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: 83 (1992)

Heft: 21

Artikel: Roboter im Postdienst : ein schnelles robustes Vision-System zur Paketvereinzelung

Autor: Rechsteiner, Martin / Schneuwly, Bruno / Guggenbühl, Walter

DOI: https://doi.org/10.5169/seals-902889

Roboter im Postdienst

Ein schnelles robustes Vision-System zur Paketvereinzelung

Martin Rechsteiner, Bruno Schneuwly, Walter Guggenbühl

Dreidimensionale Bildverarbeitung vermag die Einsatzmöglichkeiten von Robotern für Sortierprobleme stark zu erweitern. Dieser Beitrag beschreibt ein System, das mittels eines speziellen Bildverarbeitungsrechners ungeordnete Pakethaufen in Echtzeit zu vereinzeln erlaubt. Eine Stroboskop-Beleuchtung macht die Vision weitgehend unabhängig vom Umgebungslicht.

Grâce au traitement d'images en trois dimensions les possibilités de triage pour robots s'élargissent. Cet article décrit la réalisation d'un système pour l'isolement des paquets en temps réel employant un ordinateur spécialisé au traitement d'images. Grâce à la projection avec un flash l'interprétation des images est insensible à la lumière ambiante.

Adresse der Autoren

Martin Rechsteiner, Dipl. El.-Ing. ETH, *Bruno Schneuwly*, Dipl. El.-Ing. ETH, Prof. Dr. *Walter Guggenbühl*, Institut für Elektronik, ETH Zentrum, 8092 Zürich.

Es gibt viele industrielle Anwendungen, bei denen ungeordnete Objekte gezielt sortiert oder gegriffen werden müssen. Entsprechend vielseitig sind auch die konventionellen mechanischen Vereinzelungssysteme (z.B. Schüttelförderer). Da diese klassischen Methoden aber Einschränkungen bezüglich der Vielfalt der Formen und Materialien aufweisen, ist man heute noch häufig gezwungen, schwierige Sortierprobleme mit manuellen Methoden zu lösen. Solche Tätigkeiten sind jedoch oft eintönig und gefährlich, weshalb man gerne Roboter einsetzen würde. Diesem Wunsch sind Grenzen gesetzt durch die Bildverarbeitungssysteme, mit denen bis heute nur Sortieraufgaben durchgeführt werden, bei welchen die Erfassung einer zweidimensionalen Szene (2D-Vision) genügt. Dabei werden nur die Umrisse und eventuell die Helligkeit der Gegenstände erfasst und für die weitere Bildverarbeitung herangezogen. Einer einfachen 2D-Vision bereitet aber schon das Erkennen zweier sich überlappender Gegenstände erhebliche Schwierigkeiten. Vor dem Einsatz von 3D-Bildverarbeitung aber wird häufig noch zurückgeschreckt, da sie den Ruf hat, viel zu teuer, zu langsam und zu aufwendig zu sein. Dem ist entgegenzuhalten, dass die Entwicklung im Bereich der Aufnahme und Verarbeitung von dreidimensionalen Bildern (3D-Vision) in den letzten Jahren erhebliche Fortschritte gemacht hat, so dass einem industriellen Einsatz bald nichts mehr im Wege steht.

Die in diesem Artikel beschriebene Lösung zeigt, dass es heute möglich ist, viele komplexe Sortierprozesse zu automatisieren und die dazu benötigten Taktzeiten mit einem vernünftigen finanziellen Aufwand zu erreichen. Als Beispiel wurde das Sortieren von Postpaketen ausgewählt, eine Aufgabe, welche durch eine Machbarkeitsstudie [1] der Arbeitsgruppe Mechatronik an der ETH Zürich im Auftrag der Schweizerischen PTT initiiert wurde. Das realisierte Gesamtsystem basiert auf einem speziellen Bildverarbeitungsrechner für einen schnellen, billigen und präzisen Tiefensensor, bestehend aus einem Musterprojektor und einer Kamera zur Aufnahme von dreidimensionalen Bildern, sowie einem schnellen Roboter. Es ist uns bei diesem Projekt gelungen, ein 3D-Vision-System mit einem Roboter zu koppeln und die Bildverarbeitung schneller als die Greifbewegung zu machen.

Konzept

Ziel dieses Projekts war die Demonstration der Echtzeitfähigkeit eines 3D-Vision-Systems. Dies setzt voraus, dass die Bildverarbeitung schneller als die Zykluszeit des schnellen Roboters sein muss. Innerhalb der Zeitspanne, die der Roboter benötigt, um ein Paket abzulegen und während der er somit ausserhalb des Sichtbereichs der Kamera ist, sollte bereits die nächste 3D-Bildaufnahme (Tiefenbild) gemacht und ausgewertet sein.

Diese Forderung ist nur mit einer geeigneten Kombination von Bildverarbeitungs-Hardware und einem darin integrierten und auf Geschwindigkeit getrimmten Tiefensensor zu erfüllen. Da Bilder grosse Datenmengen darstellen, muss darauf geachtet werden, dass diese nicht von einem Teilsystem zum anderen übertragen werden müssen. Bei dem im Projekt eingesetzten Bildverarbeitungsrechner ist, von den AD-Wandlern für die Abtastung der Videosignale bis zur Schnittstelle, die

mit dem Roboter kommuniziert, alles auf engstem Raum beieinander.

Im weiteren sollte die Bildverarbeitung möglichst einfach gehalten werden und mit Algorithmen auskommen, die entweder in Hardware realisiert oder in Software effizient ausgeführt werden können. Grösstmögliche Sicherheit und Robustheit wurde dadurch erreicht, dass die Vision durch weitere Sensoren im Greifer des Roboters (Sensor-Fusion) unterstützt wird, welche in der Lage sind, fehlerhafte Informationen der Vision abzufangen. Da die Kamera nie alle Pakete eines komplexen Pakethaufens sehen kann, weil diese durch andere Pakete zum Teil verdeckt sind, werden die Pakethaufen immer von oben nach unten abgebaut. Nach jedem Greifen muss ein neues Tiefenbild aufgenommen werden, weil neue Pakete sichtbar werden können und weil infolge des Herausgreifens eines Pakets instabile Pakete ihre Lage verändern. Da bei diesem Projekt das Schwergewicht auf der Machbarkeit der schnellen Computer-Vision lag, wurde ein einfacher kommerzieller Parallelgreifer eingesetzt.

Der Tiefensensor

Prinzip

Die Grundlage des hier eingesetzten Tiefensensors ist die optische Triangulation. Vereinfacht lässt sich dieses Prinzip wie folgt erklären: Ein Projektor projiziert einen Lichtstreifen auf die Szene (Bild 1). Im Bild der Videokamera erscheint dieser Streifen je nach Höhe des Objekts verschieden stark verschoben. Aus der seitlichen Verschiebung (Δ_x) gegenüber der Projektion des Streifens auf eine Referenzebene und dem Winkel, unter dem der Lichtstrahl einfällt, lässt sich die Höhe h nach der vereinfachten Formel

$$h = \Delta x \bullet \tan(\alpha) \qquad (1)$$

berechnen. Um die Szene nicht streifenweise abtasten zu müssen, werden 2^n zur y-Richtung der Weltkoordinaten parallele Streifen mittels einer räumlich-zeitlichen Kodierung voneinander unterschieden. Praktisch wird das dadurch erreicht, dass n Gray-Code-Muster zeitlich hintereinander (Bild 2, Glossar) auf die Szene projiziert werden, mittels denen die Auswertelogik hinter der Kamera für jeden Punkt die Zugehörigkeit zu einem einzelnen Streifen detektieren kann (n-stelliger Code). Somit sind nur noch n Projektionen und Aufnahmen nötig, um 2^n Streifen zu unterscheiden. Dieses Verfahren ist in der Literatur als «Methode mit strukturiertem Licht» oder als «Codierter Lichtansatz» [2] bekannt.

Geometrie

Grundsätzlich kann die geometrische Anordnung von Kamera und Projektor beliebig sein, solange der Winkel zwischen Projektions- und Aufnahmerichtung grösser Null ist. In diesem allgemeinsten Fall ist aber die Höhe eine Funktion von projiziertem Code und Kamera-Koordinaten in x- und y-Richtung. Unter der geometrischen Randbedingung, dass die y-Achsen des Kamera- und Projektorkoordinatensystems parallel zueinander sind, wird die Höhe unabhängig von der y-Koordinate der Kamera. Somit kann die Höhe als

$$h = f(C_p, x_K, P_g) \qquad (2)$$

mit C_p für den projizierten Code (Zugehörigkeit zu einem bestimmten Streifen), x_B für Bildkoordinate und P_g für einen geometrischen Parameter geschrieben werden, womit sich nun diese Funktion durch eine zweidimensionale Look-Up-Table implementieren lässt. Die geometrischen Parameter müssen durch eine vorgängige Kalibrierung bestimmt werden.

In unserer Applikation war die Kamera senkrecht über der Szene montiert und der Projektor beleuchtete die Szene unter einem Winkel von 45°. Der Projektionswinkel beeinflusst sowohl die Auflösung der Höhe als auch die Grösse des von den einzelnen Objekten geworfenen Schattens. Je kleiner dieser Winkel gewählt wird, desto grösser ist die Auflösung, desto grösser wird aber auch der Schatten. In dieser Applikation wurde ein Projektionswinkel von 45° gewählt, wodurch eine gleiche Auflösung in der Höhe wie in x- und y-Richtung erreicht wurde.

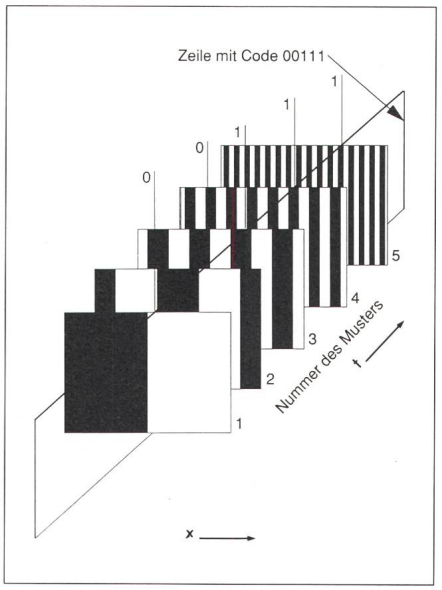

Bild 2 Beispiel von 5 Gray-Code-Mustern

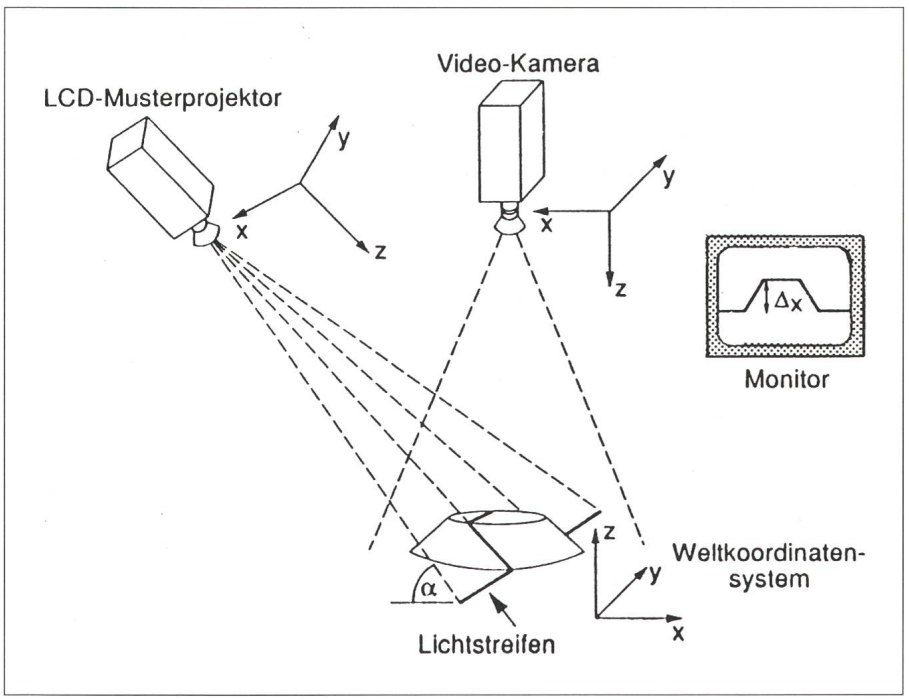

Bild 1 Grundprinzip des Tiefensensors

Tiefenbildaufnahme

Die Aufnahme eines Tiefenbildes besteht aus folgenden Schritten: Zuerst wird die Szene in unbeleuchtetem und beleuchtetem Zustand aufgenommen. Mit diesen beiden Bildern wird einerseits bestimmt, wo Schatten ist und somit keine Tiefeninformation vorhanden sein kann (Schattenmaske) und andererseits wird daraus das Schwellwertbild nach der Formel

$$S = \sqrt[\frac{1}{\gamma}]{\left(I_D^{\frac{1}{\gamma}} + I_H^{\frac{1}{\gamma}} \right)} \qquad (3)$$

berechnet, wobei γ das Gamma der Kamera (Signal in Funktion der Lichtintensität) und I_H bzw. I_D die Intensität des Hell- bzw. Dunkelbildes ist. Anschliessend werden die neun Gray-Code-Bilder projiziert, aufgenommen und pixelweise mit dem zu Beginn berechneten Schwellwertbild verglichen. Dank dem pixelweise erfassten Schwellwert wird der Vergleich unabhängig vom Reflexionsgrad der Objekte. Das binäre Resultat dieses Vergleichs ist pro empfangenes Gray-Code-Bild und pro Pixel ein Bit des Zeilen-Codes, so dass man nach Empfang und Auswertung aller neun Bilder den Zeilen-Code jedes Pixels kennt. Die einzelnen Bits werden in eine von neun Bitebenen eines bitweise beschreibbaren Speichers geschrieben (Bild 3). Zuletzt wird dann aus dem Gray-Code-Bild und der x-Koordinate mit Hilfe einer Look-Up-Tabelle (LUT) das Tiefenbild berechnet.

Projektor

Ein Tiefensensor, der in gemeinsamer Umgebung mit Menschen arbeitet, muss bei normalem Arbeitslicht funktionieren. Das heisst, dass das Projektionslicht stärker als das normale Arbeitslicht sein muss. Dazu gibt es folgende Möglichkeiten: Wahl einer sehr starken Projektionslampe oder Synchronisierung eines Stroboskop-Blitzes mit dem elektronischen Verschluss der Kamera. Bei beiden Varianten kann die Lichtunabhängigkeit durch die Projektion von zwei zueinander inversen Mustern zusätzlich verbessert werden. Dadurch wird die für einen Entscheid notwendige Beleuchtungsdifferenz zur Umgebungsbeleuchtung halbiert. Wir haben uns für den Stroboskop-Blitz entschieden, da der Einbau einer extrem starken Lampe aus thermischen und räumlichen Gründen nicht möglich war und uns zudem die Stroboskop-Version eleganter schien. Es mag auf den ersten Blick vielleicht verblüffen, aber die Stroboskop-Blitze beeinflussen die Umgebung weit weniger als ein helles (flackerndes) Halogenlicht. Die Blitze sind so kurz, dass sie bei normalem Umgebungslicht vom menschlichen Auge kaum wahrgenommen werden. Die eingesetzte Blitzröhre erlaubt eine Blitzenergie von etwa 3 J bei einer Blitzdauer von ungefähr 60 μs, was einer momentanen Leistung von 500000 W entspricht. Die Verschlusszeit des elektronischen Verschlusses der Kamera (Electronic Shutter) wurde auf 1/10000 s eingestellt. Der Blitz erreicht eine äquivalente (d.h. auf 100 μs Belichtungszeit bezogene) Be-

Glossar

(Paket-)Vereinzelung: Oft werden Gegenstände, die maschinell verarbeitet werden müssen, in einem ungeordneten Haufen angeliefert. Diese müssen deshalb vor der Weiterbehandlung in eine geordnete Anordnung (z.B. auf einem Förderband) gebracht werden. Einen solchen Hilfsprozess nennt man Vereinzelung.

Computer-Vision: Dank seinem Auge kann der Mensch nicht nur sehen, sondern auch Gegenstände gezielt greifen oder eine bestimmte Situation überwachen. Die Computer-Vision will Ähnliches auch den Maschinen (Robotern) beibringen. Dazu wird die Umwelt mittels einer Kamera erfasst und die vorhandenen Gegenstände von einem Rechnerprogramm erkannt. Da Computer-Vision meist rechenintensiv ist, wurden dafür spezielle Rechner entwickelt.

Binärbild: Bild, das nur aus zwei Farben bzw. Werten besteht, z.B. 0 und 1.

Segmentierung: Unterteilung von Bildern in bedeutungsvolle Teilbereiche, die entweder ein einheitliches Kriterium aufweisen oder durch Kanten voneinander getrennt sind. Eine Segmentierung kann, falls das Kriterium binär ist, in Transitionscodierung und Connected Component Labelling aufgespalten werden.

Transitionscode: Resultat eines Datenreduktionsverfahrens (Transitionscodierung) für binäre Bilder, bei der nur noch die Koordinaten der Übergänge von 0 auf 1 und von 1 auf 0 abgespeichert werden. Ein solcher Transitionscoder lässt sich relativ leicht in Hardware implementieren.

Connected Component Labelling: Zusammenfassung von Punkten gleicher Eigenschaften zu zusammenhängenden Gebieten. In unserem Fall werden nach der Transitionscodierung die einzelnen Zeilensegmente zu zusammenhängenden Gebieten verschmolzen.

Gray-Code: Ein Gray-Code ist ein einschrittiger Code, d.h. aufeinanderfolgende Zahlen unterscheiden sich jeweils nur in einer Binärstelle. Er ist vom Bedeutung bei der Ablesung von mehrspurigen codierten Massstäben, da er eindeutige Übergänge liefert. Auch bei der Projektion von solchen Codes werden Fehler durch nichteindeutige Übergänge vermieden.

Kirsch-Compass-Filter: Mit dem Kirsch-Compass-Filter lässt sich für jeden Bildpunkt dessen Gradient (Richtung und Steigung) berechnen. Dazu werden 8 verschiedene Filter-Masken der Form

$$P_1 = \begin{vmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{vmatrix} \qquad P_2 = \begin{vmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{vmatrix} \qquad (6)$$

benützt, die jeweils die Steigung in einer von 8 Richtungen berechnen. Die Maske P1 spricht dabei auf östliche Richtung, die Maske P2 auf nordöstliche Richtung an. Bei der Filterung werden die Ergebnisse aller Masken berechnet, und die Maske mit dem maximalen Ergebnis (Steigung) gibt die Richtung des Gradienten in diesem Bildpunkt an.

Median-Filter: Ein Median-Filter ist ein nichtlineares Filter, welches die Bildpunkte in der Umgebung des zu filternden Punktes ihrer Grösse nach sortiert und als Resultat den mittleren Wert (Median-Wert) liefert.

FIR-Filter: Ein Finite Impulse Response-Filter ist ein lineares, nicht rückgekoppeltes Filternetzwerk, mit dem sich verschiedene Filter (z.B. Hochpass, Tiefpass, Kantenextraktoren usw.) realisieren lassen. Das Netzwerk berechnet für jeden Bildpunkt ein gewichtetes Mittel der umliegenden Bildpunkte, wobei die Gewichte in einer Filter-Matrix bzw. Maske zusammengefasst sind und die Funktion des Filters bestimmen.

Yaw, Pitch, Roll: Speziell in der Robotertechnik gebrauchte Winkel, welche die Position des Roboter-Greifers bezüglich eines Koordinatensystems bestimmen. Dieses hat seinen Ursprung im Befestigungspunkt des Roboter-Greifers (Tool Center Point).

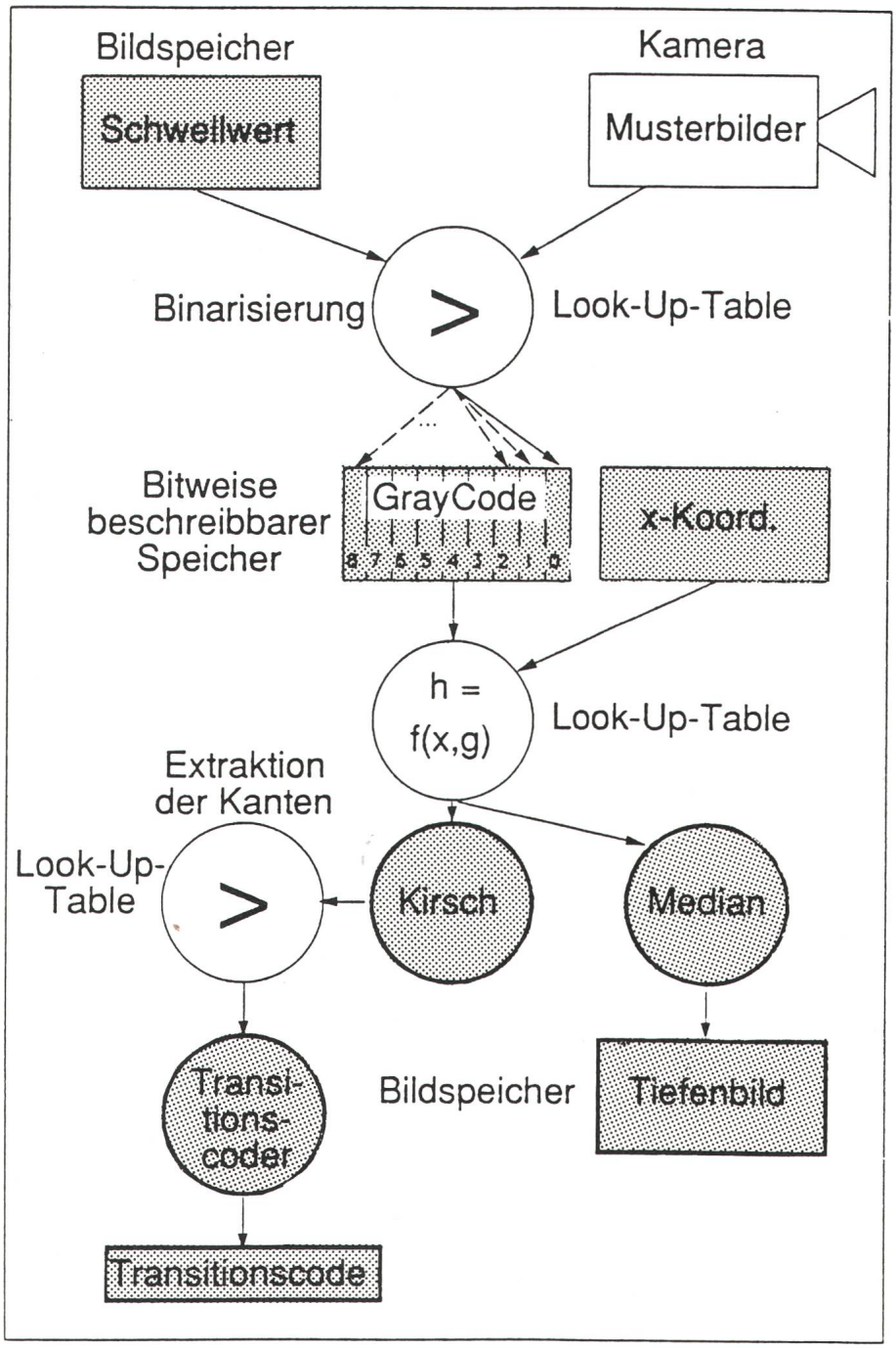

Bild 3 Vereinfachtes Datenflussdiagramm für die Tiefenbildaufnahme mit Kirschfilter und Transitionscoder
Durch pixelweisen Vergleich der 9 Musterbilder mit dem Schwellwertbild wird der vollständige 9-Bit-Gray-Code erhalten. Aus Gray-Code und *x*-Koordinate berechnet eine LUT die Höhe, welche dann median- und kirschgefiltert wird

leuchtungsdichte von 17000 lx bei einer Projektionsdistanz von etwa 1,5 m. Zum Vergleich: eine 150-W-Halogen-Glühlampe erreicht bei gleichem Abstand nur eine Beleuchtungsstärke von etwa 150 lx. Das beschriebene Verfahren erlaubt, den Einfluss des Umgebungslichtes um beinahe den Faktor 200 zu unterdrücken.

Dank der hohen Beleuchtungsstärke lieferte dieser Tiefensensor auch bei direkter Beleuchtung mit einem Scheinwerfer noch gut brauchbare Bilder, was beim Einsatz von Halogenlampen unmöglich wäre. Unter der Voraussetzung von scharfen Bildern (was nur mit sehr kleinen Blendenöffnungen erreichbar ist), kann bis etwa zu einer Umgebungslichtstärke, die der Blitzlichtstärke entspricht (unter Berücksichtigung aller im System vorhandenen Rauschquellen), noch gut binarisiert werden. Wenn der Schärfentiefebereich kleiner als der Arbeitstiefebereich ist, so wird die Binarisierung wegen den unschärferen Übergängen störungsanfälliger. Mit Umgebungsbeleuchtungsstärken von 1/4 Blitzbeleuchtungsstärke wurden so noch gute Ergebnisse erreicht. Diese Überlegungen gelten bei konstantem Umgebungslicht. Gerade in einer Umgebung, in der Roboter arbeiten, treten sich bewegende Schatten auf, welche spielend Beleuchtungsunterschiede von 50% und mehr ergeben. Verändert ein Schatten seine Position zwischen der Aufnahme des Schwellwertbildes und der Aufnahme der einzelnen Muster, so führt das zu Fehlern, falls das Umgebungslicht gegenüber dem Projektionslicht nicht genügend stark unterdrückt wird.

Als Projektor wird der LCD-Linienprojektor LCD-320 der Firma ABW, Neuhausen a.d.F. eingesetzt, wobei dieser so modifiziert wurde, dass die Halogenlampe durch eine Stroboskop-Blitzlampe ersetzt werden kann. Zur Synchronisierung wurde das VSync-Signal der Kamera in geeigneter Weise verwendet. Dank dem schnellen Bildverarbeitungsrechner konnten die Bilder mit der Halbbild-Frequenz aufgenommen werden. So benötigte dieser Sensor für die Aufnahme eines Tiefenbildes inklusive Median-Filterung nur 240 ms. Der schwerwiegendste Nachteil eines Stroboskop-Blitzes ist seine schwankende Lichtabgabe (im Mittel etwa 5%). Dies wirkt sich vor allem bei relativ geringer Blitzlichtintensität auf die Genauigkeit aus. Trotzdem lässt sich sagen, dass ein Stroboskop-Projektor für Roboter-Anwendungen bei Arbeitslicht einer Halogenlampe vorzuziehen ist.

Beim hier vorgestellten Tiefensensor sind die auftretenden Störungen vor allem punktförmiger Natur. Die Ursache dazu ist in der schwankenden Lichtintensität und im Rauschen von Kamera und A/D-Wandler einerseits und in unscharfen Mustern und Schatten andererseits zu suchen. Treten die Fehler bei den höherwertigen Bits oder bei der Schattenmaske auf, so resultiert daraus ein sehr grosser Fehler. Solche punktförmige grosse Fehler lassen sich mit einem FIR-Tiefpass-Filter (Glossar) kaum korrigieren, da dies den Fehler nur gleichmässig auf die umliegenden Pixel verteilt. Mit einem Median-Filter hingegen kann man sie grösstenteils zum Verschwinden bringen.

Bildverarbeitungsrechner Sydama

Die Tiefenbildaufnahme und vor allem die Filteroperationen benötigen eine enorme Anzahl einfacher Rechenschritte. Mit konventionellen Rechnern der ähnlichen Preisklasse würde die Bildauswertung viel zu lange dauern. Daher wurde der am Institut für Elektronik der ETH Zürich entwickelte schnelle Bildverarbeitungsrechner Sydama II (Synchronous Dataflow Machine) [3; 4; 5] eingesetzt. Dieser nach dem Datenflussprinzip gebaute Rechner erlaubt, Bilder mit einer Fernseh-Bildtaktrate zu verarbeiten. Beim Datenflussprinzip fliessen die Daten im Pixeltakt von einem Prozessor zum anderen. Der Durchsatz jedes Prozessors muss also gleich der Videodatenrate sein. Ein Algorithmus muss dazu in Einzelfunktionen (Funktionen zweier Variablen, Filtrierungen) aufgespalten werden. Jede solche Funktion wird dann einem Prozessorelement (Look-Up-Table, Transitionscoder [Glossar] oder Spezialprozessor wie FIR-, Median-, Kirsch-Filter) zugewiesen. Somit durchläuft jeder Bildpunkt pipelineartig die einzelnen Prozessorelemente (PE). Die in LCAs (Logic Cell Arrays, Xilinx) implementierten Prozessorelemente sind über einen schnellen Videobus miteinander verbunden und werden von Transputern (T805) überwacht. Einige Parameter und die Verschaltung des Videobusses können für jedes Halbbild neu gesetzt werden. Ein grosser Vorteil des Sydama ist, dass die Transputer, die für allgemeine Berechnungen verwendet werden können, direkt auf den Video-Speicher zugreifen können. Dadurch entfällt der zeitintensive Transfer von Bildern völlig und die gesamte Bildverarbeitung vom Kamerasignal bis zu den Steuerdaten für den Roboter kann in einem einzigen Rechner realisiert werden.

Der in unserem Projekt implementierte Algorithmus benötigt eine maximale Videobus-Datenrate von 36 MByte/s, was nur einen Bruchteil der bei Sydama II möglichen Datenrate darstellt. Bei einem konventionellen Rechner müssten noch ein Mehrfaches davon an Instruktions-, Zähler- und Variablendaten übertragen werden. Dies entfällt beim Sydama, da alle Adressberechnungen hardwaremässig realisiert sind und nur Bilddaten übertragen werden müssen. Zudem ist man frei, je nach Geschwindigkeitsanforderung, gewisse Algorithmen in Hardware und andere in Software zu implementieren.

In der hier vorgestellten Applikation wurden zwei Karten mit je drei Prozessorelementen und einem Transputer eingesetzt. Dazu wurden noch ein Video-I/O-Modul, ein Kirsch-Compass- und ein Median-Filter (Glossar) verwendet.

Bildinterpretation

Die Aufgabe der Bildinterpretation ist, die einzelnen Pakete zu erkennen. Da es sich bei den Paketen ausschliesslich um quaderförmige Objekte handelt, kann diese Aufgabe darauf reduziert werden, rechteckige Flächen im dreidimensionalen Bild zu suchen, wobei von jedem Paket eine bis maximal drei Flächen sichtbar sind, welche jeweils senkrecht zueinander stehen. Paketflächen sind immer durch Kanten von benachbarten Paketflächen getrennt und weisen eine konstante Steigung und Richtung auf. Es sind dabei zwei Typen von Kanten zu unterscheiden, solche, bei denen die Höhenkoordinate der Flächenfunktion sprunghaft ändert (Bild 4a, Stufenkanten) und solche, bei denen nur der Gradient der Flächenfunktion sprunghaft ändert (Bild 5a, Firstkante). Um diese beiden Arten von Kanten zu detektieren, mussten geeignete Algorithmen gefunden werden. Stufenkanten können grundsätzlich mit Hilfe der ersten Ableitung, beide Arten von Kanten mit Hilfe der zweiten Ableitung detektiert werden. Ableitungen, besonders solche höherer Ordnung, sind aber anfällig auf Störungen. Daher wurde das im folgenden beschriebene, gegenüber verrauschten Bildern robustere Verfahren angewendet. Es basiert auf der Detektion von Steigungen (1. Ableitung) und Richtungen. Das von uns verwendete Kirsch-Compass-Filter liefert für jeden Bildpunkt die Steigung und die Richtung mit einer Auflösung von 45°, wobei die Werte in einer 3×3-Umgebung (betrachteter Bildpunkt mit 8 benachbarten Punkten) berechnet werden. Da in natürlich entstandenen Pakethaufen die meisten Kanten Stufenkanten sind, wurde ein zweistufiges Verfahren an-

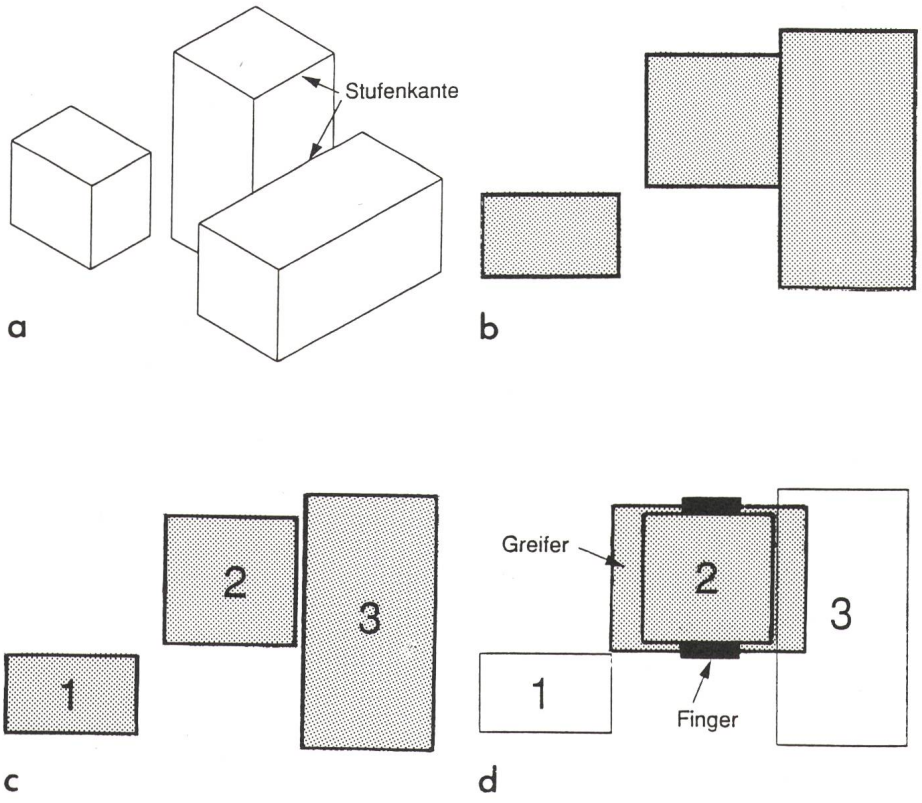

Bild 4 Bildinterpretation: Beispielszene 1
a Ansicht
b Bild mit Kanten (dicke Linien) und Flächen mit Mindesthöhe
c Binärbild der Paketflächen: Alle drei Flächen stellen genau eine Paketfläche dar
d Szene mit eingezeichnetem Greifer und Finger: Das höchste Paket wird zuerst gegriffen

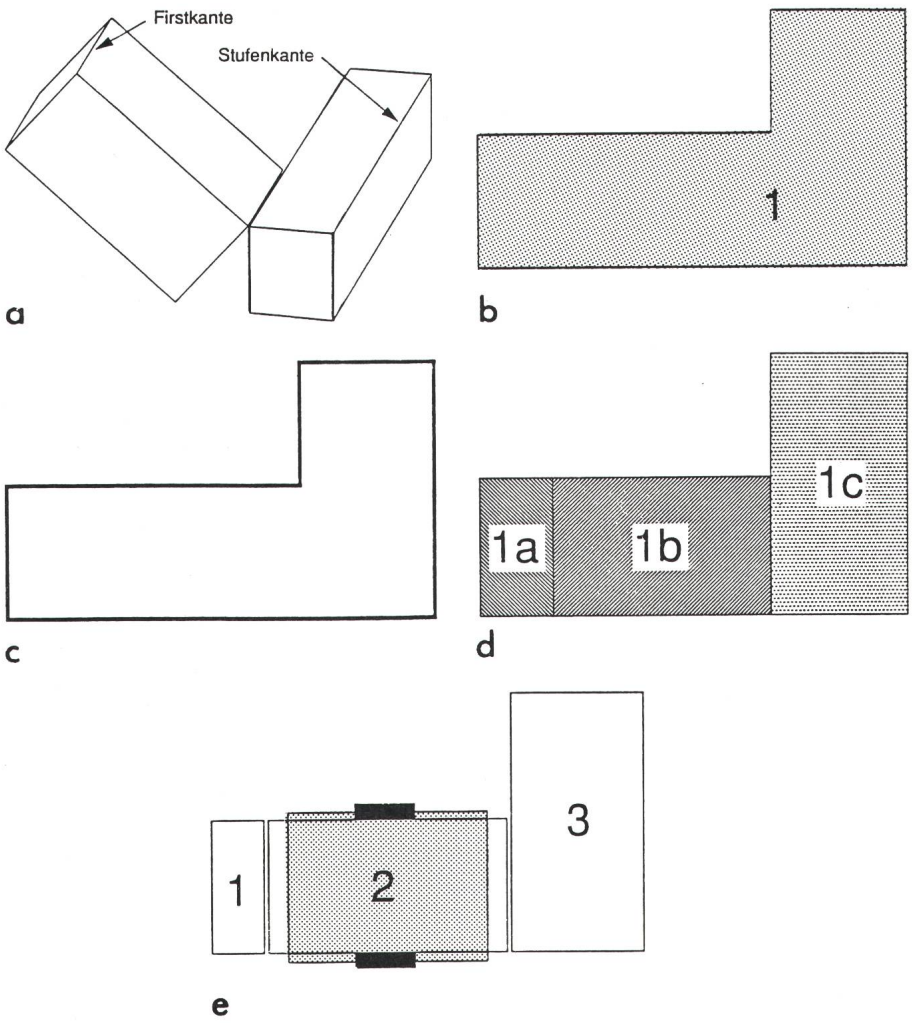

Bild 5 Bildinterpretation: Beispielszene 2

a Ansicht

b Bild mit Kanten (dicke Linien) und Flächen mit Mindesthöhe: Die beiden Paketflächen verschmelzen zu einer einzigen Fläche, da sie nicht durch Stufenkanten getrennt sind

c Binärbild der Paketflächen

d Nach der Segmentierung nach Richtungen konnte die Fläche in drei Teilflächen aufgespalten werden

e Szene mit eingezeichnetem Greifer mit Finger. Das höchste Paket wird zuerst gegriffen

gewendet, welches Stufenkanten mit höherer Priorität behandelt.

Im ersten Schritt werden Flächen, die von Stufenkanten umgeben sind, extrahiert (Bild 4a, Segmentierung aufgrund von Kanten). Stufenkanten sind dadurch charakterisiert, dass die Ebenenfunktion in einer Richtung eine sehr grosse Steigung aufweist; somit können sie durch einen Schwellwert, der auf die Steigung angewendet wird, detektiert werden. Um auch dünne Kanten besser erkennen zu können, wird die Steigung auf dem ungefilterten Bild berechnet. Für alle anderen Berechnungen wird aber das mediangefilterte Bild verwendet. Nun wird ein Binärbild (Glossar) hergestellt, das aus allen Pixeln besteht, die nicht zu einer Kante gehören und höher als eine gewisse Mindesthöhe sind, also zu Paketen gehören (vgl. Flächen in Bild 4b und 5b). Sowohl das Binärbild als auch der daraus erzeugte Transitionscode (Glossar) werden in einem Spezial-Hardwareprozessor berechnet. Im anschliessenden sogenannten Connected Component Labelling (Glossar) werden aus dem Transitionscode zusammenhängende Gebiete gesucht und gleichzeitig mehrere Momente dieser Gebiete und andere später benötigte Daten berechnet. Solche Gebiete können nun aus genau einer Fläche (Bild 4c, Flächen 1, 2, 3) oder aber auch aus mehreren zusammenhängenden Flächen bestehen (Bild 5c, die Fläche 1 besteht aus 3 Teilflächen). Um nun entscheiden zu können, ob ein Gebiet aus einer einzigen Fläche besteht, wird für jede gefundene Komponente versucht, mit einer Least Mean Square Error-Methode die Parameter der Flächengleichung der Form

$$z = A \cdot x + B \cdot y + C \qquad (4)$$

zu berechnen. Gleichzeitig wird die Varianz der Flächenschätzung berechnet. Unterschreitet die Varianz eine gewisse Schwelle, so stellt die Komponente eine einzige Paketfläche dar und wird in eine Tabelle aller Paketflächen eingetragen. Im anderen Fall besteht die Komponente aus mehreren Teilflächen (Bild 5c), die nicht durch Stufenkanten getrennt sind; sie muss noch weiteren Verarbeitungsschritten unterworfen werden. Da – wie schon erwähnt – Ableitungen höherer Ordnung störungsanfällig sind, werden für diese Segmentierung (Glossar) nicht Kanten, sondern einheitliche Eigenschaften der zu segmentierenden Flächen herangezogen. Als Kriterium wird die Richtung verwendet. Jeder Bildpunkt kann entweder als flach (kleine Steigung) taxiert oder einer von 8 Richtungen zugeordnet werden. Um zu entscheiden, nach welchen Richtungen segmentiert werden soll, wird zuerst von jeder Komponente ein Richtungs-Histogramm berechnet. Da nicht alle Flächengradienten genau in einer Richtung des Kirsch-Kompass-Filters liegen, ist es sinnvoller nach Richtungspaaren zu segmentieren. Solche zusammengehörenden Richtungspaare (jeweils α, $\alpha+45°$), welche eine bestimmte Häufigkeit nicht unterschreiten, werden aus dem Histogramm ausgesucht. In der folgenden Segmentierung werden nun Pixel, die jeweils eine der beiden Richtungen aufweisen, zu einer Unterkomponente zusammengefasst (vgl. in Bild 5d die Teilflächen 1a, 1b und 1c, entstanden aus der Komponente 1). Auch für jede Teilfläche werden wieder deren Parameter und die Varianz berechnet. Ist die Varianz genügend klein, so werden auch diese Teilflächen in die Tabelle der Paketflächen eingetragen. Flächen, die keine Paketflächen sind, werden ausser bei der Kollisionsdetektion nicht mehr weiter betrachtet.

Greifbarkeitstests

Ausgehend von der Tabelle der Flächen muss nun eine Paketfläche aus-

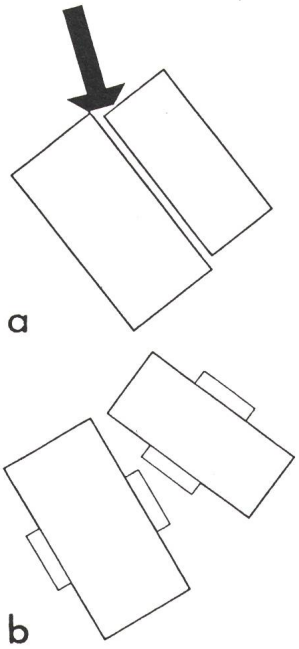

Bild 6 Greifbarkeit
a In dieser Situation kann der Roboter nicht greifen, da die Greiffinger neben den Paketen keinen Platz haben. Der Roboter verschiebt nun die Pakete in Richtung des Pfeils, um eine vereinzelbare Anordnung zu erreichen
b Nachdem die Szene verändert wurde, sind beide Pakete greifbar

gewählt werden, welche greifbar ist. Wie schon erwähnt, werden die Flächen ihrer Höhe nach einem Greiftest unterworfen. Die Pakete sollen immer parallel zur längeren Seite, auf der Höhe des Flächenschwerpunktes gegriffen werden. Sofern das Paket von seinen Dimensionen her überhaupt greifbar ist, wird nun überprüft, ob einerseits die Greiffinger genügend Platz neben dem Paket haben und andererseits der Greifer selbst nicht an andere Pakete stösst (Paket 2 in Bild 4d kann gegriffen werden, Paket 3 aber nicht). Sofern immer das höchste Paket zuerst gegriffen würde, wäre der zweite Test eigentlich nicht nötig. Es hat sich aber gezeigt, dass es immer wieder vorkommt, dass das höchste Paket nicht greifbar ist, weil es aus dem Bild ragt oder in einer sehr ungünstigen Position liegt. Dank diesem Test können viele Kollisionen mit solchen Paketen verhindert werden.

Sobald eine Fläche greifbar ist, werden die dazugehörigen Greifpunkte berechnet und in die Greiferkoordinaten (*X*, *Y*, *Z*, *yaw*, *pitch*, *roll*, Greiferöffnung, vgl. Glossar) umgerechnet, welche dem Roboter übermittelt werden. Falls keine Fläche greifbar ist, weil zum Beispiel die Pakete eng nebeneinanderliegen und keinen Platz für die Finger lassen (Bild 6a), wird versucht, durch Verschieben der Pakete die Szene so zu verändern, dass die Pakete wieder greifbar werden. Die Wahl des Verschiebungsvektors wird nach heuristischen Regeln vorgenommen. Dazu werden die nicht greifbaren Pakete verschiedenen Gruppen zugeordnet (z.B. parallel zueinander liegende Pakete, Pakete mit undefinierbaren Hindernissen, nur halb im Bildfeld liegende Pakete usw.) und für jede Gruppe besteht eine Regel, wie aus den Paketdaten der Verschiebungsvektor berechnet wird. Auch beim Verschieben wird wieder überprüft, ob der Greifer mit anderen Paketen kollidieren würde. Dank dieser Verschiebungsstrategie konnte ein Pakethaufen nur äusserst selten nicht vollständig abgeräumt werden.

Die Robustheit des Systems wird dadurch erhöht, dass der Roboter in seinen Fingern Infrarot-Abstandssensoren besitzt, welche Kollisionen verhindern und dem Roboter beim sicheren Greifen helfen sollen. Der Roboter besitzt dazu sogenannte Reflexe: Ein Fluchtreflex, der bewirkt, dass der Roboter die Szene verlässt, wenn er unerwartet auf ein Hindernis stösst, ein Greifreflex, der ihn dazu veranlasst, die Bewegung zu stoppen und den Greifer zu schliessen, wenn er im Zustand des langsamen Zufahrens auf ein Paket auf ein Hindernis stösst. Dank diesen Reflexen können Kollisionen vermieden werden, und lässt sich ein Paket auch dann sicher greifen, wenn es am Schluss des Annäherungsvorgangs zu einer Kollision kommen würde.

Kommunikation zwischen den Teilsystemen

Das gesamte System besteht aus 4 Teilsystemen: Kamera, Muster-Projektor, Sydama II und Roboter. Das System wurde so ausgelegt, dass ein

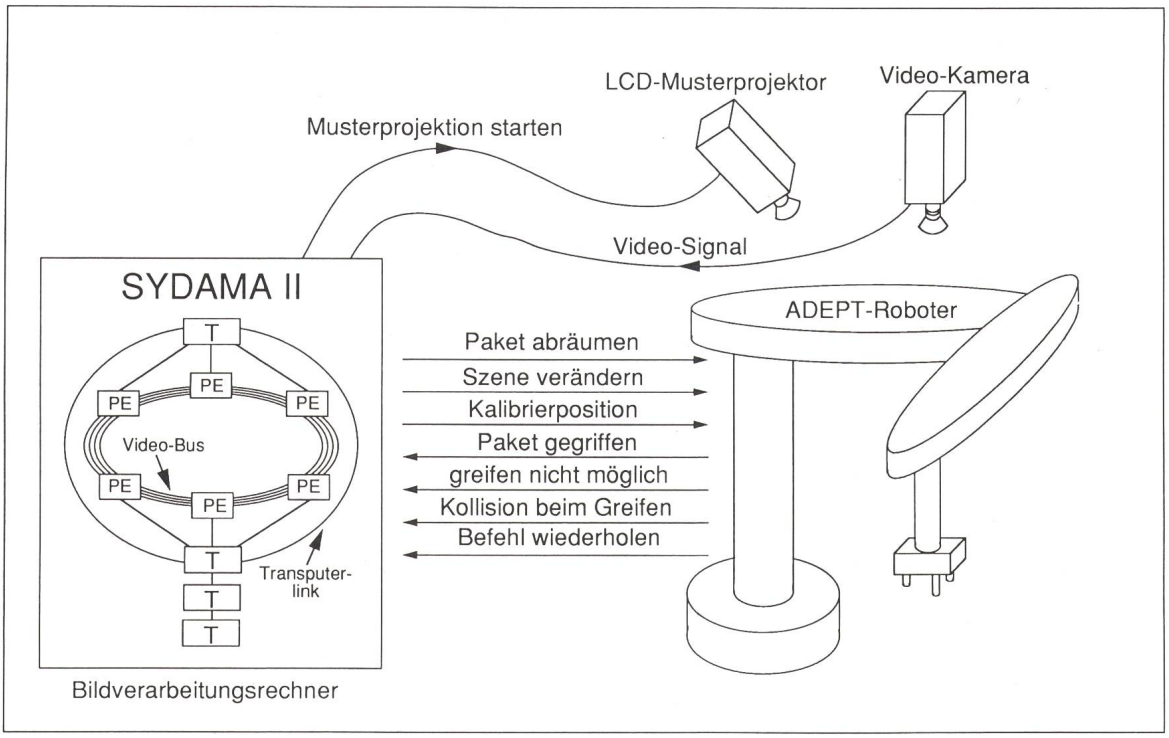

Bild 7 Gesamtsystem mit Kommunikation zwischen den Teilsystemen

Minimum an Daten von einem Teilsystem zum anderen transferiert werden muss. Zur Steuerung des Projektors und für die Kommunikation mit dem Roboter wurde eine serielle Schnittstelle (RS 232) eingesetzt. Der Bildverarbeitungsrechner übernimmt die Rolle des Masters. Er schickt dem Projektor die Befehle, um einzelne Muster oder ganze Mustersequenzen zu projizieren. Der Befehlssatz des Roboters enthält unter anderen die Befehle Greifen, Verschieben und Kalibrierposition anfahren. Sobald der Roboter nach einer Aktion den Sichtbereich der Kamera verlässt, erhält die Vision eine Antwort, welche unter anderem beinhaltet, ob der Befehl richtig ausgeführt werden konnte. Konnte ein Paket nicht gegriffen werden, so kann das System darauf in geeigneter Weise reagieren, indem es versucht, zuerst andere Pakete abzuräumen.

Kalibrierung des Systems

Der Vollständigkeit halber wird im folgenden noch kurz gezeigt, wie die Kalibrierung des Gesamtsystems vorgenommen wird, wobei aus Platzgründen nicht mehr auf Einzelheiten eingegangen werden kann. Die Kalibrierung des Gesamtsystems, welches aus dem Tiefensensor und dem Roboter besteht, erfolgt in zwei Schritten. Zuerst wird der Tiefensensor kalibriert, danach wird die Modelling-Matrix zur Umrechnung der Vision-Koordinaten in die Roboterkoordinaten bestimmt. Für die Kalibrierung gibt es verschiedene Lösungsansätze. Bei der Anwendung des Tiefensensors für den Paket-Roboter wurde mehr Gewicht auf eine schnelle, einfache Kalibrierung als auf höchstmögliche Genauigkeit gelegt. Zur Kalibrierung des Tiefensensors wird zuerst von einer Rampe, welche parallel zur y-Achse der Kamera steigt, ein Code-Bild aufgenommen. Anschliessend werden automatisch einige Punktetripel (x_{Bild}, y_{Bild} = Höhe, Code) extrahiert, wobei diese gewisse Plausibilitätskriterien erfüllen müssen, um Störungen und Schattenpunkte zu eliminieren. Nun können mit Hilfe einer Least-Square Fitting-Methode die benötigten Modell-Parameter bestimmt werden. Beim hier verwendeten Modell wurde sowohl für die Kamera wie für den Projektor von einer reinen zentralperspektivischen Abbildung ausgegangen und Objektiv-Verzerrungen und Verdrehungen zwischen den y-Achsen von Kamera und Projektor vernachlässigt. Nach dieser Kalibrierung können Tiefenbilder aufgenommen werden, welche noch nicht perspektivisch entzerrt sind und deren Höhe als Integerzahl zwischen 0 und 255 (entspricht der maximalen Höhe der Rampe) dargestellt wird. Die Umrechnung der homogenen Kamera-Koordinaten in die Weltkoordinaten des Roboters geschieht mit einer 4×4-Kamera-Transformations-Matrix der Form:

$$T = \left[\begin{array}{c|c} S*R & S*T \\ \hline Z & \end{array}\right] \qquad (5)$$

Diese Transformation beinhaltet die Perspektivenentzerrung Z (3×1-Vektor), die Skalierung S (3×3-Matrix), eine Translation T (Vektor) in alle drei Richtungen und eine Rotation R (3×3-Matrix) um die x-, y- und z-Achsen. Zur Bestimmung dieser Matrix fährt der Roboter verschiedene bekannte Punkte im Raum an, von welchen je ein Tiefenbild aufgenommen wird. Diese Punkte werden im Tiefenbild vermessen. Aus den so gewonnenen Daten werden dann in einem Least-Square Fitting die benötigten Parameter bestimmt.

Erfahrungen im praktischen Betrieb

Das in diesem Bericht vorgestellte Projekt konnte an der *Industrial Handling 1992* in Zürich und am *Technologiestandort Schweiz* an der Hannover Messe 1992 ausgestellt werden. An der IH '92 war der Paketroboter das erste Mal über längere Zeit in Betrieb. In der Zeit bis zur Hannover-Messe konnte das Vision-System aufgrund der Erfahrungen nochmals verbessert und seine Geschwindigkeit erhöht werden. Es wurde eine durch den Roboter begrenzte Zykluszeit von 3 s erreicht. Insbesondere an der Hannover Messe wurde der Paket-Roboter von den Zuschauern, welche äusserst komplexe Pakethaufen aufbauten, eingehend getestet. Obwohl dazu aufgefordert, gelang es kaum einem Zuschauer, einen Haufen zu bauen, den der Roboter nicht vereinzeln konnte. Auch die Robustheit des Systems konnte in Hannover eingehend demonstriert werden. Sowohl nachträglich von den Zuschauern in die bereits erfasste Szene geworfene Pakete als auch fremde Gegenstände konnten den Paketroboter nicht aus der Fassung bringen. In Hannover räumte der Roboter etwa 4000 Pakete pro Tag ab und benötigte dazu knapp 5000 Tiefenbildaufnahmen. Bedingt durch die geschilderten Bedingungen musste die Szene relativ oft verändert werden, was die Tagesleistung des Roboters natürlich verminderte.

Technische Daten

Der Arbeitsraum des Tiefensensors beträgt etwa 40×30×25 cm³, die Auflösung 368×286 Pixel mit einer Höhenauflösung von 8 Bit und ± ½ LSB (Least Significant Bit) Genauigkeit. Die benötigte Zeit für eine Tiefenbildaufnahme inklusive Median-Filterung beträgt 240 ms. Für zwei Kirsch-Compass-Filterungen und die Transitionscodierung wurden nochmals 40 ms benötigt. Die Rechenzeit der übergeordneten Algorithmen auf dem Transputer ist sehr abhängig von den Bilddaten und der Komplexität der Szene. Durchschnittlich beträgt die Rechenzeit für die gesamte Bildverarbeitung etwa 1 s, wobei sie von 0,6 s bis knapp 2 s bei sehr komplexen Szenen mit vielen Teilflächen variiert. Die Rechenzeit der Software-Algorithmen könnte nochmals massiv gesenkt werden, wenn statt eines Transputers mehrere Transputer oder ein Signalprozessor eingesetzt würde.

Dieses Projekt wurde im Rahmen des Projekts «Cooporating Robot» der Arbeitsgruppe Mechatronik an der ETH Zürich realisiert. Dabei waren das Institut für Elektronik (Tiefensensor, Bildverarbeitung) und das Institut für Robotik (Studie Paketvereinzelung, Roboter) beteiligt.

Literaturverzeichnis:

[1] *Baerveldt A.-J., Schweitzer G.:* Machbarkeitsstudie «Vereinzelung von Paketen mit einem Roboter», Jan. 1991.

[2] *Wahl F.M.:* A Coded Light Approach for Depth Map Acquisition. In: G. Hartmann (ed.): Mustererkennung 1986, Springer.

[3] *Gunzinger A., Guggenbühl W., Hiltebrand W., Mathis S., Schaeren P., Schneuwly B., Stokar D., Zeltner M.:* Sydama II: A Fast Computer for Machine Vision, Proc. of ISPRS Symposium: Close Range Photogrammetry Meets Machine Vision, SPIE Vol. 1395, Zürich, Sept. 3–7, 1990.

[4] *Gunzinger A.:* Der Echtzeitbildverarbeitungsrechner «Sydama II», Proc. Ident. Vision 1991, Stuttgart, 14.–17. Mai 1991.

[5] *Gunzinger A.:* «Massgeschneiderte Echtzeitbildverarbeitung – Konzept und Realisierung eines heterogenen Mehrprozessorsystems». Bull. SEV/VSE 83(1991)21.