

LEITFÄDEN DER ANGEWANDTEN
INFORMATIK

A. Meier

8768339

**Methoden
der grafischen
und geometrischen
Datenverarbeitung**



B.G. Teubner Stuttgart

0888278

Leitfäden der angewandten Informatik

Herausgegeben von

Prof. Dr. L. Richter, Zürich

Prof. Dr. W. Stucky, Karlsruhe



Die Bände dieser Reihe sind allen Methoden und Ergebnissen der Informatik gewidmet, die für die praktische Anwendung von Bedeutung sind. Besonderer Wert wird dabei auf die Darstellung dieser Methoden und Ergebnisse in einer allgemein verständlichen, dennoch exakten und präzisen Form gelegt. Die Reihe soll einerseits dem Fachmann eines anderen Gebietes, der sich mit Problemen der Datenverarbeitung beschäftigen muß, selbst aber keine Fachinformatik-Ausbildung besitzt, das für seine Praxis relevante Informatikwissen vermitteln; andererseits soll dem Informatiker, der auf einem dieser Anwendungsgebiete tätig werden will, ein Überblick über die Anwendungen der Informatikmethoden in diesem Gebiet gegeben werden. Für Praktiker, wie Programmierer, Systemanalytiker, Organisatoren und andere, stellen die Bände Hilfsmittel zur Lösung von Problemen der täglichen Praxis bereit; darüber hinaus sind die Veröffentlichungen zur Weiterbildung gedacht.

8763339

Leitfäden der angewandten Informatik

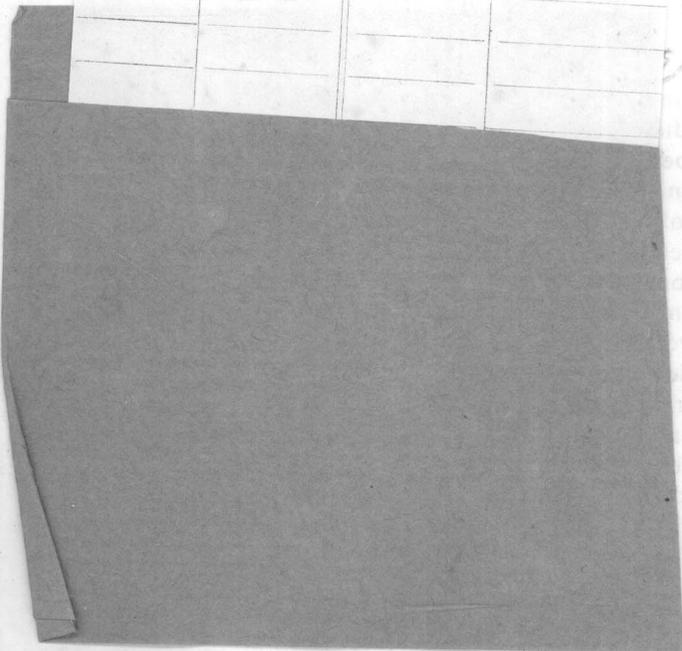
A. Meier
Methoden der grafischen
und geometrischen Datenverarbeitung

TP274
M511

8763339

7. 外借

Methoden der grafischen und
geometrischen datenverarbeit-
ung



TP 274
M 511

8763339

△
不 外借

Methoden der grafischen und geometrischen Datenverarbeitung

Von Dr. sc. techn. Andreas Meier
Eidg. Technische Hochschule Zürich

Mit 93 Abbildungen und zahlreichen Beispielen
zu Datenstrukturen und Algorithmen



E8763339



B. G. Teubner Stuttgart 1986

Dr. sc. techn. Andreas Meier

1951 geboren in Basel. 1971 bis 1972 Fagottstudien an der Musikakademie in Wien, anschließend Mathematikstudium an der Eidgenössischen Technischen Hochschule (ETH) in Zürich. 1977 Diplom in Mathematik bei B. Eckmann. 1977 bis 1980 Systemingenieur bei der IBM Schweiz. 1980 bis 1982 Assistent am Institut für Informatik der ETH Zürich und Sekretär der neugegründeten Abteilung für Informatik. 1982 Promotion bei C. A. Zehnder und J. Nievergelt. 1982 bis 1983 Forschungsaufenthalt am IBM Research Lab in San Jose, Kalifornien. Seit 1983 Oberassistent und Lehrbeauftragter am Institut für Informatik der ETH Zürich.

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Meier, Andreas:

Methoden der grafischen und geometrischen Datenverarbeitung /
von Andreas Meier. -
Stuttgart: Teubner, 1986.

(Leitfäden der angewandten Informatik)
ISBN 3-519-02482-9

Das Werk ist urheberrechtlich geschützt, Die dadurch begründeten Rechte, besonders die der Übersetzung, des Nachdrucks, der Bildentnahme, der Funksendung, der Wiedergabe auf photomechanischem oder ähnlichem Wege, der Speicherung und Auswertung in Datenverarbeitungsanlagen, bleiben auch bei Verwertung von Teilen des Werkes, dem Verlag vorbehalten.

Bei gewerblichen Zwecken dienender Vervielfältigung ist an den Verlag gemäß § 54 UrhG eine Vergütung zu zahlen, deren Höhe mit dem Verlag zu vereinbaren ist.

© B. G. Teubner, Stuttgart 1986

Printed in Germany

Gesamtherstellung: Zehnersche Buchdruckerei GmbH, Speyer

Umschlaggestaltung: M. Koch, Reutlingen

Vorwort

Grafik und Geometrie sind die siamesischen Zwillinge der technischen Informatik! Umfasst die Computergrafik Geräte und Verfahren zur Beschreibung und *Umwandlung von Daten in grafische Form*, so behandelt die geometrische Datenverarbeitung die Speicherung und *Verarbeitung geometrischer Daten*. Beinahe jede technisch-wissenschaftliche Anwendung der Informatik benötigt grafische und geometrische Methoden, um z.B. zwei- oder dreidimensionale Objekte darstellen oder deren Gestalt erfassen zu können. Das vorliegende Textbuch möchte diese Verwandtschaft zwischen Grafik und Geometrie untermauern.

Das Textbuch basiert auf der Vorlesung "Computergrafik und geometrische Datenverarbeitung" der Abteilung für Informatik der ETH Zürich. Die Stoffauswahl berücksichtigt verschiedene Ausbildungsprogramme an europäischen und amerikanischen Hochschulen, konzentriert sich aber auf die wesentlichen *Datenstrukturen und Algorithmen* für die grafische und geometrische Datenverarbeitung. Weiter haben direkte Anwenderkontakte, entstanden durch ein Kursangebot für die Praxis und unzählige Diskussionen mit Fachkollegen die Themenauswahl beeinflusst.

Nach einem einführenden Überblick im Kapitel 1 über die Entwicklung grafischer und geometrischer Datenverarbeitung werden im Kapitel 2 klassische Probleme der Computergrafik erläutert, nämlich Transformationen, grafische Primitiven und Operationen, Clipping und das Evaluieren verdeckter Kanten und Flächen. Grundsätzliche Datenstrukturen und Algorithmen zur Geometrie sind im Kapitel 3 beschrieben. Dazu gehören neuere Entwicklungen aus dem noch jungen Fachgebiet der geometrischen Algorithmik, z.B. mehrdimensionale Datenstrukturen zur Speicherung räumlicher Daten oder algorithmische Techniken für das Lokalisieren von Punkten, das Berechnen der konvexen Hülle oder für die Schnittbildung. Kapitel 4 gibt eine Einführung in die Kurven- und Flächengeometrie und erläutert vor allem Bézier- und B-Spline-Methoden. Kapitel 5 behandelt dreidimensionale Problemstellungen, wie sie z.B. beim rechnergestützten Konstruieren von geometrischen Objekten auftreten. Schliesslich rundet das Kapitel 6 über technisch-wissenschaftliche Anwendungen den Stoffinhalt ab, um zugleich auf Entwicklungstendenzen in der grafischen und geometrischen Datenverarbeitung hinzuweisen.

Das Textbuch richtet sich an Interessierte verschiedener *Ingenieurwissenschaften*, welche eine Einführung in die wesentlichen Methoden der grafischen und geometrischen Datenverarbeitung suchen. Dazu zählen Studenten oder Absolventen technischer Hochschulen und Lehranstalten sowie Ausbilder, Entwickler und Anwender in Firmen, die moderne Informatikwerkzeuge kennenlernen und einsetzen möchten. Ein umfangreiches Literaturverzeichnis mit Standardwerken und wichtigen neueren Forschungsarbeiten soll zum Weiterstudium motivieren.

Die meisten der vorgestellten Datenstrukturen und Algorithmen sind in Semester-, Diplom- und Promotionsarbeiten auf dem Arbeitsplatzrechner LILITH des Instituts für Informatik der ETH Zürich in der Programmiersprache Modula-2 implementiert worden. Diese Programme bilden einen wesentlichen Bestandteil der zitierten Vorlesung, indem

den Studenten für einzelne Probleme Programmumgebungen zur Verfügung gestellt werden. Die Vorzüge einer modularen Programmiersprache haben sich auch für Unterrichts- und Übungszwecke einmal mehr bewährt.

Eliyezer Kohen hat ein interaktives System zum Entwurf digitaler Schrift entwickelt. Josef Bösze und Reto Gilli realisierten die Clipping-Algorithmen von Cohen-Sutherland und von Liang-Barsky, um das Laufzeitverhalten bei verschiedenen Szenen vergleichen und beurteilen zu können. Die Implementation der mehrdimensionalen Gitterdatei stammt von Klaus Hinrichs. Giordano Beretta und Ernst Horber haben ein Skelett geschrieben, womit sie die ebene Durchlauftechnik an verschiedenen geometrischen Problemstellungen demonstrieren. Eine kleine Programmsammlung zur Generierung von Bézier- resp. B-Spline-Kurven und -Flächen stammt von Renzo De Maria und Erwin Petry. Im Rahmen des geometrischen Unterrichtssystems POLY haben Thomas Kohler einen Algorithmus für die Berechnung der verdeckten Kanten, Hansbeat Loacker die Mengenoperationen und Fredy Paquet einen rekursiven Algorithmus zur Evaluation von Booleschen Ausdrücken über Primitiven implementiert.

Am Institut für Informatik der ETH Zürich sind eigene Forschungsarbeiten im Bereich der geometrischen Datenverarbeitung entstanden. Zu erwähnen sind die vom Schweizerischen Nationalfonds geförderten Forschungsprojekte, nämlich in den Jahren 1982 bis 1984 das Projekt Nr. 2.533-0.82 über "Algorithmen und Datenstrukturen für geometrische Probleme" und das laufende Projekt Nr. 2.734-0.85 über "Darstellung und Speicherung von geometrischen Objekten in einer relationalen Datenbank".

Für wertvolle Anregungen und Verbesserungen zum vorliegenden Textbuch möchte ich mich herzlich bedanken. Sie stammen von F. Aurenhammer, H.-P. Bieri, H. Edelsbrunner, Ch. Eidenbenz, G. Enderle, F. Paquet, G. Heiser, K. Hinrichs, J. Nievergelt, H. Noltemeier, E. Petry und W. Strasser. Thomas Kohler hat mit grossem Einsatz und kritischem Auge meine skizzenhaften Zeichnungen ins reine gebracht. Dem Teubner-Verlag möchte ich für die reibungslose Zusammenarbeit und die rasche Drucklegung danken.

Zürich, im Februar 1986

Andreas Meier

Inhaltsverzeichnis

Vorwort	5
Inhaltsverzeichnis	7
1 Entwicklungsstand grafischer und geometrischer Datenverarbeitung	11
1.1 Historischer Überblick	11
1.2 Hardwarekomponenten	14
1.2.1 Klassifikation grafischer Geräte	14
1.2.2 Aufbau einer Grafikstation	16
1.2.3 Arbeitsweise von Vektor- und Rasterbildschirmen	18
1.3 Softwarekomponenten	20
1.3.1 Softwareaufbau grafischer Geräte	20
1.3.2 Geometrische Methoden- und Datenbanken	23
1.3.3 Gestaltung der Mensch-Maschine-Schnittstelle	24
1.4 Abgrenzungen verwandter Fachgebiete	28
2 Grundlagen der Computergrafik	30
2.1 Transformationen in homogenen Koordinaten	30
2.1.1 Definition homogener Koordinaten	30
2.1.2 Translation, Rotation und Skalierung	32
2.1.3 Koordinatentransformation Window/Viewport	34
2.1.4 Zentral- und Parallelprojektion	37
2.2 Grafische Primitiven und Operationen	40
2.3 Algorithmen zur Rastergrafik	42
2.3.1 Bresenham-Algorithmus für Geraden	42
2.3.2 Füllen von Polygonen	45
2.3.3 Erzeugen von Textsymbolen	47
2.4 Abschneiden von Figuren und Flächen am Fensterrand	49
2.4.1 Clipping von Geraden	49
2.4.2 Clipping von Polygonen, Flächen und Textelementen	53
2.4.3 Berechnen der sichtbaren Teile von Objekten im Raum	55
2.5 Evaluieren verdeckter Kanten und Flächen	57
2.5.1 Bestimmen durch wiederholte Bildunterteilung	57
2.5.2 Berechnungen im Objektraum	61
2.5.3 Prioritätsverfahren für Dreiecksflächen	65
2.6 Standard zur Grafik	68

3	Datenstrukturen und Algorithmen zur Geometrie	70
3.1	Effizienzkriterien	70
3.2	Mehrdimensionale Datenstrukturen	73
3.2.1	Baumstrukturen	74
3.2.2	Zellstrukturen	78
3.3	Interpretation mehrdimensionaler Daten	83
3.4	Inklusionsfragen	86
3.4.1	Punkt-im-Polygon-Test	86
3.4.2	Lokalisieren von Punkten	90
3.4.3	Monotone Kettenzerlegung	92
3.5	Konvexität und konvexe Hülle	95
3.5.1	Prüfen auf Konvexität	95
3.5.2	Die Fächermethode von Graham	98
3.5.3	Konvexe Hülle durch Stützgerade	99
3.5.4	Rekursive Bestimmung der konvexen Hülle	101
3.6	Schnittalgorithmen in der Ebene und im Raum	104
3.6.1	Schnitt von achsenparallelen Objekten	104
3.6.2	Schnittberechnung mit der Durchlauftechnik	107
3.6.3	Geometrische Transformationen für Halbraumschnitt	111
3.7	Nachbarschaften und Zerlegungsprobleme	114
4	Approximation von Kurven und Flächen	117
4.1	Parameterdarstellung	117
4.2	Approximation von Kurven durch Polynome	119
4.2.1	Kubische Kurven	119
4.2.2	Bézier-Kurven vom Grad m	123
4.2.3	Rekursiver Algorithmus von De Casteljau	127
4.3	Stückweise Approximation durch Polynome	130
4.3.1	B-Splinefunktionen	130
4.3.2	B-Splinekurven	134
4.4	Approximation von Flächen	137
4.5	Vergleich von Bézier- und B-Spline-Methoden	139
5	Geometrisches Modellieren	141
5.1	Analytische und approximierende Verfahren	141

5.2	Kriterien für Darstellungsformen	144
5.3	Übersicht über Darstellungsformen	146
	5.3.1 Parametrisierte Darstellung	147
	5.3.2 Enumerationsverfahren	148
	5.3.3 Zellenzerlegung	149
	5.3.4 Randdarstellung	151
	5.3.5 Konstruktion mit Raumprimitiven	152
5.4	Modellieren mit Begrenzungsflächen	154
	5.4.1 Grundlagen der Polyedertopologie	155
	5.4.2 Anwendung von Euler-Operatoren	157
	5.4.3 Datenstruktur zur Randdarstellung	159
	5.4.4 Berechnen des Produktkörpers	162
5.5	Modellieren mit Raumprimitiven	167
	5.5.1 Reguläre Mengenoperationen	168
	5.5.2 Evaluieren der Mengenzugehörigkeit	170
	5.5.3 Vereinfachen von Konstruktionsbäumen	174
5.6	Rekonstruktion von Polyedern	177
5.7	Berechnen von Volumeneigenschaften	180
5.8	Standardvorschlag zur Geometrie	182
6	Technisch-wissenschaftliche Anwendungen	183
6.1	Geographische Systeme	184
	6.1.1 Abstraktionsschritte bei flächenbezogenen Daten	184
	6.1.2 Flächenbezogene Objekte und Beziehungen	186
	6.1.3 Strukturbeschreibung eines Parzellenplans	189
	6.1.4 Konsistenzerhaltende Operationen	192
	6.1.5 Kontextbedingungen beim Ändern von Parzellen	195
6.2	Rechnergestützte Konstruktion	199
	6.2.1 Konstruktionsschritte	199
	6.2.2 Dateisysteme versus Datenbanksysteme	201
	6.2.3 Vergleich der Benutzeranforderungen	203
	6.2.4 Speicherung geometrischer Objekte in einer Datenbank	204
	6.2.5 Ein Surrogatmodell für technische Datenbanken	208
	Literaturverzeichnis	211
	Stichwortverzeichnis	222

1 Entwicklungsstand grafischer und geometrischer Datenverarbeitung

Das Kapitel gibt einen Einblick in das Gebiet der grafischen und geometrischen Datenverarbeitung. Im Abschnitt 1.1 skizzieren wir die historische Entwicklung und führen gleichzeitig die wichtigsten Begriffe ein. Die unterschiedlichen Gerätekomponenten eines grafischen Arbeitsplatzes erläutern wir im Abschnitt 1.2. Wir beschreiben die Softwarekomponenten grafischer Systeme im Abschnitt 1.3, wobei wir kurz auf die Gestaltung der Mensch-Maschine-Schnittstelle eingehen. Der Abschnitt 1.4 präzisiert die unterschiedlichen Fachgebiete der grafischen und geometrischen Datenverarbeitung.

1.1 Historischer Überblick

Seit dem Einsatz von Kathodenstrahlröhren zum Zeichnen einfacher Liniengebilde hat sich die grafische und geometrische Datenverarbeitung rasant entwickelt und ein breites Anwendungsfeld eröffnet. Dabei hat jede technische Errungenschaft bei der Entwicklung grafischer Geräte direkt die grafischen und geometrischen Methoden beeinflusst. Die Abb. 1-1 gibt einen Überblick über die Geschichte und die wichtigsten technischen Anwendungen.

Die *Vektorgrafik* der ersten Grafikgeräte ermöglicht einfache Strich- und Kurvenzeichnungen, entsprechend beschreibt man räumliche Objekte durch Linienelemente. Bei diesem sogenannten Skelett- oder *Drahtmodell* liegen keine Informationen über Flächen- oder Volumeneigenschaften vor: Operationen wie Schnittbildung oder Evaluation verdeckter Kanten sind direkt nicht möglich. Aufgrund unzureichender räumlicher Modelle ist es z.B. schwierig, Montage- und Fertigungszellen zu beschreiben oder automatische Kollisionskontrollen durchzuführen. So verwendet man für die Steuerung von Werkzeugmaschinen eigene Sprachen, wobei die Geometrie der Bearbeitungsteile jedesmal neu erfasst werden muss.

Das Abspeichern grafischer Primitiven in einem Bildwiederholpeicher lässt den Benutzer ohne grösseren Zeitverlust grafische Daten verändern oder bewegen, eine Voraussetzung für die *interaktive Computergrafik*. Das Ausblenden verdeckter Kanten verbessert zudem die Sichtbarkeit und die Anschauung. Analytische, interpolierende sowie *approximierende Verfahren* zur Flächenbeschreibung legen die Basis zu den rechnergestützten

Entwurfssystemen. Neben dem Erstellen technischer Zeichnungen setzt man die grafischen Systeme auch für Simulationen ein, um physikalische Eigenschaften der Entwurfsobjekte überprüfen zu können.

	GRAFIK	GEOMETRIE	TECHNISCHE ANWENDUNGEN
1950 -1960	Vektor- oder Liniengrafik	einfache geometrische Algorithmen, Drahtmodell für räumliche Objekte	numerische Steuerung, Fräsprogramme
1960 -1970	Interaktive Computer- grafik, Algorithmen für verdeckte Kanten und Flächen	Approximationsmethoden für Kurven und Flächen, Entwicklung geometrischer Programmiersprachen	Entwurfssysteme zum Zeichnen, Simulation, Bildverarbeitung
1970 -1980	Rastergrafik, Standardvorschläge, Animation, Computer- spiele	eindeutige Darstellung räumlicher Objekte, Komplexitätsbetrachtungen geometrischer Algorithmen	Entwurfssysteme für mechanische Teile bzw. integrierte Schaltungen, Industrieroboter, geographische Systeme
1980 -1990	Kognitive Computer- grafik, Bewegung, Computervision	Geometrische Daten- und Methodenbanken, logische Systeme, Standardisierung	Integrierte CAD/CAM- Systeme, wissensba- sierte Systeme für Produktionsplanung und Fertigung

Abb. 1-1: Entwicklungsüberblick und Anwendungsbereiche.

Seit dem vergangenen Jahrzehnt gewinnt die *Rastergrafik* an Bedeutung. Ein Bild wird durch eine Matrix von Bildpunkten beschrieben, wobei pro Bildpunkt mehrere Farbstufen möglich sind. Das Verwenden eigener Grafik- und Dialogprozessoren verhilft der Rastergrafik in vielen Anwendungsbereichen trotz anfänglichen Speicher- und Zeitengpässen zu einem Durchbruch. Zusätzlich ermöglichen *eindeutige Darstellungsformen*

für räumliche Objekte erste Resultate beim Einsatz von dreidimensionalen Entwurfssystemen sowie von Robotern.

Heutige Geräte mit leistungsfähigen Prozessoren erlauben optische Wahrnehmung und Darstellung von bewegten Objekten. Diese Geräte setzt man z.B. für die automatisierte Qualitätskontrolle oder für Steuerungsaufgaben bei der Produktherstellung ein. Die Integration der rechnergestützten Entwicklung und Fertigung (CAD/CAM) stellt ebenfalls grosse Anforderungen an die Software grafischer Systeme. Insbesondere möchte man Datenbanktechnologie und Expertenwissen im geometrischen Anwendungsbereich nutzen, um die Entwicklungs- und Produktionsverfahren zu verbessern.

1.2 Hardwarekomponenten

1.2.1 Klassifikation grafischer Geräte

In der grafischen und geometrischen Datenverarbeitung stellt sich immer wieder das Bedürfnis nach Kommunikation, da oft mehrere Fachgruppen am Entwurf und an der Entwicklung eines Produktes teilnehmen. Das Erstellen von Schaltplänen oder Konstruktionszeichnungen, das Durchführen von Simulationen oder das Kalkulieren von Herstellungskosten basiert auf gemeinsamen Daten und erfordert deshalb Datenübertragungsfunktionen. Im Gegensatz zu Rechneranwendungen ohne Grafikmöglichkeiten verlangt die dezentrale Verarbeitung von grafischen oder geometrischen Daten eigene Prozessoren, da umfangreiche Datenbestände interaktiv bearbeitet werden sollen.

Wir diskutieren im folgenden eine mögliche Klassifikation grafischer Geräte aufgrund des Funktionsumfangs (Abb. 1-2). Generell lassen sich grafische Geräte, grafische Geräte mit Intelligenz sowie lokale und dezentrale Grafikstationen unterscheiden.

Grafische Geräte verfügen über keine eigene Rechnerkapazität und verlangen eine ausreichende Datenpufferung im Hauptrechner, um die Manipulation umfangreicher grafischer Daten zu bewältigen. Arbeitsplätze mit solchen heteronomen Grafikgeräten zeigen im Fall schlechter Übertragungsraten unbefriedigend lange Antwortzeiten und verunmöglichen interaktives Arbeiten.

Geräte mit Intelligenz entlasten den Hauptrechner bei Ein- und Ausgabeaktivitäten. Sie umfassen in Hardware implementierte oder mikroprogrammierte Funktionen: Bildmanipulation und Datenzwischenspeicherung sind lokal möglich, wobei grössere Änderungen grafischer Daten periodisch zum Hauptrechner übermittelt werden müssen.

Lokale Grafikstationen besitzen eigene Prozessoren, z.B. zur Unterstützung der grafischen Ein- und Ausgabe oder zur Dialogführung. Da alle Applikationen selbständig ablaufen, d.h. ohne Verbindung zu anderen Rechnern, muss eine solche Station Speichermöglichkeiten zur Verwaltung von Daten und Programmen anbieten.

Dezentrale Grafikstationen erlauben, rechenaufwendige Programme (z.B. für Finite Elementberechnung) in andere Rechner auszulagern. Diese Arbeitsteilung kann sich über die Programmverarbeitung hinaus auf Datenspeicherung und -verwaltung ausdehnen. Falls eine zentrale Datenbank in einem Hauptrechner residiert, können die jeweiligen Daten von der Arbeitsstation angefordert, bearbeitet und später zurückgeschrieben werden.

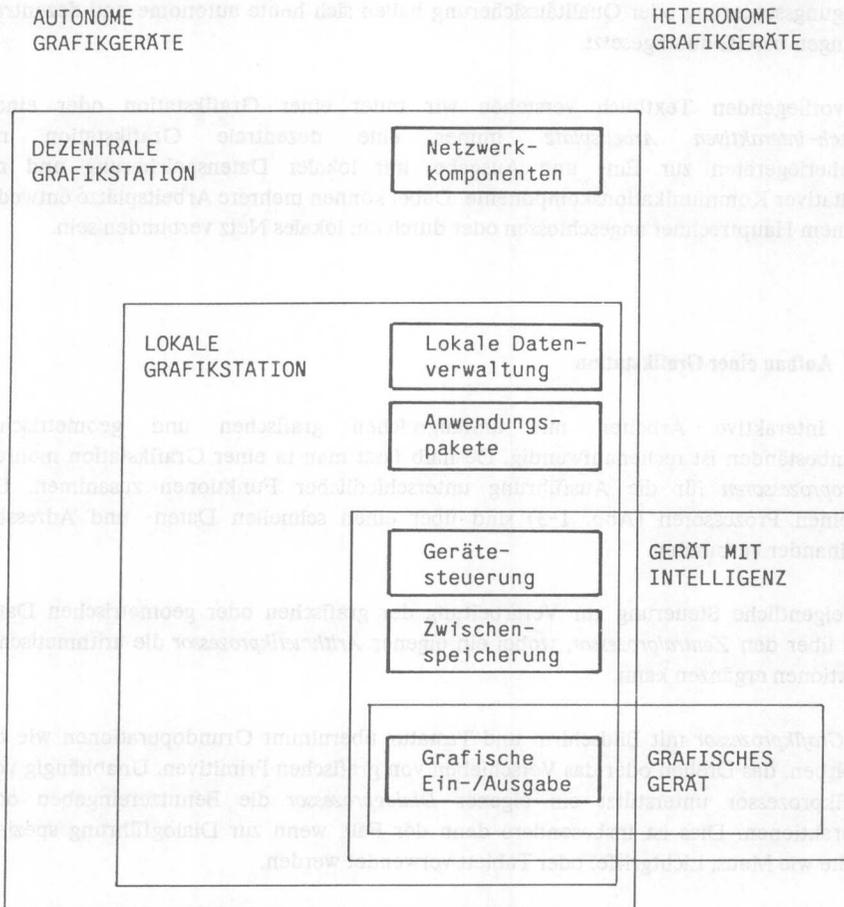


Abb. 1-2: Funktionsumfang von grafischen Geräten und Grafikstationen.

Die Auswahl einer Geräteklasse ist je nach Anwendungs- und Einsatzbereich unterschiedlich (siehe z.B. [Eigner/Maier 1982] oder [Encarnaçao et al. 1984]). Für Tätigkeiten wie Arbeitsplanung oder Auftragsabwicklung genügen meistens grafische Geräte mit Host-Anschluss, Tätigkeiten aus dem Bereich Entwicklung und Fertigung verlangen jedoch nach autonomen Rechnerlösungen. Schon beim Einsatz eines grafischen Systems für die interaktive Zeichnerstellung gehören klassische Konfigurationen mit oft überlasteten Host-Rechnern nicht zuletzt aus wirtschaftlichen Überlegungen der

Vergangenheit an. Bei vielen technischen Anwendungen wie Werkzeugkonstruktion, Fertigungssteuerung oder Qualitätssicherung haben sich heute autonome und dezentrale Lösungen bereits durchgesetzt.

Im vorliegenden Textbuch verstehen wir unter einer Grafikstation oder einem *grafisch-interaktiven Arbeitsplatz* immer eine dezentrale Grafikstation mit Peripheriegeräten zur Ein- und Ausgabe, mit lokaler Datenspeicherung und mit fakultativer Kommunikationskomponente. Dabei können mehrere Arbeitsplätze entweder an einem Hauptrechner angeschlossen oder durch ein lokales Netz verbunden sein.

1.2.2 Aufbau einer Grafikstation

Das interaktive Arbeiten mit umfangreichen grafischen und geometrischen Datenbeständen ist rechenaufwendig. Deshalb fasst man in einer Grafikstation mehrere *Mikroprozessoren* für die Ausführung unterschiedlicher Funktionen zusammen. Die einzelnen Prozessoren (Abb. 1-3) sind über einen schnellen Daten- und Adressbus miteinander verbunden.

Die eigentliche Steuerung zur Verarbeitung der grafischen oder geometrischen Daten läuft über den *Zentralprozessor*, wobei ein eigener *Arithmetikprozessor* die arithmetischen Funktionen ergänzen kann.

Ein *Grafikprozessor* mit Bildschirm und Tastatur übernimmt Grundoperationen wie das Einfärben, das Drehen oder das Verschieben von grafischen Primitiven. Unabhängig vom Grafikprozessor unterstützt ein eigener *Dialogprozessor* die Benutzereingaben oder -interaktionen. Dies ist insbesondere dann der Fall, wenn zur Dialogführung spezielle Geräte wie Maus, Lichtgriffel oder Tablett verwendet werden.

Maus, Steuerknüppel und Rollkugel unterstützen das *analoge Arbeiten* mit grafischer Information, da sie z.B. ein elektrisches Potentiometer integrieren. Die Stellung dieses Potentiometers ist ein Mass für die x- und y-Position eines Fadenkreuzes oder eines Cursors. Neben der Positionierungsaufgabe können diese grafischen Eingabegeräte durch zusätzliche Knöpfe weitere Funktionen übernehmen, z.B. analoge Eingabe von Drehwinkeln oder Verschiebeabständen.

Ein Potentiometer als Eingabegerät wirkt indirekt, d.h. grafische Primitiven lassen sich nur über Positionsänderungen eines Fadenkreuzes oder eines Cursors ansprechen. Demgegenüber ermöglicht ein Lichtgriffel ein *direktes Identifizieren*, indem er ein