Bildes für den Rasterscan-Monitor darstellen. Die Struktur und die Wirkungsweise der Echtzeit-Transformationseinheit, die im Zu-

sammenhang mit der dynamischen Zuweisung der optischen Attribute eine wichtige Rolle spielt, wird im nächsten Abschnitt detailliert beschrieben.

- Die Cursor-Einheit dient der Einblendung des Cursors in das optische Bild und der Auswertung von Cursor-Kommandos, die im Rahmen der Kommunikation mit der interaktiven Bildmanipulationsinstanz auftreten. Die Cursor-Manipulation findet mit Hilfe eines ergonomischen Eingabegerätes - z. B. Joystick oder Tablet - über ein paralleles Interface direkt im Mikroprozessorsystem statt. Das bedeutet, daß kurze Reaktionszeiten bei der Cursor-Manipulation erwartet werden können und gleichzeitig der zentrale Rechner entlastet wird.

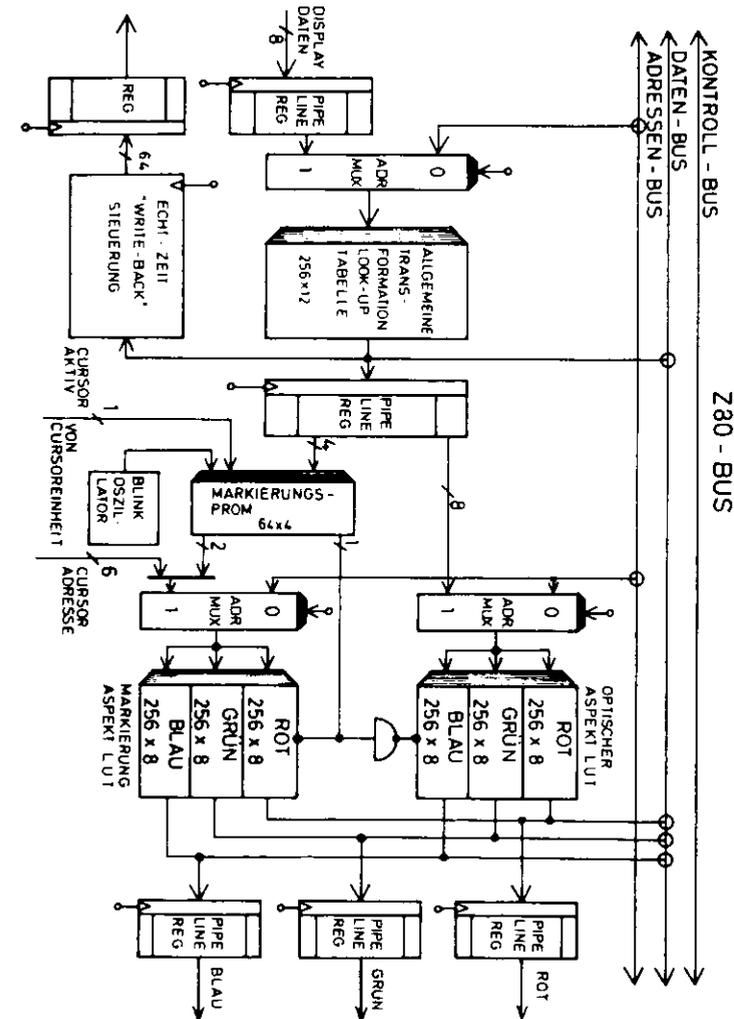
Für die Kommunikation des gesamten Monitorprozessors mit den anderen Prozessoren im System ist ein spezielles RTA-Interface vorgesehen.

5.2.4.2. Programmierbare Echtzeit-Transformationseinheit

Die Struktur der programmierbaren Echtzeit-Transformationseinheit ist in Bild 5.10 veranschaulicht. Es zeigt zwei kaskadierte Look-up-Tabellen, die vom Mikroprozessorsystem über zwischengeschaltete Adreßmultipllexer und abschaltbare Datentreiber programmiert - d. h. geschrieben und/oder gelesen - werden können.

Die erste Look-up-Tabelle realisiert die allgemeine Transformation der logischen Indizes. Sie beinhaltet 256 Wörter mit jeweils 12 Bits. Davon werden 8 Bits für die tabellarische Spezifikation der allgemeinen Transformation verwendet, während die restlichen 4 Bits zur Angabe von assoziativen Markierungseigenschaften, wie Blinken, Hell oder Dunkel, benutzt werden. Ein Bit von diesen zusätzlichen Markierungsbits ist während des Darstellungsprozesses schreibbar und dient zur automatischen Herstellung der Index-Belegungstabelle (siehe Kapitel 4). Mit Hilfe der allgemeinen Transformations-Look-up-Tabelle können sämtliche monadische Bildpunkttransformationen durchgeführt werden. Beispielsweise zeigen die Bilder 5.11a) und b) die logarithmische bzw. die exponentielle Transformation der in Bild 5.2 dargestellten Aufnahme. Die allgemeine Transformations-Look-up-Tabelle wird in Verbindung mit der "WRITE-BACK"-Steuerung auch für die zeitsparende Realisierung der Index-Kompression im virtuellen Schirm verwendet. Sie erlaubt in dieser Konfiguration auch die Durchführung iterativer Verarbeitungsalgorithmen, bei denen nach jedem Verarbeitungsschritt die transformierten Indexwerte zur weiteren Verwendung im nächsten Schritt in den Bildspeicher zurückgeschrieben werden müssen.

Bild 5.10 Programmierbare Echtzeit-Transformationseinheit





a) Logarithmische Transformation



b) Exponentielle Transformation

Bild 5.11 Zur Veranschaulichung der monadischen Bildpunkttransformationen

Die zweite Look-up-Tabelle, hier Aspekt-Look-up-Tabelle genannt, realisiert die optische Transformation der eventuell transformierten logischen Indexwerte. Sie besteht aus zwei parallel geschalteten Look-up-Tabellen, die die optischen Normal- und Markierungsattribute getrennt enthalten. Die Umschaltung der beiden Aspekt-Look-up-Tabellen wird von einem vorgeschalteten Markierungs-PROM kontrolliert, der die Markierungsbits der allgemeinen Transformationstabelle und die Cursor-Information auswertet. Mit der in Bild 5.10 angegebenen Organisation der Aspekt-Look-up-Tabellen von jeweils $3 \times (256 \text{ Wörtern} \times 8 \text{ Bits})$ können insgesamt 2^{24} verschiedene Farbwerte oder 2^8 Graustufen erzeugt werden. Die Wirkungsweise der dynamischen Zuweisung optischer Attribute kann mit Hilfe einfacher Beispiele aus der Darstellung der Abbildinformation deutlich gemacht werden. Beispielsweise zeigt das Bild 5.12 ein pseudofarbcodiertes Funktionsbild einer Niere, das die Verteilung der Blutgeschwindigkeit in diesem Organ darstellt. Die Pseudofarbcodierung, die die Zuordnung von willkürlichen Farbwerten zu einem schwarz/weißen Bild im virtuellen Schirm bedeutet, eröffnet die Möglichkeit, die vorhandene Bildinformation durch die visuell leichter identifizierbaren Farbwerte auch quantitativ zu beurteilen. Das Bild 5.13 zeigt ein echtfarbcodiertes Bild einer willkürlich gewählten Szene, die im virtuellen Schirm auch schwarz/weiß angegeben ist. In diesem Falle wird jedem Bildpunkt ein Farbwert zugeordnet, wobei die ursprüngliche Helligkeit jedes Bildpunktes nicht verändert wird (siehe Gleichung 2.2). Das Abbild wird also nachträglich mit Farbinformation ergänzt, ohne daß dadurch die ursprüngliche Helligkeitsinformation verlorenght. Da die Look-up-Tabellen innerhalb eines Videobildes neu programmiert werden können, d. h. sie dürfen alle 20 msek verändert werden, können Playback-Effekte leicht realisiert werden. Bild 5.14 zeigt beispielsweise die Emulation des Blutflusses durch die Niere in Bild 5.12, wobei hier nur zwei unterschiedliche Phasen dargestellt sind. Obwohl sich im Bildspeicher die gleichen Bilddaten wie in Bild 5.12 befinden, kann durch sequentielle Zuordnung der Farbe ROT zu den verschiedenen Blutgeschwindigkeiten in ihrer steigenden Reihenfolge der Blutfluß leicht sichtbar gemacht werden.

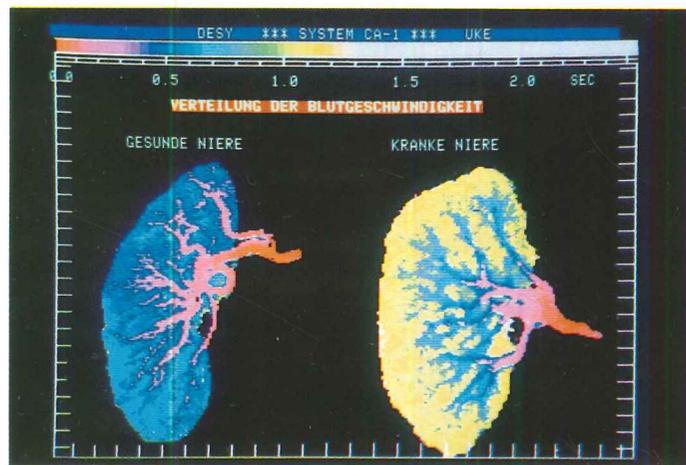


Bild 5.12 Beispiel eines pseudofarbcodierten Bildes



Bild 5.13 Beispiel eines echtfarbcodierten Bildes

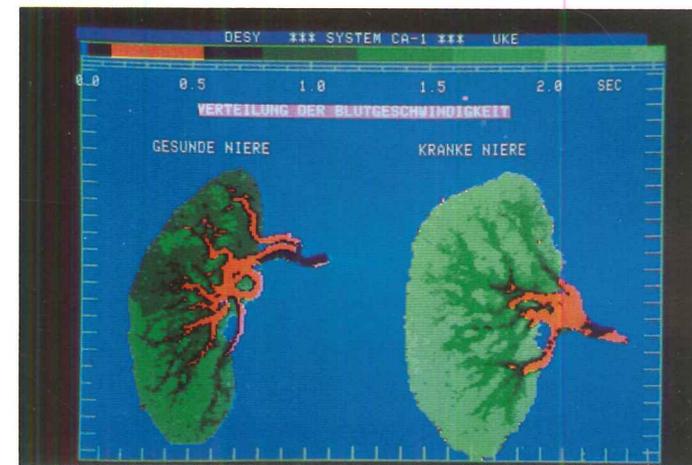


Bild 5.14 Zwei Phasen der Echtzeit-Emulation des Blutdurchflusses durch die Niere

6. Anhang

6.1. Algorithmus für die Vektorgenerierung

```

;AUFRUF: PROC VEKTOR(Xo,Yo,dX,dY)

;DEKLARATIONEN

var Xo,Yo :integer ;VEKTORANFANGSPUNKT
    dX,dY :integer ;VEKTORINKREMENT
    MIN,MAX :integer ; LAUFENDER VEKTORPUNKT
    dmax,dmin :integer ;INTERNE VEKTORINKREMENTE
    CNT,MAX1 :integer ;INTERNE VARIABLEN

;INITIALISIERUNG

begin
if ABS(dX) > ABS(dY) then begin
    dmax=dX,MAX=Xo
    dmin=dY,MIN=Yo
end
else begin
    dmax=dY,MAX=Yo
    dmin=dX,MIN=Xo
end

CNT=dmax div 2
MAX1=MAX

;VEKTORGENERIERUNG

while MAX in MAX1 to MAX1+dmax do
begin
write(MAX,MIN) ;OUTPUT VEKTORPUNKT
MAX=MAX+1
CNT=CNT+dmin
if CNT > 0 then MIN=MIN+1
    CNT=CNT-dmax
else MIN=MIN
end
end
end

```

6.2. Algorithmus für die Flächengenerierung

```

;AUFRUF: PROC AREA (I(N),Xmin,Xmax,Ymin,Ymax)

;DEKLARATIONEN

var Xmin,Xmax,Ymin,Ymax :integer ;Flächenarbeitsfenster
    I(N) :integer ;Indexwert der polygonalen Kontur
    I(x,y) :integer ;Indexwert des Pixels P(x,y)
    CNT :integer ;interne Variable mit CNT=2*+1 fuer flaeche
    U,D :integer ;interne variablen: U:=up,D:=down
    X,Y :integer ;interne Variablen

;FLAECHEGENERIERUNG

begin
for Y in Ymin to Ymax do
begin
set CNT=0
set X=Xmin
while X < Xmax do
begin
if I(x,y)neqI(N) then if CNT=2*+1, set I(x,y)=I(N)
else
begin
set U=0,D=0
if I(x-1,y-1)=I(N) , U=U+1
if I(x-1,y+1)=I(N) , D=D+1
while I(x,y)=I(N) do
begin
if I(x,y-1)eqI(N)andI(x-1,y-1)eqI(N) ,U=U+1
if I(x,y+1)eqI(N)andI(x-1,y+1)neqI(N),D=D+1
X = X+1
end
if I(x,y)neqI(N)andI(x,y-1)eqI(N), U=U+1
if I(x,y)neqI(N)andI(x,y+1)eqI(N), D=D+1
if Ueq1andDeq1, CNT=CNT+1 ;normaler Pixel
if U+D>2orU+D=1, CALL ERROR ; singularer Pixel
end
end
end
end
end
end

```

Die Bedeutung der internen Variablen U (UP) und D (DOWN) ist wie folgt:

UP	DOWN	KONTURPIXEL
1	1	Normal
0	2	Konturmaximum (y_{\max})
2	0	Konturminimum (y_{\min})
1	0	Benachbarte
0	1	
		Konturpixel

7. Literaturverzeichnis

- /ABRAMS-71/ Abrams, M.D., 'Datastructure for Computer Graphics', National Bureau of Standards, SIGPLAN-ACM, Vol. 6, No. 2, 1971
- /ACM-79/ Status Report of the Graphic Standards Planning Committee, Vol. 13, No. 3, August 1979
- /ANDREWS-79/ Andrews, H.C., 'Digital Image Processing', IEEE-Spectrum, April 1979
- /BAECKER-79/ Baecker, R., 'Digital Video Display Systems and Dynamic Graphics', Proceedings of SIGGRAPH '79, Chicago, Ill., August 1979
- /BLINN-77/ Blinn, J., 'Models of light Reflection for Computer Synthesized Pictures', Proc. SIGGRAPH, BP 192-198, 1977
- /BRESENHAM-65/ Bresenham, J.E., 'Algorithm for Computer Control of a Digital Plotter', IBM Systems Journal, Vol. 4, No. 1 pp 25-30, 1965
- /CLARK-77/ Clark, J.H., 'Designing Surface in 3-D', Communications of ACM, Vol. 19, No. 8, August 1977
- /CODD-70/ Codd, E.F., 'A Relational Model of Data for Large Shared Data Banks', ACM Communications, Vol. 13, June 1970
- /CROW-77/ Crow, F.C., 'Shadow Algorithms for Computer Graphics', Proc. SIGGRAPH, pp 242-248, 1977
- /DUDANI-76/ Dudani, S.A., 'Region Extraction using Boundary Following', Pattern Recognition and Artificial Intelligence (C.H. Chen ed.), Academic Press, pp 216-232, 1976
- /EARNSHAW-79/ Earnshaw, R.A., 'Line Generation for Incremental and Raster Devices', Proc. SIGGRAPH 77, San Jose, July 20-22, pp 199-205, 1977

- /ENCARNACAO-72/ Encarnacao, J., Giloi, W., et al., 'Programmierungs- und gerätetechnische Realisierung einer 4x4-Matrix für Koordinatentransformation auf Computer-Bildschirmgeräten', Elektr. Rechenanlagen, Heft 5, pp 206-219, 1972
- /ENCARNACAO-75/ Encarnacao, J., 'Computer-Graphics: Programmierung und Anwendung von graphischen Systemen', R. Oldenburg Verlag, München, Wien, 1975
- /FRY-76/ Fry, J.P., Sibley, E.H., 'Evolution of Data-Base Management Systems', ACM Computer Surveys, Vol. 8, No. 1, March 1976
- /FU-76/ Fu, K.S., Rosenfeld, A., 'Pattern Recognition and Image Processing', IEEE Transactions on Computers, Vol. C-25, No. 12, December 1976
- /FULTON-74/ Fulton, D.L., 'A Plasma-Panel Interactive Graphic System', Proc. of S.I.D., Vol. 15-2, Second Quarter, pp 74-80, 1974
- /GHEST-71/ Ghest, R.C., 'New IC Multipliers 2-Bit by 4-Bit', Electronics, Nov. 22, 1971
- /GILOI-70/ Giloi, W., and Waldschmidt, K., 'Neue Dimensionen und Geschwindigkeit analoger Rechenelemente', 6th Intern. Hybrid Computation Meeting, München, 1970
- /GILOI-74/ Giloi, W., 'On High-Level Language Programming Systems for Interactive Computer Graphics', Technical Report No. 72-74, Missouri University, Nov. 1974
- /GILOI-78/ Giloi, W.K., 'Interactive Computer Graphics: Data Structures, Algorithms, Languages', Prentice Hall, 1978
- /GONAUSER-78/ Gonauser, M., Schinner, P., Weiß, J., 'Hierarchical Data Structures for Graphic Program Languages', Siemens Forschungs- und Entwicklungs-Bereich, Band 7, No. 1, 1978

- /GREENBERG-77/ Greenberg, D.P., et al., 'An Interactive Computer Graphics Approach to Surface Representation', Communications of ACM, Vol. 20, No. 10, October 1977
- /GROSSER-78/ Grüsser, O.-J., 'Grundlagen der Informationsverarbeitung in den Sinnesorganen und Gehirn', GI-8, Jahrestagung, Springer-Verlag 1978
- /HARTENSTEIN-74/ Hartenstein, R., 'Microprogramming Concepts - A Step towards Structured Hardware Design', Proc. of 7th annual Workshop on Microprogramming, ACM, New York, 1974
- /HARTENSTEIN-77/ Hartenstein, R., 'Fundamentals of Structured Hardware Design', North Holland, Amsterdam - New York, Oxford, 1977
- /HÖHNE-78/ Höhne, K.H., Böhm, M., Nicolae, G.C., Pfeiffer, G., Sonne, B., 'Computer Angiography: A New Tool for X-Ray Functional Diagnostic', Med. Progr. Techn. 6, pp23-28, 1978
- /HÖHNE-79/ Höhne, K.H., Böhm, M., Nicolae, G.C., 'The Processing of X-Ray Image Sequences', Advances in Digital Image Processing, pp 147, Plenum Press, New York and London, 1979
- /HUSSON-70/ Husson, S.S., 'Microprogramming: Principles and Practices', Englewood Cliffs, Prentice Hall, 1970
- /JOBLOVE-78/ Joblove, G.H., Greenberg, D., 'Color Spaces for Computer Graphics', SIGGRAPH - ACM, 1978
- /KELLER-74/ Keller, A., 'Erstellung eines Dialogprogramms zur Generierung und Manipulation einer Datenstruktur für Hierarchien gerichteter Graphen', Diplomarbeit, Universität Hamburg, 1974

- /KELLER-78/ Keller, A., 'Informationelle Strukturen in Daten und ihr Einfluß auf die Programmierung', Dissertation, Kaiserslautern, 1978
- /LINDNER-79/ Lindner, R., 'Rasterdisplay-Prozessoren: ihre Bedeutung, Konzepte und Verfahren', Dissertation, TH Darmstadt, 1979
- /MCKEOWN-77/ McKeown, D.H., Reddy, D.R., 'A Hierarchical Symbolic Representation for an Image Database', Proc. on the Workshop on 'Picture Data Description and Management', Chicago, 1977
- /MOSTEK-76/ MK 4116 = 16.384 x 1 Bit Dynamic, RAM, Datenblatt, 1976
- /MUSHANN-79/ Musmann, H.G., 'Digital Coding of Television Signals', Advances in Digital Image Processing, Plenum Press, 1979
- /NATHERAT-78/ Natherat, P., 'A Low-Cost Raster-Scan Graphic Display', Proc. SIGGRAPH 78, August 23-25, pp 181-186, 1978
- /NEWMAN-76/ Newman, W.M., 'Trends in Graphic Display Design', IEEE Transactions on Computers, Vol. C-25, No. 12, December 1976
- /NEWMAN-78/ Newman, W.M., Dam, A., 'Recent Efforts Towards Graphics Standardization', ACM Computing Surveys, Vol. 10, No. 4, December 1978
- /NEWMAN-79/ Newman, W.M., Sproul, R.F., 'Principles of Interactive Computer Graphics', 2. Ed. McGraw-Hill, 1973
- /NICOLAE-79-1/ Nicolae, G.C., 'Digital Video System: Ein Multiprozessor-System für Echtzeitverarbeitung von Videobildern', Interner DESY-Bericht, 1979
- /NICOLAE-79-2/ Nicolae, G.C., Höhne, K.H., 'Multiprocessor System for the Real-Time Digital Processing of Video-Image Series', Elektron. Rechenanlagen 21, Heft 4, pp 171-183, 1979

- /NICOLAE-79-3/ Nicolae, G.C., Wendt, S., 'Design of a Microprogrammed Video Display Processor for Real-Time Image Processing', Euromicro Symposium on Microprocessing and Microprogramming, August 28-30, Göteborg, 1979
- /PAVLIDIS-78/ Pavlidis, T., 'Filling Algorithms for Raster Graphics', Technical Report No. 238, Dept. of Electrical Eng. and Computer Science, Princeton University, Princeton, N.Y., 1978
- /PETERSON-77/ Peterson, J.L., 'Petri-Nets', ACM Computing Surveys, Vol. 9, No. 3, September 1977
- /PETRI-62/ Petri, C.A., 'Kommunikation mit Automaten', Dissertation, Darmstadt, 1962
- /ROSENFELD-77/ Rosenfeld, A., 'Digital Image Processing and Recognition', Digital Image Processing, Gi/NTG Fachtagung, Springer Verlag, 1977
- /SCHÖNFELDER-72/ Schönfelder, H., Fernsehtechnik: Vorlesungsskript TU Braunschweig, von-Liebig-Verlag, Darmstadt, 1972
- /SENKO-75/ Senko, M.E., 'The DDL in the Context of a Multilevel Structured Description DIAM II with Foral', Data Base Description, B.C.M., Donque and Nijssen, G.M. (Eds.), North-Holland Publ. Co., Amsterdam, 1975
- /SHAPIRO-79/ Shapiro, L.G., 'Data Structures for Picture Processing: A Survey', Computer Graphics and Image Processing No. 11, pp 162-184, Academic Press, 1979
- /SLOAN-77/ Sloan, K.R., Bajcsy, R., 'World Model Driven Recognition of Natural Scenes', Proc. of the Workshop on 'Picture Data Description and Management', Chicago, 1977
- /SLOAN-79/ Sloan, K.R., Brown, C.M., 'Color Map Techniques', Computer Graphics and Image Processing, No. 10, pp 297-317, Academic Press, 1979
- /SPROUL-68/ Sproul, R.F., Sutherland, J.E., 'A Clipping Divider', PJCC, Thompson Books, Washington, D.C., pp 76, 1968

- /THOMPSON-64/ Thompson, J.R., 'Straight Lines and Graph Plotters', Computer Journal, Vol. 4, No. 3, pp 227, 1964
- /ULLRICH-76/ Ullrich, G., 'Der Entwurf von Steuerstrukturen für parallele Abläufe mit Hilfe von Petri-Netzen', Dissertation, Universität Hamburg, 1976
- /WENDT-71/ Wendt, S., 'On Structures of Microprogram Control Units', Elektr. Rechenanlagen, Vol. 13. pp 22-26, 1971
- /WENDT-74-1/ Wendt, S., 'Entwurf komplexer Schaltwerke', Springer-Verlag, 1974
- /WENDT-74-2/ Wendt, S., 'Petri-Nets and Asynchronous Sequential Circuits', Elektr. Rechenanlagen, Vol. 16, pp 208-216, 1974
- /WENDT-77/ Wendt, S., 'Using Petri-Nets in the Design Process for Interacting Asynchronous Sequential Circuits', Proc. of IFAC Symposium on Discrete Systems, Dresden, 1977
- /WENDT-79-1/ Wendt, W., 'On the Partitioning of Computing Systems into Communicating Agencies', GI-Fachtagung: Struktur und Betrieb von Rechensystemen, Kiel, März 1980
- /WENDT-79-2/ Wendt, S., 'The Programmed Action Modul: An Element for System Modelling', Digital Processes, Vol. 4, 1979
- /WHITTED-79/ Whitted, T., 'An Improved Illumination Model for Shaded Display', Proc. SIGGRAPH, 1979

Diese Arbeit wurde vom Fachbereich Elektrotechnik der Universität Kaiserslautern als Dissertation zur Verleihung des akademischen Grades DOKTOR-INGENIEUR genehmigt.

Dekan des Fachbereichs Elektrotechnik : Prof. Dr.-Ing. W. Rupprecht
Vorsitzender der Promotionskommission : Prof. Dr.-Ing. D. Schröder
1. Berichterstatter : Prof. Dr.-Ing. S. Wendt
2. Berichterstatter : Prof. Dr.-Ing. R. Hartenstein

