

Zeitschrift:	Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses
Herausgeber:	Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen
Band:	75 (1984)
Heft:	11
Artikel:	Adaption eines Videosystems an einem Roboter
Autor:	Bichsel, J. / Buess, C.
DOI:	https://doi.org/10.5169/seals-904412

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 26.01.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Adaption eines Videosystems an einen Roboter

J. Bichsel, Ch. Buess

Ein Grossteil der heute eingesetzten Roboter arbeitet nach Programmen, die voraussetzen, dass sich die zu bewegenden Objekte an einer vorbestimmten Stelle in einer vorbestimmten Lage befinden. Kann dies nicht gewährleistet werden, müssen die Roboter mit Erkennungssystemen, z.B. Videosystemen, ausgerüstet werden, die nach Auswertung von Bildinformationen die Bewegung steuern. Im folgenden werden die Resultate der Realisierung eines kameragesteuerten Robotersystems gezeigt.

Die Arbeit entstand im Wintersemester 1983 am Centre for Robotics and Automated Systems des Imperial College London im Rahmen einer Zusammenarbeit mit der Professur für elektrotechnische Entwicklungen und Konstruktionen der ETH Zürich.

Les robots actuellement employés fonctionnent en grande partie selon des programmes exigeant des objets à déplacer se trouvant à un emplacement et dans une position déterminés. Lorsque cela n'est pas le cas, les robots doivent être équipés de systèmes de reconnaissance, par exemple de systèmes vidéo, qui commandent le mouvement après interprétation des informations affichées. Les résultats de la réalisation d'un système de robot commandé par caméra sont présentés.

1. Einleitung

Zuerst eine Definition des im folgenden verwendeten Begriffes «Roboter». Darunter soll eine Maschine verstanden werden, die in der Lage ist, Gegenstände wie Bauteile oder Werkzeuge zu greifen, zu halten, im dreidimensionalen Raum zu bewegen und an bestimmten Positionen abzulegen. Die hierfür erforderliche mechanische Struktur muss Bewegungsvorgänge mit mehreren Freiheitsgraden ausführen können, deren Zahl von der Komplexität der gestellten Aufgabe abhängt. Figur 1 zeigt ein mögliches Schema eines Robotersystems. Für eine vorgesehene Aufgabe erstellt ein Operateur ein Programm der Roboterarmbewegungen. Dieses Programm wird schrittweise vom Kontroller abgearbeitet, der einerseits Signale zur Bewegung des Roboterarms sendet, andererseits aber Rückmeldungen der räumlichen Lage des Roboterarms erhält.

Soll das Gerät zusätzlich bestimmte Situationen in der Umgebung erkennen und berücksichtigen (Selbstanpassung an die Umgebung), so muss das System durch eine computerunterstützte Auswertung von Rezeptorinformationen ergänzt werden. Diese Daten werden dem Kontroller als Steuerinformationen eingegeben. Als

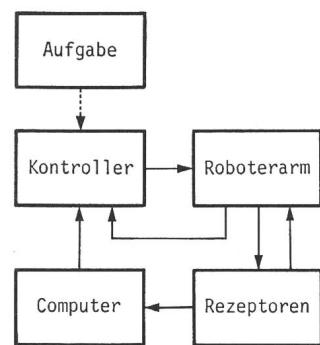


Fig. 1 Blockschema eines Robotersystems

Rezeptoren, in der Technik meist als Sensoren bezeichnet, eignen sich beispielsweise optische Erkennungsgeräte oder Tastfühler.

Die Auswahlkriterien für den Roboterarm, wie Traglast (100 g bis 100 kg), Positioniergenauigkeit (0,1 mm bis 1 cm) und der durch die Bewegung erfassbare Raum hängen vom Anwendungsbereich des Gerätes ab. Das Mass für die Beweglichkeit ist der Freiheitsgrad. Er gibt an, in wie vielen Achsrichtungen ein Roboter bewegt werden kann, wobei zu beachten ist, dass im allgemeinen diese Freiheitsgrade nicht unabhängig voneinander sind. In Tabelle I finden sich einige Anwendungsbeispiele von Industrierobotern.

Einige Anwendungsbeispiele für Industrieroboter

Tabelle I

Anwendungsgebiet	Eigenschaften	Traglast	Genauigkeit	Beweglichkeit
Giessen, Materialtransport	gross	wenig präzis	gross	
Schweißen, Montage	mittel	präzis	sehr gross	
Leiterplattenherstellung	klein	sehr präzis	mittel	

Adresse der Autoren

Jürg Bichsel und Christian Buess, Professur für elektrotechnische Entwicklungen und Konstruktionen, ETH-Zentrum, 8092 Zürich.

2. Problemstellung

Viele der heute verwendeten Industrieroboter können ihre Aufgabe ohne Rezeptoren erfüllen, da ihnen von jedem relevanten Gegenstand die räumliche Position bekannt ist. Andere, bei denen diese Voraussetzung nicht erfüllt ist, müssen mit Sensoren, z.B. optischen Aufnehmern zum Erkennen der Gegenstände, ausgerüstet werden, was allerdings heute in Anbetracht des grossen Aufwandes noch relativ selten geschieht. Da aber anderseits solche Systeme sich durch einen hohen Grad von Anpassungsfähigkeit an die verschiedensten Erkennungsaufgaben bei einheitlicher Ausführung der Hardware auszeichnen, lohnt es sich, diese Technik näher zu untersuchen, um deren Möglichkeiten und Grenzen zu erkennen. Bei der praktischen Realisierung interessieren die Verwendung einfachster Komponenten und ein günstiges Verhältnis von Leistung zu Aufwand. Die Untersuchungen wurden an einem System durchgeführt, das aus folgenden Komponenten aufgebaut ist:

- Roboter mit fünf Freiheitsgraden (Fig. 2)
- Videokamerasystem mit 128×100 Bildpunkten und mit je 16 Helligkeitsstufen pro Punkt
- 8-Bit-Computer mit 32-K-RAM zur Informationsverarbeitung des Videobildes und Steuerung des Roboters

Mit diesen Komponenten sowie mit der nötigen Software sollten nun Gegenstände im Arbeitsbereich des Roboters mit dem Videosystem detektiert, und anschliessend der Roboter

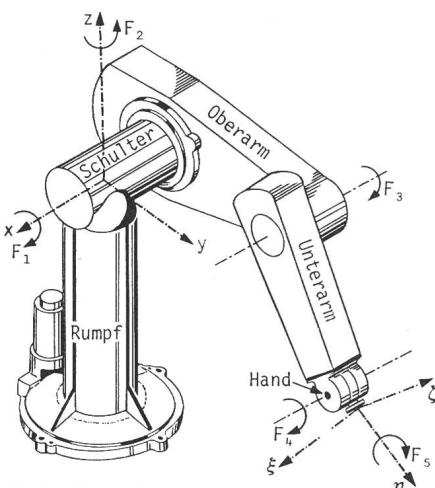
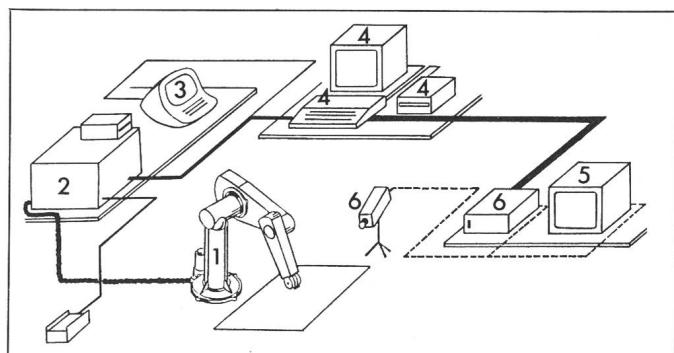


Fig. 2 Der Roboter

x, y, z Hauptkoordinatensystem
 ξ, η, ζ Werkzeugkoordinatensystem
 $F_1 \dots F_5$ Freiheitsgrade

Fig. 3
 Darstellung des Gesamtsystems

- 1 Roboter
- 2 Kontroller
- 3 zusätzliches Terminal
- 4 Minicomputer
- 5 zusätzlicher Monitor
- 6 Videosystem



zu diesen Gegenständen bewegt werden.

3. Problemlösung

3.1 Komponenten des Gesamtsystems

In Figur 3 sind die verwendeten Komponenten des Gesamtsystems sowie die elektrischen Verbindungen zwischen den einzelnen Bestandteilen dargestellt. Wie diese Komponenten zusammenarbeiten, soll im folgenden erläutert werden:

- Die Kamera ist auf die Arbeitsfläche des Roboters gerichtet; der Roboterarm hat freien Zugriff zu dieser Fläche. In einem ersten Schritt wird ein Bild dieses Arbeitsbereichs über das Interface des Videosystems in den Kleincomputer eingelesen.
- Das interessierende Bild befindet sich nun in digitalisierter Form im Speicher des Kleincomputers. Man muss davon ausgehen, dass dieses Abbild der realen Arbeitsfläche Redundanz wie auch Fehlinformation enthält. Der zweite Verarbeitungsschritt besteht deshalb in einer Informationsreduktion, welche das Auffinden der relevanten Punkte des Bildes ermöglicht.
- Hierauf kann der Kleincomputer aufgrund dieser Informationen Befehle an den Kontroller geben, die den Roboter veranlassen, die gefundenen Gegenstände aufzunehmen und zu entfernen, oder damit andere Manipulationen auszuführen.

Der Informationsfluss geht somit vom Videosystem über den Kleincomputer zum Roboter. In Figur 3 befinden sich noch weitere Komponenten, welche bis jetzt noch nicht erwähnt wurden. Diese sind: Erstens ein zusätzlicher, an die Kamera des Videosystems angeschlossener Monitor, der die direkte Kontrolle ermöglicht und die Bildeinstellung erleichtert. Zweitens ein Terminal in der Leitung zwischen Kleincomputer und Roboterkontroller, welches zur allfälligen Prü-

fung der Datenübertragung dient, und drittens eine kleine Bedienungseinheit, welche ein beliebiges Bewegen des Roboters ausserhalb der Steuerprogramme ermöglicht.

3.2 Nebenbedingungen und Teilprobleme

Aus der Tatsache, dass es sich beim Videosystem um eine relativ einfache Anlage handelt und dass der zur Verfügung stehende Computer nur eine geringe Speicherkapazität besitzt, ergeben sich folgende Randbedingungen:

- Es sollen Objekte detektiert werden, die praktisch als flach gelten können und die in einer bestimmten Arbeitsebene des Roboters liegen, was eine Beschränkung auf zweidimensionale Bildverarbeitung bedeutet.
- Verzicht auf eine Echtzeitkontrolle der Roboterbewegungen, da die Information vom Videosystem nicht genügend schnell geliefert werden kann.
- Die Kamera wird ausserhalb des Arbeitsbereichs des Roboters an einer bestimmten Stelle fixiert, wobei ein beliebiger Blickwinkel zugelassen ist.

Bereits die ersten Versuche zeigten, dass die mit dem Bilderfassungssystem gelieferte Software zur Lösung der gestellten Aufgabe nicht geeignet war. Es wurden deshalb spezielle Rechenprogramme zur Verarbeitung der Bildinformation entwickelt, wobei darauf geachtet wurde, mit Hilfe von Assemblerprogrammen die grösstmögliche Arbeitsgeschwindigkeit in diesem zeitkritischen Bereich zu erzielen.

Das zu lösende Problem lässt sich in Teilprobleme zerlegen, wobei immer höchstens zwei der Hauptkomponenten des Systems beteiligt sind (Tab. II). Im weiteren werden die einzelnen Probleme etwas detaillierter geschildert.

3.3 Einlesen eines Videobildes

Die zur Verfügung stehende Hardware erlaubt Punkt-für-Punkt-Einle-

Teilprobleme und ihre beteiligten Komponenten

Tabelle II

Teilprobleme	Beteiligte Komponenten	Kamera	Computer	Roboter
Einlesen und Verarbeiten des Bildes	●	●	○	
Informationsreduktion	○	●	○	
Koordinatentransformation	○	●	○	
Bewegen des Roboters	○	●	●	

sen des Videobildes. Dazu wird dem Videointerface vom Computer signalisiert, dass ein neues Bild benötigt wird, worauf dieses einen Bildpunkt nach dem anderen sendet, die vom Computer an einem Parallelport eingelesen werden. Die Bildgrösse beträgt 100×128 Punkte mit jeweils sechzehn Helligkeitsstufen, welche durch 4 Bit dargestellt werden. Die Bildinformation umfasst also $100 \times 128 \times 4 = 51200$ Bit oder 6400 Byte. Der Kleincomputer benötigt für das Einlesen eines Bildes etwa fünf Sekunden, woraus ersichtlich wird, das man hier nicht mehr von Echtzeitdatenverarbeitung reden kann.

Zur Kontrolle der Kameraeinstellung (Fokussierung, Blende, Kamerablickpunkt) wird das digitalisierte Bild noch in einer Ecke des Computerbildschirms dargestellt. Fehler in der Kamerajustierung oder der Datenübertragung können auf diese Weise vom Benutzer erkannt werden.

3.4 Informationsreduktion

Die Informationsreduktion des digitalisierten Bildes hat folgende Aufgaben: Es sollen Fehler beziehungsweise Störungen im Bild eliminiert und zudem wesentliche Punkte erkannt werden. Das Vorgehen wird anhand von Figur 4 erläutert.

Figur 4a zeigt einen Ausschnitt des Originalbildes in seiner ursprünglichen Form. Neben dem Hauptobjekt, welches als grosser schwarzer Fleck sichtbar ist, sind auch zwei Störungen im Bild enthalten. Unten links zeigt sich eine etwas dunklere Region, beispielsweise ein Schatten, und oben links ist eine kleine Verunreinigung zu erkennen. Zur Steuerung des Roboters ist nur die Mittelpunktskoordinate des Hauptobjektes wichtig. Im folgenden müssen die beiden Störungen eliminiert werden. Figur 4b zeigt das Bild

nach seiner Digitalisierung in der Darstellung mit vier Helligkeitsstufen: weiss, schraffiert, punktiert und schwarz. Die beiden Störungen im Originalbild sind noch vorhanden. Die Darstellung in Figur 4c verwendet nur noch die Helligkeitstufen weiss und

schwarz. Der Übergang vom vorherigen Bild zu diesem erfolgt dadurch, dass jeder Punkt, der dunkler als schraffiert ist, als schwarz definiert wird. Punktierter oder schwarze Punkte in Figur 4b werden also zu schwarzen Punkten in Figur 4c. Die Störung durch den Schatten ist eliminiert, die Verunreinigung oben links ist aber immer noch vorhanden. Der eben beschriebene Verarbeitungsschritt bedeutet eine Reduktion der Information von vier Bit (16 Helligkeitsstufen) auf ein Bit (schwarz-weiss). Die Schwelle zwischen schwarz und weiss kann vom Benutzer interaktiv gewählt werden und ist eine Funktion von Kontrast und Helligkeit. Im nächsten Schritt wird die Verunreinigung oben links, welche auch ein Fehler in der Kamerahardware sein könnte, beseitigt. Das geschieht dadurch, dass in Figur 4d nur noch diejenigen Punkte

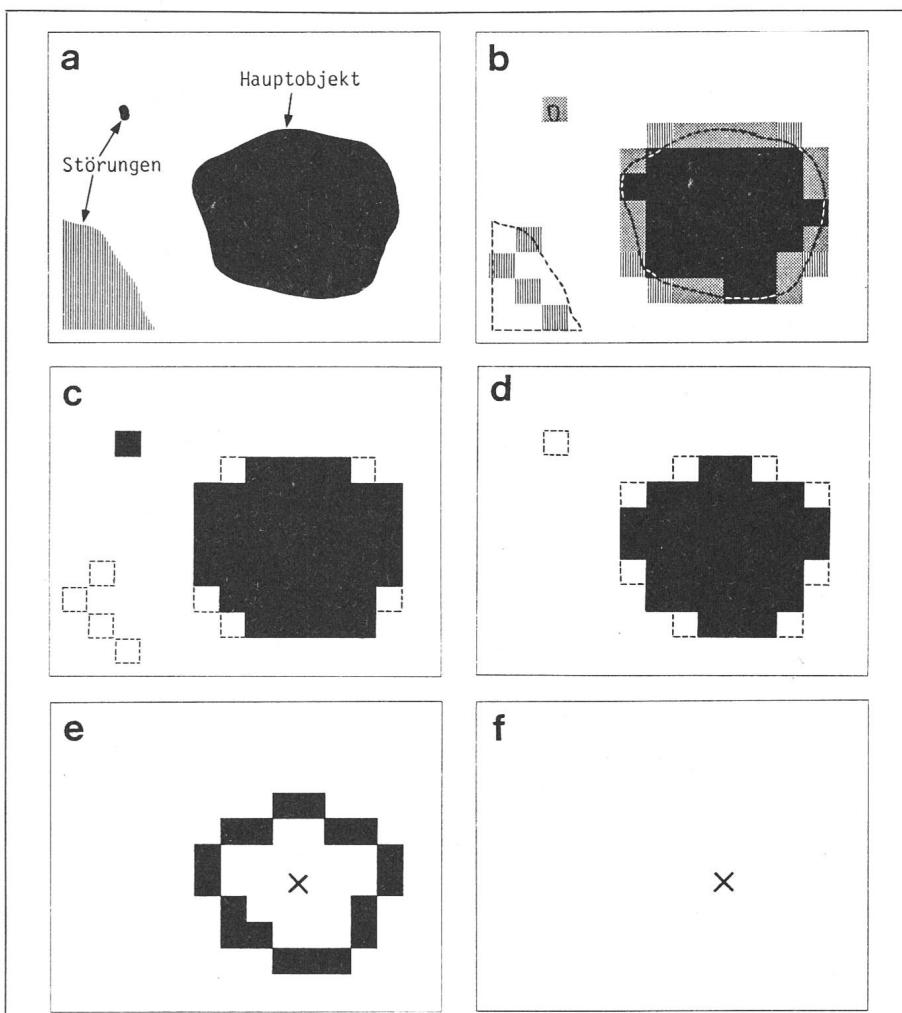


Fig. 4 Informationsreduktion eines Bildes

- a Originalbild
- b Digitalisierung mit vier Helligkeitsstufen
- c Reduktion auf zwei Helligkeitsstufen
- d Korrelation mit Umgebung
- e Randbildung
- f Schwerpunktbildung
- Konturen der vorgehenden Reduktionsstufe

schwarz sind, welche in Figur 4c über eine gewisse Mindestanzahl von schwarzen Nachbarn, im Beispiel mindestens fünf, verfügen. Fehler werden dadurch weitgehend eliminiert, und die Bilddaten können von nun an als richtig betrachtet werden.

Wie schon die vorherigen Schritte reduziert auch der nächste die Information weiter, indem das gesamte Bild nochmals so weiterverarbeitet wird, dass in Figur 4e nur noch die Grenzen der Objekte übrig bleiben. Diese Massnahme führt bei grossen schwarzen Objekten zu einer beträchtlichen Informationsreduktion und beschleunigt alle nachfolgenden Operationen. Im letzten Bearbeitungsschritt werden aus den Konturen die Schwerpunkte berechnet. Im Beispiel (Fig. 4d) stellt die Schwerpunktskoordinate des Hauptobjekts, durch ein Kreuz gekennzeichnet, die Endinformation dar.

Die beschriebenen Bildverarbeitungsschritte reduzieren die Bildinformation beträchtlich. Geht man von einem Bild mit vier schwarzen Hauptobjekten aus, so enthält dieses 12 800 Punkte mit 16 Helligkeitsstufen. Nach sämtlichen Verarbeitungsschritten erhält man als Endinformation die vier Schwerpunktskoordinaten der Hauptobjekte. Der Zeitbedarf für diese Informationsreduktion ist je nach Bild unterschiedlich. Der eingesetzte Kleincomputer benötigt für die Detektion von vier mittelgrossen schwarzen Punkten auf einer hellen Grundfläche etwa zwei Sekunden.

3.5 Koordinatentransformation

Damit der Roboter durch die Bildinformation gesteuert werden kann, muss der Zusammenhang zwischen den zweidimensionalen Bild- und den dreidimensionalen Roboterkoordinaten hergestellt werden. Dies kann durch ein sogenanntes Referenzbild erfolgen, das dem Computer erlaubt, die Bild- und Raumkoordinaten einander zuzuordnen. Zur Vereinfachung weiterer Berechnungen sollte es sich bei diesem Referenzbild um eine einfache geometrische Figur handeln, beispielsweise um ein Rechteck beliebiger Kantenlänge, wobei allerdings die Einschränkung gilt, dass das gesamte Rechteck durch das Videosystem erfasst wird und dass der Roboter zu den Eckpunkten bewegt werden kann. Durch manuelles Bewegen des Roboters zu den Eckpunkten werden dem Computer die Raumkoordinaten des Originals eingegeben. Aus dem

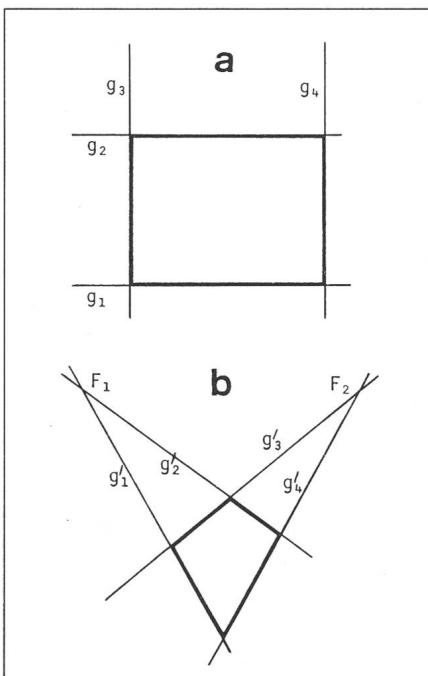


Fig. 5 Transformation von Original- in Bildkoordinaten
 a Original
 b Bild
 g_1, \dots, g_4 Originallinien
 g'_1, \dots, g'_4 Bildlinien
 F_1, F_2 Fernpunkte

Kamerabild werden die vier Eckpunktskoordinaten automatisch bestimmt, wobei eine korrekte Zuordnung von Original- zu Bildpunkten durch den Benutzer zu erfolgen hat. Da die Kamera nicht rechtwinklig zur Arbeitsfläche steht, ergeben sich perspektivische Verzerrungen, die durch eine bilineare Koordinatentransformation korrigiert werden. Wie aus Figur 5 ersichtlich ist, werden parallele Geraden des Originals in sich schneidende im Bild überführt. Um die Transformation durchführen zu können, sind zuerst die Fernpunkte F_1 und F_2 zu berechnen.

3.6 Bewegen des Roboters

Nachdem die Objekte auf der Arbeitsfläche detektiert sind, muss der Roboter zu ihnen hin bewegt werden. Die Lösung dieses Problems ist stark vom benutzten Robotertyp abhängig. Das hier beschriebene Modell ist mit einem relativ komfortablen Befehlsatz ausgestattet. Zwei der am häufigsten benutzten Befehle sollen als Beispiele herausgegriffen werden: *Drive* erlaubt das direkte Drehen eines Gelenks um einen bestimmten Winkel mit einer spezifizierten Geschwindigkeit. Ein anderer Befehl, genannt

Draw, leistet wesentlich mehr. Mit ihm kann die Spitze des Roboterarms in einem kartesischen Koordinatensystem mit einer definierten Geschwindigkeit entlang einer Geraden bewegt werden. Dieser Befehl ist komplexer, da zur Führung der Hand entlang einer Geraden im Raum alle Gelenke des Roboters gleichzeitig bewegt werden müssen. Andere Befehle erlauben, die Koordinaten des Roboterarms abzufragen. Da die Argumente der Kommandos sich aus der Bildinformation ergeben, kann die Roboterhand zum Objekt hin bewegt werden. Die Verbindung zwischen Kleincomputer und Kontroller wurde mit einer RS-232-C Schnittstelle realisiert.

4. Software

Zur Steuerung des Systems wurde ein handelsüblicher, preisgünstiger 8-Bit-Kleincomputer verwendet. Als Programmiersprachen standen ein komfortables Basic-System sowie ein einfacher Assembler zur Verfügung. Die Aufteilung der Aufgabe in getrennt zu lösende Teilprobleme findet sich in der Struktur der Software wieder. Die Programmiersprache Basic zwingt zwar nicht unbedingt zu einer strukturierten Programmierung. Trotzdem lassen sich auch hier die verschiedenen Routinen klar voneinander trennen. Eine zweckmässige Gliederung und Trennung der einzelnen Probleme erleichtert nämlich einerseits das Schreiben der Software und andererseits die Anpassung des Programmsystems an eine modifizierte Aufgabe.

Eine weitere Anforderung an die Software war, dass der Computer unter keinen Umständen abstürzen darf. Da der Systemaufbau aber ziemlich komplex ist, galt es deshalb, eine ganze Reihe von Fehlermöglichkeiten in Betracht zu ziehen, wie Schwierigkeiten der Kommunikation mit dem Roboter, falsche Benutzereingaben, Probleme in der Ansteuerung der Kamera und nicht zuletzt auch unentdeckte Programmierfehler. In all den beschriebenen Fällen sollte ein sauberer Programmabbruch sichergestellt sein. Auch sollte der Benutzer jederzeit die Möglichkeit haben, die Programmausführung anzuhalten. In den Programmen wurden diese Gesichtspunkte berücksichtigt. Nach jedem Programmstopp, sei er bewusst oder fehlerbedingt, wird eine vollständige Fehlermeldung ausgedruckt; der Computer

geht dann sogleich in einen definierten Wartezustand über.

Weiter musste überlegt werden, welche Programmteile zweckmässig in Basic und welche in Assembler verfasst werden sollen. Basic hat den Vorteil der besseren Übersichtlichkeit und einfacheren Formulierbarkeit von arithmetischen und logischen Abläufen, während Assemblerprogrammierung einen beachtlichen Geschwindigkeitsvorteil verschafft. Diese Überlegungen führten zu einer Aufteilung, in der der Grossteil der Programme in Basic geschrieben werden konnte, darunter das Hauptprogramm, der Benutzerdialog, die logische Programmsteuerung und sämtliche arithmetischen Berechnungen. Mit Assemblerprogrammen laufen die Kommunikation mit dem Roboter, die mit Basic nicht gut möglich ist, sowie die wesentlichen Schritte der Bildinformationsreduktion. Vor allem der letztgenannte Aufgabenbereich ist in Anbetracht der grossen Datenmengen ein typisches Anwendungsgebiet der MaschinenSprache.

Als interessantes Detail ist noch anzumerken, dass durch die Einführung paralleler Prozesse sämtliche Abläufe

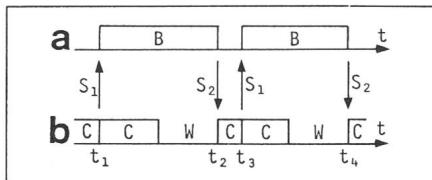


Fig. 6 Parallele Prozesse Roboter-Computer

a Roboter

b Computer

S₁ Befehl an Roboter

S₂ Signal an Computer zur Synchronisation

B Bewegen des Roboters

C Berechnungen des Computers

W Warten auf Synchronisation mit Roboter

wesentlich beschleunigt werden können. Das Vorgehen soll anhand der Figur 6 erläutert werden: Zum Zeitpunkt t_1 sendet der Computer dem Roboter einen Befehl. Der Computer wartet nun nicht, bis der Roboter diesen Befehl zu Ende ausgeführt hat, sondern berechnet schon die Argumente für den nächsten Roboterbefehl. Zum Zeitpunkt t_2 synchronisieren sich Roboter und Computer wieder, und der Computer kann zum Zeitpunkt t_3 den eben berechneten Befehl an den Roboterkontroller senden. Diese parallele Verarbeitung führt zu einer beträchtlichen Geschwindigkeitssteigerung.

5. Schlussfolgerungen

Mit den in Abschnitt drei und vier vorgestellten Grundideen konnte das Ziel, ein videogesteuertes Robotersystem aufzubauen, realisiert werden. Es zeigte sich, dass es mit verhältnismässig einfachen Mitteln möglich ist, Gegenstände im Bereich des Roboters mit Hilfe einer Kamera zu detektieren und anschliessend den Roboterarm zu diesen Objekten zu führen. Allerdings ergaben praktische Versuche oft kleine Ungenauigkeiten beim Positionieren des Roboters. Dies hatte zwei Ursachen. Einerseits ist die Genauigkeit der Objekterkennung durch die Kameraauflösung beschränkt, und andererseits zeigt der Positioniermechanismus des Roboters selbst noch grössere Toleranzen. Die über Drehgelenke positionierenden Roboter mit Schwenkarm sind für Bewegungen im kartesischen Koordinatensystem einer Fläche nicht besonders gut geeignet. Im weiteren hat sich gezeigt, dass bei komplizierterer Objektgeometrie der Rechenaufwand sehr stark ansteigt, weshalb von grosser Wichtigkeit ist, dass bei der Steuerung mit einer Videokamera die Bildauswertung auf wesentliche Merkmale des Objekts beschränkt bleibt.