

Arnd Müller

**Ein Billardroboter -
praktische Realisierung von ausgewählten Konzepten der Robotik**

Diplomarbeit

**eingereicht am 30. April 1998
an der**

**Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik**

Betreuer: PD Dr. habil. A. Schierwagen

Simulationen zählen nicht.

(erster von "10 Gründen, Roboter zu bauen",
aus einem Katalog der Firma Mondotronics, USA)

Zusammenfassung

Die Arbeit beschreibt Konzipierung, Bau und Untersuchung eines Robotersystems, welches die grundlegenden Züge des Billardspiels beherrscht. Die Realisierung erfolgte in Form eines an die Charakteristika von mobilen Robotern angelehnten Fahrzeuges, welches sich auf einem Billardtisch bewegt und mit einer Art Queue Billardstöße ausführt. Alle Abläufe werden mit Hilfe einer über dem Tisch montierten Digitalkamera erfaßt und überwacht. Die Steuerung erfolgt durch einen Personalcomputer.

Schwerpunkte der Arbeit bilden Probleme der Bilderkennung, Wegplanung und technischen Umsetzung.

Die Lösung wurde mit Hilfe relativ einfacher technischer Mittel und geringem finanziellen Budget erstellt.

Inhaltsverzeichnis

Zusammenfassung	3
Inhaltsverzeichnis	4
Übersicht	7
1 Einführung und Motivation	8
1.1 Kurzvorstellung des Projektes	8
1.1.1 Der Stoßzyklus	8
1.2 Erläuterungen zur Wahl des Arbeitsthemas	9
1.3 Die Robotik und ihre Teilgebiete	11
1.3.1 Mobile Roboter	11
1.3.2 Fortbewegung und Wegplanung	12
1.3.3 Sensorik und Bilderkennung	13
1.3.4 Planung und Simulation	13
1.3.5 Intelligentes Verhalten und Lernen	14
1.4 Besonderheiten der praktischen Umsetzung von Robotik-Themen	14
1.4.1 Bauen vs. Simulieren	16
1.4.2 Auftretende Probleme	17
1.5 Einordnung der Arbeit und Bezüge zu anderen Projekten	18
1.5.1 Billardsystem und mobile Roboter - ein prinzipieller Vergleich	18
1.5.2 Forschungsarbeiten und Industrielle Anwendungen	19
1.6 Spielregeln des Poolbillard	20
1.7 Unterschiede zwischen realem Spiel und Billard-Roboter	21
2 Bilderkennung	23
2.1 Die Kamera - allgemeine und spezielle Probleme der Bildaufnahme	24
2.1.1 Technisch bedingte Aufnahmefehler	25
2.1.2 Die Connectix-ColorQuickCam	26
2.2 Anforderungen an die Bilderkennung	27
2.3 Erfassung der Kugelpositionen (Groberkennung)	28
2.3.1 Eliminieren der "Christmas-Lights"	29
2.3.2 Shading-Korrektur	30
2.3.3 Binarisieren	32
2.3.4 Erosion der Kugelgebiete	34
2.3.5 Ermittlung der groben Mittelpunktkoordinaten der Kugeln	35

2.4 Bestimmung der Kugelfarben	36
2.5 Genaue Positionsbestimmung von Kugeln (Feinerkennung)	37
2.5.1 Aufnahme des Bildausschnittes	38
2.5.2 Kantendetektion mit Sobel-Kompaß-Operator	38
2.5.3 Einsatz des "Kreisfilters" zur Mittelpunktfindung	39
2.5.4 Korrektur des perspektivischen Fehlers	40
2.6 Optische Erkennung des Roboterfahrzeuges	41
2.6.1 Schnellere Variante	42
2.6.2 Genauere Variante	44
3 Stoßwahl	46
3.1 Physik und Geometrie des Billardstoßes	46
3.2 Restriktionen und Parameter der Stoßwahl	48
3.3 Implementierung des Algorithmus	50
4 Wegplanung	53
4.1 Grundlegende Überlegungen	53
4.1.1 Gemeinsamkeiten von Wegplanungs-Szenarien	53
4.1.2 Varianten der Bewertung von Wegen	54
4.2 Anforderungen an die Wegplanung	55
4.3 Bestimmung der potentiellen Wege	55
4.3.1 Modellierung von Hindernissen und Fahrzeug	55
4.3.2 Voronoi-Parkettierung	56
4.3.3 Bestimmung der sicheren Teilwege	58
4.4 Auswahl des optimalen Weges	58
4.5 Details der praktischen Anwendung des Algorithmus	60
5 Kamera-Fahrzeug-Koordination	61
5.1 Fahrt zur Stoßposition	61
5.1.1 Vereinfachung der Fahrtroute	61
5.1.2 Drehen des Fahrzeuges	61
5.1.3 Geradeaus Fahren	63
5.2 Ausrichtung des Fahrzeuges zum Stoß	64
6 Technische Details	65
6.1 Der Billardtisch	65

6.2 Die Kamera-Mechanik	66
6.2.1 Prinzipielle Überlegungen	66
6.2.2 Realisierung	67
6.3 Das Roboterfahrzeug	69
6.3.1 Der Stößel	69
6.3.2 Grundlegende Anforderungen an das Roboterfahrzeug	71
6.3.3 Wahl der Antriebsart	71
6.3.4 Technische Umsetzung	73
6.4 Die Verbindung zum Steuercomputer	74
7 Analyse der Leistungsfähigkeit des Billardsystems	76
7.1 Punkte 1 bis 4 des Stoßzyklus	76
7.1.1 Bilderkennung	76
7.1.2 Stoßwahl	77
7.1.3 Wegplanung	77
7.1.4 Fahrt	77
7.2 Treffergenauigkeit	78
7.2.1 "Benchmark"-Test	78
7.2.2 Fehlerquellen bei der Fahrzeugausrichtung	80
8 Ausblick	81
Literaturverzeichnis	82
Verzeichnis von Internet-Links	84
Anhang Farbbilder	
Anhang Kostenübersicht	

Übersicht

Das Kapitel "Einführung und Motivation" stellt das Gesamtprojekt vor, erläutert den für die Arbeit grundlegenden Begriff des Stoßzyklus, gibt einen Überblick über das Forschungsgebiet der mobilen Roboter und versucht, die Arbeit innerhalb dieses Gebietes einzuordnen. Desweiteren wird das Spiel Poolbillard erläutert und die Realitätsnähe des Billardroboters untersucht.

Im Kapitel "Bildererkennung" werden zunächst allgemeine Probleme der computergestützten Bildererkennung dargelegt. Im Anschluß werden die im Billardroboter-System eingesetzten Algorithmen zur Erkennung der Kugeln und des Fahrzeuges ausführlich beschrieben.

Das Kapitel "Stoßwahl" beschäftigt sich mit physikalischen Grundlagen des Billardstoßes und erläutert den eingesetzten Algorithmus, welcher aus einer gegebenen Spielkonstellation einen erfolgversprechenden Stoß auswählt.

Das Kapitel "Wegplanung" erläutert die in der Robotik weitverbreitete Aufgabe der Planung von Wegen und beschreibt den für den Billardroboter entwickelten Algorithmus.

Das Kapitel "Kamera-Fahrzeug-Koordination" geht spezifisch auf die Aktionen des Fahrzeuges ein, wenn dieses sich auf dem Billardtisch fortbewegt bzw. zu einem Stoß ausgerichtet wird.

Das Kapitel "Technische Details" gibt eine zusammenfassende Übersicht über alle im System integrierten technischen Komponenten und Problemlösungen.

Im Kapitel "Analyse der Leistungsfähigkeit des Billardsystems" wird dargestellt und untersucht, wie zuverlässig die einzelnen Teilalgorithmen arbeiten. Den Schwerpunkt bildet dabei die Treffergenauigkeit des Roboters beim eigentlichen Billardstoß.

Den Abschluß der Arbeit bildet das Kapitel "Ausblick".

Abbildungen, welche mit "F" gekennzeichnet sind, befinden sich im Farbbild-Anhang. Quellenangaben der Form [WWW...] bezeichnen Internet-Seiten und sind im Anschluß an das Literaturverzeichnis aufgeführt.

1 Einführung und Motivation

1.1 Kurzvorstellung des Projektes

Das Ziel der Arbeit besteht in der praktischen Realisierung eines Robotersystems, welches in der Lage ist, die grundlegenden Züge des menschlichen Billardspieles nachzuvollziehen. Dabei kommen Verfahren und Algorithmen zum Einsatz, die in verschiedenen Gebieten der Robotik angesiedelt sind und zur Erfüllung der gestellten Aufgabe auf sinnvolle und teilweise neuartige Weise miteinander kombiniert werden.

Die konkrete Umsetzung besteht in einem kleinen Roboterfahrzeug (Abb. F1.1), welches mit einer Vorrichtung zum Anstoßen der Billardkugeln, im folgenden als Stößel bezeichnet, ausgerüstet ist. Dieses Fahrzeug bewegt sich, über ein Kabel ferngesteuert, auf einem miniaturisierten Billardtisch (ca. 100×50cm). Als einziger Sensor im gesamten System fungiert eine digitale Kamera, die über der Spielfläche beweglich montiert ist und dem steuernden Personalcomputer eine Erfassung der Vorgänge mittels Bilderkennungsalgorithmen ermöglicht. Als Verbindung zwischen dem Rechner und der externen Technik dient ein spezielles elektronisches Interface. Das System arbeitet weitgehend selbständig. Einen Überblick über die gesamte Anordnung gibt Abb. F1.2.

1.1.1 Der Stoßzyklus

Die Handlungen bzw. Schritte, welche sich für jeden einzelnen Stoß einer Billardpartie (vgl. Abschnitt über Spielregeln) neu wiederholen, werden innerhalb dieser Arbeit als **Stoßzyklus** bezeichnet. Die folgende Aufstellung zeigt dessen prinzipiellen Ablauf. Zusätzlich sind die jeweils korrespondierenden Forschungsgebiete der Robotik angegeben.

1. visuelle Erfassung der Konstellation auf dem Spielfeld
 - Positionen und Farben der Kugeln
 - Position und Ausrichtung des Roboterfahrzeuges*Bilderkennung*
2. Auswahl des besten, d.h. erfolgversprechendsten Stoßes
Suche nach optimaler Handlungsalternative durch Simulation
3. Berechnung des sichersten Weges für das Fahrzeug zur späteren Stoßposition
Wegplanung
4. Fahrt des Fahrzeuges auf diesem Weg, visuelle Überwachung des Vorganges
Echtzeit-Koordination von motorischen und sensorischen Komponenten
5. Manövrieren des Fahrzeuges in die exakte Stoßposition mit Hilfe visueller Rückkopplung
präzise optische Vermessung und motorische Steuerung
6. Ausführung des Stoßes, d.h., Aktivierung des Stößels

1.2 Erläuterungen zur Wahl des Arbeitsthemas

Bei der Bearbeitung einer Problemstellung auf dem Gebiet der Robotik bestehen die folgenden prinzipiellen Alternativen:

1. Bearbeitung von theoretischen Grundlagen und Teillösungen mit Hilfe von Computermodellen und -simulationen
2. Untersuchung von theoretisch untermauerten Algorithmen an bereits existierenden Robotern. Diese können i.w.S. im Eigenbau entstanden oder von kommerziellen Anbietern erworben worden sein.
3. Konzipierung, Implementierung und Untersuchung von einzelnen mechanischen, elektronischen und/oder Software-Komponenten, evtl. um diese in ein zukünftiges oder existierendes Gesamtsystem zu integrieren
4. Konstruktion und Bau eines vollständigen Roboters sowie Implementierung der für diesen notwendigen Algorithmen

Da beim Autor der Wunsch nach einer praktischen Arbeit im Vordergrund stand, welche physisch vorhandene Roboter einbezieht, wäre ein in die erste Kategorie einzuordnendes Projekt wahrscheinlich unbefriedigend gewesen.

Leider bestanden zum Zeitpunkt der Themenwahl an der Universität Leipzig auch noch keine Voraussetzungen, die Projekte der 2. oder 3. Kategorie ermöglicht hätten.

Somit blieben als Alternativen ein externes Projekt an einer anderen Forschungseinrichtung oder Punkt 4 mit all seinen Konsequenzen.

Die Suche nach einer interessanten Problemstellung, die trotz schmaler technologischer Basis und niedrigem finanziellem Budget in vertretbarer Zeit umsetzbar schien, war schließlich mit der Idee eines "Roboters, der Billard spielen kann" erfolgreich.

Die Aufgabe verfügt u.a. über die folgenden wichtigen Aspekte:

- Sie wurde, nach Kenntnisstand des Autors, bisher noch nicht bearbeitet.
- Zu untersuchen ist eine ganze Reihe verschiedener Problemfelder, welche für die Robotik relevant sind und zusammen einen recht guten Querschnitt darstellen. Besondere Schwerpunkte bilden dabei die Konstruktion und Fertigung der mechanischen und elektronischen Komponenten, die Auseinandersetzung mit Methoden der Bildverarbeitung und Mustererkennung, die Lösung von Wegplanungsaufgaben und die praxistaugliche Verschmelzung dieser Teilkonzepte.
- Die einzelnen Probleme erscheinen nicht zu anspruchsvoll, d.h. sie erfordern kein Betreten von bisher unbekanntem Gebiet. Dadurch sinkt die Wahrscheinlichkeit, daß das Gesamtprojekt an der prinzipiellen Unlösbarkeit einer seiner Teilaufgaben scheitern könnte.
- Trotzdem existieren interessante Herausforderungen dahingehend, die üblichen Verfahrensweisen auf die konkreten Anforderungen des Projektes zu fokussieren und sie in geeigneter Weise zu erweitern und zu kombinieren. Dabei sollte allerdings von vornherein angestrebt werden, sich nicht einfach aus dem Angebot von vorhandenen Algorithmen zu bedienen, sondern vorrangig eigene Lösungen zu entwickeln.

- Das Projekt stellt eine in sich abgeschlossene Gesamtlösung dar, d.h., bei Erfolg entsteht ein konkretes "Endprodukt".

Dabei wird nicht angestrebt, daß die Lösung den Anforderungen an ein marktfähiges Produkt gerecht wird oder kommerziell verwertet werden kann.

- Die Aufgabenstellung und zum Teil auch deren Lösung sind auch für außenstehende Personen, die nicht mit dem Fachgebiet vertraut sind, problemlos nachvollziehbar. Insofern ist das Projekt geeignet, wissenschaftliche und technische Sachverhalte einer breiteren Öffentlichkeit nahezubringen.

Dem stehen auch einige negative Gesichtspunkte gegenüber:

- Die zum Bau der technischen Komponenten zur Verfügung stehende technische und technologische Basis ist auf eine Grundausstattung begrenzt und beinhaltet keine speziellen, hochwertigen Bauteile, Werkzeuge oder Maschinen (vgl. Kap. "Technische Details").

Diese Tatsache führt zu dem berechtigten Einwand, das Projekt "doch lieber gleich richtig", also mit professionelleren bzw. fortschrittlicheren Mitteln zu bearbeiten. Dies würde allerdings mit entsprechend höheren Kosten einhergehen, vorausgesetzt, die betreffenden Möglichkeiten stünden dem Autor überhaupt zur Verfügung.

Deshalb soll bewußt versucht werden, die Aufgabe gerade unter scheinbar schlechten Ausgangsbedingungen zu lösen, was in gewisser Weise der Philosophie "cheaper, faster, better" (billiger, schneller, besser) entspricht. Dabei wird angestrebt, aktuelle technische Probleme auch ohne die Verwendung der momentan fortgeschrittensten, dafür aber kostspieligen Technologien zu lösen.

- Es ist kein offensichtlicher Praxisbezug des Themas, also kein wirtschaftlich umsetzbarer Nutzen, zu erkennen. (Von der Vorstellung eines Billardroboters als Konsumgut, ähnlich einem Schachcomputer, sei hier abgesehen.)

Dieser Fakt ist für die experimentelle Robotik durchaus charakteristisch, da viele Projekte der Grundlagenforschung oft dem, im wörtlichen Sinne, "spielerischen" Sammeln von Erfahrungen dienen.

Die innerhalb des Projektes zur Anwendung kommenden Teillösungen (Bildererkennung, Wegplanung, optische Überwachung der Motorik) besitzen jedoch wirtschaftliche Relevanz.

- Die Realitätsnähe des "Roboter-Billards" ist relativ gering. (Weitere Erläuterungen zu diesem Punkt folgen im Abschnitt "Unterschiede zwischen realem Spiel und Billard-Roboter".)

1.3 Die Robotik und ihre Teilaspekte

Die Robotik ist eines der in der heutigen Forschungslandschaft auffälligsten Gebiete. Dies erklärt sich allerdings keineswegs dadurch, daß ihr prozentualer Anteil an laufenden Projekten besonders hoch wäre. Vielmehr gilt sie in ihrer Gesamtheit nach wie vor als exotisch, übt aber durch die von ihr hervorgebrachten Ergebnisse sowohl auf Fachleute als auch Laien eine hohe Anziehungskraft aus.

Wie ist es zu erklären, daß die meisten Menschen von Themen der Robotik deutlich stärker angesprochen werden, als von anderen Gebieten der Informatik, die sich bereits fest etabliert haben? (Als Beispiel seien Datenbanksysteme in Finanzwesen, Verwaltung und Handel genannt.)

Der Grund liegt vor allem wohl darin, daß Roboter physisch existieren, wie der Mensch auch, und daß sie Dinge tun, die uns allen vertraut sind. Zu nennen sind der Umgang mit Werkzeugen und Werkstücken, das taktile, akustische oder visuelle Erkennen von Objekten, das Bewegen in der Umgebung und schließlich, die Voraussetzung für vieles davon - das Denken.

Womit das vielleicht wichtigste Hindernis dafür angesprochen ist, warum Roboter in unserem Alltag noch nicht die Rolle spielen, die ihnen Zukunftsprognosen vergangener Jahrzehnte zuschrieben. Sie kranken an den gleichen Problemen wie die hinter frühen Erwartungen weit zurückbleibende Künstliche Intelligenz, da sie sich letztlich durch diese von "gewöhnlichen Maschinen" unterscheiden ([Drey85], [Var90]).

Weitere entscheidende Faktoren sind der Bedarf an speziellen mechanischen Lösungen (Antriebe, Werkstoffe), finanzielle Gründe und schließlich eine nicht zu vernachlässigende sozialpsychologische Komponente.

Wenn in dieser Arbeit von Robotern die Rede ist, so bezieht sich dieser Begriff fast ausschließlich auf (autonome) mobile Roboter und nicht auf die seit vielen Jahren etablierten Industrieroboter, die zwar bedeutend flexibler sind als herkömmliche Maschinen, deren Einsatz- und Leistungsspektrum aber vergleichsweise begrenzt ist.

1.3.1 Mobile Roboter

Die folgenden Teilabschnitte sollen eine kurze Einführung in das Gebiet der mobilen Roboter geben. Zunächst wird der Versuch einer verallgemeinerten Definition unternommen, gefolgt von Ausführungen zu den zahlreichen Facetten der in mobilen Robotern vereinten Problematiken. Dabei wird kein Anspruch auf Vollständigkeit erhoben. Zum Thema existiert eine recht umfangreiche Auswahl an Literatur, welche sich aber meist nur auf spezielle Teilaspekte konzentriert. Ein einführendes Standardwerk liegt leider noch nicht vor.

Ein **mobiler Roboter** ist ein technisches Gerät, welches sich in einer natürlichen (menschlichen) Umwelt selbständig orientieren und bewegen kann und dabei in der Lage ist, auf sinnvoller Art und Weise zielgerichtet bestimmte Aufgaben zu erfüllen.

Operiert der Roboter dabei ohne eine physische oder anders geartete Verbindung zu weiteren Komponenten, die für seine Funktion zwingend notwendig sind, so wird er als autonom bezeichnet.

Der Aspekt der Orientierung umfaßt die im Roboter vorhandene Sensorik, die Bewegung wird durch motorische Komponenten repräsentiert. Die Selbständigkeit und das sinnvolle Handeln werden durch eine mehr oder weniger komplexe informationsverarbeitende Komponente ermöglicht. Eine zusätzliche Funktionalität kann mit der Erweiterung des Systems durch Manipulatoren oder Werkzeuge, wie z.B. Greifarme oder Schweißgeräte, erreicht werden.

Das Spektrum der Geräte, welche all diesen Punkten genügen, ist sehr umfangreich und ermöglicht eine Vielzahl von Varianten der weiteren Klassifizierung nach verschiedenen Gesichtspunkten. Einige Beispiele für existierende Roboter werden weiter unten angegeben.

1.3.2 Fortbewegung und Wegplanung

Ein charakteristisches Merkmal mobiler Roboter ist das schon an ihrer Bezeichnung erkennbare Element der Fortbewegung in der Umwelt. Diese Umwelt umfaßt neben dem Inneren von Gebäuden oder Produktionsanlagen auch weniger strukturierte und menschenfeindliche Umgebungen. So sind mobile Roboter geplant bzw. im Einsatz, welche sich in unwegsamem Gelände, in Vulkankegeln, unter Tage, auf Schlachtfeldern (!?), in der Kanalisation und in Rohrleitungen, auf bzw. unter der Wasseroberfläche, in der Luft und sogar im freien Weltraum und auf anderen Planeten bewegen. Entsprechend vielfältig sind auch die Arten der Fortbewegungseinrichtungen. Sie reichen von Rädern, Raupenketten und Beinen über Luftkissen, Schiffsschrauben und künstliche Flossen bis hin zu Tragflügeln und Raketentriebwerken.

So unterschiedlich all diese Beispiele sein mögen, so ähnlich sind die in jedem Fall zu behandelnden Fragestellungen. Dabei stellt sich vor allem das Problem der Wegplanung. Welche Steuersignale müssen die motorischen Komponenten erhalten, um bestimmte Bewegungen auszulösen und wie kann eine geeignete Folge von Teilbewegungen erstellt werden, die den Roboter zu einem anderen Punkt (in der Ebene oder im Raum) gelangen läßt? Bei dieser Bewegung darf der Roboter nicht mit anderen Objekten kollidieren oder sich in Situationen begeben, welche zu seiner Beschädigung bzw. Zerstörung führen könnten.

Für die Planung von Wegen in der Robotik stehen prinzipiell zwei verschiedene Methoden zur Verfügung.

Zum einen kann der Roboter über eine geeignet kodierte Karte der Umgebung verfügen. Aufgrund dieses Umweltmodells ist es möglich, eine Route zu planen, noch bevor die eigentliche Fortbewegung beginnt. Für dynamische, d.h. sich schnell verändernde Umgebungen ist diese Vorgehensweise allerdings nicht geeignet.

Die andere Möglichkeit besteht darin, auf die absoluten Koordinaten einer Karte zu verzichten und den Roboter seine nähere Umgebung ausschließlich durch seine Sensoren explorieren zu lassen.

In der Praxis werden die beiden Prinzipien meist miteinander kombiniert. Bei sogenannten Service-Robotern, welche z.B. in Krankenhäusern, Hotels und Bürogebäuden zum Einsatz kommen, wird der Weg mit Hilfe von Karten geplant und während der Bewegung den plötzlich auftretenden Hindernissen angepaßt. Die Karte kann explizit vorgegeben sein oder während einer Lernphase vom Roboter erstellt werden, indem er die Umgebung selbständig erkundet.

Die Realisierung der Wegplanungskomponente im Billardsystem wird im Kapitel "Wegplanung" beschrieben.

1.3.3 Sensorik und Bildererkennung

Der Roboter muß für die Erfüllung der ihm gestellten und sich ihm stellenden Aufgaben über eine geeignete Kombination von externen Sensoren verfügen. (Interne Sensoren, wie z.B. Winkelgeber, sollen an dieser Stelle unberücksichtigt bleiben.)

Eine grundlegende Aufgabe von Sensoren liegt in der Messung der Distanz zu anderen Objekten, um eine Kollision mit diesen zu vermeiden. Dabei kommen, je nach vorhandenen Anforderungen und Budgets, z.B. Ultraschalldetektoren oder Laserscanner zum Einsatz. Weitere wichtige Aufgaben von Sensoren sind die Lagebestimmung des Roboters (Kompaß, Neigungsmesser) und das Identifizieren bestimmter Objekte, wie z.B. von Personen (Infrarot-Bewegungssensor) oder anderen Robotern (Funksignale). Eine Darstellung von verschiedenen Sensorarten und deren Zusammenwirken findet sich in [Jörg94].

Mit zunehmender Perfektionierung von optischen Sensoren, insbesondere von Kameras, wird die Wahrnehmung der Umgebung auf visuellem Wege praktikabel. Optische Informationen sind aufgrund ihrer großen Aussagekraft meist anderen Varianten vorzuziehen. Auch der Mensch bezieht etwa 80 Prozent seines (der Wissenschaft bekannten) "Inputs" über die Augen. Die Anforderungen, welche die Verwendung von Kameras an die Hardware und die Erkennungsalgorithmen stellt, sind enorm. Ein vielversprechender Fortschritt ist die Integration von grundlegenden Verarbeitungsschritten (z.B. Kantendetektion) in die Sensoren selbst. So könnte eine Art von künstlichen Augen entstehen ([Maho91]).

Mit den Aspekten der Sensorik innerhalb des Billard-Projektes beschäftigt sich das Kapitel "Bildererkennung".

1.3.4 Planung und Simulation

Menschen sind in ihrem Leben fast ständig mit dem Problem konfrontiert, Entscheidungen treffen zu müssen. Diese können in ihrer Tragweite sehr bedeutend sein (Hochzeit, Arbeitswechsel) oder uns kaum bewußt werden (ein Frühstücksbrötchen mehr oder weniger essen?). In fast allen Fällen denken wir darüber nach, was passieren wird, wenn man die eine oder andere Wahl trifft. Die Entscheidung wird auf Basis von "objektivem", allgemein zugänglichem Wissen, eigener subjektiver Erfahrung und gefühlsbestimmter Intuition getroffen. Man spielt die möglichen Szenarien in Gedanken durch, simuliert sie sozusagen. Die Qualität der schließlich getroffenen Entscheidung wird in hohem Maße von der Güte des zugrundeliegenden **Umweltmodells** beeinflusst, da der Mensch i.A. nur auf der Basis der ihm zur Verfügung stehenden Informationen operieren kann.

Bei mobilen Robotern verhält es sich (mit Ausnahme der Intuition) im Prinzip nicht anders. Der entscheidende Unterschied besteht in deren stark eingeschränktem Umweltmodell und ihren geringen sensorischen Möglichkeiten. Jedoch wäre auch ein noch so komplexes Weltmodell ohne eine gewisse Fähigkeit zu intelligentem Verhalten wertlos. Der enormen Herausforderung, realistische, alltagstaugliche Modelle "der realen Welt" zu entwerfen und sie mit leistungsfähigen kognitiven Algorithmen zu verknüpfen, ist die Wissenschaft heute noch nicht gewachsen. Weitere Ausführungen dazu folgen im nächsten Abschnitt.

Die Planung und die Simulation verhalten sich in gewisser Weise komplementär zueinander. Abstrakt betrachtet kann von folgenden Begriffen ausgegangen werden:

Einem aktiv handelnden **Agenten** (Mensch, Tier, Roboter) stehen verschiedene **Handlungen** (gehen, sich drehen, greifen oder komplexeres Verhalten) zur Verfügung. Desweiteren

existieren **Ziele** (Zustände), welche durch das Ausführen von geeigneten Handlungen möglicherweise erreicht werden können.

Die Planung hat die Aufgabe, eine Folge von Handlungen zu generieren, die zu einem gewünschten Ziel führen. Im Gegensatz dazu wird bei der Simulation eine Handlung oder eine Folge von Handlungen hypothetisch ausgeführt und der erreichte Zustand bewertet.

In vielen praktischen Anwendungsfällen werden beide Verfahren kombiniert und ergänzen sich wechselseitig. Eine deutliche Trennung zwischen Planungs- und Simulationskomponente ist dann meist gar nicht möglich und auch nicht sinnvoll.

Innerhalb der Robotik tritt der Begriff der Simulation noch in einem anderen, wichtigen Kontext auf. Da die praktische Realisierung, also der Bau von Robotern, eine sehr arbeits- und kostenintensive Angelegenheit ist, werden viele Handlungsabläufe und technische Fragestellungen mit Hilfe von Software nur simuliert. Der Fakt, daß diese Vorgehensweise nur bis zu einem gewissen Punkt akzeptabel ist, wird im Abschnitt "Bauen vs. Simulieren" erörtert.

Die diesem Teilgebiet zuzuordnende Komponente des Billardsystems ist die Stoßwahl, bei der es sich allerdings keineswegs um eine echte Simulation handelt.

1.3.5 Intelligentes Verhalten und Lernen

Als das Forschungsgebiet der Künstlichen Intelligenz entstand, wurde bei der Mehrzahl der Forschungsprojekte der Versuch unternommen, kognitive Probleme durch Computer lösen zu lassen, welche dem Menschen besonders schwer fallen, deren Behandlung also "intelligenten" Menschen vorbehalten ist. Zu nennen sind das Paradebeispiel des Schachspieles und das Führen von mathematischen Beweisen. Man ging zunächst davon aus, daß Handlungen, die unseren Intellekt weniger stark belasten (Erkennen von Gesichtern, einfache Konversation) auch für die Rechenmaschinen unproblematischer sind. Diese Annahme erwies sich als Trugschluß.

Bis heute, nach fast fünfzigjähriger Forschung auf dem Gebiet der KI, existiert kein System, das die Intelligenz eines Kleinkindes oder wenigstens die einer Stubenfliege besitzt. All die Kleinigkeiten, die wir täglich bewältigen, stellen Computer und Roboter vor eine Flut von Problemen. Die Erfolge bei den oben genannten "schwierigen" Aufgaben beruhen überwiegend darauf, daß diese recht gut zu formalisieren, also in Form von Regeln beschreibbar sind. Dies läßt den Schluß zu, daß die Mehrzahl der menschlichen Denkprozesse nicht darauf beruht, sich in einer Menge von expliziten Regeln zu bewegen. Vielmehr ist anzunehmen, daß unsere kognitiven Prozesse auf einer eher implizit "codierten" Basis arbeiten.

In der Robotik werden momentan sowohl klassische KI-Algorithmen eingesetzt, die regelbasiert arbeiten, als auch neuere Konzepte des Softcomputing, welche sich stark an biologischen Vorbildern orientieren. Diese Konzepte, welche z.B. auf künstlichen neuronalen Netzen, evolutionären Algorithmen oder Fuzzy-Logik aufbauen, stellen oft Ausprägungen des Bottom-Up-Ansatzes dar. Dabei wird versucht, kleine, einfache Einheiten kooperieren zu lassen, so daß auf einer höheren Betrachtungsstufe sinnvolle Operationen oder Verhaltensweisen i.w.S. entstehen. Diese Strategie folgt dem Leitsatz "Das Ganze ist mehr als seine Teile." Diese einfachen Einheiten können sowohl physischer Natur sein ("Hardware"), wie z.B. einzelne künstliche Neurone oder parallele Prozessoren, als auch Verhaltensbausteine, welche durch ihre Kombination komplexere Handlungen ermöglichen

("Software"). Ein vielversprechendes Konzept ist z.B. die sogenannte subsumierende Architektur ([Brooks91]). Die relativ komplexen und teilweise recht intelligent anmutenden Verhaltensweisen, die sich schon auf der Basis einfacher Grundbausteine realisieren lassen sind beeindruckend (vgl. [Brai93]).

Im übrigen ist sorgfältig abzuwägen, ob es notwendig und sinnvoll ist, ein Robotersystem mit den gewünschten intelligenten Eigenschaften ausschließlich auf einem bestimmten Paradigma basierend aufzubauen. Die Erfahrung zeigt, daß es meist verschiedene Möglichkeiten gibt, die gleichen Effekte bzw. Funktionen zu realisieren. Kriterien für deren Auswahl können z.B. ihre Funktionssicherheit und der Aufwand für ihre Umsetzung sein.

Bei Robotern, welche mit Menschen in Kontakt kommen können, spielen auch Fragen der Sicherheit (vgl. Asimovs Grundgesetze für Roboter, z.B. in [Rand97]) sowie ergonomische (Mensch-Maschine-Interface) und ästhetische Gesichtspunkte eine Rolle.

Ein hoher Stellenwert kommt innerhalb der Robotik Verfahren zu, welche die Möglichkeit adaptiver Veränderungen im Verhalten des Roboters beinhalten, ihn also befähigen, in gewisser Weise zu lernen. Ein flexibel gestaltetes System ist viel besser in der Lage, entsprechend den Gegebenheiten seiner Umwelt zu operieren als eines, dem ausschließlich eine feste Wissensbasis zur Verfügung steht.

Die für die Konstrukteure und Programmierer des Roboters oft sehr schwierige Aufgabe, die für seine Arbeit notwendigen Informationen zu erfassen und in einer geeigneten Darstellung bereitzustellen, wird damit dem Roboter selbst (und seiner Umwelt!) übertragen. Dies betrifft vor allem solche Probleme, deren formale Beschreibung sehr schwierig oder unmöglich ist. Ein Beispiel ist die Anforderung an den Roboter, seine Aktoren in sinnvoller Weise einzusetzen (Laufen, Greifen, Blickwinkelsteuerung). Ein Algorithmus, der in der Lage ist, diese Aufgaben für alle möglichen (vorher zu katalogisierenden) Konstellationen zu lösen, erfordert einen äußerst hohen Arbeitsaufwand.

Die konkrete Umsetzung von adaptiven (Lern-)Verfahren reicht vom sukzessiven Füllen von LookUp-Tabellen über die Erstellung von Kohonen-Netzen ([Ritter91] Kap.8-13) bis zu selbstmodifizierendem Programmcode. Grundlegend neue Möglichkeiten der Hardware-Realisierung von lernenden Systemen könnten FPGAs (Field Programmable Gate Arrays) bieten, die im Prinzip Schaltkreise darstellen, deren logische Funktionen, d.h. ihre "Verdrahtung", in Echtzeit manipuliert werden kann. Damit wäre es möglich, die Vorteile der Hardware-Implementierung (hohe Verarbeitungsgeschwindigkeit) mit der für adaptive Verfahren notwendigen strukturellen Flexibilität zu vereinigen ([Villa97]).

Aus Gründen des Arbeitsumfanges muß innerhalb der vorliegenden Arbeit auf eine Umsetzung des Konzeptes "Lernen" verzichtet werden (vgl. Kapitel "Ausblick").

1.4 Besonderheiten der praktischen Umsetzung von Robotik-Themen

1.4.1 Bauen vs. Simulieren

Wie bereits dargelegt wurde, kann auf dem Gebiet der Robotik sowohl rein theoretisch, als auch praktisch gearbeitet werden. Da die grundlegende Aufgabe des Gebietes aber in der Schaffung von echten, also physisch existenten Robotern besteht, führt kein Weg am Bau der Roboter und ihrer Erprobung unter Realwelt-Bedingungen vorbei.

Aufgrund des i.A. relativ hohen finanziellen und technologischen Aufwandes der technischen Umsetzung wird oft angestrebt, einen großen Anteil der Entwicklungsarbeit in Form von Computer-Simulationen durchzuführen. Dazu werden zunächst Datenmodelle erstellt, die den Roboter und die Umwelt abbilden, in welcher er sich bewegen soll.

Bei der Auswahl der Parameter, die in die Modellierung einbezogen werden sollen, beschränkt man sich oft auf jene, welche für das konkret zu untersuchende Teilproblem relevant scheinen. Aber selbst wenn man versuchte, *alle* offensichtlichen und vorstellbaren Faktoren in das Modell zu integrieren, kann keinesfalls davon ausgegangen werden, daß bei der späteren realen Umsetzung unangenehme Überraschungen ausbleiben.

Diese Überraschungen können sich z.B. in einer der folgenden Formen manifestieren:

- Die konstruktive Umsetzung der Roboter-Motorik unterliegt Toleranzen, die eine genaue Umsetzung der Algorithmen, insbesondere von Steuerbefehlen, erschweren oder verhindern, d.h. die Algorithmen erweisen sich im Fall der vorliegenden Konstruktion als nicht tauglich.
- Die Sensorik wird durch verschiedene Einflüsse (unerwartet verrauschte Meßgrößen, elektrische Störfelder, ungenügende Helligkeit bei Kameras o.ä.) in ihrer Arbeit behindert, wodurch die Meßergebnisse nicht die für die vorgesehenen Auswertungs-Algorithmen notwendige Verlässlichkeit besitzen.
- Bei realen Umgebungen, die also nicht standardisiert bzw. vereinfacht sind, wie es unter Laborbedingungen bzw. in Computersimulationen möglich und notwendig ist, können Situationen (Zustände) auftreten, die zuvor nicht in Betracht gezogen wurden.

Diese Fakten liegen in der Komplexität der Realwelt begründet, die sich offensichtlich nicht mit einer überschaubaren Anzahl von Regeln und Parametern erfassen läßt. Je komplexer die Szenarien bzw. Umwelten werden, in denen der Roboter agieren soll, desto weniger können wir davon ausgehen, daß theoretisch gemachte Vorhersagen bezüglich des Roboter-"Verhaltens" in jedem Fall korrekt sind. Ansätze zur Erklärung dieser Effekte liefert die Chaos-Theorie (z.B. [Briggs90]), eine Lösung des Problems ist aber wenig wahrscheinlich.

Der Begriff der Umwelt reduziert sich dabei im Übrigen keinesfalls nur auf statische Umgebungen und materielle Objekte im engeren Sinne. Jede Art von Quellen, von denen Informationen ausgehen, die von der Roboter-Sensorik erfaßt werden können, sind zur Umwelt zu zählen (Menschen, andere Roboter, Geräusche, Licht). Gleiches gilt für physische Einflüsse, deren Erfassung die vorhandenen Sensoren nicht ermöglichen, die aber eine Gefahr für den Roboter darstellen können (Treppenstufen, Abhänge, Feuer).

Auch die Bauteile des Roboters selbst können in gewisser Weise als Umwelt betrachtet werden, da deren Parameter die Qualität der Roboter-Umwelt-Interaktion bestimmen und eine Veränderung dieser Parameter (Defekte, Unzuverlässigkeit) unter Umständen erhebliche Konsequenzen nach sich zieht.

Als Analogie sei der menschliche Körper im Falle des Fehlens oder Verlustes eines Körperteiles bzw. Sinnes angeführt. Auch dabei werden viele angeborene bzw. bisher erlernte Verhaltensweisen ("Algorithmen") untauglich. Durch die veränderte Interaktionsfähigkeit hat sich auch die subjektive äußere Umwelt verändert.

Aufgrund der dargestellten Unzulänglichkeiten bei der Übertragung von simulierten Algorithmen auf Realwelt-Bedingungen ist es erforderlich, durch Simulation gewonnene Erkenntnisse frühestmöglich auf ihre Korrektheit in der Praxis zu testen. Dieser Test setzt natürlich die Existenz der Technik voraus.

Somit gilt es ständig, einen vernünftigen Kompromiß zwischen der kostengünstigen, aber unter Umständen nicht korrekten Simulation und dem arbeits- und kostenintensiven Bau der Technik zu finden.

Innerhalb der vorliegenden Arbeit wurden alle Komponenten parallel zur Entwicklung der zugehörigen Algorithmen gebaut, so daß Schwierigkeiten relativ früh erkennbar wurden und die wechselseitigen Erfordernisse permanent abgeglichen werden konnten.

1.4.2 Auftretende Probleme

Das erste Hauptproblem, welches die praktische Umsetzung mit sich bringt, ist die Suche nach geeigneten mechanischen bzw. elektronischen Lösungen, die eine bestimmte Funktionalität ermöglichen. Als Beispiele seien die beiden Roboter-Funktionen "geradeaus fahren" und "drehen auf der Stelle" genannt. Im Rahmen einer Computer-Simulation genügen wenige Zeilen Programmcode zu deren Beschreibung. In der Realität gilt es jedoch, eine Vielzahl von potentiellen Varianten der technischen Umsetzung gegeneinander abzuwägen.

Diese Entscheidung ist wiederum davon abhängig, welche Bauteile, Materialien und Fertigungsverfahren zur Verfügung stehen und ob bestimmte Randbedingungen einzuhalten sind. Dabei kann es sich z.B. um finanzielle Beschränkungen, Anforderungen an Abmessungen, Energieverbrauch und Geschwindigkeit sowie Fragen der Sicherheit handeln.

Die Materialbeschaffung stellt ein weiteres häufiges Problem dar. Da der Markt an Robotik-spezifischen Artikel begrenzt ist und die Produkte relativ teuer sind, bedarf es oft einer langwierigen Suche, verschiedenste Materialien und Produkte zu finden, deren Eigenschaften an die geforderten Funktionen angepaßt werden können. Viele Anregungen dazu bietet der Spielwaren- und speziell der Modellbauhandel.

Diese Ausführungen beziehen sich überwiegend auf mechanische Lösungen. Elektronische Komponenten sind, wenn sie nicht allzu speziellen Anforderungen genügen sollen, in den entsprechenden Fachgeschäften erhältlich.

Eine ausführlichere Erörterung der Thematik findet sich in [Jones96].

Schließlich ist die Funktionsstabilität der technischen Komponenten ein entscheidendes Kriterium für den Erfolg eines praktischen Robotik-Projektes. Alle technischen Teile unterliegen diversen Effekten, die bedacht und, wenn möglich, kompensiert werden müssen. Exemplarisch seien Verschleiß, Reibung, Schlupf, Temperatur- und Stoßempfindlichkeit, Funktionsunterschiede zwischen eigentlich gleichen Bauteilen (Motoren, Schaltkreise), Anlaufverzögerungen, Übersprechen bei elektrischen Leitungen sowie elektromagnetische Störungen genannt.

Somit ist bei Konstruktion und Bau genau auf eine möglichst durchdachte und exakte Fertigung zu achten, damit die Bearbeitung der eigentlichen wissenschaftlichen Inhalte auf einer verlässlichen Basis aufbauen kann.

Im vorliegenden Projekt konnte die gewünschte Qualität der Technik nicht in jedem Fall zufriedenstellend realisiert werden. (Weitere Ausführungen dazu folgen in den Kapiteln "Technische Details" und "Analyse der Leistungsfähigkeit des Billardsystems".)

1.5 Einordnung der Arbeit und Bezüge zu anderen Projekten

Im folgenden soll versucht werden, die vorliegende Arbeit in mehrere Kontexte einzuordnen und damit bezüglich verschiedener Gesichtspunkte mit anderen existierenden Robotik-Projekten vergleichbar zu machen.

1.5.1 Billardsystem und mobile Roboter - ein prinzipieller Vergleich

Offensichtlich handelt es sich bei dem Billard-Roboter nicht um einen typischen Vertreter der Gattung Mobiler Roboter. Trotzdem gibt es eine Reihe von auffälligen Gemeinsamkeiten.

Der eigentliche "Roboter" im Gesamtsystem ist das Fahrzeug, welches den Stößel trägt. Legt man die weiter oben angeführte Definition eines mobilen Roboters zugrunde, so entspricht das Fahrzeug dieser in den Punkten der Fortbewegung und der damit verbundenen Wegplanung mit Hilfe einer Karte. Der Stößel stellt den Manipulator dar, welcher dem Roboter überhaupt erst ermöglicht, seine eigentliche Aufgabe, also das Stoßen der Billardkugeln, zu erfüllen.

Der entscheidende Unterschied zu einem echten mobilen Roboter besteht darin, daß das Fahrzeug über keinerlei eigene Sensorik verfügt. Alle für seine Aktionen notwendigen Informationen über seine Umgebung werden durch die Kamera bzw. die Software bereitgestellt. Diese Art des Zusammenspiels von visueller Erfassung und Motorik schafft einen starken Bezug zu sogenannten Hand-Auge-Systemen. Diese kommen als Einzellösungen, z.B. zum Sortieren von Abfällen, oder als Teile von umfassender ausgestatteten Robotern zum Einsatz. Auch dort wird in hohem Maße die visuelle Information zur Steuerung der Motorik oder zumindest zu deren Überwachung genutzt. Interessant wäre die Vorstellung eines Billard-Roboters, der die Kamera mit sich führt und die Spielfläche selbst nur aus der Perspektive der Kugeln betrachten kann.

Ein weniger starker Unterschied zu typischen mobilen Robotern besteht in der Begrenztheit der Umwelt, in der das Fahrzeug agiert, also dem Billardtisch. Die Komplexität der Umgebung ist einerseits beschränkt, da sich in dieser nur die Kugeln befinden, andererseits aber auch sehr hoch, da die Kugeln in unendlich vielen Konstellationen angeordnet sein können.

Die Kriterien für einen autonomen Roboter erfüllt das Billardsystem definitiv nicht, da die steuernde, informationsverarbeitende Komponente (PC) vollständig vom Fahrzeug getrennt ist.

Die Beantwortung der Frage, ob das Billardsystem als mobiler Roboter bezeichnet werden kann oder nicht, liegt im Ermessen des Betrachters. Wie schon verdeutlicht wurde, war der Bau eines solchen auch nicht das erklärte Ziel der vorliegenden Arbeit. Vielmehr ging es darum, in der Robotik gebräuchliche Verfahren und Konzepte unter dem Gesichtspunkt der Lösung der speziellen Aufgabe "Billard spielen" in einen neuen Kontext zu setzen.

1.5.2 Forschungsarbeiten und Industrielle Anwendungen

Die Robotik erlebte in den letzten Jahren einen starken Aufschwung, welcher wohl in erster Linie an der verbesserten technologischen Basis liegt, die zur Verfügung steht. Die Verfügbarkeit neuartiger und kostengünstiger Bauteile (Mikrocontroller, verschiedene Arten von Sensoren, genaue und robuste Antriebe) und neue theoretische Grundlagen (Softcomputing, Artificial Life [Levy93]) ermöglicht einer Vielzahl von Institutionen, Unternehmen und Privatleuten, sich im engeren oder weiteren Sinne mit dem Bau von Robotern und deren Komponenten zu beschäftigen.

Auch in Deutschland existieren Robotik-Projekte an nahezu jeder Hochschulfakultät, die auf den Gebieten Informatik, Automatisierungstechnik oder Maschinenbau arbeitet. Die weltweit führenden Zentren der Roboterforschung befinden sich aber in den USA (MIT, Carnegie-Mellon-University) und in Japan.

Die folgende Aufstellung enthält einige Beispiele aus einer großen Fülle von bearbeiteten Robotik-Projekten. Diese Arbeiten scheinen dem Billardsystem bezüglich ihres Inhaltes bzw. Umfangs in etwa vergleichbar.

"Stabbalance"

Ein auf einer beweglichen Plattform an einem seiner Endpunkte befestigter Stab soll in senkrechter Position gehalten, also balanciert werden. Je nach Art der Befestigung kann der Stab in einer oder zwei Dimensionen kippen. Die bei verschiedenen Abweichungen von der Senkrechten auszuführenden Ausgleichsbewegungen sollen durch Lernen bestimmt werden.

"taumelndes Labyrinth"

Ein hölzernes Labyrinth von der Größe eines Tellers ist auf einer Mechanik montiert, durch die es um zwei Achsen gekippt werden kann. Darüber ist eine Kamera fest installiert. Im Labyrinth befindet sich eine Kugel. Die Aufgabe besteht darin, die Kugel an eine bestimmte Position des Labyrinths zu bewegen, indem dieses geeignet gekippt wird. Die Kamera dient zur optischen Erfassung des Labyrinths, ein Computer berechnet die Steuerbefehle für die Motoren.

"Fußball"

Ein mit einer Kamera ausgestatteter Roboter soll innerhalb eines Spielfeldes einen Ball und ein Tor finden. Anschließend soll der Ball durch Anschieben in das Tor befördert werden. Der Roboter soll seine Fähigkeiten bei jedem weiteren Versuch verbessern, also lernen.

"Golfbälle einsammeln"

Auf einem Golfplatz erkennt ein Roboter mit Hilfe einer Kamera umher liegende Golfbälle, fährt auf diese zu und nimmt sie mit einer Art Korb auf.

"Ball fangen"

Eine Kombination von Kamera und Roboterarm fängt zugeworfene Bälle auf.

Diese Aufgabe stellt natürlich weitaus höhere technische Anforderungen als das Billardprojekt.

Eine starke Analogie besteht zum Projekt MONAMOVE der Technischen Universität Braunschweig ([WWW1]). Dieses beschäftigt sich mit der Steuerung von führerlosen Transportfahrzeugen in Werkhallen. Derartige Fahrzeuge werden gewöhnlich durch Induktionsschleifen oder andere geeignete Markierungen geführt.

Als Alternative dazu sind bei MONAMOVE an der Hallendecke Kameras montiert, welche die Fahrzeuge erkennen und mittels Funksignalen steuern können. Dadurch ist keine Beschränkung der Fahrtrouten auf feste Wege mehr notwendig.

Weltweit existieren inzwischen eine ganze Anzahl von Unternehmen, welche sich fast ausschließlich mit Robotik (keine Industrieroboter) beschäftigen. Einige ihrer Roboter sind bereits zur Marktreife gelangt und lösen verschiedene Aufgaben.

Als Beispiel sei der Serviceroboter HELPMATE der US-amerikanischen Firma "Transition Research" genannt, der vorwiegend in Krankenhäusern zum Einsatz kommt. Dieser wie eine Art fahrender Kühlschrank aussehende Roboter operiert auf der Basis einer Karte des Gebäudes und kann mittels Ultraschallsensoren vor auftauchenden Hindernissen stehenbleiben bzw. sie umgehen. Dabei gibt er akustische Meldungen von sich. Über Funksignale kann er den Fahrstuhl bedienen. HELPMATE wird überwiegend für Botengänge eingesetzt, z.B. den Transport von speziellen Essenportionen in verschiedene Teile des Krankenhauses.

Schließlich sei noch der vielleicht populärste Vertreter von existierenden Robotern, das Marsfahrzeug "Sojourner", genannt, welches im Juli 1997 mit der NASA-"Pathfinder"-Mission auf dem Roten Planeten landete. Dabei handelt es sich nicht im eigentlichen Sinne um einen mobilen Roboter, sondern eher um ein ferngesteuertes Fahrzeug mit Kamera und wissenschaftlichen Sensoren. Aufgrund seiner mechanischen Konstruktion und des enormen Popularitätsgewinnes, den er der Robotik brachte, kann er aber mit ruhigem Gewissen als mobiler Roboter "ehrenhalber" betrachtet werden.

1.6 Spielregeln des Poolbillard

Die existierenden Varianten des Billardspieles sind recht zahlreich. Für diese Arbeit war es notwendig, eine dieser Varianten auszuwählen und an die technischen Möglichkeiten und inhaltlichen Ansprüche anzupassen. Die Wahl fiel auf das Poolbillard, da die spieltechnischen Anforderungen und die Komplexität des Spielablaufes recht ausgewogen sind. Auch die Tatsache, daß im Handel eine entsprechende Palette von Spieltischen angeboten wird und der Autor über grundlegende eigene Spielerfahrung verfügt, waren ausschlaggebend.

Poolbillard hat seine Wurzeln, im Gegensatz zu Karambolage und Snooker, überwiegend in Nordamerika. Gespielt wird von zwei Spielern mit insgesamt 16 Kugeln, einer weißen, der sogenannten Spielkugel, einer schwarzen und 14 farbigen Kugeln. Diese sind je zur Hälfte einfarbig bzw. weiß mit einer farbigen Banderole und werden meist als "volle" und "halbe" Kugeln bezeichnet. Mit dem Queue angestoßen wird stets die weiße Spielkugel.

Der Spieltisch besitzt in den vier Ecken und in der Mitte der Längsseiten Öffnungen, welche Taschen oder Löcher genannt werden. Das Ziel der einzelnen Stöße besteht darin, die eigenen farbigen Kugeln mit Hilfe der Spielkugel und eventuell unter Einbeziehung der Spielfeldbanden in die Löcher zu befördern. Dieser Vorgang wird als "Versenken" bezeichnet.

Zu Beginn einer Partie werden alle Kugeln, außer der Spielkugel, in Form eines Dreiecks angeordnet und dieses in Höhe eines Viertels der Tischlänge plaziert. Auf der gegenüberliegenden Seite wird die weiße Kugel positioniert und der erste Spieler eröffnet die Partie, indem er die Spielkugel in Richtung des Kugeldreiecks stößt. Wenn durch diese Aktion bereits eine der farbigen Kugeln in einem der 6 Löcher versenkt wurde, ist entschieden, ob der eröffnende Spieler ab sofort auf die vollen oder die halben Kugeln spielt, also diese

oder jene in den Löchern versenken muß. Dem anderen Spieler ist dann die andere Hälfte der farbigen Kugeln zugeordnet.

Einen Sonderfall stellt die schwarze Kugel dar. Sollte ein Spieler diese versenken, bevor er alle seine farbigen Kugeln versenkt hat, so gilt die Partie als verloren. Wird die weiße Spielkugel unbeabsichtigt versenkt, so gilt dies als Foul und das Spiel wird vom Gegner fortgesetzt. Gleichfalls verboten ist, daß die Spielkugel nach ihrem Anstoßen als erstes eine Kugel des Gegners berührt.

Die beiden Spieler stoßen abwechselnd. Wer eine Kugel versenkt hat, erhält das Recht, einen weiteren Stoß auszuführen. Hat ein Spieler alle seine farbigen Kugeln vom Tisch entfernt, ist es seine Aufgabe, die schwarze Kugel in einem bestimmten der sechs Löcher zu versenken. Für die Festlegung dieses Loches gibt es zahlreiche Varianten. Gemeinsam ist ihnen jedoch, daß der Spieler verloren hat, sollte sich die schwarze Kugel in ein anderes Loch bewegen.

1.7 Unterschiede zwischen realem Spiel und Billard-Roboter

Wie leicht einzusehen ist, muß die Umsetzung des Billardspieles in der angestrebten Form mit einer Vielzahl von Einschränkungen gegenüber dem realen Spiel einhergehen. Dies betrifft sowohl technische und konstruktive als auch solche Aspekte, die sich auf Spielregeln und Taktik beziehen. In diesem Abschnitt sollen derartige Unterschiede erläutert und begründet werden.

Konkret geht es um:

1. Abmessungen von Tisch und Kugeln sowie Kugelfarben,
2. Konsequenzen, die sich aus der Tatsache ergeben, daß der Roboter sich ständig auf der Spielfläche befindet,
3. prinzipiell unmögliche Stöße
4. Stöße, die wegen der beschränkten Genauigkeit des Systems oder aus anderen Gründen nicht ausführbar sind.

zu Punkt 1:

Beim verwendeten Billardtisch handelt es sich um einen von der Firma PotBlack UK (Großbritannien) produzierten Snooker-Tisch. (Snooker ist eine besonders in Großbritannien verbreitete Billard-Variante.) Er wird im Spielwarenhandel angeboten und ist mit seiner Größe von ca. 100×50cm eigentlich auf die Benutzung durch Kinder zugeschnitten. Seine Vorteile bezüglich der gestellten Anforderungen liegen in seinen Abmessungen, dem Preis und den mitgelieferten Kugeln. Echte Poolbillardkugeln sind (auch in der maßstäblich verkleinerten Kinder-Version) größer als Snookerkugeln und zudem teilweise mit der oben erwähnten Farbbänderole und aufgedruckten Ziffern versehen. Snookerkugeln sind durchweg einfarbig. Dadurch wird ihre optische Erkennung bedeutend vereinfacht.

Trotzdem sind nicht alle dem Spiel ursprünglich beiliegenden Kugeln zufriedenstellend einsetzbar. Die dunkleren Farben blau, grün, braun und schwarz sind für die Bildverarbeitungsalgorithmen nur mit sehr hohem Aufwand vom Hintergrund zu trennen, so daß sie keine Verwendung finden.

Gespielt wird mit je fünf gelben und roten Kugeln, welche die "vollen" bzw. "halben" ersetzen, sowie einer weißen Spielkugel. Auf die oben angedeutete Funktion der schwarzen Kugel wird verzichtet.

Da die Kugeln beim Billardspiel stark beschleunigt werden müssen, muß der Stoßende (Mensch oder Roboter) eine beträchtliche Kraft aufbringen können. Während ein menschlicher Spieler echten Poolkugeln (Durchmesser ca. 58 mm) relativ mühelos eine hohe Geschwindigkeit geben kann, entstünden bei deren Verwendung im vorliegenden System gravierende Probleme. Eine technische Lösung zu finden, die bei relativ kleinen eigenen Abmessungen genügend Kraft entwickeln kann, ist sehr schwierig. Bei den verwendeten kleinen Kugeln waren diese Schwierigkeiten bedeutend leichter zu beherrschen. Weitere Ausführungen zu diesem Punkt enthält das Kapitel "Technische Details".

zu Punkt 2:

Wenn ein menschlicher Spieler einen Billardstoß ausführt, so führt er die auf der Spielfläche aufgesetzte Hand und das Queue unmittelbar nach dem eigentlichen Stoßen nach oben. Somit können sich die Kugeln auf der Fläche ungehindert bewegen. Im Gegensatz dazu bleibt das Roboterfahrzeug auf dem Tisch stehen. Dadurch besteht die Möglichkeit, daß es von der einen oder anderen Kugel getroffen wird und dadurch deren eigentliche Bahn verfälscht.

Als Lösung dieses Problems wäre es z.B. denkbar, das Fahrzeug mit Hilfe einer Feder oder eines Seilzuges nach dem Stoß sehr schnell vom Untergrund abzuheben. In Betracht käme auch die grundlegend andere Ausführung des Roboters in Form eines Armes, welcher den Stößel von oben auf den Tisch herabsenkt. Mit dieser Variante wären aber für diese Arbeit interessante Aspekte von mobilen Robotern und Bilderkennung (Phasen 3 bis 5 des Stoßzyklus) ausgeschlossen.

Die möglicherweise interessanteste Lösung wäre, das System mit einer zusätzlichen Simulationskomponente auszustatten, die den geplanten Stoß vor seiner Ausführung auf fragliche Kollisionen testet und ihn bei positivem Ergebnis verwirft. Diese Aufgabe ist aus modelltheoretischer und programmiertechnischer Sicht interessant, aber aufgrund des chaotischen Charakters der Kugelbewegungen sehr anspruchsvoll. Einige schon existierende Billard-Simulationen (z.B. "VirtualPool" der Firma InterPlay, [WWW2]) erreichen bereits eine große Annäherung an die Realität. Die eigentliche Herausforderung bestünde jedoch darin, die Parameter der Simulation mit den Gegebenheiten des konkreten Systems, wie z.B. Material und Justierung, exakt abzustimmen, um eine zuverlässige Stoßprognose zu ermöglichen.

Aufgrund des offensichtlichen Umfangs dieser Aufgabe muß im Rahmen der vorliegenden Arbeit auf deren Umsetzung verzichtet werden. Die durch Kollisionen von Kugeln und Fahrzeug entstehenden Einschränkungen werden, im Bewußtsein der verringerten Realitätstreue, in Kauf genommen.

zu Punkt 3:

Dem Billardspieler dient bei der Mehrzahl seiner Stöße eine Hand zur Führung des Queues in Nähe der Spielkugel, während die andere dasselbe weiter hinten ergreift und die Stoßkraft einbringt. Die führende Hand kann meist vor der Kugel aufgesetzt werden, da dazu ausreichend Platz vorhanden ist. In der Roboter-Variante ist an dieser Stelle das Fahrzeug positioniert.

Nun können aber Fälle auftreten, in denen die Spielkugel von anderen Kugeln umgeben ist oder sich nahe an einer Spielfeldbande befindet. Der menschliche Spieler ist also gezwungen, die Führungshand auf andere Art und Weise oder gar nicht aufzusetzen. Für das Roboterfahrzeug bestehen keine derartigen Alternativen, da es sich immer in einer von der angestrebten Bewegungsrichtung der Spielkugel determinierten Position und Ausrichtung befinden muß. Im Kapitel "Stoßwahl" werden sowohl diese, als auch die folgenden Probleme weiter behandelt.

zu Punkt 4:

Dem geübten Billardspieler steht zum Erreichen seiner Ziele eine Vielzahl technischer und taktischer Feinheiten zur Verfügung. Zum Beispiel ist es möglich, der Spielkugel ein Effet, also einen zusätzlichen Rotationsimpuls zu verleihen und damit z.B. deren "Zurückkommen" oder eine bogenförmige Bewegungstrajektorie der Kugelbewegung zu erreichen. Angestrebt wird auch ein "Zurechtlegen" der Kugeln für den nächsten Stoß, bei gleichzeitigem Versenken einer Kugel. Solche teilweise atemberaubenden Stoßtechniken werden in der entsprechenden Fachliteratur (z.B. [Schiff94]) ausführlich beschrieben.

Das Nachvollziehen dieser anspruchsvollen Stoßvarianten kann (im Moment) kein realistisches Ziel für ein künstliches System mit den vorliegenden Möglichkeiten sein. Sie erfordern ein sehr gutes Auge, eine hohe Fingerfertigkeit und eine großes Maß an Wissen und Erfahrung, welche durch Kamera, Roboterfahrzeug und Steuersoftware nur schwach angenähert werden können. Die Algorithmen, welche sowohl die genannten Effekte, als auch die bewußte Einbeziehung der Banden ins Spiel realisieren könnten, sind sicher realisierbar, würden aber den Rahmen dieser Arbeit sprengen.

Aus diesen Gründen beschränkt sich das Repertoire des Billardroboters auf einfache, direkte Stöße, wie in Abb. 1.3 illustriert.

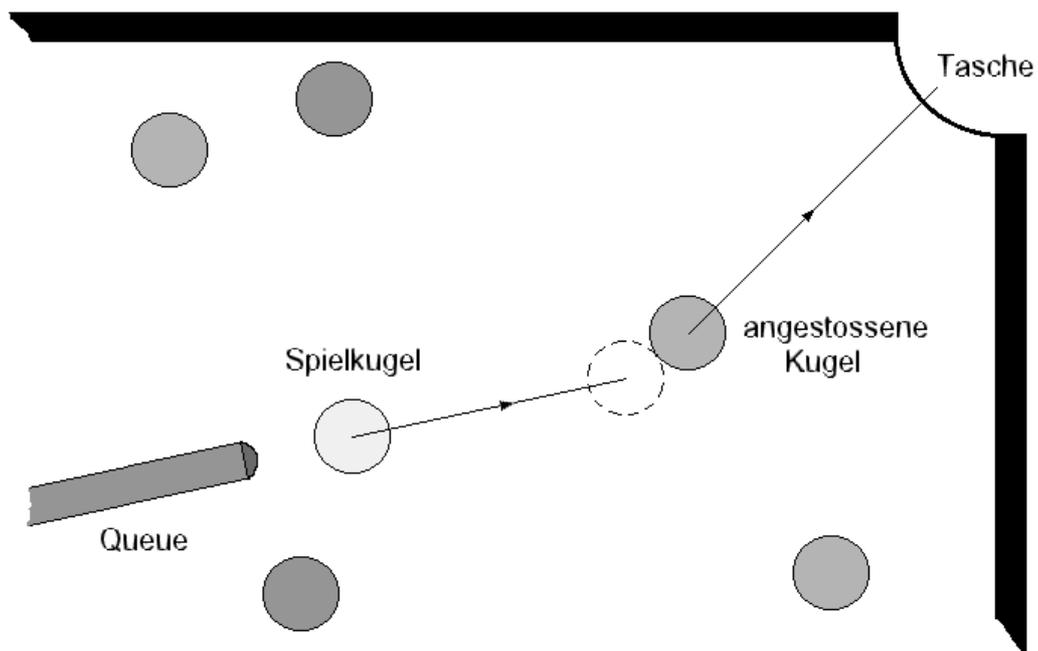


Abb. 1.3 einfacher, direkter Billardstoß

Mit dem Queue (bzw. dem Roboter-Stößel) wird die weiße Kugel angestoßen. Diese stößt direkt jene Kugel an, welche versenkt werden soll.

2 Bilderkennung

Dieses Kapitel behandelt alle Teile der Gesamtlösung, die eine visuelle Komponente besitzen, in denen also die Kamera und Algorithmen der Bildverarbeitung und Objekterkennung zum Einsatz kommen. Die Reihenfolge der einzelnen Abschnitte ist nicht streng am chronologischen Ablauf des Stoßzyklus (vgl. Kap. "Einführung und Motivation") ausgerichtet, es sollte aber stets ersichtlich sein, wann und wofür die verschiedenen Verfahren angewandt werden.

Zunächst werden einige Erläuterungen zur verwendeten Kamera und den damit verbundenen Vor- und Nachteilen gegeben. Es folgen Überlegungen zu den an die Bildverarbeitung gestellten Anforderungen. Danach werden die einzelnen Verarbeitungsschritte der Spielfeld-Erfassung (Punkt 1 des Stoßzyklus) detailliert erläutert. Anschließend geht es um die exakte Lageermittlung der am bereits ausgewählten Stoß beteiligten Kugeln. Den Abschluß des Kapitels bildet die Darstellung der Algorithmen, welche bei der optisch rückgekoppelten Steuerung der Fahrzeugbewegungen eingesetzt werden (Punkte 4 und 5 des Stoßzyklus).

2.1 Die Kamera - allgemeine und spezielle Probleme der Bildaufnahme

In Projekten, welche sich mit Themen der Bildverarbeitung beschäftigen, werden zur Zeit überwiegend Videokameras zur Erfassung der benötigten Bilder eingesetzt. Dazu stellt die entsprechende Industrie eine große Auswahl von in Qualität und Preis unterschiedlichen Modellen zur Verfügung. Das mit einem CCD-Sensor (ladungsgekoppelte Sensorzelle) aufgenommene (und damit digitale) Signal wird elektronisch in ein Standard-Video-Signal (analog) umgewandelt. Um diesen analogen Signalfluß für einen Computer bearbeitbar zu machen, muß er wieder digitalisiert werden. Dies geschieht meist mit Hilfe von Frame-Grabber-Karten, um die der verwendete Computer ergänzt werden muß. Durch ihre komplexe Hardware sind diese Karten in der Lage, die sehr große Datenmenge einer Videoquelle in Echtzeit zu digitalisieren und dem Rechner über den Systembus bereitzustellen.

Die zunehmend aufkommenden Digital-Kameras liefern sofort ein digitales Bildformat, können aber mit der Kombination Video-Kamera/Frame-Grabber bezüglich Bildauflösung und Geschwindigkeit noch nicht mithalten.

Die für die Aufgaben der Bildverarbeitung entscheidenden Eigenschaften einer Kamera sind in erster Linie die Bildauflösung und die Fähigkeit, Farben zu erfassen.

Die Auflösung einer Kamera ergibt sich aus der Anzahl der im Bild enthaltenen Bildpunkte (**Pixel**, Abkürzung Px), welche entscheidend die Detailgenauigkeit des Bildes bestimmen. Von Vorteil ist, wenn die Bildquelle unterschiedliche Auflösungen zur Auswahl bereitstellt, da eine hohe Bildpunktzahl einen hohen Rechenaufwand bei der Verarbeitung nach sich zieht und die zu lösende Erkennungsaufgabe eventuell auch schon mit einer niedrigeren Auflösung zu bewältigen ist.

Im menschlichen Auge ist auch nur ein bestimmtes Gebiet der lichtempfindlichen Rezeptoren mit der maximalen Dichte besetzt (Fovea oder gelber Fleck). Für Beobachtungen "aus dem Augenwinkel" reicht bereits eine geringere Sensordichte, respektive Auflösung, aus.

Farbkameras bieten gegenüber Graustufen-Kameras die Möglichkeit, Objekte nicht nur bezüglich ihrer Helligkeit, sondern auch ihres optischen Reflektionsspektrums, also ihrer Farbe, zu unterscheiden. Diesem Vorteil steht allerdings gegenüber, daß die Bilder statt nur eines Graustufenkanals drei Farbkanäle (rot, grün, blau) umfassen und somit etwa die dreifache Datenmenge eines Graustufenbildes zu übertragen und zu bearbeiten ist.

2.1.1 Technisch bedingte Aufnahmefehler

Für die praktische Bilderkennung ist wesentlich, welche technischen Effekte die Darstellungstreue eines Objektes auf einem Bild beeinflussen, d.h. ob das Objekt auf dem Bild entsprechend seiner tatsächlichen Geometrie, Helligkeit und Farbe dargestellt wird.

Diese Probleme sind bei der praktischen Arbeit keineswegs vernachlässigbar, obwohl sie in der Literatur zum Thema nur selten behandelt werden.

Im einzelnen zu nennen sind der Verzeichnungsfehler, der Shadingeffekt und weitere technologisch bedingte Einflüsse, welche bei der Vorstellung der in dieser Arbeit verwendeten Kamera behandelt werden.

Der **Verzeichnungsfehler** liegt in der optischen und geometrischen Beschaffenheit des Aufnahmesystems begründet. (Eine exakte Ausrichtung und Justierung der gesamten Anordnung sei vorausgesetzt.) Durch die im Objektiv der Kamera verwendeten Linsen tritt ein mehr oder weniger starker "Fischaugen-Effekt" auf, d.h. daß gerade Linien in Nähe des Bildrandes zunehmend gekrümmt erscheinen. Vor allem dann, wenn im Bild bestimmte Vermessungen durchgeführt werden sollen, kann sich diese Störung erheblich auswirken. Gegenmaßnahmen sind z.B. die Verwendung von verzeichnungsfreien Optiken (teuer), das exakte Berechnen eines korrigierten Bildes mit Hilfe von optischen Formeln (Parameter der Linsen müssen bekannt sein) oder Kalibrierobjekten bzw. die empirische Ermittlung des Fehlers und das Erstellen einer korrigierenden Matrix bzw. Karte.

Die einfachste Lösung besteht darin, nur die inneren Bildteile zur Weiterverarbeitung zu verwenden, welche kaum verzerrt werden.

Zu Verzeichnungsfehlern führen auch Montagetoleranzen der Kamera, wenn z.B. die optischen Achsen der Einzelteile (Linsen und CCD-Sensor) nicht exakt aufeinander ausgerichtet sind.

Der **Shadingeffekt** bezeichnet allgemein das Auftreten von Schattierungen, welche nicht ursächlich mit den zu erkennenden Objekten in Verbindung stehen.

Wiederum durch die Krümmung der Linsen im Kamera-Objektiv ist das aufgenommene Bild zum Rand hin dunkler. Somit werden gleichartige Gegenstände, abhängig von ihrer Position im Bild, mit unterschiedlicher Helligkeit bzw. Farbe wiedergegeben. Bei einer angestrebten Vordergrund-Hintergrund-Trennung verursacht dies u.U. große Probleme.

Auch eine ungleichmäßige Ausleuchtung der aufzunehmenden Szene trägt stark zum Shading bei.

Der Abschnitt "Shading-Korrektur" beschreibt den in dieser Arbeit eingesetzten Algorithmus zur Kompensation dieser Effekte.

Der für das Billardsystem ebenfalls relevante **perspektivische oder Parallaxe-Fehler** wird im Abschnitt „Korrektur des perspektivischen Fehlers“ erläutert.

2.1.2 Die Connectix-ColorQuickCam

Bei der für das Billard-System verwendeten Kamera handelt es sich um eine ColorQuickCam der Firma Connectix (Abb. 2.1, [Conn96], [WWW3]). Sie gehört zur oben erwähnten Gruppe der Digitalkameras und wurde u.a. für rechnergestützte Videokonferenzen entwickelt. Ein Frame-Grabber ist nicht notwendig, die PC-Variante der Kamera wird an einen parallelen (Drucker-) Port des Rechners angeschlossen. Diese praktische und kostengünstige Lösung wird allerdings mit einer vergleichsweise langsamen Übertragung der Bilder von der Kamera in den Speicher erkauft. Für die im Rahmen dieser Arbeit auftretenden Aufgaben fällt dieser Nachteil aber kaum ins Gewicht.

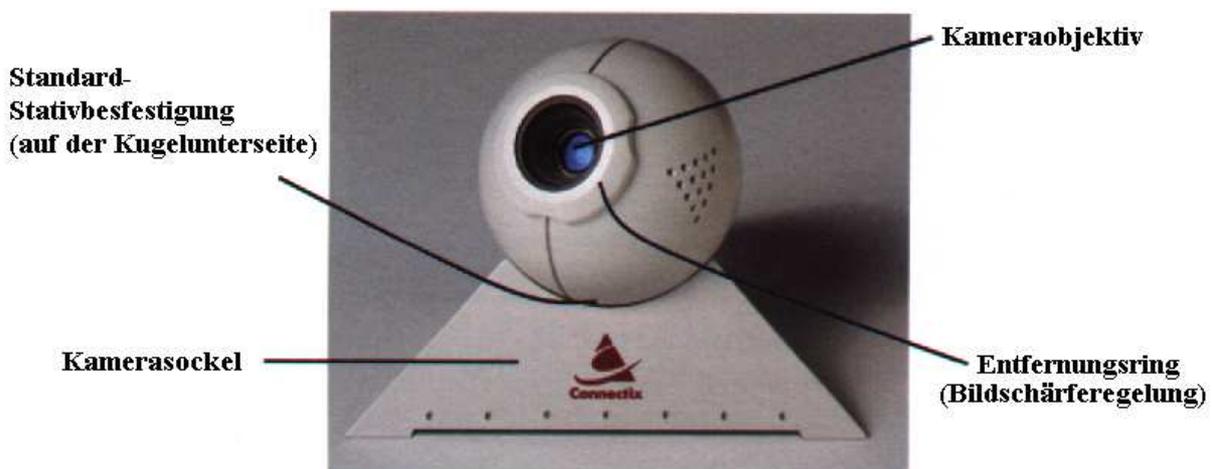


Abb. 2.1 Connectix-ColorQuickCam
Bild übernommen aus [Conn96]

Die bereitgestellten Bildgrößen reichen von 80×60 bis 320×240 Pixel. Im Anwenderhandbuch [Conn96] und der mitgelieferten Software steht eine weitere Größe von 640×480 Pixel zur Auswahl, welche sich aber als nicht wirklich aufgenommen, sondern als interpoliert herausstellte und somit für die Zwecke dieser Arbeit nicht geeignet ist.

Die Kamera ist in der Lage, TrueColor-Aufnahmen (Farbtiefe 24 Bit) zu liefern. Somit stehen in jedem der drei Farbkanäle Intensitäten von 0..255 zur Verfügung.

Gesteuert wird die ColorQuickCam unter Verwendung der im "Video for Windows"-API der Firma Microsoft bereitgestellten Fensterklasse AVICap ([MiSo92]). Dadurch ist es mit Hilfe weniger Steuerbotschaften möglich, die Kamera zu veranlassen, ein Bild in einer gewünschten Größe aufzunehmen und an den Rechner zu übertragen. Über die Windows-Zwischenablage werden die Bilddaten schließlich in die eigentliche Bildverarbeitungssoftware übernommen.

Das eigenwillige kugelförmige Design der Kamera macht es notwendig, sie in einer zusätzlichen Halterung zu fixieren. Diesbezügliche Einzelheiten werden innerhalb des Kapitels "Technische Details" näher erläutert.

Die bereits beschriebenen Effekte des Verzeichnungsfehlers und des Shadings wirken sich in den mit der ColorQuickCam aufgenommenen Bildern deutlich aus (Abb. 2.2).

Als weitere Störung treten bei dieser speziellen Kamera sogenannte "**Christmas-Lights**" auf. In Abhängigkeit von der gewünschten Helligkeit des aufzunehmenden Bildes und der realen Ausleuchtung der Szene muß der CCD-Sensor unterschiedlich lange aktiviert sein, also elektrische Ladungen akkumulieren. Das entspricht der Belichtungszeit beim herkömmlichen Fotografieren. Ist diese Dauer relativ lang, die Szene also dunkel, laden sich einige der CCD-Zellen aufgrund ihrer Halbleitereigenschaften von allein auf und verursachen einzelne hell

leuchtende Pixel im Bild, die aussehen wie die Lichter an einem Weihnachtsbaum (Abb. F2.3). Das Kompensation dieser Störung wird im Abschnitt "Elininieren der 'Christmas-Lights'" beschrieben.

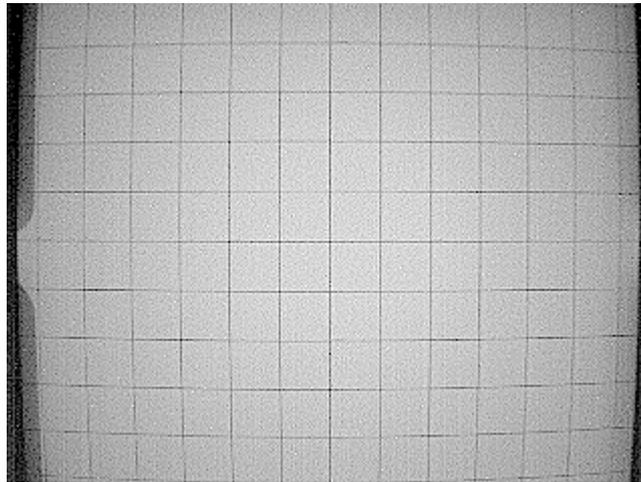


Abb. 2.2 Verzeichnungsfehler und Shading bei der ColorQuickCam

Das Farbbild (320x240 Pixel) wurde in Graustufen umgewandelt und die Schärfe rechnerisch erhöht. Zu erkennen ist die nach außen zunehmende Krümmung der eigentlich geraden Linien und die abnehmende Helligkeit. Die dunklen Streifen am rechten und linken Bildrand sind Banden des Billardtisches. Links ist außerdem ein Schattenwurf der Bande zu erkennen.

Die Kamera verfügt über eine automatische Helligkeits- und Farbsteuerung. Obwohl diese zu für den menschlichen Betrachter schöneren Bildern führt, bleibt sie während der Billard-Prozeduren deaktiviert, da die Algorithmen der Bilderkennung eine gewisse Konstanz der Aufnahmeparameter voraussetzen.

2.2 Anforderungen an die Bilderkennung

Die Aufgaben der Bilderkennungskomponente sind die Ermittlung der Positionen und Farben der jeweils auf dem Spielfeld vorhandenen Kugeln sowie der Position und Ausrichtung des Fahrzeuges zu verschiedenen Zeitpunkten. Dabei werden unterschiedlich hohe Genauigkeiten der Erkennung gefordert. Die theoretisch erreichbare Genauigkeit ist abhängig von der maximalen Bildauflösung, dem Abstand der Kamera zum Spielfeld und der Größe der zu erkennenden Objekte. Zudem soll das gesamte Spielfeld erfaßt werden können.

Zunächst sollen diese Werte bezüglich der Kugelerkennung betrachtet werden. Es reicht dazu aus, eine Dimension, also die Länge oder die Breite zu untersuchen. Die maximale Anzahl von Bildpunkten B_{\max} beträgt 320 Px, die Länge des Spielfeldes L ca.1000 mm und der Kugeldurchmesser D_K 28 mm. Gesucht ist die Anzahl der Bildpunkte B_K , die einer Kugel entsprechen. Es gilt $B_K = B_{\max} \times D_K / L$.

Montiert man die Kamera so, daß sie das gesamte Spielfeld "überblickt", dann ergibt sich $B_K=9$ Px bzw. 1 Px entspricht 3,125 mm. Dieser Wert ist offensichtlich nicht akzeptabel, da hier durch die mit Ungenauigkeiten behafteten Erkennungsergebnissen schon geringe Unterschiede zu erheblichen Positionsfehlern führen. Werden die 320 Pixel auf die Breite des Spielfeldes von ca. 500 mm verteilt, verdoppelt sich die Genauigkeit, allerdings müssen drei Bilder aufgenommen werden, um die gesamte Fläche zu erfassen. Das macht ein mechanisches Positionieren der Kamera oder die Verwendung von drei Kameras notwendig.

Will man die Genauigkeit weiter verbessern, so ist es notwendig, die Kamera sowohl in Richtung der Längsachse als auch der Breite des Spielfeldes zu verschieben. Das erfordert jedoch eine sehr aufwendige Mechanik.

Im vorliegenden System stellte sich die Variante der alleinigen Längsbewegung der Kamera als umsetzbar und ausreichend heraus. Somit erscheint eine Kugel im Bild mit einer Größe von ca. 19×19 Px. Für die Erfassung des gesamten Feldes wird eine Bildgröße von 320×160 Px (3,19 mm/Px, Kugel ca.9×9 Px, Aufnahmegröße der Rohbilder 120×160 Px) verwendet. Die mindestens notwendigen 3 Einzelbilder werden auf 8 erweitert, wobei jeweils nur das mittlere Drittel der Rohbilder (40×160 Px) in das Gesamtbild des Spielfeldes eingeht.

Es existieren 8 feste Positionen, welche von der Kamera angefahren werden können. Die Einzelheiten der Positionier-Mechanik sind im Kapitel „Technische Details“ beschrieben. Durch die Beschränkung auf das mittlere Bilddrittel wird der Effekt der Kamera-Verzeichnung zumindest in Richtung der Spielfeld-Längsachse nahezu ausgeschaltet.

Auf die Unterstützung der Bilderkennungsalgorithmen durch besondere Markierungen (Landmarken) oder andere optische Hilfsmittel wurde, mit Ausnahme der Fahrzeugerkennung (siehe Abschnitt „Optische Erkennung des Roboterfahrzeuges“), völlig verzichtet.

Auch die meist übliche zusätzliche Ausleuchtung der Szene, in diesem Falle also der Spielfläche, ist nicht vorhanden. Daraus ergibt sich natürlich die Konsequenz, daß die Algorithmen relativ robust gegenüber ungleichmäßigem Umgebungslicht und Schattenwürfen sein müssen.

Eine wichtige Anforderung an viele Bilderkennungsverfahren ist eine hohe Verarbeitungsgeschwindigkeit. Für die erfolgreiche Funktion des Billardsystems ist die Dauer der Bearbeitung der aufgenommenen Bilder unkritisch. Abstriche können bei der Wegkontrolle des Fahrzeuges (vgl. Kapitel „Kamera-Fahrzeug-Koordination“) gemacht werden, aber im Prinzip ist das schwächste Glied im zeitlichen Ablauf die Geduld des menschlichen Beobachters.

Generell gilt, daß diese Arbeit in erster Linie die Lösbarkeit der Aufgabe "Billard spielen" mit den vorhandenen technischen Voraussetzungen untersuchen soll und Performance-Aspekte (bzgl. der Geschwindigkeit) zweitrangig sind. Mit höherem technischen Aufwand (bessere Kamera und schnellerer Computer) wären viele Probleme natürlich einfacher oder eleganter zu lösen, bzw. würden gar nicht auftreten.

2.3 Erfassung der Kugelpositionen (Groberkennung)

Die Kugelerkennung erfolgt mit dem Ziel der Zeit- und Speicherplatzersparnis in zwei Stufen:

1. Erfassung des gesamten Spielfeldes mit einer relativ geringen Bildauflösung und Ermittlung der Kugelpositionen mit einer für die folgenden Schritte des Spielzyklus (Stoßwahl, Wegplanung, Fahrt des Roboterfahrzeuges) ausreichenden Genauigkeit
2. genaue Erfassung der Positionen der am folgenden Stoß beteiligten (zwei) Kugeln mit der maximalen Bildauflösung

Die erste Stufe wird als **Groberkennung** bezeichnet und in diesem Abschnitt behandelt. Später folgt die Erläuterung der zweiten, als **Feinerkennung** bezeichneten Stufe.

Wie bereits beschrieben, wird zunächst das gesamte Spielfeld in Form eines einzigen Bildes erfaßt, indem acht aneinander grenzende Streifen mit einer Kameraauflösung von 120×160 Pixel aufgenommen und zusammengesetzt werden. Die Überblick-Bilder liegen im Format

320×160 Pixel (320=8×mittleres Drittel von 120) vor, getrennt in die drei Farbkanäle Rot, Grün und Blau. Um aus diesen Rohbildern die Koordinaten des Fahrzeuges und der Kugeln sowie deren Farben zu erhalten, sind die folgenden Schritte auszuführen :

1. Eliminieren der "Christmas-Lights"
2. Shading-Korrektur
3. Binarisieren, d.h., Trennung in Vorder- (Kugeln) und Hintergrund (Spielfeld)
4. Erosion (Schrumpfen) der Kugelgebiete
5. Ermittlung der Mittelpunktkoordinaten
6. Bestimmung der Kugelfarben
7. Ermittlung von Position und Ausrichtung des Fahrzeuges

Die Schritte 1 bis 5 werden nun detailliert beschrieben. Die Beschreibungen der Schritte 6 und 7 folgen in den entsprechenden Abschnitten.

2.3.1 Eliminieren der "Christmas-Lights"

Der diesen Störungen zugrundeliegende elektrische Effekt wurde bereits erläutert. Das Beispiel eines Rohbildes ist in Abb. F 2.4 zu sehen. Dabei ist zu erkennen, daß die Störungen auffallend hell sind, sich immer nur auf ein einzelnes Pixel erstrecken und sich innerhalb der 8 Bildstreifen wiederholen. Der zum Entfernen der Störungen eingesetzte Algorithmus arbeitet wie folgt:

Teste für jedes Pixel, ob es heller ist als alle seine Nachbarn (3x3-Umgebung).
Wenn ja, dann ersetze seinen Wert durch die durchschnittliche Helligkeit der Nachbarn.

Damit werden natürlich auch Punkte manipuliert, die keine Störungen darstellen, was aber keine nachteiligen Wirkungen mit sich bringt. Die Filterung muß für alle drei Farbkanäle getrennt erfolgen. Das Ergebnis ist in Abb. F 2.5 dargestellt.

Alle weiteren Schritte der Kugel-Koordinatenermittlung werden ausschließlich auf dem Rot-Kanal des aufgenommenen Bildes, d.h., auf einem Graustufen-Bild, ausgeführt. Für das Ziel der deutlichen Trennung von Kugeln und Untergrund erweist sich der Rotkanal als am kontrastreichsten, da die roten Spektralanteile der Kugelfarben weiß, gelb und rot sich gut vom weit geringeren Rot-Anteil der grünen Bespannung abheben (Abb 2.6).

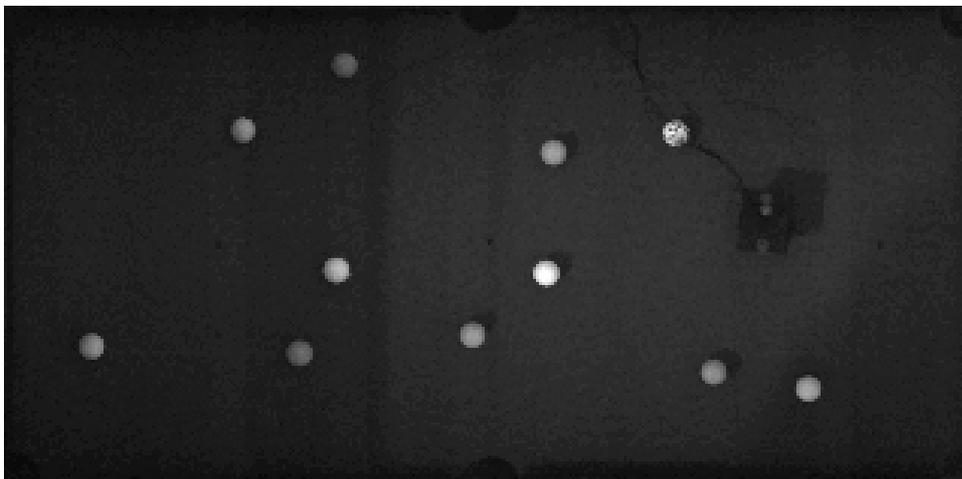


Abb. 2.6 Rotkanal des Ausgangsbildes

2.3.2 Shading-Korrektur

Das hier zur Verminderung von Shading-Effekten verwendete Verfahren lehnt sich an die in [Voss91] beschriebene Vorgehensweise an.

Ähnlich einem Fußballfeld, welches teilweise im Schatten der Stadionüberdachung oder von Wolken liegt, sind großflächige dunklere Gebiete vorhanden. Diese Helligkeitsdifferenzen müssen weitgehend eliminiert werden, da im folgenden Schritt das gesamte Bild mit einer Schwellwertbildung in (helle) Kugeln und (dunklen) Untergrund getrennt werden soll. Würde diese Trennung ohne vorherige Shading-Korrektur ausgeführt, bestünde die Möglichkeit, daß die absoluten Helligkeitswerte einer im Schatten liegenden Kugel geringer sind, als die einer besser beleuchteten Stelle des Spielfelduntergrundes.

Zunächst wird eine sogenannte Shading-Maske benötigt. Dabei handelt es sich um ein in der Größe dem Originalbild entsprechendes Bild, das die globale Helligkeitsverteilung darstellt. Das heißt, daß die Helligkeit jedes einzelnen Pixels der Maske von der in seiner weiteren Umgebung vorhandenen Helligkeit bestimmt wird. Dadurch gehen im Vergleich zum Originalbild kleinere und örtlich begrenzte Helligkeitsunterschiede verloren (hohe Ortsfrequenzen werden unterdrückt).

Zur Erstellung der Shading-Maske wird das Verfahren des gleitenden Minimums genutzt. Dabei ist zunächst die mittlere Helligkeit des gesamten Bildes zu bestimmen (Durchschnittlicher Wert aller Pixel des Bildes). Anschließend wird das Folgende getan:

Für jede Zeile des Bildes:

Setze die Variable *Shade* gleich der mittleren Helligkeit des Gesamtbildes.

Bewege dich von links nach rechts über die gesamte Zeile und

teste für jeden Punkt:

Ist der Punkt heller als *Shade*, dann erhöhe *Shade*,

ist er dunkler, dann verringere *Shade*

Setze den entsprechenden Punkt in der Shading-Maske gleich *Shade*.

Die so entstehende Shading-Maske ist in Abb. 2.7 zu sehen. Die entstehenden "Grauwert-Fahnen" werden dadurch kompensiert, daß das Verfahren auch in den drei anderen Richtungen (recht-links, oben-unten und unten-oben) angewandt und das Minimum aus diesen vier Werten in die endgültige Shading-Maske eingetragen wird. Diese ist in Abb. 2.8 dargestellt.

Schließlich sind das eigentliche Bild und die Shading-Maske miteinander zu verknüpfen, was im konkreten Fall durch eine Mittelwertbildung zwischen beiden für jedes Pixel geschieht.

Abb. 2.9 zeigt das korrigierte Bild.

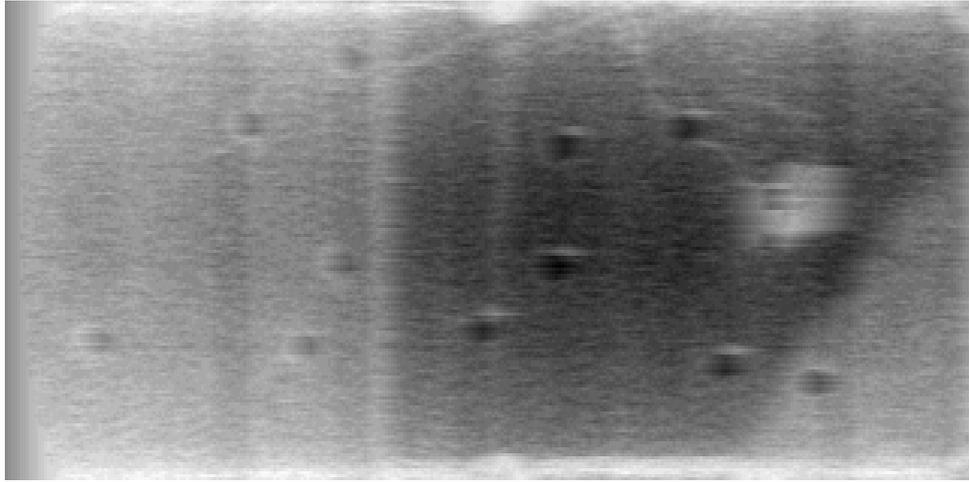


Abb. 2.7 Zwischenstufe der Shadingmaske nach Anwendung des Algorithmus in Richtung links→rechts
Es ist zu erkennen, daß die Maske bis zu einem gewissen Grade der Invertierung des zugrundeliegenden Originalbildes entspricht.

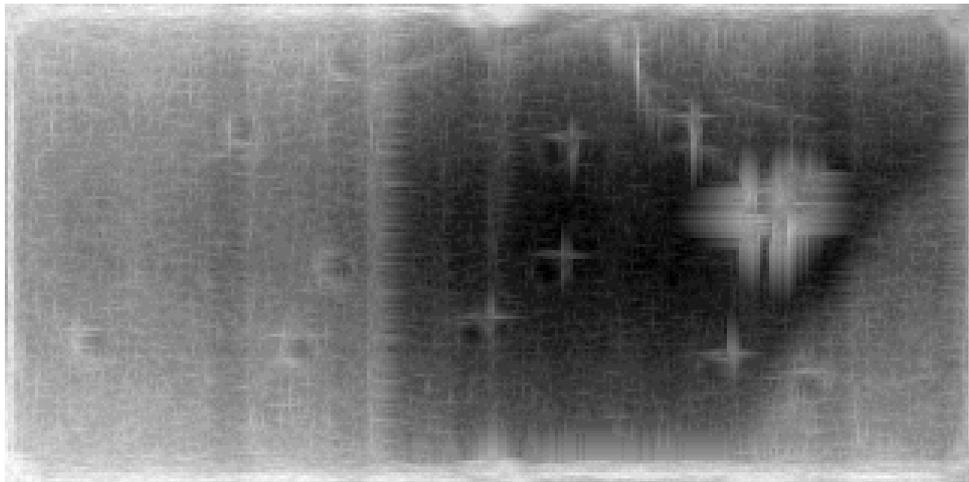


Abb. 2.8 endgültige Shadingmaske

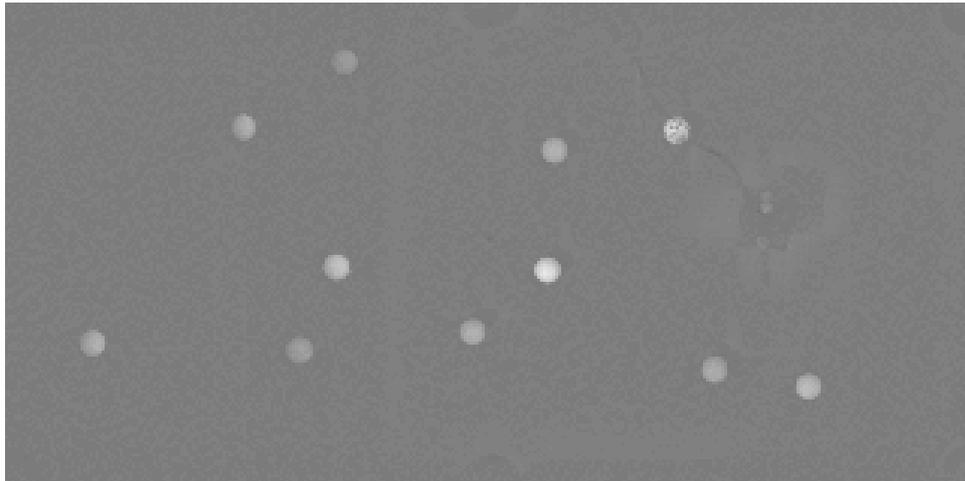


Abb. 2.9 Der Rotkanal nach erfolgter Shading-Korrektur

Die für den menschlichen Betrachter dramatische Verschlechterung der Bildqualität hat für die Bilderkennung den gegenteiligen Effekt. Der gesamte Hintergrund (bzw. Untergrund) weist kaum noch Helligkeitsdifferenzen auf und auch die Helligkeitswerte der Kugeln haben sich einander stärker angenähert.

2.3.3 Binarisieren

Der nächste Verarbeitungsschritt besteht in der Trennung der Pixel, welche zu den Kugeln gehören, von denen, die dem Spielfeld zuzuordnen sind. Das Ziel ist ein reines Schwarz-Weiß-Bild mit weißen Kugeln und schwarzem Untergrund.

Eine derartige Trennung wird meist auf der Basis eines Histogramms des Bildes vorgenommen. Erläuterungen zu vielfältigen derartigen Operationen mit Grauwert-Histogrammen finden sich z.B. in [Voss91]. Ein Grauwert-Histogramm ist eine Tabelle, in der jeder darstellbaren Graustufe bzw. Helligkeit (meist 0..255) die Anzahl der Bildpunkte zugeordnet ist, welche diese Helligkeit besitzen. Die räumliche Lage der Punkte im Bild ist dabei nicht von Interesse. Das Histogramm des Beispielbildes (Abb. 2.6) ist in Abb 2.10 dargestellt.

Abb. 2.11 zeigt das Histogramm des Bildes (Abb. 2.9) nach der Shading-Korrektur.

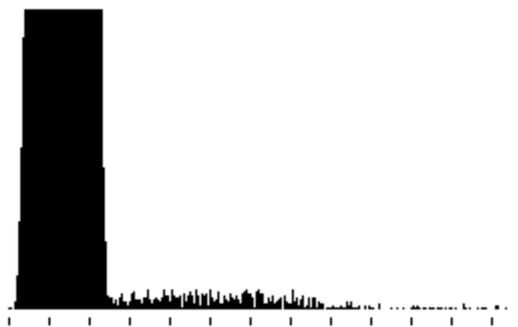


Abb. 2.10 Histogramm des Rotkanals des Ausgangsbildes
Skalierung von 0 (links) bis 255 (rechts), dargestellt ist nur der untere Abschnitt

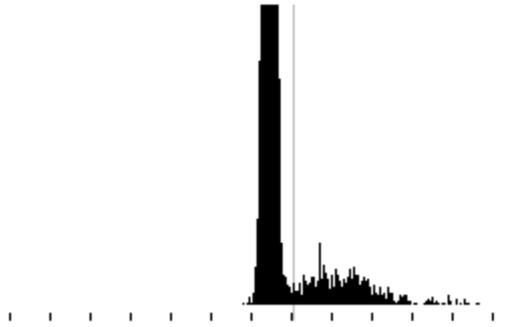


Abb. 2.11 Histogramm des Rotkanals nach der Shading-Korrektur

Skalierung von 0 (links) bis 255 (rechts), dargestellt ist nur der untere Abschnitt

Die graue Linie markiert den bei der späteren Binarisierung verwendeten Schwellwert (Eräuterungen folgen im Text).

Das eigentliche Problem beim Binarisieren besteht darin, einen geeigneten Schwellwert zu bestimmen, der alle helleren Graustufen dem reinen Weiß und alle anderen dem reinen Schwarz zuordnet. Ideal sind diejenigen Histogramme, welche einen deutlichen bimodalen Charakter besitzen, d.h., in denen lediglich zwei lokale Maxima der Verteilung auftreten.

In diesem Fall kann der Schwellwert im lokalen Minimum zwischen den beiden Maxima festgelegt werden (vgl. Abb. 2.12).

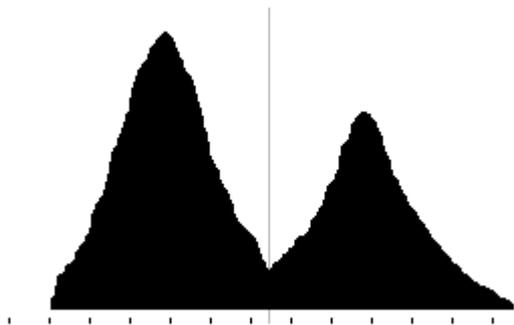


Abb. 2.12 Beispiel eines bimodalen Histogramms

Skalierung von 0 (links) bis 255 (rechts)

Die graue Linie markiert die offensichtliche Trennung der beiden Gebiete.

Im realen Histogramm wird zwar der menschliche Betrachter mühelos die Bimodalität wahrnehmen, für eine einfache algorithmische Suche nach dem lokalen Minimum weist es jedoch noch zu viele kleinere "Peaks" auf. Es erscheint also sinnvoll, daß Histogramm zunächst durch Mittelwertbildung zu glätten. Dabei entsteht allerdings die Frage, wie oft und mit welcher seitlichen Einflußspanne geglättet werden soll. Außerdem verzerrt die Glättung das Histogramm u.U., so daß das Minimum des geglätteten relativ weit von dem des Original-Histogramms entfernt sein kann. Aus diesen Gründen wird im vorliegenden Fall auf eine Glättung verzichtet.

Der eingesetzte Algorithmus ermittelt zunächst die Stelle im Histogramm, an welcher die stärkste Verringerung der Anzahl von zugeordneten Bildpunkten zu verzeichnen ist (in Richtung der aufsteigenden Grauwerte gesehen). Diese Stelle befindet sich aufgrund der großen Anzahl von Hintergrundpunkten stets an der im Bild des Histogramms rechten Flanke des linken Maximums (vgl. Abb 2.11). Von hier aus wird weiter "abgestiegen" bis der nächste Anstieg der Histogrammeinträge auftritt. Der nun erreichte Intensitätswert wird zur Schwelle bestimmt.

Abb. 2.13 zeigt das daraufhin binarisierte Bild.

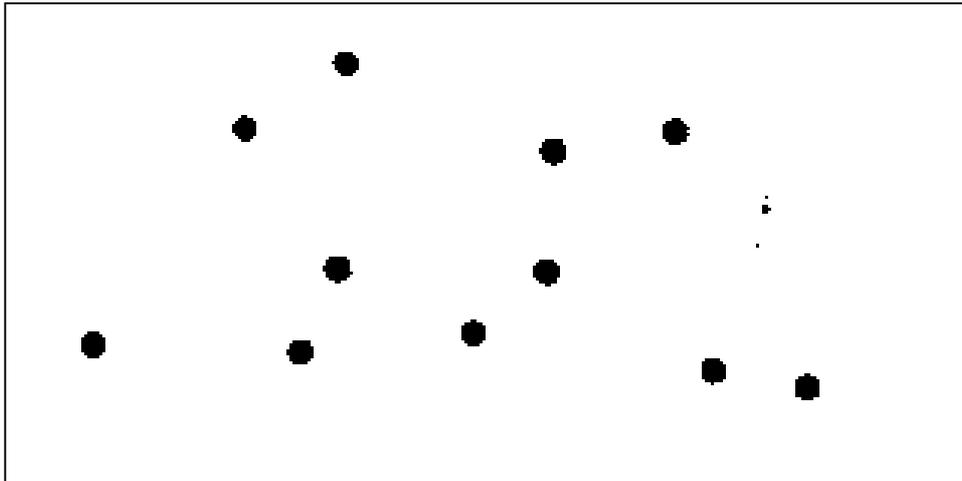


Abb. 2.13 Binarisiertes Bild von Kugeln und Hintergrund

Das Bild wurde zur besseren Darstellung invertiert. Rechts sind einige Artefakte zu erkennen, welche vom Fahrzeug herrühren.

2.3.4 Erosion der Kugelgebiete

Für den im Anschluß beschriebenen letzten Schritt der Koordinatenermittlung sind nur solche Bilder geeignet, in denen die Kugeln deutlich voneinander getrennt sind, also keine "Brücken" zwischen Kugelgebieten bestehen. Ein diesbezüglicher Extremfall ist in Abb. 2.14 dargestellt.

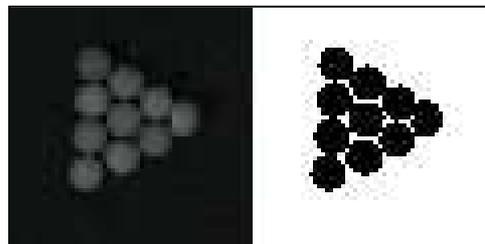


Abb. 2.14 Bildung von "Brücken" bei der Binarisierung

Dieses Beispiel zeigt die dreiecksförmige Anordnung der Kugeln zu Beginn einer Billard-Partie. Aufgrund der relativ geringen Bildauflösung ist eine deutliche Trennung der einzelnen weißen Kugelgebiete nicht möglich. Es ist also notwendig, diese Brücken zu eliminieren. Das dafür gewählte Verfahren wird als Erosion bezeichnet und ist dem Gebiet der morphologischen Bildoperationen zuzuordnen. Diese werden z.B. in [Klette95] eingehend behandelt.

Ähnlich einem Zellularautomaten wird für jeden Bildpunkt die Konfiguration der ihn umgebenden Nachbarpunkte untersucht, um anschließend einen neuen Zustand für den Bildpunkt festzulegen. Die möglichen Zustände sind im vorliegenden Fall "weiß" oder "schwarz".

Die Operation hat die folgende Gestalt:

Teste für jeden weißen Punkt $[x, y]$,
ob seine 5×5 Umgebung (also $[x-2..x+2, y-2..y+2]$)
ausschließlich aus weißen Punkten besteht.
Wenn ja, dann bleibt der Punkt weiß,
sonst setze den Punkt auf den Zustand "schwarz".

Diese Operation bewirkt eine Schrumpfung der weißen Gebiete, wodurch die Brücken eliminiert werden. Die für die Mittelpunkttermittlung entscheidende Information geht nicht verloren, da die Erosion die Gebiete relativ gleichmäßig verkleinert. Das Ergebnis für den oben genannten "worst case" ist in Abb. 2.15 zu sehen, wogegen Abb. 2.16 die bereits bekannte Beispiel-Situation nach der Erosion zeigt.

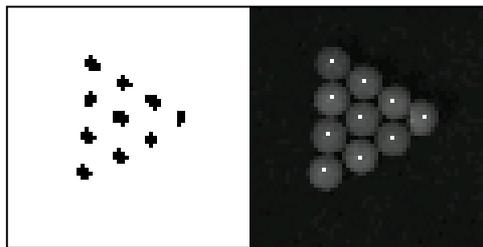


Abb. 2.15 Der Extremfall aus Abb. 2.14 nach der Erosion

Links das Ergebnis der Erosion (Darstellung invertiert). Rechts dargestellt sind die später ermittelten Kugelmittelpunkte (Erläuterungen folgen im Text).

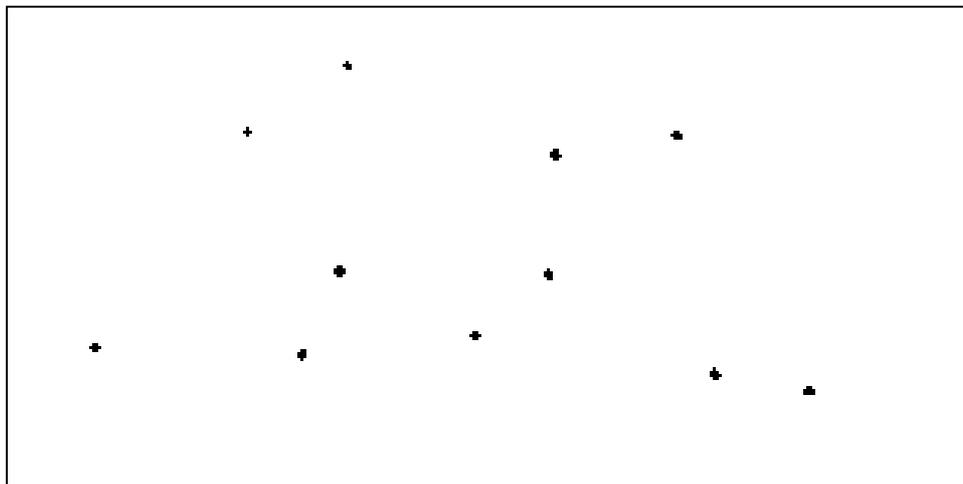


Abb. 2.16 Ergebnis der Erosion

Die Darstellung ist wiederum invertiert. Die Artefakte aus Abb. 2.13 wurden eliminiert.

2.3.5 Ermittlung der groben Mittelpunktkoordinaten der Kugeln

Für jedes weiße Gebiet im Gesamtbild werden dessen minimale und maximale Ausdehnung in x - und y -Richtung bestimmt und daraus die Mittelwerte gebildet. Jedes der dadurch erhaltenen Koordinatenpaare $[x,y]$ stellt den Mittelpunkt einer Kugel dar. (Exakt handelt es sich um den Mittelpunkt des durch senkrechte Projektion der Kugel auf die Spielfläche entstehenden Kreises). Abbildung F 2.17 zeigt nochmals das Ausgangsbild, diesmal mit hervorgehobenen (groben) Kugelmittelpunkten.

Die relativ geringe Genauigkeit dieser Werte ist zum Ausführen des endgültigen Stosses nicht

ausreichend, genügt aber für die im Stoßzyklus folgenden Phasen der Stoßwahl und der Wegplanung vollauf.

2.4 Bestimmung der Kugelfarben

Nachdem nun bekannt ist, wo Kugeln liegen, ist es erforderlich, deren Farben zu bestimmen, da diese für die später folgende Stoßwahl ausschlaggebend sind. Zuzuordnen sind die Farben weiß (1 mal) sowie rot und gelb (jeweils 0.5 mal). Es wird davon ausgegangen, daß sich mindestens die weiße Kugel und ein Vertreter der als nächstes anzupielenden Kugeln auf dem Spielfeld befinden.

Die Farbenbestimmung bezieht alle drei Farbkanäle des aufgenommenen Überblick-Bildes ein. Die Werte des Rot-, Grün- und Blau-Anteils der einzelnen Pixel spannen einen dreidimensionalen Farbraum $[0..255,0..255,0..255]$ auf. Im Idealfall wäre eine einzelne Kugel darin als genau ein Punkt repräsentiert, vorausgesetzt alle ihr zugehörigen Pixel hätten die gleiche spektrale Zusammensetzung. Die Realität weicht davon natürlich erheblich ab. Dies resultiert zum einen aus der Kugelform, die eine Schattierung nach sich zieht und aus einem verhältnismäßig starken Rauschen in den Aufnahmeelementen der Kamera. Somit bilden sich im Farbraum mehr oder weniger kompakte Cluster von Punkten aus. Für die Analyse derartiger Cluster existiert eine Vielzahl von Algorithmen, welche einen Repräsentanten, also die "durchschnittliche" Farbe des Clusters bestimmen (z.B. Kohonen-Netze [Ritter91]).

Bei der Suche nach einer geeigneten Möglichkeit der Farbermittlung zeigte sich aber, daß das Problem auch mit relativ geringem Aufwand zu lösen ist.

Das zur Unterscheidung der Kugelfarben angewandte Verfahren soll anhand folgender Tabelle veranschaulicht werden:

K	R	G	B	q1	q2	q3	d	q4	F
1	255	246	253	1.04	1.01	0.97	0.07	1.07	W
2	87	29	29	3.00	3.00	1.00	4.00	3.00	R
3	90	32	34	2.81	2.65	0.94	3.52	2.99	R
4	70	24	24	2.92	2.92	1.00	3.83	2.92	R
5	115	35	38	3.29	3.03	0.92	4.39	3.57	R
6	90	28	30	3.21	3.00	0.93	4.28	3.44	R
7	138	69	43	2.00	3.21	1.60	3.81	1.25	G
8	247	146	98	1.69	2.52	1.49	2.70	1.14	G
9	222	114	80	1.95	2.78	1.42	3.15	1.37	G
10	224	113	79	1.98	2.84	1.43	3.25	1.39	G
11	240	126	84	1.90	2.86	1.50	3.26	1.27	G

Die einzelnen Spalten haben folgende Bedeutung:

K	laufende Nummer der erkannten Kugeln (Die Tabelle ist bereits geordnet.)
R	durchschnittlicher Wert des Rotkanals für die Pixel der 5x5-Umgebung um den (groben) Kugelmittelpunkt
G, B	analog für Grün- bzw. Blaukanal
q1	R/G

q2	R/B
q3	G/B
d	$ (q1-1) + (q2-1) + (q3-1) $
q4	$q1/q3 = R \cdot B / G^2$
F	ermittelte Farbe (<u>W</u> eiß, <u>R</u> ot, <u>G</u> elb)

Die Quotienten q1, q2 und q3 dienen dazu, relative Werte für die spektrale Zusammensetzung der Pixel zur Verfügung zu stellen. Eine Untersuchung der absoluten Werten der Kanäle würde eine gleichmäßige Helligkeit im gesamten Bild voraussetzen, die nicht gegeben ist.

Der Wert d dient der Identifizierung der weißen Kugel. Da im (reinen) Weiß alle Farben des Spektrums gleich stark vertreten sind, sollten die Quotienten q1..q3 gleich 1 sein.

Der Parameter d ist ein einfaches Abstandsmaß, welches die Abweichung von diesem Ideal bestimmt. Die Kugel mit dem minimalen Wert d wird als weiße Kugel identifiziert.

Zur Unterscheidung zwischen roten und gelben Kugeln wird der Quotient q4 herangezogen. Aus der Tabelle ist ersichtlich, daß sich charakteristische Werte für q4 ergeben, welche sich für rote Kugeln zwischen ca. 2,8 und 4,0 und für gelbe zwischen 1,0 und 2,2 bewegen. Als Schwelle wurde nach eingehender empirischer Untersuchung der Wert 2,5 gewählt.

Voraussetzung für den Erfolg dieses einfachen Algorithmus ist eine vorausgegangene Einstellung der Aufnahme-Helligkeit der Kamera. Dabei muß eine Übersteuerung ausgeschlossen werden, d.h., die maximal mögliche Intensität pro Farbkanal (255) darf nicht erreicht bzw. überschritten werden. Ist dies garantiert und die Umgebungshelligkeit während des Betriebs des Billardsystems relativ konstant, so arbeitet die Farberkennung sehr zuverlässig.

2.5 Genaue Positionsbestimmung von Kugeln (Feinerkennung)

Die in diesem Abschnitt beschriebenen Verarbeitungsschritte kommen zur Anwendung, bevor das Roboterfahrzeug vor dem eigentlichen Stoß exakt ausgerichtet wird (Punkt 5 des Stoßzyklus). Zur möglichst genauen Bestimmung der optimalen Stoßposition und des Stoßwinkels wird die Lage der am folgenden Stoß beteiligten zwei Kugeln (weiße Spielkugel und zu versenkende Kugel) nochmals bestimmt.

Wie bereits dargelegt wurde, ist die Genauigkeit der bis dahin nur grob ermittelten Koordinaten der Kugeln aufgrund der geringen Auflösung der Bilder relativ gering. Deshalb werden die beiden am bevorstehenden Stoß beteiligten Kugeln (weiße Spielkugel und anzuspielende Kugel) nochmals mit der doppelten Bildauflösung (320×240 Px, ca.19×19 Px Kugelgröße) aufgenommen.

Prinzipiell wäre es auch möglich, von vornherein diese Genauigkeit für alle Algorithmen der Bilderkennung zu verwenden. Dies würde aber eine mindestens vierfache Zeit- und Platzkomplexität nach sich ziehen und wäre damit unnötig ineffizient.

Die Kugel-Feinerkennung besteht aus den folgenden, im Anschluß erläuterten Schritten:

1. Aufnahme des Bildausschnittes
2. Kantendetektion mit Sobel-Kompaß-Operator
3. Einsatz eines "Kreisfilters" zur Mittelpunktsfindung
4. Korrektur des perspektivischen Fehlers

2.5.1 Aufnahme des Bildausschnittes

Entsprechend der bekannten groben Koordinaten wird die Kamera an die Position (von 8 möglichen) bewegt, welche die Kugel möglichst nah der Mitte (bzgl. der x-Koordinate) des aufzunehmenden Rohbildes abbildet. Das Rohbild (240×320 Px) wird aufgenommen und ein Bild mit einer Größe von 30×30 Px um die Grobkoordinaten ausgeschnitten. Aufgrund der kleinen Abmessungen dieses Ausschnittes können bei der Feinerkennung bedeutend zeitaufwendigere, aber bessere Algorithmen als bei der Groberkennung zum Einsatz kommen.

Als Vorverarbeitung wird der bereits beschriebene "Christmas-Filter" auf allen drei Kanälen angewandt, vor allem um eine bessere optische Darstellung im Steuerprogramm zu erzielen. Die weiteren Verarbeitungsschritte werden, wie schon bei der Groberkennung, ausschließlich auf dem Rot-Kanal des Bildes ausgeführt.

2.5.2 Kantendetektion mit Sobel-Kompaß-Operator

Die nächste Aufgabe besteht darin, die Umrisse der zu erkennenden Kugel zu identifizieren, da mit deren Hilfe auf die Lage des Kugelmittelpunktes geschlossen werden kann. Das Auffinden von Objektkanten in Bildern ist eine der in der Bilderkennung am häufigsten auftretenden Problemstellungen. Daher existiert eine Vielzahl von möglichen Lösungswegen, welche auf verschiedenen Bildverarbeitungs-Operatoren beruhen ([Klette95]).

Eine sehr weit verbreitete Methode ist die Anwendung eines sogenannten **Sobel-Operators**. Dabei werden für jeden Bildpunkt die Helligkeitswerte in seiner 3×3-Umgebung entsprechend einer bestimmten Matrix gewichtet und summiert. Die Koeffizienten der Matrix sind so gewählt, daß die Summe für Bildpunkte, welche an Kanten, also starken räumlichen Grauwertänderungen, liegen, sehr hohe Werte annimmt. Dagegen führen gleichmäßige Helligkeiten innerhalb der 3×3-Umgebung zu niedrigen Summen.

Die Verwendung einer einzelnen Sobel-Matrix führt immer zur Detektion der Kanten, welche in einer bestimmten Richtung verlaufen (z.B. vertikal). Um generell Kanten in beliebigen Richtungen zu finden, sind mehrere Sobel-Operatoren, d.h., verschiedene Koeffizienten-Matrizen, auf das zu untersuchende Bild anzuwenden. Aus den Ergebnissen wird für jeden Bildpunkt das Maximum dieser Einzelergebnisse ermittelt, um seine Zugehörigkeit zu einer mehr oder weniger scharfen Kante festzustellen. Durch die Untersuchung verschiedener Richtungen von möglichen Kanten wird diese Vorgehensweise auch als **Sobel-Kompaß-Operator** bezeichnet.

Die für die Kugel-Feinerkennung verwendeten Matrizen haben die folgende Gestalt:

vertikale Kanten

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

horizontale Kanten

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

diagonale Kanten (links unten → rechts oben)

diagonale Kanten (links oben → rechts unten)

$$\begin{bmatrix} 0 & -1 & -2 \\ +1 & 0 & -1 \\ +2 & +1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & +1 \\ 0 & +1 & +2 \end{bmatrix}$$

In Abbildung 2.18 ist die Wirkung des Sobel-Kompaß-Operators innerhalb der Kugel-Feinerkennung dargestellt.

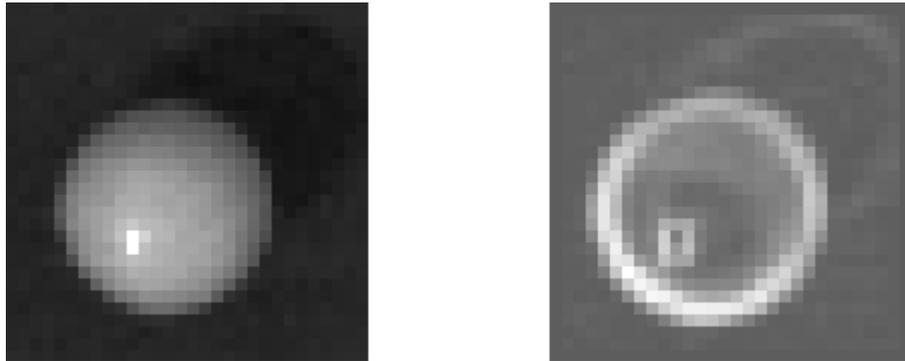


Abb. 2.18 Rotkanal-Bild einer einzelnen Kugel und Ergebnis des Sobel-Kompaß-Operators

Durch einen stärkeren Lichteinfall von links unten wird in diesem Gebiet auch eine stärkere Kante detektiert. Zu erkennen sind außerdem die Auswirkungen eines Glanzlichtes und des Schattenwurfes der Kugel.

2.5.3 Einsatz des "Kreisfilters" zur Mittelpunktfindung

Der abschließende Schritt der Feinerkennung nutzt die Tatsache aus, daß der Durchmesser der Kugeln im Bild (ca. 19 Pixel) bekannt ist, und läuft in folgender Art und Weise ab:

- Für alle Punkte in einer 21×21 Umgebung um den bekannten groben Mittelpunkt:
- Betrachte den aktuellen Punkt als Mittelpunkt eines Kreises mit 9 Px Radius.
- Summiere die Werte der Pixel, welche auf dem gedachten Kreis liegen, in Schritten von 10 Grad und normalisiere die Summe (Division durch 36).
- Trage den erhaltenen Wert in ein neues Bild ein.

Die für jeden Punkt ermittelte Summe ist um so größer, je mehr Punkte, welche auf dem Kugelrand liegen, "eingesammelt" wurden. Aus dem entstandenen Bild wird der Punkt mit dem maximalen Wert ermittelt und zum Mittelpunkt der Kugel bestimmt (Abb. 2.19).

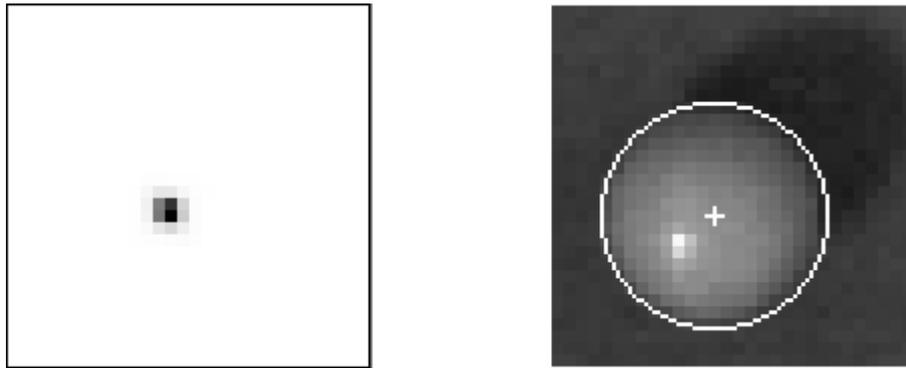


Abb. 2.19 Ergebnis des Kreisfilters und gefundener Kugelmittelpunkt für das Beispiel von Abb. 2.18

Aus drucktechnischen Gründen wurde das linke Bild invertiert, d.h., je dunkler der Punkt, desto größer die Summe. Das rechte Bild ist ein in Graustufen umgewandeltes Farbbild der im Original roten Kugel, ergänzt um den gefundenen Mittelpunkt und einen nachgezeichneten Kugelrand.

2.5.4 Korrektur des perspektivischen Fehlers

Die nunmehr ermittelten Koordinaten sind noch mit einem u.U. erheblichen Fehler behaftet, welcher auf die perspektivische Verzerrung des von der Kamera aufgenommenen Bildes zurückzuführen ist. Dieser resultiert daraus, daß das Bild nicht wirklich eine senkrechte Projektion auf die Spielfläche darstellt. Abb. 2.20 soll diese Zusammenhänge verdeutlichen.

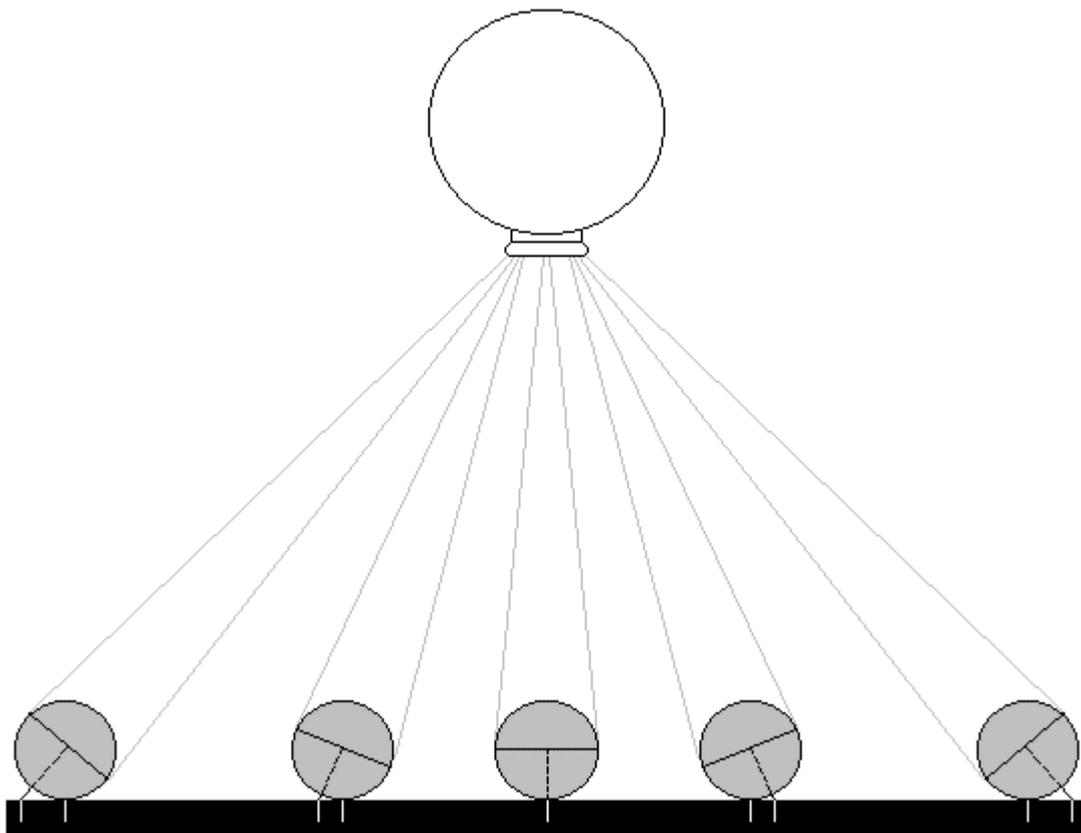


Abb. 2.20 perspektivischer Abbildungsfehler

Die weißen Linien markieren die Abweichungen der wirklichen von den im Bild erscheinenden Kugelmittelpunkte. Nur die genau senkrecht unter der Kamera liegende Kugel wird korrekt abgebildet.

Der Abbildungsfehler führt dazu, daß die Kugelmittelpunkte im Bild nicht an der Stelle erscheinen, an der sie sich real befinden. Je weiter die Kugel vom Lotpunkt der Kameraachse entfernt liegt, desto stärker wirkt sich die Verzerrung aus.

Um die groben Auswirkungen des Effektes zu kompensieren, wurden an regelmäßig über das Spielfeld verteilten Positionen Kugeln plziert und ihre Mittelpunktkoordinaten mit dem bisher beschriebenen Verfahren ermittelt, um sie anschließend einer Matrix der realen Positionen zuzuordnen. Auf diese Weise werden auch die bereits beschriebenen Verzeichnungseffekte des Kameraobjektives zusätzlich unterdrückt. Dies ist auch der Grund dafür, die Perspektiven-Korrektur nicht auf rein rechnerischem Wege durchzuführen.

2.6 Optische Erkennung des Roboterfahrzeuges

Im Vergleich zu den geometrisch einfach gestalteten Kugeln wäre der Aufwand zur Erkennung des Fahrzeuges und seiner Ausrichtung sehr hoch, wenn die Algorithmen sich ausschließlich an dessen Umrissen orientierten. Der zu erwartende hohe Zeitaufwand würde sich gravierend auf die Geschwindigkeit der Fahrzeug-Fortbewegung auswirken.

Deshalb ist eine andere Lösung zum schnellen Auffinden des Fahrzeuges und zur Bestimmung seiner räumlichen Ausrichtung notwendig. Um das Fahrzeug mit geringem Aufwand in einem Rohbild identifizieren, ist es von Vorteil, wenn es über einfach strukturierte, aber unverwechselbare optische Charakteristika verfügt. Im vorliegenden Fall werden dazu verschiedene Markierungen genutzt, welche auf der Oberseite des Fahrzeuges angebracht sind (Abb. 2.21).

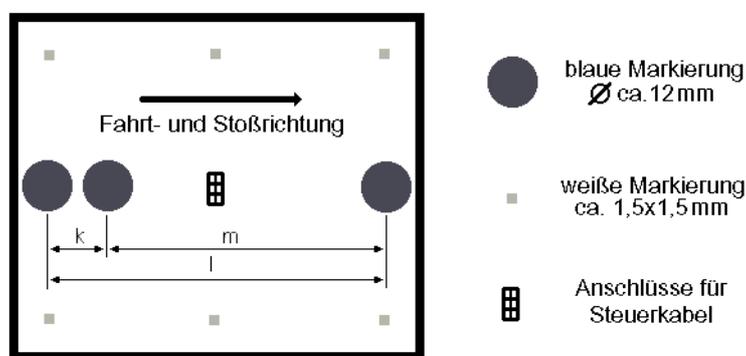


Abb. 2.21 Schematische Darstellung der Fahrzeug-Oberseite

Die Oberseite des Roboterfahrzeuges ist in Wirklichkeit schwarz. Alle weiteren Erläuterungen folgen im Text.

Die Anforderungen an die Genauigkeit der ermittelten Positionen variieren, je nachdem, ob sich das Fahrzeug auf einem (durch die Wegplanung ermittelten) relativ breiten Korridor zwischen den Kugeln hindurch bewegt oder möglichst präzise vor der Spielkugel ausgerichtet werden soll. Ähnlich der Grob- und Feinerkennung bei den Billardkugeln sind deshalb zwei unterschiedliche Varianten (schnell/relativ ungenau und langsam/genau) zu implementieren.

2.6.1 Schnellere Variante

Für diese Variante der Fahrzeugerkennung werden die drei blauen, kreisförmigen Markierungen genutzt. Diese sind entlang der Längsachse des Fahrzeuges angeordnet und erscheinen bei der hier verwendeten Rohbild-Auflösung von 120×160 Px mit einem "Durchmesser" von ca. 4 Px. Die mittlere Markierung, welche einer der beiden äußeren deutlich näher liegt als der anderen, dient zur Identifikation des Fahrzeughecks und somit der Fahrt- und Stoßrichtung. Die Farbe Blau führt zu einer von vornherein garantierten Eindeutigkeit der Markierungen im optischen Erscheinungsbild der gesamten Spielfläche.

Unter der Voraussetzung, daß sich das Fahrzeug im aktuellen Erfassungsbereich der Kamera befindet, setzt sich die schnelle Fahrzeugerkennung aus den folgenden Teilschritten zusammen:

1. Aufnahme von Blau- und Rot-Kanal und Differenzbildung zur Kontrastverstärkung
2. Auffinden der drei hellsten Stellen (nicht Pixel!) im Differenzbild
3. Zuordnung der ermittelten Punkte zu den drei Markierungen

zu Punkt 1:

Würde sich der Erkennungsalgorithmus zum Auffinden der blauen Markierungen allein auf den Blau-Kanal des Bildes beschränken, so wären die drei Punkte auf dem Fahrzeug mit hoher Wahrscheinlichkeit nicht immer die hellsten Stellen im Bild. da z.B. im Fall der weißen Kugel alle drei RGB-Kanäle etwa die gleiche, hohe Intensität haben.

Aus diesem Grund wird nach Pixeln gesucht, deren Blau-Anteil deutlich höher als ihr Rot-Anteil ist. Dazu wird das Rot-Bild vom Blau-Bild subtrahiert, wobei negative Werte durch 0 ersetzt werden. Im entstehenden Differenzbild heben sich die Markierungen deutlich ab.

Als zusätzliche Verbesserung dieses Bildes im Sinne der vorliegenden Aufgabe wird es einer Mittelwertberechnung für jedes Pixel (3×3-Umgebung) unterzogen. dadurch verschieben sich die Maxima der Helligkeitsintensitäten weiter in Richtung der Mittelpunkte der ca. 4×4 Px großen Markierungen. Außerdem werden durch Rauschen entstandene "stark blaue" Pixel im gesamten Bild unterdrückt.

Die Ergebnisse dieser Vorgehensweise sind in Abb. 2.22 dargestellt.

zu Punkt 2:

Die nun folgende Aufgabe besteht darin, aus dem Differenzbild die Koordinaten der Markierungen zu ermitteln. Dabei wird davon ausgegangen, daß die drei Stellen mit der höchsten Helligkeitsintensität mit den Markierungen identisch sind. Diese Annahme hat sich nach intensiver Untersuchung als zulässig herausgestellt.

Die folgende Prozedur wird dreimal hintereinander auf das Differenzbild angewandt:

Ermittle den Punkt des Bildes, welcher die maximale Intensität besitzt.

Speichere diesen als Koordinaten von einer der Markierungen.

Setze den Punkt und dessen 5×5-Umgebung auf die Intensität 0.

Durch das "Schwarzmalen" der Umgebung der jeweils gefundenen Maxima wird vermieden, daß ein in deren unmittelbarer Nähe liegendes Pixel das Rennen um die zweit- bzw. dritthöchste Intensität für sich entscheiden kann. Mathematisch betrachtet werden die drei höchsten lokalen Maxima ermittelt.

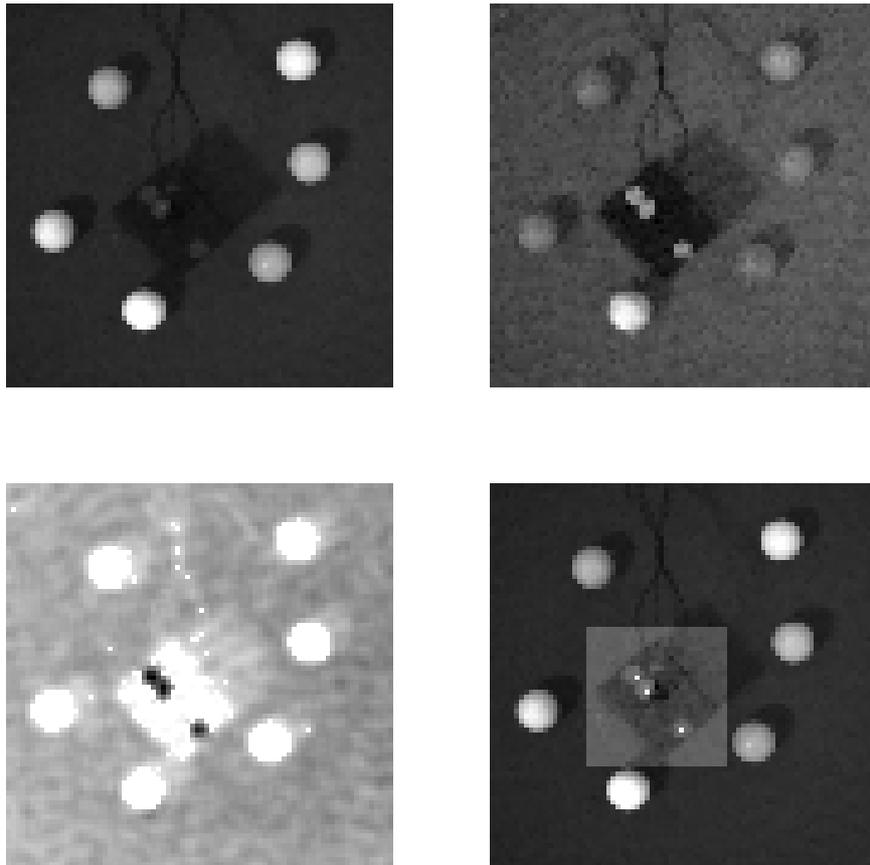


Abb. 2.22 Differenzbildung zum Auffinden der Fahrzeug-Markierungen

Beim Rot-Kanal (oben links) und Blau-Kanal (oben rechts) wurde eine Grauwertspreizung durchgeführt, d.h., die im Bild vorhandenen Grauwerte wurden so "gestreckt", daß sie das mögliche Intervall von 0..255 voll ausnutzen. Im Blaukanal erscheint die weiße Kugel (unteres Bildviertel) heller als die drei Markierungen. Der unruhige Gesamteindruck des Bildes ist technisch bedingt.

Das entstehende Differenzbild (links unten) wurde zur besseren Darstellung invertiert, schwarze Stellen bedeuten hohe Differenzen. Die Wirkung der weißen, wie auch der anderen Kugeln ist völlig ausgeschaltet und die drei Markierungen heben sich deutlich ab. Rechts unten ist wiederum der Rot-Kanal, diesmal mit den gefundenen "Mittelpunkten" der Markierungen zu sehen (Erläuterungen im Text). Zur besseren Erkennung der Fahrzeug-Umrisse wurde der Kontrast des betreffenden Bildausschnittes erhöht. Deutlich sichtbar ist, daß die drei gefundenen Koordinaten nicht in jedem Fall wirklich in der Mitte der Markierungen liegen. Für die gestellten Anforderungen ist diese Genauigkeit aber ausreichend.

(Die sechs weißen Markierungen waren zum Zeitpunkt der Aufnahme dieser Bilder noch nicht vorhanden.)

zu Punkt 3:

Nunmehr liegen drei Koordinatenpaare vor, die mit hoher Wahrscheinlichkeit den drei Markierungen auf dem Fahrzeug entsprechen. Aufgrund der bereits beschriebenen Anordnung der Markierungen ergeben sich drei charakteristische Abstände zwischen jeweils zwei der Koordinaten. Diese sollen als l , m und k ("längster", "mittlerer" und "kürzester" Abstand) bezeichnet werden (siehe Abb. 2.21). Eine einfache Fallunterscheidung ordnet die ermittelten Koordinaten den drei Markierungen zu.

Als zusätzliche Sicherheitsmaßnahme wird der Abstand l zuvor darauf geprüft, ob er eventuell einen Wert hat, welcher viel größer ist als der wirkliche, bekannte Abstand zwischen den zwei äußeren Markierungen. Sollte dies der Fall sein, wird die gesamte Fahrzeugerkennung erneut durchgeführt. Der praktische Einsatz des Algorithmus hat gezeigt, daß derartige Fälle nur durch Rauschen im Bild verursacht werden. Da sich dessen Auswirkungen von einer Aufnahme zur nächsten ändern, ist gewährleistet, daß die Erkennung spätestens nach einigen

wenigen Versuchen sicher zum Erfolg führt. Die Implementierung weiterer Überprüfungen der drei Koordinatenpaare, wie z.B. ein Test ihrer Kolinearität, hat sich als nicht notwendig erwiesen.

An die aufrufende Prozedur in der Steuersoftware zurückgegeben werden die Durchschnittswerte der x- bzw. y-Koordinaten der beiden äußeren Markierungen (als Fahrzeug-Mittelpunkt) und der Winkel der Gerade, welche durch die beiden äußeren Markierungen verläuft (Orientierung des Fahrzeuges). Alle mit der Fahrzeugposition in Zusammenhang stehenden Winkelangaben beziehen sich auf ein gedachtes Koordinatensystem, dessen Ursprung sich in der linken, unteren Ecke des Bildes der gesamten Spielfläche befindet (0° = "nach rechts", 90° = "nach oben" usw.).

2.6.2 Genauere Variante

Die Ausrichtung des Fahrzeuges vor der Spielkugel erfordert eine höchstmögliche Präzision der Positionsermittlung. Dafür werden, wie schon bei der zuvor beschriebenen schnellen Erkennung, Markierungen genutzt. In diesem Falle wird mit den 6 weißen Quadraten gemäß Abb. 2.21 gearbeitet.

Diese erscheinen im Bild (240×320Px Auflösung) in einer Größe von etwa einem Pixel. Dadurch wird eine maximale Genauigkeit bei der Zuordnung der Bildpunkte zu den wirklichen Markierungen erreicht.

Die Fahrzeug-Feinerkennung setzt sich aus den folgenden Schritten zusammen:

1. Grobe Ermittlung von Position und Ausrichtung (Winkel) des Fahrzeuges mittels der schnellen Erkennungs-Variante
2. Aufnahme eines RGB-Bildes (240×320Px) und Ausschneiden eines Bereiches von 65×65Px um den groben Fahrzeugmittelpunkt
3. Erstellung eines Graustufenbildes des Ausschnittes durch Berechnung der durchschnittlichen Intensität aller drei Farbkanäle in jedem Pixel (In diesem Bild erhalten die weißen Markierungen einen sehr hohen Wert und eventuell vorhandene "Christmas-Lights" werden unterdrückt, da sie stets nur in einem der drei Farbkanäle auftreten.)
4. Rotieren der bekannten wirklichen Koordinaten der weißen Markierungen um den grob ermittelten Winkel (Somit existieren 6 Punkte, in deren näherer Umgebung jeweils eine der aufgenommenen Markierungen im Bild zu finden sein müßte.)
5. Ermittlung des jeweils hellsten Bildpunktes in der 7×7-Umgebung eines jeden dieser 6 Punkte.
6. Diese 6 neuen Punkte werden nun, wie auch die Kugeln bei der Feinerkennung, einer Korrektur des perspektivischen Fehlers unterworfen. Die Basis dafür bildet eine mit Hilfe eines Kalibrierkörpers einmalig erstellte Zuordnung von im Bild ermittelten zu realen Markierungspositionen. Die Höhe des Kalibrierkörpers stimmt mit der des Fahrzeuges überein und auf seiner Oberfläche befinden sich ebenfalls Markierungen der Größe 1,5×1,5 mm

in regelmäßiger Anordnung. (vgl. Abschnitt „Korrektur des perspektivischen Fehlers“ in diesem Kapitel)

7. Berechnung der durchschnittlichen x- und y-Koordinatenwerte der so modifizierten 6 Punkte und Festlegung des berechneten Koordinatenpaares als Mittelpunkt des Fahrzeuges
8. Ermittlung der zwei Winkel von diesem Mittelpunkt zu den beiden in Fahrtrichtung liegenden weißen Markierungen und Festlegung von deren Durchschnittswert als Winkel (Richtung), in dem das Fahrzeug ausgerichtet ist

Abb. F 2.23 dient der Veranschaulichung der beschriebenen Vorgehensweise.

3 Stoßwahl

Dieses Kapitel beschreibt den Algorithmus, der aus einer gegebenen Konstellation von Kugeln auf dem Billardtisch den als nächsten auszuführenden Stoß auswählt. Zunächst werden kurz die physikalischen und geometrischen Eigenschaften eines einfachen Billardstoßes erläutert.

3.1 Physik und Geometrie des Billardstoßes

Die beim realen Billard auftretenden physikalischen Effekte sind recht vielfältig, besonders wenn der Spieler in der Lage ist, der Kugel ein zielgerichtetes Effet, also eine Eigenrotation, zu verleihen. Die folgende Darstellung beschränkt sich auf den vom Billardroboter ausschließlich angewandten einfachen Stoß, bei dem davon ausgegangen wird, daß die weiße Spielkugel vom Queue (bzw. Stößel) zentral getroffen wird und somit kein Effet besitzt.

Der für die Stoßwahl relevante Abschnitt des gesamten Stoßes ist die Kollision der Spielkugel mit der (ruhenden) zu versenkenden Kugel. Physikalisch betrachtet handelt es sich dabei um einen dezentralen, elastischen Stoß. Dessen allgemeiner Fall ist z.B. in [Paus95] ausführlich beschrieben und soll an dieser Stelle kurz nachvollzogen werden.

In Abb. 3.1 sind die geometrischen Zusammenhänge zunächst graphisch dargestellt.

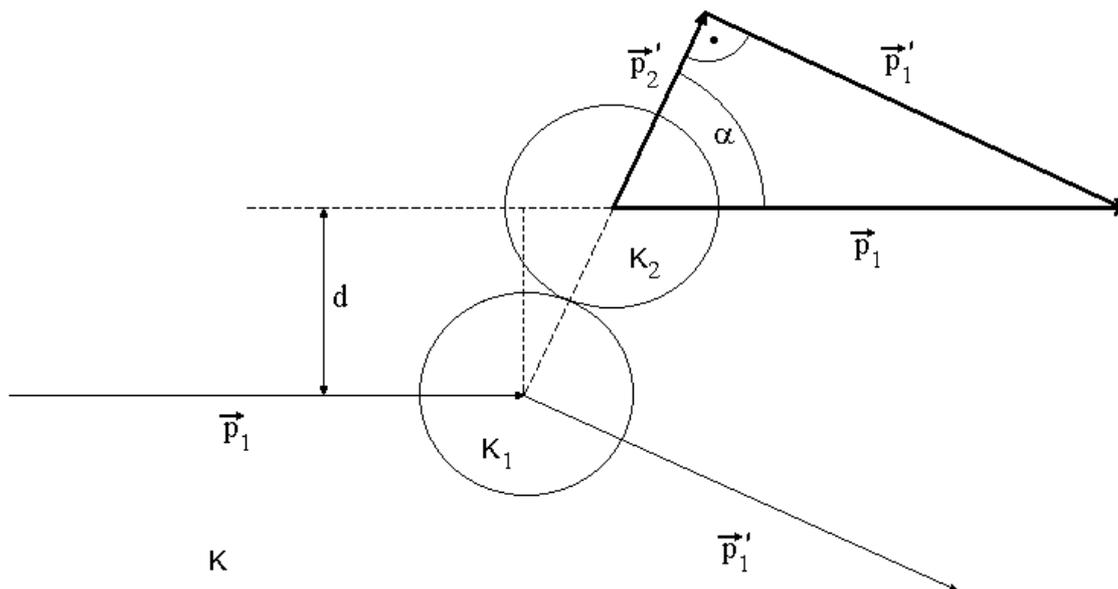


Abb. 3.1 Dezentraler elastischer Stoß

Die weiße Spielkugel K₁ verfügt über den Impuls p_1 und trifft auf die ruhende Kugel K₂. Dabei überträgt sie ihre kinetische Energie in Abhängigkeit vom Dezentralitätsparameter d auf K₂. Für beide Kugeln ergeben sich neue Impulse p'_1 und p'_2 und damit neue Bewegungsrichtungen. α ist der für die Stoßwahl interessante Auftreffwinkel. Er ist identisch mit der resultierenden Bewegungsrichtung von K₂. Der Punkt, an dem sich K₁ im hier dargestellten Moment des Kontaktes befindet, ist der Auftreffpunkt A.

Für den beim Billard vorliegenden Fall, daß die beiden am Stoß beteiligten Körper die gleiche Masse besitzen, gilt nach dem Energiesatz die Gleichung

$$p_1^2 = p_1'^2 + p_2'^2 \quad (1).$$

Daraus ist unmittelbar ersichtlich, daß zwei Billardkugeln sich nach einer Kollision stets im rechten Winkel voneinander entfernen. Die einzige Ausnahme von dieser Regel bildet ein zentraler Stoß, bei dem der Impuls der sich bewegenden Kugel vollständig auf die ruhende Kugel übertragen wird. In diesem Fall bewegt sich die angestoßene Kugel in der ursprünglichen Bewegungsrichtung der nunmehr zum Stillstand gekommenen ersten Kugel weiter. Auch die Bewegungsgeschwindigkeit ist (theoretisch) identisch (Impulserhaltungssatz).

Im Billard ist es wünschenswert, der angestoßenen Kugel eine definierte Richtung zu geben, d.h., der in Abb. 3.1 dargestellte **Auftreffwinkel** α zwischen dem ursprünglichen, vom Queue bestimmten Impuls p_1 und dem resultierenden Impuls p_2' ist von Interesse.

Für diesen gilt die Beziehung

$$\sin \alpha = \frac{d}{2r} \quad (2) ,$$

wobei r der Radius der Kugeln und d der in Abb. 3.1 dargestellte Dezentralitätsparameter ist.

Daraus ergeben sich für die resultierenden Impulse die folgenden Gleichungen:

$$p_1' = \frac{d}{2r} \cdot p_1 \quad (3)$$

$$p_2' = \sqrt{1 - \left(\frac{d}{2r}\right)^2} \cdot p_1 \quad (4).$$

Diese stellen Spezialfälle der allgemeinen, in [Paus95] hergeleiteten Gleichungen

$$p_1' = \left\{ \sqrt{1 - \frac{4m_1m_2}{(m_1 + m_2)^2} \cdot \left[1 - \left(\frac{d}{r_1 + r_2}\right)^2\right]} \right\} \cdot p_1 \quad \text{und}$$

$$p_2' = \left\{ \frac{2m_2}{m_1 + m_2} \cdot \sqrt{1 - \left(\frac{d}{r_1 + r_2}\right)^2} \right\} \cdot p_1$$

mit $m_1 = m_2$ (Kugelmassen) und $r_1 = r_2 = r$ (Kugelradien) dar.

3.2 Restriktionen und Parameter der Stoßwahl

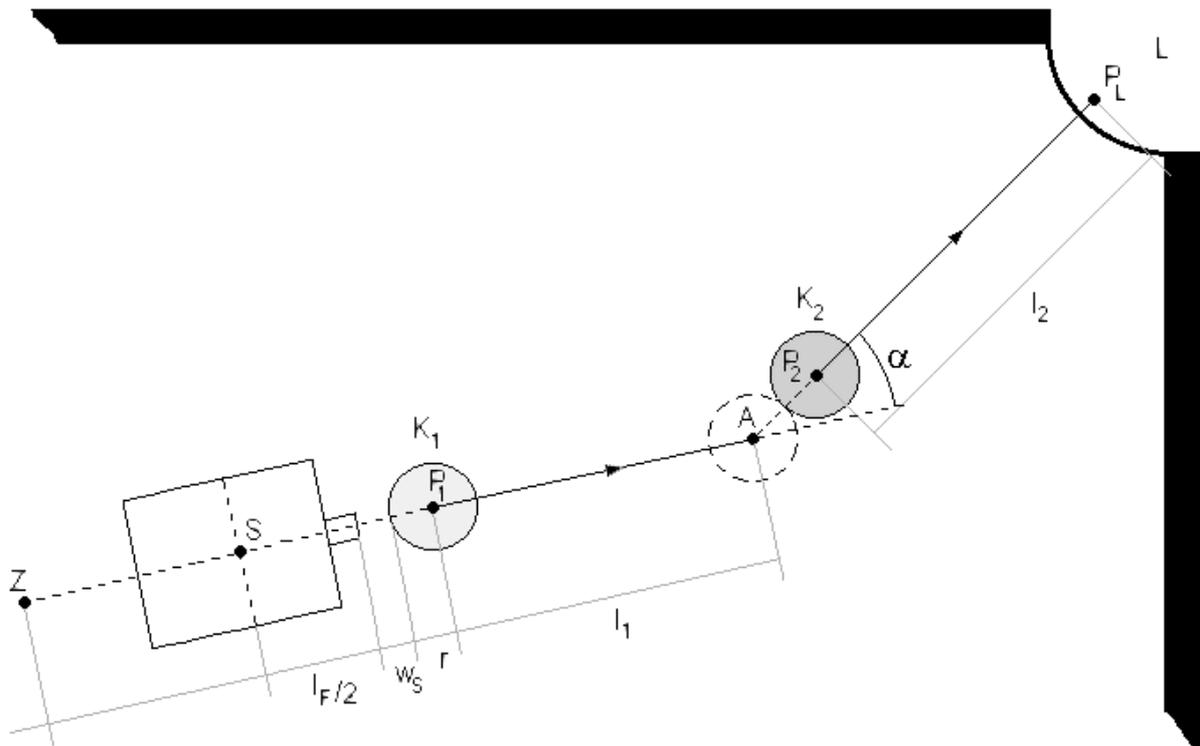


Abb. 3.2 Parameter der Stoßwahl
Erläuterungen im Text

Die bisherigen Darstellungen sind zunächst geeignet, für einen gegebenen Impuls p_1 der Spielkugel und einen bekannten **Auftreffpunkt** A die resultierenden Impulse (bzw. Geschwindigkeiten und Bewegungsrichtungen) der beiden Kugeln zu berechnen. Dabei ist der Auftreffpunkt der Punkt, an dem sich die Spielkugel im Moment des Kontaktes mit der zu versenkenden Kugel befindet (vgl. Abb. 3.2).

Im Gegensatz dazu ist es für die Auswahl eines Stoßes erforderlich, aus einer gewünschten Richtung der zweiten Kugel (von ihrer momentanen Position zu einem Loch) den dazu passenden Auftreffpunkt der Spielkugel und daraus den Stoßwinkel und die erforderliche Position des Roboterfahrzeuges zu berechnen.

Desweiteren sind die oben angegebenen Beträge der Impulse, also die Geschwindigkeiten der Kugeln, nur von untergeordnetem Interesse. Solange der Impuls der Spielkugel stark genug ist, um der zu versenkenden Kugel genügend „Schwung“ mit auf den Weg zu geben, ist sein konkreter Wert unerheblich, da dieser auf die Richtung der resultierenden Bewegungen keinen Einfluß hat.

Wie im Kapitel „Technische Details“ dargelegt wird, ist die vom Stößel bereitgestellte Stoßkraft in jedem Falle ausreichend.

Die folgende Aufstellung beinhaltet die grundsätzlichen Voraussetzungen, welche zur Ausführung eines bestimmten Stoßes in jedem Falle erfüllt sein müssen.

Dabei sei die weiße Spielkugel mit K_1 , die potentiell zu versenkende Kugel mit K_2 , und das potentielle Ziel-Loch mit L bezeichnet.

Die Bezeichnungen P_1 , P_2 und P_L stehen für die Positionen (Koordinaten) der drei Objekte.

- 1.) Liegt K_2 „zwischen“ K_1 und L , d.h., ist der für die Bewegung von K_2 in Richtung L notwendige Auftreffpunkt A von K_1 auf der dieser zugewandten Seite von K_2 ?

Diese Bedingung ist im Allgemeinen erfüllt, wenn der Winkel zwischen den Vektoren P_1P_2 und P_2P_L deutlich kleiner als 90° ist.

Die hinreichende Bedingung lautet $\alpha < 90^\circ$ und bezieht sich auf den etwas aufwendiger zu berechnenden Auftreffwinkel (siehe nächster Abschnitt).

- 2.) Ist L von K_2 auf gerader Linie erreichbar, d.h., liegen evtl. andere Kugeln im Weg?
- 3.) Ist K_2 von K_1 auf gerader Linie erreichbar, d.h., liegen evtl. andere Kugeln im Weg?
- 4.) Ist „hinter“ der Spielkugel genügend Platz für das Roboterfahrzeug vorhanden?

Erläuterungen dazu finden sich auch im Abschnitt „Unterschiede zwischen realem Spiel und Billard-Roboter“ im Einführungskapitel.

Sind diese vier Bedingungen erfüllt, so ist ein Stoß $K_1 \rightarrow K_2 \rightarrow L$ prinzipiell möglich.

Ob der Stoß jedoch auch erfolgversprechend ist, hängt von den folgenden Größen ab:

- 5.) Welchen Wert hat der notwendige Auftreffwinkel α ?

Dieser Winkel ist in zweifacher Hinsicht interessant.

Je kleiner er ist, desto größer ist die Wahrscheinlichkeit, K_2 auch dann zu treffen, wenn der **reale Auftreffpunkt** A_R , vom theoretisch idealen Auftreffpunkt A abweicht, also die Bewegungsrichtung der Spielkugel mit einem Fehler behaftet ist.

Außerdem hat er gravierenden Einfluß auf die „Aufteilung“ des von K_1 eingebrachten Impulses p_1 auf beide Kugeln. Hat er z.B. den Wert 0, so liegen K_1 , K_2 und L auf einer Linie und es gilt $p_2' = p_1$, d.h. die Energie wird, wie oben bereits erwähnt, vollständig auf K_2 übertragen. Im anderen Extremfall ist $\alpha > 90^\circ$, was bedeutet, daß die Kollision gar nicht stattfindet und K_1 somit auch keinerlei Energie auf K_2 überträgt.

- 6.) Wie groß ist der Abstand zwischen K_1 und K_2 , respektive P_1 und P_2 (im Folgenden mit l_1 bezeichnet)?

Je größer diese Entfernung ist, desto stärker wirken sich Fehler in der Richtung des **realen Stoßvektors** P_1A_R aus.

- 7.) Wie groß ist der Abstand zwischen K_2 und L , respektive P_2 und P_L (im Folgenden mit l_2 bezeichnet)?

Mit zunehmendem Abstand l_2 wirken sich Fehler in der Bewegungsrichtung von K_2 stärker aus.

Für die Punkte 6. und 7. gilt gleichermaßen, daß bei langen Wegen die Eigenschaften des Untergrundes (Reibung, Unebenheiten) sowie ungewolltes Effet in größerem Maße die Bahnstabilität der Kugeln beeinflussen können.

Aus diesen Erläuterungen kann geschlußfolgert werden, daß ein potentieller Stoß offensichtlich dann besonders günstig ist, wenn beide Kugeln sich in der Nähe eines Loches befinden und mit diesem eine Linie bilden.

3.3 Implementierung des Algorithmus

Der Gesamtalgorithmus der Stoßwahl lautet wie folgt:

Teste
für Loch 1..6
für jede Kugel der gewünschten Farbe (rot bzw. gelb)
Sind die Bedingungen 1, 2, 3 und 4 erfüllt ?
Wenn ja, dann berechne l_1 und l_2
und füge den Stoß in die Liste möglicher Stöße ein.
Sollte die Liste leer sein, so brich ab und gib die Meldung aus,
daß kein ausführbarer Stoß gefunden werden konnte.
Wichte die Parameter α , l_1 und l_2 für jeden Stoß der Liste.
Wähle den am besten bewerteten Stoß aus.

Die einzelnen Schritte werden im folgenden erläutert:

Test der Bedingung 1:

Zu prüfen ist, ob für den durch den Vektor P_2P_L vorgegebenen Auftreffwinkel α ein Auftreffpunkt A existiert, ob also eine Kollision zwischen K_1 und K_2 überhaupt stattfinden kann.

Die Punkte P_1 , P_2 und P_L sind bekannt. Vom Dreieck P_1P_2A sind die Längen der Seiten P_1P_2 und $P_2A=2r$ sowie der Winkel im Punkt P_2 (α) bekannt.

Die Anwendung des Kosinussatzes führt auf eine quadratische Gleichung, deren Lösungen einer der folgenden drei Konstellationen entsprechen:

- 1. keine positiv reelle Lösung → Es findet keine Kollision statt.**
- 2. eine positiv reelle Lösung → Der Wert entspricht der Länge der Seite P_1A .**
- 3. zwei positiv reelle Lösungen → Der kleinere Wert entspricht der Länge der Seite P_1A .**

Daraus resultierend können die Koordinaten von A und schließlich auch der Winkel des Vektors P_1A berechnet werden, welcher dem

Stoßwinkel entspricht, in dem das Roboterfahrzeug vor dem Stoß auszurichten ist.

Test der Bedingung 2:

Analog zum vorhergehenden Test ist für jede auf dem Spielfeld befindliche Kugel (außer K_1 und K_2) zu ermitteln, ob für den vorgegebenen Stoßwinkel (die Bewegungsrichtung von K_1) ein Kollisionspunkt existiert, also die entsprechende Gleichung eine positiv reelle Lösung besitzt. Trifft dies für mindestens eine Kugel zu, so ist Bedingung 2 verletzt.

Test der Bedingung 3:

Nach dem gleichen Prinzip ist für alle in Frage kommenden Kugeln zu ermitteln, ob für den vorgegebenen Auftreffwinkel α (die Bewegungsrichtung von K_2) ein Kollisionspunkt existiert.

Trifft dies für mindestens eine Kugel zu, so ist Bedingung 3 verletzt.

Test der Bedingung 4:

Um diese Bedingung zu überprüfen, sind zunächst der erforderliche Standpunkt des Roboterfahrzeuges (**Stoßpunkt S**) und der entsprechende **Zielpunkt Z** der Fahrt des Roboters (Punkt 4 des Stoßzyklus) zu berechnen.

Beide befinden sich auf der Verlängerung der Verbindungslinie P_1A über den Punkt P_1 hinaus.

Die Distanz zwischen S und P_1 ergibt sich aus der Summe $r+l_F/2+w_s$, wobei r den Kugeldurchmesser, l_F die Länge des Roboterfahrzeuges und w_s den Weg bezeichnet, welchen der Stößel beim Stoß zurücklegt. Die realen Werte betragen ca. 14, 65 und 10mm. Z liegt entsprechend weiter von P_1 entfernt (ca. 30mm).

Die weitere Vorgehensweise beim Test, ob hinter der Spielkugel hinreichend viel Platz für das Fahrzeug zur Verfügung steht, wird von Abb. 3.3 illustriert. Für jede außer der Spielkugel auf der Spielfläche befindliche Kugel wird geprüft, ob ihre Position einen bestimmten **Mindestabstand** d_{\min} von den Punkten Z und S besitzt.

Dieser Mindestabstand d_{\min} ergibt sich wie folgt:

$$d_{\min} = \sqrt{l_F^2 + b_F^2} + r + s$$

Dabei bezeichnen l_F und b_F die Länge und Breite des Roboterfahrzeuges, r den Radius der Billardkugeln und s einen zusätzlichen Sicherheitspuffer.

Verfügt mindestens eine der Kugeln nicht über den Mindestabstand, so ist Bedingung 4 verletzt und der entsprechende Stoß demzufolge nicht durchführbar.

Wichtung der Parameter α , l_1 und l_2 und Ermittlung des besten Stoßes

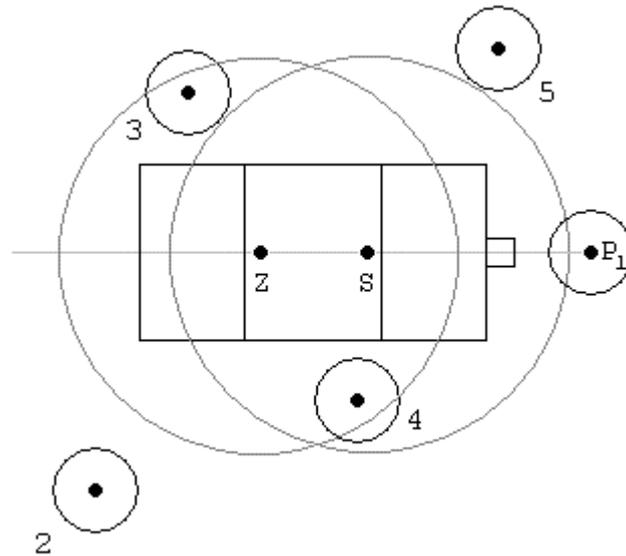
Die in der Liste der möglichen Stöße enthaltenen Einträge sind für die Auswahl des günstigsten Stoßes in geeigneter Weise vergleichbar zu machen. Dazu wird für jeden einzelnen Stoß S_i ein **Bewertungskoeffizient** k_i ermittelt, der sich wie folgt berechnet:

$$k_i = l_{1i} / l_{\max} + l_{2i} / l_{\max} + \alpha_i / \alpha_{\max}$$

Dabei sind l_{1i} , l_{2i} und α_i die Parameter des Stoßes, l_{\max} ist gleich der Länge der Spielfeld-Diagonale, also dem theoretischen Maximalwert, den l_{1i} und l_{2i} annehmen können und $\alpha_{\max} = 90^\circ$, was dem theoretischen Maximum für α_i entspricht.

Abschließend ist der Stoß mit dem niedrigsten Koeffizienten k_i zu ermitteln und zum auszuführenden Stoß zu bestimmen.

Die Abbildungen 3.4 bis 3.9 zeigen einige Beispiele für Ergebnisse des Stoßwahl-



Algorithmus. Gespielt wurde jeweils auf die gelben Kugeln (hellgrau).

Abb. 3.3 Zum Test von Bedingung 4

Die eingezeichneten Kreise um die Punkte S bzw. Z haben den Radius d_{\min} . P_1 bezeichnet die Position der weißen Spielkugel, welche in den Test nicht einbezogen wird. Die Kugeln 2 und 5 haben den erforderlichen Mindestabstand, die Kugeln 3 und 4 jedoch nicht.

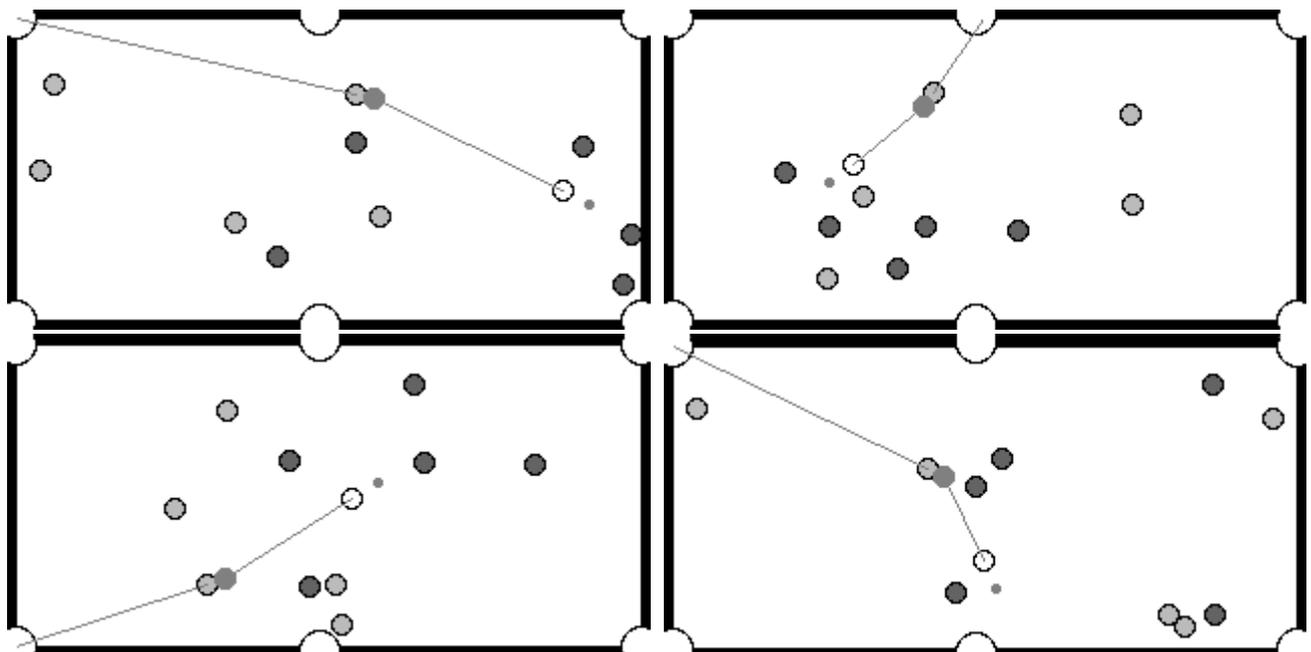


Abb. 3.4 bis 3.9 Beispiele für gewählte Stöße

Die kleinen Kreise markieren den Stoßpunkt S. Eingezeichnet ist auch der jeweilige Auftreffpunkt der Spielkugel.

4 Wegplanung

4.1 Grundlegende Überlegungen

Das Problem, in einer bestimmten Umgebung einen Weg zu bestimmen, der von Punkt A nach Punkt B führt, tritt in den verschiedensten Bereichen des Alltags auf. Zunächst sollen die Gemeinsamkeiten all dieser Szenarien und anschließend Unterschiede zwischen ihnen, besonders bzgl. der Bewertung entstehender Alternativen, dargelegt werden.

Die Betrachtung beschränkt sich auf statische Umgebungen, welche durch eine feste Karte beschrieben werden können. Diese Einschränkung ist möglich, da es sich beim Billardsystem um eine (innerhalb eines Stoßzyklus) nicht veränderliche Umgebung handelt. Desweiteren wird von einer Bewegung in der Ebene ausgegangen.

In Anlehnung an die Begriffe der Graphentheorie soll im folgenden von **Knoten** (Punkten, an denen Richtungsänderungen erfolgen können), **Kanten** (geradlinige Verbindungen zwischen Knoten) und **Wegen** (Ketten von miteinander verbundenen Kanten) die Rede sein.

4.1.1 Gemeinsamkeiten von Wegplanungs-Szenarien

In jedem Fall existiert ein Agent (Person, Kraftfahrzeug, Roboter), welcher sich aufgrund seiner konstruktiven Eigenschaften prinzipiell in alle Richtungen der Ebene bewegen kann. Ist die Ebene (Fläche) völlig frei von anderen, statischen Objekten und bezüglich der Fortbewegung qualitativ an jeder Stelle gleichwertig, so kann die direkte Verbindung von Start- und Zielpunkt als Weg genutzt werden.

Dieser einfachste Fall ist allerdings ohne praktische Bedeutung, da in der Realität stets massive Hindernisse, also nicht durchdringbare physikalische Körper, oder zur Fortbewegung nicht geeignete Flächen auftreten. Als Beispiele sind Wände, Möbel, Bäume und Billardkugeln bzw. Gräben, Gewässer und unbefestigtes Gelände zu nennen. Auf einer abstrakten Betrachtungsstufe sind diese beiden Arten von Hindernissen als gleichwertig zu betrachten.

Es existiert somit eine zu betrachtende Gesamtfläche, welche den Start- und den Zielpunkt sowie nicht begehbare Teilflächen enthält. Der diesem allgemeinen Fall entsprechende Typ von Karte soll als **Flächenkarte** bezeichnet werden.

Eine schärfere Einschränkung liegt vor, wenn die begehbaren Teile der Fläche keine zweidimensionale Ausdehnung mehr besitzen, also innerhalb dieser keine alternativen Richtungen eingeschlagen werden können. Ein Beispiel dafür ist die Bindung von Schienenfahrzeugen an das vorhandene Schienennetz. Ein Richtungswechsel ist dort nur an relativ wenigen Punkten der Gesamtfläche möglich. Die der Wegplanung zugrunde liegende Topologie hat von vornherein die Gestalt eines mathematischen Graphen, wobei sich die begehbare Fläche auf dessen Kanten reduziert. Die entsprechende Karte soll als **Graphenkarte** bezeichnet werden.

Eine Flächenkarte besitzt, im Gegensatz zu einer Graphenkarte, eine unendliche Anzahl von potentiellen Wegen vom Start- zum Zielpunkt. Voraussetzung dafür ist, daß der entsprechende topologische Zusammenhang der begehbaren Fläche gegeben ist. Sollte der Fall eintreten, daß die begehbare Fläche durch die Anordnung der Hindernisse in mehrere

voneinander getrennte Teile zerlegt wird, dann existiert u.U. keine einzige Verbindung vom Start zum Ziel.

Die unendlich vielen Wege lassen sich bezüglich ihrer Topologie in eine endliche Anzahl von Äquivalenzklassen einordnen. Allen Wegen einer Äquivalenzklasse ist die Art der Umgehung der vorhandenen Hindernisse gemeinsam, d.h., ob ein bestimmtes Hindernis "links- oder rechtsherum" umgangen wird.

In der Praxis ist es sinnvoll, aus jeder Äquivalenzklasse einen geeigneten Repräsentanten auszuwählen und somit die Flächenkarte in eine Graphenkarte zu überführen.

Abb. 4.1 veranschaulicht die beschriebenen Sachverhalte.

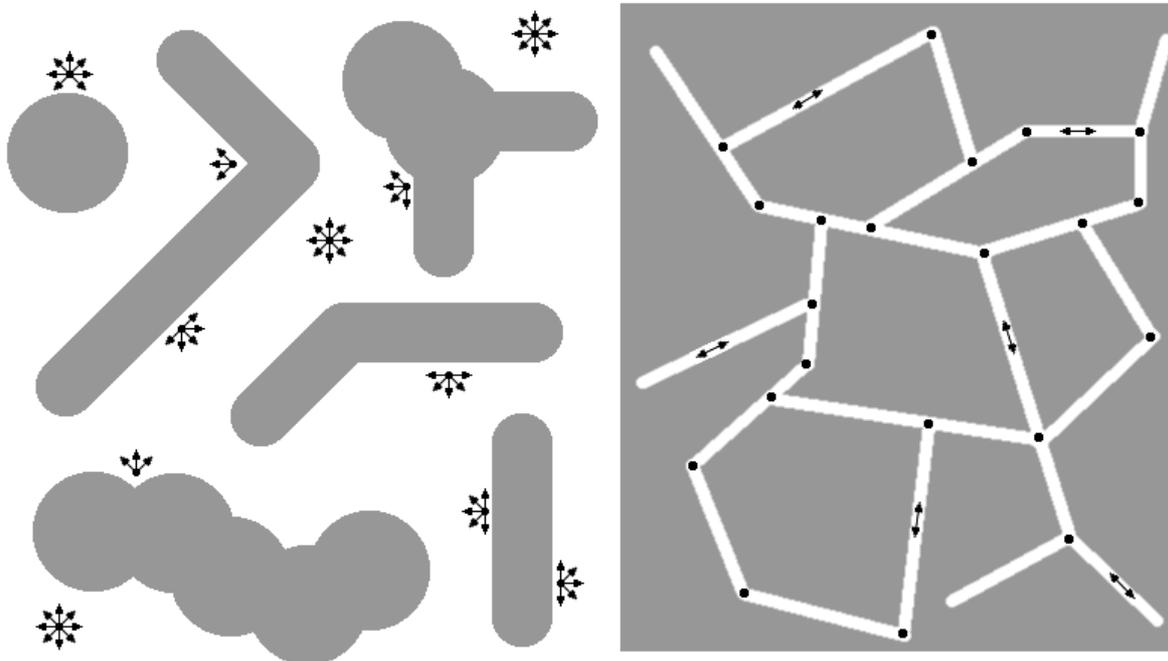


Abb. 4.1 Flächenkarte (links) und Graphenkarte (rechts)

Hindernisse sind grau und begehbare Gebiete weiß dargestellt. Die Pfeile symbolisieren die möglichen Bewegungsrichtungen an verschiedenen Stellen. In der Graphenkarte sind nur an den gekennzeichneten Punkten Richtungswechsel ungleich 180 Grad möglich.

Eine bei Flächenkarten zu beachtender Parameter ist der Platz, welcher vom sich bewegenden Agenten selbst eingenommen wird. Da der Agent bei der Wegplanung durch seine Position in der Ebene, d.h., als Punkt repräsentiert wird, muß seine Ausdehnung auf andere geeignete Weise in die Modellierung einbezogen werden. Eine praktikable Variante besteht darin, alle Hindernisse um die Hälfte des Betrages der maximalen Ausdehnung des Agenten zu erweitern, also sozusagen "aufzublähen". Dadurch ist sichergestellt, daß im Modell die von Hindernissen und Agent belegten Flächen in jedem Falle disjunkt sind.

4.1.2 Varianten der Bewertung von Wegen

Die bisher gemachten Ausführungen beziehen sich überwiegend auf die Modellierung von Wegen, welche innerhalb des konkreten Szenariums potentiell vorhanden sind. Da es sich in der Regel um mehr als einen möglichen Weg handelt, besteht nun die Notwendigkeit, aus

dieser Menge von Alternativen die beste auszuwählen. Die Suche nach der optimalen Verbindung verlangt nach einer Möglichkeit, die einzelnen Wege in geeigneter Form zu vergleichen und zu bewerten.

Die für die Güte der Bewertung entscheidenden Kriterien können, je nach konkreter Problemstellung, von verschiedenster Art sein. Oft geht es um den kürzesten, schnellsten oder kostengünstigsten Weg. Entsprechend werden den einzelnen Kanten euklidische Längen, Zeiten oder finanzielle Kennziffern zugeordnet. Für einige Szenarien steht die genaue Konfiguration von nutzbaren Kanten erst nach einer vorausgehenden Bewertung der Kanten bezüglich einer weiteren Größe fest. So muß bei der Planung eines Schwerlasttransportes eine große Anzahl von Kanten (Straßen) aus dem zugrunde liegenden Graphen (der Straßenkarte) gestrichen werden, da diese nicht über die erforderliche Breite oder Belastbarkeit verfügen. Eine völlig andere Anforderung liegt vor, wenn die Gesamtlänge des Weges einen bestimmten Wert haben soll, z.B. bei der Planung eines Marathonlaufes durch die Straßen einer Stadt.

Die bei der vorliegenden Wegplanungsaufgabe verwendete Bewertung ist darauf ausgerichtet, den sichersten Weg zu bestimmen. Was im konkreten Fall unter Sicherheit zu verstehen ist, wird in den nächsten Abschnitten erläutert.

4.2 Anforderungen an die Wegplanung

Die Aufgabe der Wegplanungskomponente des Billardsystems besteht darin, eine Folge von Punkten auf der Spielfläche zu generieren, welche vom Roboter nacheinander angefahren werden, so daß er sich von seiner momentanen Position (Startpunkt) zu dem Punkt, von dem aus der nächste Stoß erfolgen soll (Zielpunkt), bewegt. Die Bewegung zwischen zwei derartigen Punkten folgt der Verbindungsgerade zwischen beiden. Nach Erreichen eines Punktes dreht sich das Fahrzeug um einen bestimmten Winkel und wird dadurch auf den nächsten anzufahrenden Punkt ausgerichtet.

Auf seinem Weg darf der Roboter keine der auf der Spielfläche liegenden Kugeln berühren. Das Billardsystem ist mit einer schwer einzuschätzenden und nur teilweise zu kompensierenden Ungenauigkeit bei der Ermittlung der Positionen von Kugeln und Fahrzeug behaftet. Gleiches gilt für die Umsetzung der geplanten Bewegungen. Aus diesem Grund muß die Wahrscheinlichkeit, eine Kugel zu berühren, minimiert werden.

Am sichersten ist ein Weg dann, wenn die Entfernung jedes einzelnen zum Weg gehörenden Punktes von den Hindernissen möglichst groß ist. Das heißt im Besonderen, daß die Durchfahrt zwischen zwei Hindernissen (Kugeln, Bande) in deren Mitte erfolgt und breite Passagen schmalere vorzuziehen sind.

4.3 Bestimmung der potentiellen Wege

4.3.1 Modellierung von Hindernissen und Fahrzeug

Die Ausgangssituation für die Wegplanung besteht in einer Flächenkarte, welche den Start- und den Zielpunkt sowie die Hindernisse (Kugeln, Banden) enthält (Abb. 4.2).

Die von Hindernissen belegten Gebiete sind in der bereits beschriebenen Art und Weise ausgedehnt worden, wodurch das Roboterfahrzeug als einzelner Punkt betrachtet werden kann. (Dieser Punkt entspricht dem Mittelpunkt des Fahrzeuges.) Die Breite des zusätzlichen Randes entspricht dem halben Wert der größten Ausdehnung des Fahrzeuges (halbe Diagonale, zuzüglich eines gewissen "Sicherheitspuffers", vgl. auch Parameter d_{\min} im Kap. "Stoßwahl"). Somit kann davon ausgegangen werden, daß sich der Roboter auf allen in der

Abbildung weiß dargestellten Punkten sowohl bewegen als auch vollständig um seine Hochachse drehen kann.

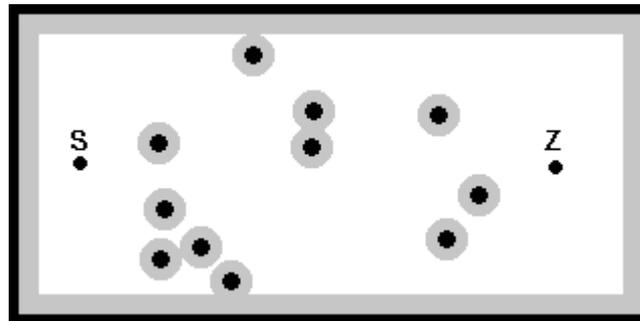


Abb. 4.2 Beispiel einer der Wegplanung zugrunde liegenden Karte (S=Start, Z=Ziel)

Start- und Zielpunkt wurden aufgrund der besseren Übersichtlichkeit willkürlich platziert und entsprechen keiner echten Spielsituation. Diese und die folgenden Abbildungen sind direkt aus dem Berechnungsprozeß übernommen, welcher auf Byte-Arrays von 320×160 Elementen, analog der Bildgröße bei der Kugel-Groberkennung, beruht.

Für die Aufgabe, zwischen zwei Hindernissen nur hindurch zu fahren, wäre es ausreichend, die Breite des Fahrzeuges zu betrachten. Da Fahrzeugdiagonale und -breite im Sinne der vorliegenden Problemstellung aber nicht wesentlich differieren, wurde auf eine derartige Unterscheidung verzichtet. Dies vereinfacht die Algorithmen relativ stark.

4.3.2 Voronoi-Parkettierung

Zunächst ist es notwendig, alle mögliche Teilwege (Kanten) zu bestimmen, welche die topologischen Besonderheiten der vorliegenden Konstellation, also alle "Durchfahrten" zwischen je zwei Hindernissen, repräsentieren. Die gesuchten Kanten sollten auch den Sicherheitsaspekten genügen und in der Mitte zwischen den Hindernissen verlaufen.

Eine Lösung, die diesen Anforderungen gerecht wird, basiert auf der sogenannten **Voronoi-Parkettierung** (z.B. [Sedg92], S.465ff). Eine derartiger Algorithmus ordnet die Punkte einer Ebene verschiedenen Äquivalenzklassen zu. Es bilden sich um die Hindernisse Gebiete aus, welche als "Einflußbereiche" gedeutet werden können.

Im konkreten Fall lautet die entsprechende Vorschrift wie folgt:

Ordne

jeder Kugel,
dem Startpunkt,
dem Zielpunkt und
der Bande

unterschiedliche Farben zu.

Für jeden Punkt:

Bestimme die Abstände zu
jeder Kugel,
dem Startpunkt,
dem Zielpunkt und
der Bande.

Bestimme das Minimum dieser Abstände.

Färbe den Punkt in der Farbe des Objektes ein, dessen Abstand minimal ist.

Die zum Beispiel in Abb. 4.2 gehörende parkettierte Fläche ist in Abb. 4.3 dargestellt.

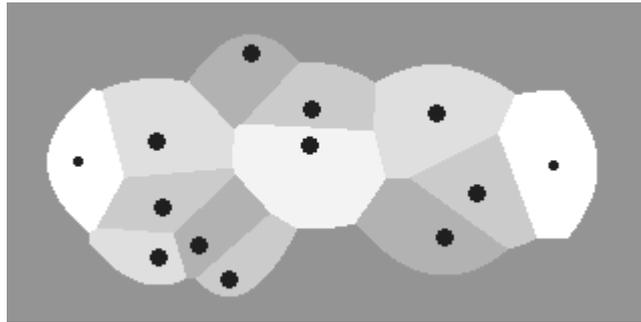


Abb. 4.3 Voronoi-Parkettierung

Die Grenzen zwischen den zu Kugeln gehörenden Gebieten verlaufen immer in der Mitte zwischen den Kugeln und sind gerade, also ohne weiteres in die späteren Steuerbefehle für das Fahrzeug umsetzbar.

Im Gegensatz dazu sind die Grenzen zwischen Kugeln und Banden mehr oder weniger stark gekrümmt, wodurch sie durch das Fahrzeug nur schwer verfolgt werden könnten.

Zur Beseitigung dieses nicht unwesentlichen Effektes wird der oben dargestellte Algorithmus modifiziert. Um gerade Grenzen zu erzielen, wie sie zwischen den zu Kugeln gehörenden Gebieten existieren, werden auf Höhe der den Banden nächsten Kugeln zusätzliche virtuelle Kugeln positioniert.

Schließlich wird die oben beschriebene Voronoi-Parkettierung ausgeführt, allerdings ohne Berücksichtigung der Banden. Diese Vorgehensweise und den daraus resultierenden Unterschied illustriert Abb. 4.4.

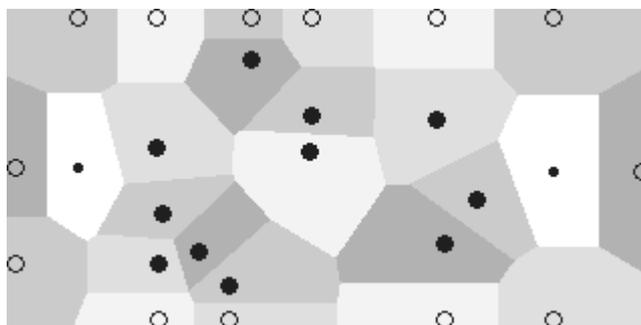


Abb. 4.4 Verbesserte Voronoi-Parkettierung mit eingefügten virtuellen Kugeln

Anschließend werden alle Punkte der entstandenen Karte, an denen drei Gebiete (Farben) zusammentreffen, als Knoten deklariert. Kanten entstehen durch das Verbinden von Knoten, bei denen zwei der drei angrenzenden Gebiete übereinstimmen, sofern sie sich nicht an den Gebieten von Start- oder Zielpunkt befinden. Diese Knoten werden durch eine Kante jeweils direkt mit dem Start- bzw. Zielpunkt verbunden, welche selbst zu Knoten werden.

4.3.3 Bestimmung der sicheren Teilwege

Jeder Kante wird eine Bewertung zugeordnet, welche dem Abstand der Kugeln entspricht, zwischen denen sie verläuft (sowohl reale als auch virtuelle Kugeln). Die Kanten, welche zum Start- und Zielknoten gehören, erhalten eine von den anderen Kanten nicht erreichbare, besonders hohe Bewertung, da sie immer als sicher gelten können (Abb. 4.5).

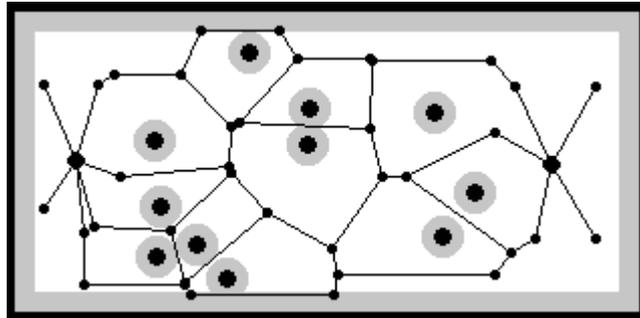


Abb. 4.5 Alle prinzipiell vorhandenen Kanten (Teilwege)

Es liegen nun eine Liste von Knoten sowie eine Liste bewerteter Kanten vor, welche in ihrer Gesamtheit einen ungerichteten, bewerteten Graph darstellen.

Der letzte Schritt vor der eigentlichen Auswahl des Weges besteht im Entfernen aller Kanten, deren Bewertung kleiner als der erforderliche Mindestwert (Fahrzeuglänge zuzüglich Sicherheitspuffer) ist. Die dabei entstehenden "Sackgassen" sind ebenfalls zu löschen.

Das jetzt vorliegende Netz enthält alle Kanten, entlang derer sich das Fahrzeug sicher bewegen kann (Abb. 4.6).

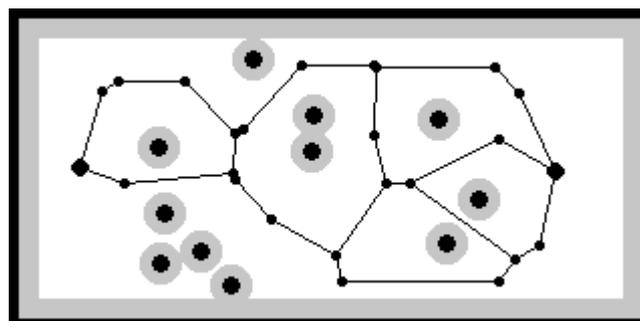


Abb. 4.6 Alle sicheren Kanten

4.4 Auswahl des optimalen Weges

In diesem Stadium der Wegplanung existiert (in den meisten Fällen) eine relativ große Anzahl von verschiedenen Kantenfolgen, welche vom Start zum Ziel führen. Die nunmehr zu lösende Aufgabe besteht darin, aus den vorhandenen Weg-Varianten die zu bestimmen, welche aus möglichst gut bewerteten Kanten zusammengesetzt ist.

Entscheidungen zur Wegwahl müssen nur an Knoten getroffen werden, deren Grad größer 2

ist, die also mindesten 3 Nachbarknoten besitzen. Sie sollen **Abzweige** genannt werden. Start- und Zielknoten sind ebenfalls Abzweige, auch wenn sie weniger als 3 Nachbarn haben. Alle Kantenfolgen zwischen zwei Abzweigen werden als **Wege** bezeichnet. Jeder Weg erhält eine Bewertung, die gleich dem Minimum der Bewertungen seiner Kanten ist.

Für das weitere Verfahren wird zunächst ein einfacher Algorithmus benötigt, mit dem überprüft werden kann, ob zwischen Start und Ziel überhaupt eine Verbindung existiert:

Teste, ob der Start zu mindestens einem Weg gehört.

Wenn nicht, dann Abbruch: Es existiert keine Verbindung vom Start zum Ziel.

Ordne jedem Abzweig eine natürliche Zahl >0 zu, dabei erhält der Start die Nummer 1.

Initialisiere eine (zunächst leere) Menge M von natürlichen Zahlen.

Füge die Nummer 1 der Menge M hinzu.

Wiederhole

Für jedes Element A von M:

Füge die Nummern aller Abzweige, mit denen Abzweig A über einen direkten Weg verbunden ist, der Menge M hinzu.

solange, bis M sich nicht mehr ändert.

Teste, ob die Nummer des Zielknotens sich in M befindet.

Wenn ja, dann existiert eine Verbindung vom Start zum Ziel, sonst nicht.

Aus der vorhandenen Liste von Wegen, werden nun sukzessive solche Wege entfernt, die eine schlechte (niedrige) Bewertung haben. Dies erfolgt unter Berücksichtigung der geforderten Erreichbarkeit des Zieles vom Start aus. Der Algorithmus lautet:

Teste, ob eine Verbindung zwischen Start und Ziel existiert (s.o.).

Wenn nein, dann ist das Ziel vom Start aus nicht erreichbar. Abbruch.

Jeder Weg sei nicht markiert.

Wiederhole

Bestimme aus der Menge der nicht markierten Wege den am schlechtesten bewerteten Weg W.

Entferne diesen aus der Menge.

Teste, ob eine Verbindung zwischen Start und Ziel existiert (s.o.).

Wenn nein, dann füge W wieder der Wegmenge hinzu und markiere ihn.

bis alle Wege markiert sind.

Entferne alle die Kanten aus der Kantenliste, die nicht zu einem der verbliebenen

Wege gehören.

Falls überhaupt eine Verbindung existiert, so ist das Ergebnis des gesamten Algorithmus eine Kette von Wegen, welche ihrerseits einer durchgehenden Kette von Kanten entspricht, die vom Start zum Ziel führt. Anders ausgedrückt sind in der endgültigen Kantenliste genau die Kanten (Strecken) enthalten, welche der Roboter (in der richtigen Reihenfolge) befahren muß (Abb. 4.7).

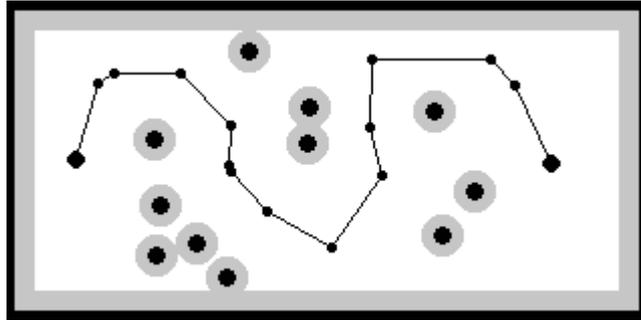


Abb. 4.7 Der gefundene sicherste Weg

4.5 Details der praktischen Anwendung des Algorithmus

Die bisherige Erläuterung der Wegplanung beschrieb den allgemeinen Fall der Bestimmung eines Weges auf dem Billard-Tisch. Ergänzend sollen nun noch einige Ausführungen zum praktischen Einsatz des Algorithmus folgen.

Der Zielpunkt des Weges, den das Fahrzeug zu absolvieren hat, ist der Punkt, auf dem es stehen muß, um den angestrebten Stoß erfolgreich durchzuführen. Dieser Punkt wird durch die Stoßwahl-Komponente vorgegeben.

Wie bereits dargelegt wurde, operieren die Algorithmen der Kugel-Groberkennung, Stoßwahl, Wegplanung und Fahrt (Punkte 1 bis 4 des Stoßzyklus) alle auf der relativ geringen Bild- bzw. Koordinatenauflösung von ca. 3mm pro (diskretem) Bild- bzw. Koordinatenpunkt. Für die Einnahme der Stoßposition muß sich das Fahrzeug der weißen Spielkugel bis auf ca. 10 mm annähern und sich eventuell noch auf der Stelle drehen. Würde nun der Stoßpunkt direkt als Zielpunkt der Fahrt gewählt, so bestünde aufgrund der Erkennungs- und Berechnungstoleranzen die Wahrscheinlichkeit, die Spielkugel zu berühren und zu bewegen. Dieser Fall darf nicht eintreten.

Deshalb wird der Zielpunkt der Fahrt, wie bereits im Kapitel „Stoßwahl“ beschrieben, etwa 30mm "hinter" den Stoßpunkt verlegt, d.h. entgegen der Stoßrichtung von der Spielkugel weg. Ein zusätzlicher positiver Effekt dieser Vorgehensweise besteht darin, daß bei der später folgenden exakten Ausrichtung des Fahrzeuges nur mehr oder weniger geradeaus fahren muß und keine größeren "Rangierarbeiten" notwendig werden.

Eine weitere kleiner, aber für die praktische Durchführung der Fahrt sinnvoller Schritt besteht in der Umwandlung der ungeordneten Kantenliste, welche nach dem bisher beschriebenen Verfahren vorliegt. Programmtechnisch gesehen wird aus ihr eine Liste von n Punkten $[x_i, y_i]$ erzeugt, bei der $[x_1, y_1]$ dem Startpunkt und $[x_n, y_n]$ dem Zielpunkt entspricht. Alle dazwischen liegenden Punkte sind die Zwischenstationen in der richtigen Reihenfolge.

5 Kamera-Fahrzeug-Koordination

Dieses Kapitel beschäftigt sich mit den Problemen und Algorithmen, welche mit der Bewegung des Fahrzeuges und deren Überwachung mittels der Kamera in Verbindung stehen. Die dabei zugrunde liegenden Aspekte der Fahrzeugerkennung, der Wegplanung sowie der konstruktiven Eigenschaften des Fahrzeuges werden in den entsprechenden Abschnitten der Kapitel "Bildererkennung", "Wegplanung" und "Technische Details" behandelt.

Getrennt voneinander werden die Fahrt des Roboterfahrzeuges in die Nähe der Stoßposition (Punkt 4 des Stoßzyklus) und die exakte Ausrichtung des Fahrzeuges bzw. das "Zielen" (Punkt 5 des Stoßzyklus) behandelt.

5.1 Fahrt zur Stoßposition

Die zu lösende Aufgabe besteht darin, das Roboterfahrzeug so zu steuern, daß es sich entlang der durch die Wegplanungskomponente bereitgestellten Folge von Punkten bewegt und somit von seinem aktuellen Standpunkt zu seinem Zielpunkt in der Nähe der weißen Spielkugel gelangt. Dabei soll dem berechneten Weg so genau gefolgt werden, daß keine der auf der Spielfläche liegenden Kugeln berührt und dadurch in ihrer Lage verändert wird. Außerdem sollte der gesamte Vorgang der Fahrt in einem relativ zügigen Tempo vor sich gehen.

5.1.1 Vereinfachung der Fahrtroute

Der letzte Schritt vor dem Beginn der eigentlichen Fahrt besteht darin, die **Route** noch ein wenig zu vereinfachen, d.h. die Anzahl ihrer einzelnen **Abschnitte** zu reduzieren. Das Ziel dieser Vorgehensweise besteht in der Verringerung der Zeit, die für das Zurücklegen des Weges benötigt wird. Diese ist u.a. auch von der Anzahl der Abschnitte abhängig, wie später ersichtlich werden wird.

Die Vereinfachung besteht darin, **Stationen**, d.h., Punkte, an denen Richtungswechsel erfolgen, zusammenzufassen, wenn sie sehr dicht beieinander liegen. Im konkreten Fall geschieht das, wenn der Abstand kleiner als 10 Px (bzw. Koordinateneinheiten) ist. Dies entspricht etwa 30 mm oder einer halben Fahrzeuglänge. Ist diese Bedingung für zwei Stationen erfüllt, dann werden diese gelöscht und durch eine neue Station ersetzt, die in der Mitte zwischen ihnen liegt.

5.1.2 Drehen des Fahrzeuges

Der Vorgang des Drehens wird sowohl zur Richtungsänderung an den einzelnen Stationen als auch zur Richtungskorrektur innerhalb eines gerade zurückzulegenden Abschnittes benötigt. Der letztgenannte Fall tritt während des Geradeausfahrens auf und wird im nächsten Abschnitt wieder aufgegriffen.

Die Drehung des Fahrzeuges von seiner momentanen Ausrichtung (**Startwinkel**) in eine neue Richtung (**Zielwinkel**) wird in mehrere Teildrehungen zerlegt, welche mit kleiner werdendem Abstand zum Zielwinkel (**Restwinkel**) immer geringere Richtungsänderungen umfassen. Der Zielwinkel wird also nicht mit einer einzigen kontinuierlichen Drehung erreicht. Diese Vorgehensweise ist insofern vorteilhaft, als damit schlecht erfaßbare Effekte beim Beschleunigen und beim Abbremsen der Drehung ausgeschaltet werden. Das heißt, daß die notwendige Dauer der Aktivierung der Motoren keine lineare Funktion des gewünschten

Drehwinkels ist. Die Erstellung einer dementsprechenden Wertetabelle wäre sehr aufwendig und die vorhandenen technischen Unwägbarkeiten (Reibungseffekte, Anlaufverzögerungen der Motoren, Spannungsschwankungen) würden ihre Verlässlichkeit stark einschränken. Somit wäre wahrscheinlich ohnehin meist eine Korrektur des erreichten Winkel notwendig.

Der für eine einzelne Drehung eingesetzte Algorithmus hat die folgende Gestalt:

Ermittle den momentanen Winkel, in dem das Fahrzeug orientiert ist (Bilderkennung).
 Berechne den Differenzwinkel (Restwinkel) zum gewünschten Zielwinkel.
 Ermittle die erforderliche Drehrichtung (positiver Winkel = positiver Drehsinn = links) (Gedreht wird immer um den Winkel, der $\leq 180^\circ$ ist).

Durchlaufe die folgende Schleife je einmal mit den Werten 90, 45, 22, 11, 5 (Grad) für die Variable *Schwelle*:

Wenn $|\text{Restwinkel}| > \text{Schwelle}$, dann
 Drehe für die Dauer $t(\text{Schwelle})$ in Drehrichtung.
 Ermittle den erreichten Winkel und berechne den neuen Restwinkel und die neue Drehrichtung. (Falls über das Ziel hinaus gedreht wurde)

Durchlaufe obige Schleife für $\text{Schwelle}=3$ solange, bis der Fehlwinkel kleiner als 3 ist.

Die Werte $t(\text{Schwelle})$ wurden experimentell ermittelt und stellen die Anzahl von Durchläufen durch eine bestimmte Zählschleife dar. Eine Angabe in Sekunden oder Millisekunden ist nur schwer möglich und auch nicht notwendig. Natürlich gelten diese Werte nur für den speziell verwendeten PC und eine bestimmte Motorspannung (ca. 8V). Um die oben erwähnte Nichtlinearität zu belegen, seien sie an dieser Stelle dennoch angegeben:

Schwelle in Grad	90	45	22	11	5	3
$t(\text{Schwelle})$	3900	2000	1250	800	400	200

Bei all dem ist zu bedenken, daß die von der Bilderkennungskomponente gelieferten Werte bei jeder neuen Aufnahme unterschiedlich stark fehlerbehaftet sind. Deshalb ist eine Genauigkeit von unter 3 Grad nicht sinnvoll und könnte zu Endlosschleifen führen. Abb. 5.1 verdeutlicht dies nochmals.

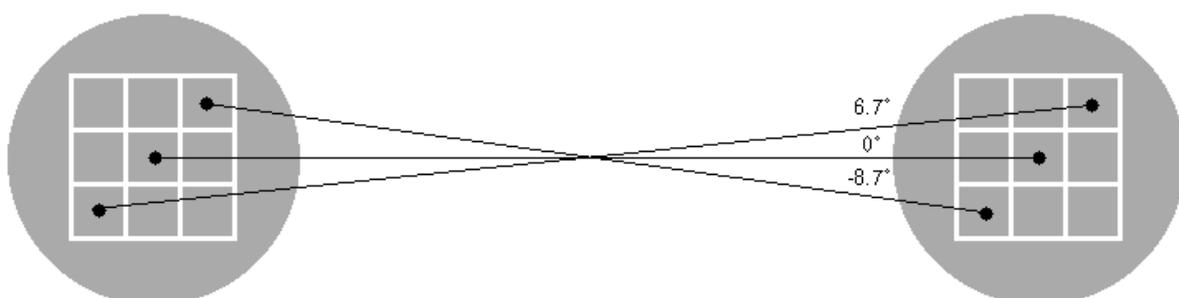


Abb. 5.1 Fehler bei der Grob-Ermittlung der Fahrzeug-Ausrichtung

Schematisch dargestellt sind die beiden äußeren blauen Markierungen auf dem Fahrzeug und stilisierte Pixel. Es wird angenommen, daß die jeweils mittleren der neun Pixel das Optimum der Erkennung darstellen. In Form zweier Beispiele ist der Fehler bei der Winkelberechnung dargestellt, wenn bei der Positionsbestimmung jeder Markierung nur um ein Pixel vom Ideal abgewichen wird. (Der Abstand der Mittelpunkte beträgt in der Realität etwa 15 Px bzw. 45 mm.)

5.1.3 Geradeaus Fahren

Die Durchführung der Geradeausfahrt des Roboterfahrzeuges gestaltet weit weniger trivial, als zunächst zu vermuten ist. Dies liegt in der Bauweise des Antriebes begründet.

Da das Fahrzeug mit zwei getrennt anzusteuern Rädern ausgestattet ist (**Differentialantrieb** vgl. [Jones96], siehe auch Kapitel „Technische Details“), müssen diese sich mit der gleichen Geschwindigkeit drehen, um eine geradlinige Fahrzeugbewegung zu erzielen. Außerdem ist es notwendig, daß sie ihre Drehungen synchron beginnen und beenden. Aufgrund unvermeidbarer mechanischer und elektrischer Unterschiede zwischen den beiden Motor/Getriebe/Rad-Systemen ist eine derartige Übereinstimmung in der Praxis nicht gegeben.

Bei den meisten Robotern, welche mit einem solchen Antrieb ausgestattet sind (z.B. "Khepera", "Rug Warrior"), ist deshalb an jedem Rad ein sogenannter **Rad-Encoder** angebracht. Dabei handelt es sich um einen Sensor, der die vom Rad tatsächlich absolvierten Umdrehungen mißt und an den steuernden Mikroprozessor bzw. Computer meldet. Dieser kann dann die Geschwindigkeit der beiden Räder dynamisch angleichen.

Der Billard-Roboter verfügt über keine derartigen Encoder. Dafür sind mehrere Gründe anzuführen:

1. Der Platzbedarf der Encoder steht den geforderten geringen Abmessungen des Fahrzeuges entgegen.
2. Da auf dem Fahrzeug keinerlei Elektronik montiert ist, müßten die Signale der Encoder über (mindestens 3) weitere Leitungen zum Computer übertragen werden.
3. Der Nutzen der Encoder würde den Aufwand wahrscheinlich nicht rechtfertigen.

Wie im folgenden beschrieben, ist das Problem ist auch auf andere Weise lösbar, welche außerdem der Aufgabenstellung der Sensorik-Motorik-Koordination entgegenkommt.

Die implementierte Lösung beinhaltet wiederum die Kamera als Sensor zur Messung der Abweichung von Ideallinie und die schon bei der Fahrzeug-Drehung eingesetzte "Taktik der kleinen Schritte". Der Algorithmus zum Anfahren eines bestimmten **Zielpunktes**, d.h. einer Station der Route, lautet wie folgt:

Wiederhole für sich verringernde Werte von $Schwelle=20,10,5$ (Pixel als Längenmaß):
Erfasse Position und Ausrichtung des Fahrzeuges.
Berechne den Winkel und die Entfernung zum Zielpunkt (Restweg).
Drehe das Fahrzeug in Richtung Zielpunkt.
Wenn $Schwelle > Restweg$,
dann fahre für die Zeit $t(Schwelle)$ vorwärts. (Aktivierung beider Räder)
sonst verringere $Schwelle$. Abbruch, wenn $Schwelle < 5$.

Bei $t(Schwelle)$ handelt es sich wiederum um experimentell ermittelte Anzahlen von Wiederholungen einer Zählschleife.

Da das Fahrzeug immer nur eine relativ kleine Strecke zurücklegt und seine Fahrtrichtung anschließend wieder korrigiert wird, werden Schwächen bei der Geradeausfahrt kompensiert. Bei den ersten praktischen Versuchen mit diesem Algorithmus betrug der geforderte Restweg 0, d.h. der angestrebte Punkt mußte exakt angefahren werden. Diese Genauigkeit wurde vom System auch ohne Probleme erzielt, allerdings erhöhte sich die Fahrtdauer durch die zahlreichen kleinen Bewegungen gegen Ende der Fahrt beträchtlich. Die Hauptursache dafür ist das relativ langwierige Einlesen der Kamera-Daten über den Parallelport des Computers, denn es muß nach jeder Einzelbewegung (Fahrt und Teildrehung!) ein neues Bild aufgenommen und ausgewertet werden.

Im Interesse eines zügigen Ablaufes der Fahrt nimmt das Fahrzeug deshalb schon dann Kurs auf die nächste Station der Gesamtroute, wenn es von der gerade anzufahrenden Station weniger als 5 Pixel (oder Koordinateneinheiten) bzw. weniger als 15mm (ein Viertel der Fahrzeuglänge) entfernt ist. Die dadurch entstehenden Fehler wirken sich praktisch nicht aus (vgl. Kap. "Analyse der Leistungsfähigkeit des Billardsystems").

Durch das schrittweise Vorwärtstasten werden die oben erläuterten Nachteile des Differentialantriebes im Rahmen der gestellten Anforderungen voll kompensiert.

Eine vollständige Fahrt des Roboters ist in Abb. F 5.2 dargestellt.

5.2 Ausrichtung des Fahrzeuges zum Stoß

Die Algorithmen der Fahrzeugausrichtung entsprechen im wesentlichen den im vorangegangenen Abschnitt beschriebenen. Das Fahrzeug bewegt sich vom Zielpunkt der vorangegangenen Fahrt zum eigentlichen Stoßpunkt, analog der Bewegung zwischen zwei Stationen. Die durchgeführten Aktionen sind wiederum zusammengesetzt aus sich abwechselnden Phasen der Vorwärtsbewegung und der Drehung. Auch werden dabei die Parameter der Bewegungen (zurückzulegender Weg bzw. Drehwinkel) zum Ende der jeweiligen Teilbewegung hin in ihrem Wert verringert.

Im Gegensatz zur reinen Fahrt basiert die Fahrzeugausrichtung auf der genaueren Variante der Fahrzeug-Erkennung (vgl. Kapitel "Bildererkennung"). Durch den notwendigen Einsatz derjenigen Algorithmen, welche die höchstmögliche Genauigkeit bereitstellen, ist die Ausrichtung eine relativ zeitaufwendige Prozedur (bis ca. 3min).

Das Ergebnis der Fahrzeugausrichtung ist von vielen verschiedenen Faktoren abhängig und nicht immer optimal.

Das Thema wird im Kapitel "Analyse der Leistungsfähigkeit des Billardsystems" nochmals aufgegriffen und ausführlicher behandelt.

6 Technische Details

In diesem Kapitel werden technische und konstruktive Einzelheiten des Billardsystems behandelt, welche den Billardtisch, die Mechanik zur Positionierung der Kamera, das Roboterfahrzeug und die Interface-Elektronik betreffen.

Viele der im Folgenden dargestellten konstruktiven Details wurden bereits innerhalb der vorhergehenden Kapitel erwähnt bzw. als existent vorausgesetzt. Insofern kann dieses Kapitel auch als ausführliche Zusammenfassung der technischen Aspekte angesehen werden.

Grundsätzlich kann vorausgeschickt werden, daß außer einigen grundlegenden Bestandteilen des Systems (Kamera, Schrittmotor, Servo-Motoren und Elektronik-Bauteile) auf keinerlei vorgefertigte Komponenten zurückgegriffen wurde. Die speziellen Anforderungen an die Konstruktionen machten es notwendig, die Mehrzahl der verwendeten Einzelteile selbst zu entwerfen und anzufertigen bzw. anfertigen zu lassen. In vielen Fällen wurden auch Bauelemente zweckentfremdet, was eine in der experimentellen Robotik sehr verbreitete Vorgehensweise ist.

Bei der Anfertigung der verschiedenen Bauteile des Billard-Systems kam nur ein Grundstock von einfachen Werkzeugen und Maschinen zum Einsatz. Es handelte sich um eine Tischbohrmaschine, eine elektrische Säge, einen Lötkolben und diverse Werkzeuge zur Bearbeitung von Metall und Holz sowie zur Fertigung elektronischer Schaltungen. Die einzigen nicht vom Autor selbst gefertigten Bauteile sind die beiden Achsen des Roboterfahrzeuges, welche in der erforderlichen Präzision nur von einer Metallbau-Firma auf einer Drehmaschine hergestellt werden konnten.

6.1 Der Billardtisch

In der ursprünglich im Spielwarenhandel erworbenen Form war der Billardtisch für die Zwecke dieser Arbeit nur bedingt geeignet. Die mitgelieferten Beine besaßen nicht die erforderliche Stabilität und wurden deshalb nicht montiert. Statt dessen wurde die eigentliche Spielfläche (im Prinzip eine Holzplatte) mit kleinen Füßen versehen, welche in ihrer Höhe verstellbar sind und dadurch eine exakte waagerechte Justierung der Spielfläche erlauben. Die so modifizierte Platte kann auf einem beliebigen Tisch aufgestellt werden.

An den Taschen (Löchern) des Spielfeldes waren ursprünglich kleine weiße Netze vorhanden, welche die versenkten Kugeln aufnehmen sollten. Da die Aussparungen für die Taschen aber in die Spielfläche hinein ragen und damit Teil des von der Kamera erfaßten Gesamtbildes sind, war es notwendig, diesen hellen Untergrund durch einen dunkleren zu ersetzen. Damit wird vermieden, daß die weißen Netze fälschlich als Kugeln erkannt werden könnten. Natürlich wäre dieser Effekt auch durch die Erkennungsalgorithmen zu vermeiden gewesen, aber eine kleine konstruktive Veränderung am Tisch war einfacher und zeitsparend. Unter den Löchern wurden waagrecht dunkle Holzplatten montiert. Zum Auffangen der versenkten Kugeln wurden zusätzlich transparente Kunststoffplatten senkrecht an den Außenkanten der Platte angebracht.

Jeweils in der Mitte der beiden Stirnseiten der Platte sind senkrechte Träger befestigt, auf denen sich ca. 70 cm über der Spielfläche die waagerechte Schiene befindet, welche den Schlitten mit der Kamera trägt.

6.2 Die Kamera-Mechanik

6.2.1 Prinzipielle Überlegungen

Die Notwendigkeit, die Kamera längs des Spielfeldes beweglich zu montieren, wurde bereits im Abschnitt "Anforderungen an die Bilderkennung" erläutert.

Die Kamera-Mechanik stellt einen bedeutenden Mehraufwand an konstruktiver Arbeit dar, der mit der Lösung der inhaltlichen bzw. wissenschaftlichen Anforderungen an das Projekt Billard-Roboter eigentlich nichts zu tun hat.

Zudem entstehen durch die Lösung dieser Aufgabe eine Vielzahl zusätzlicher Fehlerquellen für die korrekte Arbeit der eigentlich wichtigen und interessanten Algorithmen.

Wie im Kapitel "Analyse der Leistungsfähigkeit des Billardsystems" dargelegt wird, liegt in der beschränkten Präzision der Kamera-Mechanik eine Hauptursache für Fehler bei der Ausrichtung des Fahrzeuges an der Spielkugel.

Zu lösen war die Aufgabe, die lineare Bewegung einer Art von Schlitten über eine Entfernung von ca. 1m zu realisieren, welche eine Positionierung mit hoher **Wiederholgenauigkeit** einschließt. Die Wiederholgenauigkeit bezeichnet die durchschnittliche Abweichung einer erreichten Position vom gewünschten Wert und spielt besonders bei Druckern und Plottern eine große Rolle. Sie sollte beim geplanten Billardsystem deutlich unter 1mm betragen.

Zur Auswahl standen die in Abb. 6.1 veranschaulichten drei prinzipiellen Varianten:

Ein auf einer Schiene rollender oder gleitender Schlitten, dessen Antrieb über einen Seilzug oder Zahnriemen (1) bzw. mit einem direkt montierten Motor (2) erfolgt, oder eine Lösung in Form eines Spindeltriebes (3).

Zahnriemen und -stange bieten, wie auch der Spindeltrieb mittels einer Gewindestange, den Vorteil eines schlupffreien Antriebs. Da aber keines der drei Bauteile in erforderlicher Länge zu einem vertretbaren Preis zu beschaffen war, wurde zunächst mit einem Seilzug experimentiert. Diese Lösung erwies sich, inklusive einiger spezieller Ergänzungen, als praktikabel und ausreichend.

Als Antriebsmotor kam von vornherein nur ein Schrittmotor in Frage (Erläuterungen dazu z.B. in [Prau96]). Nur ein solcher kann, aufgrund seiner aufwendigen Mechanik und der speziellen elektronischen Ansteuerung, zufriedenstellende Ergebnisse bezüglich der nötigen Präzision liefern. Er ermöglicht Positionsveränderungen des Schlittens in der Größenordnung von wenigen Zehntelmillimetern.

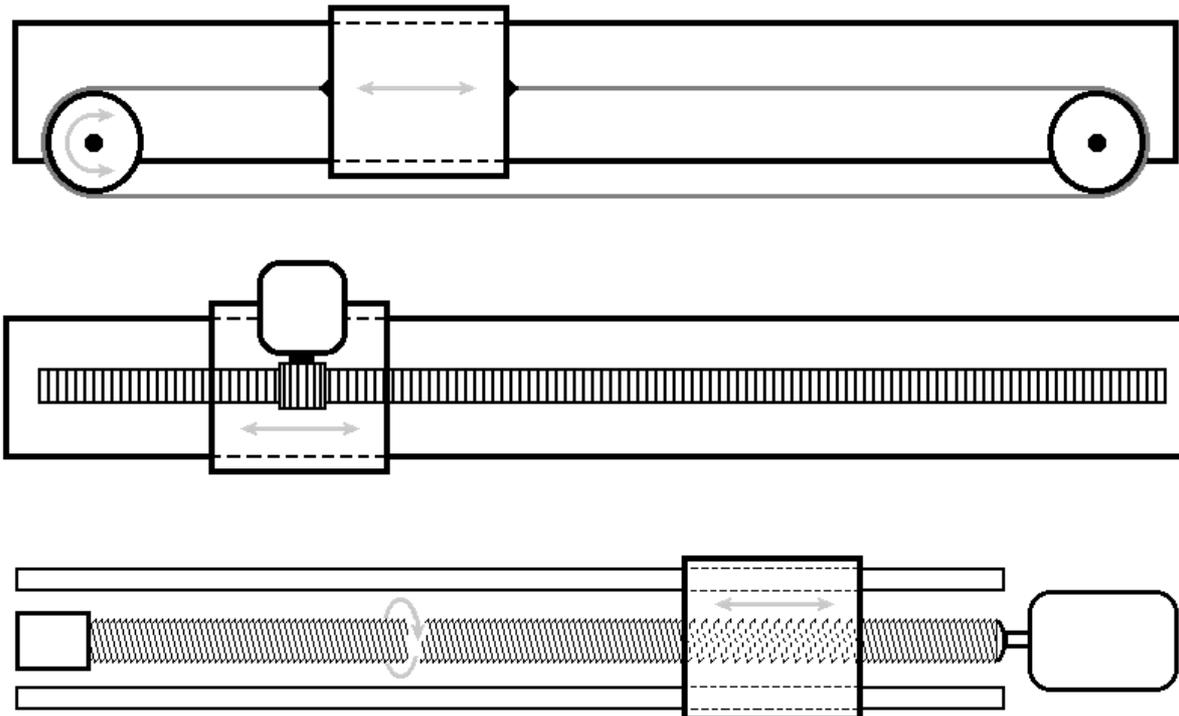


Abb. 6.1 Alternativen für die Realisierung der Linearbewegung des Kamera-Schlittens
 Seilzug/ Zahnriemen mit angetriebener Rolle und Umlenkrolle (oben), Schlitten mit aufgesetztem Motor und Zahnstange (Mitte) sowie Spindeltrieb mit zusätzlichen Führungsstangen (unten)

6.2.2 Realisierung

Der Schlitten selbst ist ein zweiachsiges Fahrzeug, dessen Räder aus vier auf den (feststehenden) Achsen aufgesteckten Kugellagern bestehen. Er bewegt sich auf einer Profilschiene, welche normalerweise in einem Regal-Montagesystem verwendet wird. Das Seil wird über eine Rolle auf der Motorachse und eine Umlenkrolle geführt. Seine Enden sind am Schlitten befestigt. Die Aufhängung der Kamera befindet sich unter dem eigentlichen Schlitten unter der Schiene. Die Anordnung ist vergleichbar mit einer Schwebbahn, deren Gondel durch die Kamera gebildet wird.

Um das eigenwillige kugelförmige Design der Kamera zu kompensieren, ist diese zwischen zwei waagerechten Aluminiumplatten eingespannt, welche mit entsprechend großen runden Aussparungen versehen sind. Diese Lösung macht eine genaue Justierung der Kamera-Ausrichtung möglich, aber auch notwendig.

Wie bereits angedeutet, tritt beim Betrieb des Seilzuges ein nicht zu vernachlässigender Schlupf auf. Die Hauptursache dafür ist die nicht hinreichend reibungsfreie Fahrt des Schlittens auf der Schiene, welche in schwer kompensierbaren Montagetoleranzen begründet liegt. Daraus ergibt sich die Konsequenz, daß eine genaue Positionierung des Schlittens allein aufgrund einer entsprechenden Bewegung der Schrittmotorachse nicht möglich ist. Es ist notwendig, die Position des Schlitten in einer geeigneten Form an den Rechner zurückzumelden.

Zur Erfüllung dieser Aufgabe existieren zwei prinzipielle Möglichkeiten:

1. Eine annähernd kontinuierliche Messung der Position mit Hilfe eines Drehgebers oder fein gestaffelter optischer oder elektrischer Marken.
2. Die Beschränkung auf eine relativ geringe Anzahl von festen anfahrbaren Positionen, welche, wie in der ersten Variante, durch geeignete Marken identifiziert werden.

Die erste Möglichkeit ist zwar im Sinne des Gesamtsystems komfortabler, aber auch bedeutend aufwendiger umzusetzen. Deshalb werden die vom Kamera-Schlitten anfahrbaren Positionen auf die (für die Bild-Groberkennung) notwendigen acht beschränkt. Diese Anzahl ist auch für alle anderen Belange ausreichend.

Für die Rückmeldung beim Erreichen einer Marke war eine berührungslose Variante zu favorisieren. Der zunächst naheliegende Gedanke mehrerer Lichtschranken wurde allerdings zugunsten einer viel einfacheren und kostengünstigeren Lösung verworfen.

Unterhalb der Führungsschiene sind an den entsprechenden Positionen kleine Magnete angebracht. Am Schlitten befindet sich ein sogenannter "Reed"-Kontakt (z.B.[Miel85]). Dieser schließt sich, wenn er in den Einflußbereich eines Magnetfeldes gerät. Die dadurch erzielte Wiederholgenauigkeit ist außerordentlich gut. Aufgrund eines gewissen Hysterese-Effektes ist es allerdings notwendig, daß der Schlitten sich immer von der gleichen Seite annähert. Der Steueralgorithmus wurde deshalb wie folgt implementiert:

1. Fall: Die nächste Markierung befindet sich *rechts* von der aktuellen Schlittenposition.

Wiederhole

 Bewege den Schlitten einen (Motor-) Schritt nach rechts.
 bis der Reed-Kontakt geschlossen ist.
Markierte Position ist erreicht.

2. Fall: Die nächste Markierung befindet sich *links* von der aktuellen Schlittenposition.

Wiederhole

 Bewege den Schlitten einen Schritt nach links.
 bis der Reed-Kontakt geschlossen ist.

Wiederhole

 Bewege den Schlitten einen Schritt nach links.
 bis der Reed-Kontakt wieder geöffnet ist.
Bewege den Schlitten noch eine feste Anzahl Schritte nach links (ca. 2cm)
Weiter wie in Fall 1.

Da die Rückmeldung des Reed-Kontaktes nicht darüber informiert, welche Markierung jeweils erreicht wurde, "tastet" sich die Steuer-Software immer von einer Markierung zur nächsten. Dafür wird beim Start des Programmes eine interne Variable für die Kamera-Position mit einem Anfangswert initialisiert, nachdem der Schlitten zuvor manuell an die entsprechende Position bewegt wurde.

Das Anschlußkabel der Kamera ist an der linken Seite des waagerechten Trägers fixiert und wird nicht gesondert geführt, sondern hängt, je nach aktueller Schlittenposition, mehr oder weniger tief nach unten durch. Dabei gerät sie aber nie in das Blickfeld der Kamera.

Die Abbn. F 6.2 und F 6.3 zeigen den Kameraschlitten bzw. den Schrittmotor.

6.3 Das Roboterfahrzeug

Dieser Abschnitt behandelt die konstruktive Gestaltung des Roboterfahrzeuges. Dabei geht es zunächst um den Stößel, da dieser letztlich das primäre Bauteil des Fahrzeuges darstellt und seine Eigenschaften dessen konstruktive Aspekte weitgehend bestimmen.

6.3.1 Der Stößel

Der Stößel übernimmt im vorliegenden System die Rolle des Billard-Queues und die des stoßenden Armes des menschlichen Spielers. Er wird somit sowohl zur Erzeugung als auch zur Übertragung der Stoßkraft eingesetzt.

Die verwendeten Kugeln bestehen, wie auch die meisten echten Billardkugeln, aus Kunstharz und besitzen eine Masse von ca. 14 Gramm. Die Kraft eines Stoßes muß ausreichen, um die weiße Spielkugel so stark zu beschleunigen, daß sie einen ausreichend starken Impuls auf die von ihr zu treffende Kugel überträgt, so daß diese die ausgewählte Tasche (Loch) des Billardtisches erreichen kann. Dabei muß u.U. eine Entfernung zurückgelegt werden, die der Diagonalen des Spielfeldes entspricht (ca. 111 cm).

Benötigt wird also eine technische Einrichtung, die für eine kurze Zeitdauer eine ausreichend starke, in Längsrichtung wirkende Kraft bereitstellt und dabei selbst möglichst wenig Raum einnimmt. Außerdem sollte die Lösung konstruktiv gut beherrschbar sein.

Der naheliegende Gedanke, eine radiale Motorbewegung mittels einer Zahnstange in eine axiale Bewegung umzuwandeln, erscheint nicht realisierbar, da die Mechanik den hohen auftretenden Kräften sicher nicht gewachsen wäre. Eine Konstruktion, welche auf hydraulischer oder pneumatischer Basis arbeitet, scheidet aufgrund des technischen Aufwandes und des hohen zu erwartenden Platzbedarfs ebenfalls aus.

Als praktikable Lösung erwies sich die Verwendung der elektromagnetischen Kraft, welche von einer stromdurchflossenen Spule entwickelt wird. Im Fachhandel wird eine Vielzahl von sogenannten Zugmagneten angeboten, welche zum Beispiel als elektrisch steuerbare Türöffner oder Magnet-Ventile zum Einsatz kommen.

Die Funktionsweise eines Zugmagneten besteht darin, daß ein magnetischer, zylinderförmiger Kern beim Anlegen einer elektrischen Spannung an eine Spule heran bzw. in diese hinein gezogen wird. Die Spule ist dabei um die Längsachse eines Kunststoffröhrchens gewickelt, dessen Innendurchmesser dem Durchmesser des Kerns entspricht.

Für die Kraft F , welche von einer stromdurchflossenen Spule ausgeübt werden kann, gilt die folgende Beziehung:

$$F \sim H = I \cdot N / l$$

Dabei ist H die magnetische Feldstärke, I die Stromstärke, N die Windungszahl der Spule und l die Länge der Spule. Der Strom I ist proportional zur angelegten Spannung, woraus folgt, daß die erreichbare Stoßkraft des Stößels von den Kennwerten der Spule und der Betriebsspannung abhängig ist.

Nach der Untersuchung der Eigenschaften verschiedener Zugmagnete fiel die Wahl auf einen mit Netzspannung (220 Volt Wechselspannung) zu betreibenden Typ, der zu einem sehr günstigen Preis angeboten wurde. Der Nachteil dieser Lösung ist offensichtlich die Bereitstellung der Spannung, welche praktisch nur mit Hilfe eines Kabels erfolgen kann. Damit war es unumgänglich, das Roboterfahrzeug mit einem Steuerkabel zu versehen.

Aus dem ursprünglichen Zugmagneten wird lediglich die eigentliche Spule (ein Grundkörper aus Kunststoff und die Wicklungen aus Kupferdraht) sowie der Stahlkern verwendet.

Am Kern wurde als Verlängerung ein schmaleres Stück Rundstahl angebracht, an dessen Ende sich ein Holzstück befindet, welches der Spitze eines Queues entspricht. Zur Verbesserung der Kraftübertragung zur Spielkugel wurde die Holzspitze mit einer Gummiauflage versehen.

Abb. F 6.4 zeigt die verwendete Spule und den modifizierten Kern.

In Ruhestellung befindet sich der Stahlkern teilweise außerhalb der Spule. Wird die Spannung angelegt, so wird der Kern in den Spulenkörper hinein gezogen und dadurch die Spitze vorwärts geschoben. Der zurückgelegte Weg beträgt nur etwa 10mm. Trotzdem ist der dabei entstehende mechanische Impuls des Kernes für die gestellten Aufgaben ausreichend.

Zu Beginn der Versuche mit der Magnetspule wurden auch einige Tests zur **Dosierbarkeit** der Stoßkraft durchgeführt.

Dabei kam zunächst eine mittels Phasenanschnittsteuerung realisierte Spannungsregelung zum Einsatz, welche sich in diesem Stadium allerdings als nicht sinnvoll erwies, da ihre Aktivierung manuell über einen Taster erfolgte. Es zeigte sich, daß die Dauer der Spannungsaktivierung einen entscheidenden Einfluß auf den resultierenden Stoß hat. Durch eine Steuerung dieser Dauer durch den Computer kann die Stoßkraft in gewissen Grenzen ebenfalls dosiert werden.

Problematisch dabei ist, daß der praktisch in der Spule liegende Stahlkern erst eine gewisse Gleitreibung überwinden muß, bevor er sich bewegt. Somit ergibt sich eine geknickte Zeit-Kraft-Kennlinie, d.h. unterhalb einer bestimmten Zeitschwelle bewegt der Kern sich gar nicht und bei deren Überschreitung wird er sofort relativ stark beschleunigt. Aus diesem Grund ist die erreichbare "Sanftheit" eines Stoßes beschränkt. Somit ist das im realen Billard bisweilen angewandte "Zurechtlegen" von Kugeln mit der vorliegenden Konstruktion nicht umsetzbar, da es oft ein sehr saches Anspielen voraussetzt.

Für den im Billardsystem durchgeführten einfachen, direkten Stoß ist eine Dosierung der Stoßkraft nicht zwingend notwendig. Es muß lediglich sichergestellt sein, daß die erreichbare Kraft groß genug ist, um die Kugeln in der oben beschriebenen Art zu bewegen. Dies ist jedoch gewährleistet.

Ohne eine Begrenzung der Aktivierungsdauer reicht die vom Stößel entwickelte Kraft aus, um eine einzelne Kugel mehr als dreimal in Spielfeld-Längsrichtung hin und zurück zu bewegen (ca. 350 cm bei 3 Bandenkontakten, welche den Impuls der Kugel zusätzlich zur Rollreibung verringern).

Auf die Verwendung einer Spannungsregelung wurde schließlich verzichtet. Die Zuschaltung der Netzspannung erfolgt durch ein elektronisches Relais, welches direkt mit den am Parallelport des Computers bereitstellbaren Signalspannungen gesteuert werden kann.

6.3.2 Grundlegende Anforderungen an das Roboterfahrzeug

Die Aufgabe des Roboterfahrzeuges besteht darin, den Stößel zur Position des beabsichtigten Stoßes zu transportieren und dort (möglichst präzise) auszurichten. Dabei darf während der Transportphase keine der auf dem Spielfeld liegenden Kugeln berührt werden.

Für die konstruktive Gestaltung des Fahrzeuges waren u.a. die folgenden Kriterien maßgebend:

1. **geringe Abmessungen und Kompaktheit**

Da das Fahrzeug sich zwischen den Kugeln bewegen muß, sollten seine Abmessungen den Wert von etwa 2 Kugeldurchmessern (2x ca. 28 mm) nicht wesentlich überschreiten.

2. **technologische Beherrschbarkeit**
Die gewählte Konstruktion muß mit den zur Verfügung stehenden Bauteilen und Werkzeugen umsetzbar sein.
3. **mechanische Robustheit und Zuverlässigkeit**
Die Konstruktion soll auch eine längere Beanspruchung ohne Defekte meistern und besonders den in Augenblick des Stoßes auftretenden Kräften gewachsen sein.
Die Steuerbefehle sollen in gleichbleibender Qualität in die gewünschte Bewegung umgesetzt werden.
4. **gute Manövrierfähigkeit mittels einfacher Algorithmen**
Das Antriebsprinzip soll es ermöglichen, die notwendigen Steuerbefehle mit geringem algorithmischen Aufwand zu berechnen, ohne dabei zu einer eingeschränkten Bewegungsfähigkeit des Fahrzeuges zu führen.
5. **einfache elektronische Ansteuerung über wenige Leitungen**
Die zur Steuerung des Antriebes benötigte Hardware (Rechner-Fahrzeug-Schnittstelle) sollte sich auf ein Minimum beschränken. Insbesondere soll die Anzahl der direkt zum Fahrzeug führenden Kabel so gering wie möglich sein.
6. **gute optische Erkennbarkeit**
Die Oberseite des Fahrzeuges sollte so gestaltet sein, daß die für die Bilderkennung verwendeten Markierungen (vgl. Abschnitt "Optische Erkennung des Roboterfahrzeuges") problemlos angebracht werden können.

Die zu lösende Aufgabe bestand darin, den bereits existierenden Stößel um einen Antrieb und eine mechanische Verkleidung zu ergänzen, welche diesen Kriterien gerecht werden.

6.3.3 Wahl der Antriebsart

Wie schon weiter oben ausgeführt wurde, können mobile Roboter mit den verschiedensten Antriebsmechanismen ausgestattet sein. Aus Gründen der technologischen Beherrschbarkeit, der geforderten Abmessungen und der Verfügbarkeit von Bauteilen kamen für den Billardroboter aber von vornherein nur Räder und Elektromotoren in Frage.

Im Folgenden sollen kurz verschiedene zur Auswahl stehende Prinzipien der Radanordnung und Steuerung erläutert werden. Die Darstellung lehnt sich an [Jones96] an.

Differentialantrieb: Je ein von einem eigenen Motor angetriebenes Rad an der rechten und an der linken Seite sowie stützende Freilaufäder vorn und/oder hinten.
Lenken erfolgt durch Drehung auf der Stelle, indem beide Räder gegenläufig arbeiten.

Prinzipiell vergleichbar mit einem muskelbetriebenen Rollstuhl.

Nachteile: - Für Geradeausfahrt müssen beide Räder exakt synchron laufen.
- relativ geringe Eigenstabilität im Stand

Vorteile: - wenige Bauteile und damit geringer Platzbedarf und Fehlerquellen.

- einfache Steueralgorithmen (Turtle-Prinzip).

Dreiradantrieb/

Ackermannlenkung: Eine angetriebene Hinterachse mit zwei Rädern und ein lenkbares Vorderrad (Dreirad) bzw. zwei miteinander verbundene lenkbare Vorderräder. Lenken ist nur während der Fahrt möglich. Es entsteht ein Lenk(Wende-)radius. Kommt beim Trike (dreirädriges Motorrad) bzw. bei (heckgetriebenen) Autos zur Anwendung.

- Nachteile: - relativ aufwendige Lenkmechanik, erhöhter Platzbedarf,
- schwierige Steueralgorithmen (Beispiel Einparken !)
- Vorteile: - Ist in vielen Spielzeugautos bereits "implementiert".
- Kann problemlos geradeaus fahren.

Synchronantrieb: Drei oder mehr Räder, von denen mindestens eins angetrieben ist, werden zur Richtungsänderung alle synchron um denselben Winkel geschwenkt. Das Fahrzeug kann damit praktisch "aus dem Stand" in jede beliebige Richtung fahren.

- Nachteile: - sehr aufwendige Mechanik
- Wenn eine Ausrichtung des Fahrzeuges in eine bestimmte Richtung erforderlich ist (trifft auf den Billard-Roboter zu), muß eine zusätzliche Möglichkeit zur Drehung des Fahrzeug-Oberteiles auf dem eigentlichen Antrieb vorgesehen werden.
- Vorteile: - sehr elegante und einfache Steuerung

Wie leicht einzusehen ist, fiel die Wahl auf den Differentialantrieb. Auf zusätzliche Stützrollen oder -räder wurde allerdings verzichtet. Das Fahrzeug gleitet an seiner Heckseite auf einer schmalen Auflagefläche aus Acrylglas.

Die Umgehung der dem Differentialantrieb eigenen Nachteile genannten Nachteile wurde bereits im Abschnitt "Fahrt zur Stoßposition" (Kap. "Kamera-Fahrzeug-Koordination") erläutert.

6.3.4 Technische Umsetzung

Das Hauptproblem beim Entwurf des Roboterfahrzeuges bestand in den geforderten geringen Abmessungen bei gleichzeitiger mechanischer Robustheit und Zuverlässigkeit. Geringe Abmessungen erfordern möglichst kleine Bauteile, deren Stabilität und Belastbarkeit wiederum begrenzt ist.

Desweiteren sind aufwendige mechanische Konstruktionen auf kleinem Raum natürlich schwieriger zu integrieren, so daß eine möglichst geringe Anzahl von Einzelteilen anzustreben ist. Wenige Einzelteile reduzieren außerdem die Störanfälligkeit von technischen Systemen.

In das Roboterfahrzeug waren die folgenden Komponenten zu integrieren:

1. der Stößel
2. 2 Antriebsräder
3. 2 Antriebsmotoren
4. 2 Getriebe zur Kraftübertragung zwischen den Motoren und den Rädern
5. Einrichtungen zur Stabilisierung (Abstützung) des Fahrzeuges vorn und/oder hinten
6. eine Oberseite (Abdeckung) zur Aufnahme der Markierungen und elektrischen Anschlüsse

All diese Teile waren mit Hilfe eines geeigneten Chassis miteinander zu verbinden.

Dabei wirkten sich die gegebenen Abmessungen der Kugeln und der Stößel-Spule erschwerend aus. Die Ausführung des im Billardsystem vorgesehenen zentralen Stoßes setzt voraus, daß sich die Längsachse des Stößels in der Höhe des Kugelmittelpunktes befindet. Zufällig entspricht die Höhe des Spulenkörpers fast genau dem Kugeldurchmesser, so daß diese Bedingung bereits erfüllt ist. Allerdings war es dadurch nicht möglich, die Spule und das gesamte Fahrzeug auf einer Grundplatte als Basis aufzubauen, da dadurch die Stößel-Achse über den Kugelmittelpunkt angehoben worden wäre.

Somit mußte das gesamte Fahrzeug im wörtlichen Sinne "um den Stößel herum" gebaut werden.

Die Umsetzung des Chassis besteht in einem, von oben betrachtet, U-förmigen Grundgerüst aus drei Aluminiumplatten, welche miteinander vernietet wurden. Dabei zeigt die mittlere Platte in die Fahrt- bzw. Stoßrichtung des Roboters. Sie verfügt in ihrer Mitte über eine Bohrung, welche den Stößel führt.

In jede der beiden seitlichen Platten ist eine Stahlachse eingeschraubt, welche ein Rad trägt. Das heißt, daß das Fahrzeug über keine durchgängige Achse verfügt, da sich die Spule zwischen den Rädern befindet. Die beiden einzelnen Achsen stellten erhöhte Anforderungen an die Fertigung des Chassis, da letztlich gewährt sein muß, daß die Räder sich auf einer gemeinsamen geometrischen Achse befinden, welche im rechten Winkel zur Stoßachse steht.

Die Räder selbst sind Zahnräder aus Stahl, welche in sich sowohl die Funktion der Kraftübertragung vom jeweiligen Motor, als auch die des auf der Spielfläche rollenden Rades selbst vereinigen. Diese unkonventionelle Lösung, die auf eine Kombination eines Zahnrades mit einem "richtigen" Rad (z.B. mit Gummibereifung) verzichtet, verringerte die Anzahl der Bauteile und die Fahrzeugbreite, ein entscheidendes Kriterium für eine mögliche Passage des Roboters zwischen zwei Kugeln.

Die Stößel-Spule ist in das U-Teil eingesetzt und wird durch eine vierte Aluminiumplatte fixiert, welche von hinten, an die offene Seite des "U", angeschraubt ist. Auch diese Platte besitzt in ihrer Mitte eine Bohrung, aus der in Ruhestellung der Stahlkern des Stößels herausragt.

Über der Spule befinden sich die beiden Antriebsmotoren, welche an den nach oben verlängerten Seitenplatten befestigt sind. Es handelt sich dabei um Servo-Motoren (siehe u.a. [Jones96]), die im Modellbauhandel angeboten und normalerweise z.B. für die Steuerung von Schiffs- oder Flugzeugrudern eingesetzt werden. Ihre Hauptvorteile liegen in den kleinen Abmessungen und dem bereits integrierten Getriebe.

Servomotoren enthalten eine eigene Elektronik, die dazu dient, mittels eines pulsbreitenmodulierten Steuersignals die Motorachse auf einen bestimmten Winkel zu drehen und in dieser Position zu halten. Um sie für ihre im Billardprojekt vorliegende

Zweckbestimmung als Antriebsmotoren umzurüsten, wurden die Elektronik und einige mechanische Sperren aus den Gehäusen entfernt, so daß wieder eine kontinuierliche Rotationsbewegung der Motorachse entsteht. Die verwendeten Servomotoren gehören zur kleinsten erhältlichen Bauart "Super-Micro" und sind nur etwa 22×11×20mm groß. Auf jeder der Motorachsen ist ein Zahnrad aus Kunststoff montiert, welches in das jeweilige Antriebs(Zahn-)rad greift.

Der für einen stabilen Stand des Fahrzeuges notwendige dritte Stützpunkt (außer den zwei Rädern) befindet sich am Heck und damit unter dem relativ schweren Stahlkern des Stößels. Er ist in Form einer Kunststoffleiste realisiert, welche über die gesamte Fahrzeugbreite reicht. Während seiner Fahrt gleitet der Roboter auf dieser Fläche, ohne daß sich die auftretende Reibung negativ auswirkt.

Die Oberseite des Fahrzeuges besteht aus einer Art Dach aus schwarzem Kunststoff, auf dem die zur Fahrzeugerkennung notwendigen Markierungen aufgebracht sind (siehe Abschnitt "Optische Erkennung des Roboterfahrzeuges").

Die zur Steuerung des Fahrzeuges und des Stößels benötigten Anschlußleitungen (2×2 für die Motoren und 2 für die Spule) werden in der Mitte der Fahrzeugoberseite aufgesteckt. Für die Leitungen selbst kommt kein gemeinsames Kabel mit sechs Adern zum Einsatz, da dieses eine eingeschränkte Flexibilität mit sich bringen würde. Außerdem könnte der Durchmesser eines solchen Kabels dazu führen, daß bestimmte für die Bilderkennung wichtige Objekte (Fahrzeugmarkierungen, Kugeln) ganz oder teilweise verdeckt würden. Zum Einsatz kommen sechs einzelne flexible Drähte, welche aller 10cm mit Klebeband zusammengehalten werden. Dadurch wird dieses Steuerkabel sehr flexibel und die Kamera kann sozusagen zwischen den einzelnen Leitungen "hindurch sehen".

Die konstruktiven Details des Roboters zeigt Abb. F 6.5.

6.4 Die Verbindung zum Steuercomputer

Die Steuerung des Roboterfahrzeuges und des Schrittmotors zur Kamerabewegung erfolgt über eine parallele Schnittstelle (vgl. [Mess97]). Diese wurde zusätzlich zur standardmäßig vorhandenen, aber bereits durch die Kamera belegten, in den steuernden Personalcomputer eingebaut. Genutzt werden insgesamt 8 Bit des Daten- bzw. Steuerports der Schnittstelle als Ausgänge. Dabei entfallen 4 Bit auf die Servomotoren, 3 Bit auf den Schrittmotor und 1 Bit zur Aktivierung des Stößels. Ein Bit des Statusports dient als Eingang zur Rückmeldung der Kamera-Position (vgl. Abschnitt "Die Kamera-Mechanik").

Die unmittelbare Ansteuerung der Servomotoren übernimmt ein spezieller Motortreiber-Schaltkreis ([Thom93]), die der Stößelspule ein elektronisches Halbleiterrelais, welches unmittelbar an den Computerport angeschlossen werden kann und die benötigte Netzspannung schaltet. Diese Bauteile sind, inklusive einer Spannungsstabilisierung für den Motortreiber-IC und eines "Not-Aus-Schalters" in einem eigenen Kunststoffgehäuse zusammengefaßt.

Zum Betrieb des Billardsystems ist ein zusätzliches stabilisiertes Netzteil erforderlich, welches die Versorgungsspannungen für die Servos und den Schrittmotor bereitstellt.

Der verwendete PC hat folgende technische Kennwerte:

Prozessor	Cyrix 686 120+ (entspr. etwa Pentium 150 MHz)
L2-Cache	512 KByte
Hauptspeicher	16 MByte
Bussystem	PCI
Grafikkarte	2 MByte Speicher
Betriebssystem	Microsoft Windows 3.1 .

Die Programmierung erfolgte unter Verwendung von Borland-Pascal 7.0.

Zur Steuerung und Überwachung der Abläufe im Billardsystem wurde eine eigene Programmapplikation erstellt und eingesetzt.

7 Analyse der Leistungsfähigkeit des Billardsystems

In diesem Kapitel wird der Versuch unternommen, die Leistungsfähigkeit und Funktionssicherheit der einzelnen Teilalgorithmen des Billardsystems einzuschätzen.

Die Frage nach geeigneten Kriterien, aufgrund derer die verschiedenen Verfahren und das Gesamtsystem beurteilt werden können, ist nicht leicht zu beantworten. Es kann aber die allgemeine Aussage zugrunde gelegt werden, daß das System dann zufriedenstellend arbeitet, wenn die geforderten **Restriktionen erfüllt** werden (korrektes Erkennen aller Kugeln, Bewegung des Fahrzeuges ohne Kollisionen mit Kugeln, alle Abläufe terminieren) und ein hinreichend großer Anteil von Stößen zum erwünschten **Versenken von Kugeln** führt.

In diesem Sinne beschäftigen sich die zwei Abschnitte des Kapitels mit den Schritten, welche zur Vorbereitung von Stößen absolviert werden müssen bzw. mit der Güte der Stöße selbst. Wenn im Folgenden Fehler beschrieben werden, ohne deren Häufigkeit quantitativ auszudrücken, so handelt es sich in jedem Fall um Effekte, welche während der gesamten Zeit der Entwicklung und Erprobung des Systems so selten auftraten, daß sie im Prinzip vernachlässigbar sind und ihre geringe Signifikanz keine entsprechenden, gezielten Versuchsreihen rechtfertigen würde.

7.1 Punkte 1 bis 4 des Stoßzyklus

7.1.1 Bilderkennung

Die **Groberkennung der Kugeln** erfolgt nahezu ausnahmslos fehlerfrei, d.h. es werden nur in einer vernachlässigbar geringen Anzahl von Fällen Kugeln übersehen, Farben falsch zugeordnet oder Kugeln an Stellen erkannt, an denen keine Kugel vorhanden ist.

Voraussetzung für diese Ergebnisse ist die im Kapitel "Bilderkennung" erwähnte manuelle Justierung der Kamera-Parameter, welche aber unkritisch ist, da sie keine engen Toleranzen aufweist.

Die wenigen beobachteten Fehler entstehen durch eine unzureichende Justierung, bei extremen Schatten auf der Spielfläche oder mitunter bei starker Verdeckung von Kugeln durch die Steuerleitungen des Roboterfahrzeuges.

Die Genauigkeit der ermittelten (groben) Kugelkoordinaten kann nicht explizit angegeben werden, ist aber in jedem Fall ausreichend, da die vorhandenen Fehler sich nicht erkennbar auf die folgenden Schritte des Stoßzyklus auswirken.

Die Komponente der **Kugel-Feinerkennung** arbeitet ebenfalls praktisch fehlerfrei in dem Sinne, daß die Umrisse der jeweiligen Kugel zuverlässig erkannt werden und daraus der im Bildausschnitt erscheinende Kugelmittelpunkt berechnet wird.

Geringe Abweichungen in den Koordinatenwerten entstehen dadurch, daß die zugrundeliegenden Bezugskoordinaten (Position des Bildausschnittes im Gesamtkoordinatensystem) bereits fehlerbehaftet sind. Dies ist aber überwiegend auf konstruktive Aspekte der Kameramechanik zurückzuführen und die Fehler somit nicht der eigentlichen Bilderkennung anzulasten.

Die **Fahrzeug-Groberkennung** erfüllt ihre Aufgabe nicht immer im ersten Versuch, was jedoch in den meisten Fällen erkannt und durch ihre wiederholte Durchführung kompensiert wird (vgl. Abschnitt "Optische Erkennung des Roboterfahrzeuges / Schnellere Variante").

In einer nicht signifikanten Anzahl von Erkennungsversuchen wird die Richtung des Fahrzeuges falsch ermittelt, was aber während der Fahrt des Fahrzeuges nicht zu

Fehlbewegungen im engeren Sinne führt. Sollte sich das Fahrzeug in eine falsche Richtung gedreht bzw. bewegt haben, wird es durch den Algorithmus der Fahrtsteuerung (vgl. Kap. "Kamera-Fahrzeug-Koordination") automatisch wieder auf "den richtigen Weg" zurück geführt.

Die Ursache dieser Fehler liegt in der recht einfachen Verifizierung der ermittelten Koordinaten der blauen Markierungen (vgl. entspr. Abschnitt im Kap. "Bilderkennung").

Die sechs weißen Markierungen des Fahrzeuges werden von der **Fahrzeug-Feinerkennung** nahezu fehlerfrei erkannt. Abweichungen von den Idealwerten entstehen zum einen dadurch, daß der von der vorausgehenden Fahrzeug-Groberkennung ermittelte Orientierungswinkel einen relativ hohen Fehler aufweist und die weißen Markierungen nicht im jeweils untersuchten 7×7 Px-Fenster liegen (vgl. Abschnitt "Optische Erkennung des Roboterfahrzeuges / Genauere Variante"). Zum anderen haben das Bildrauschen und die auflösungsbedingten Diskretisierungsfehler der Kamera bei diesem Algorithmus einen relativ großen Einfluß, da die Markierungen in ihrer Größe nur etwa einem Pixel entsprechen.

7.1.2 Stoßwahl

Der relativ simple Stoßwahl-Algorithmus (siehe auch Kap. "Ausblick") erfüllt seine Aufgabe seinen Möglichkeiten entsprechend und fehlerfrei. Da er streng deterministisch arbeitet, ermittelt er in jedem Falle den aufgrund seiner Bewertungsfunktion optimalen Stoß. Dabei ist er natürlich voll und ganz auf die von der Bilderkennungskomponente zur Verfügung gestellten Daten angewiesen.

Sollte kein durchführbarer Stoß vorhanden sein (vgl. Abschnitt "Restriktionen und Parameter der Stoßwahl"), so wird zuverlässig eine entsprechende Meldung ausgegeben.

7.1.3 Wegplanung

Die Komponente Wegplanung arbeitet ebenfalls rein deterministisch und liefert stets das der ihr übergebenen Spielfeldkonstellation entsprechende Ergebnis in Form des sichersten Weges (vgl. Kapitel "Wegplanung"). Auch hier wird in den Fällen, in denen kein für das Fahrzeug gangbarer Weg existiert, eine Meldung ausgegeben.

7.1.4 Fahrt

Die im Kapitel "Kamera-Fahrzeug-Koordination" beschriebenen Algorithmen, welche die Bewegung des Fahrzeuges von einem Startpunkt bis in die Nähe des berechneten Stoßpunktes steuern und überwachen, arbeiten innerhalb der geforderten Restriktionen und sind in der überwiegenden Mehrzahl der Fälle zuverlässig.

Innerhalb dieses Abschnittes des Stoßzyklus kommt es (neben der folgenden Fahrzeug-Ausrichtung) zur höchstmöglichen Integration der einzelnen Systemkomponenten in eine gemeinsame Aufgabe. Daher pflanzen sich eventuelle Fehler schnell fort und potenzieren sich unter Umständen.

Die Berührung von Kugeln während der Fahrt tritt sehr selten auf und ist dann zurückzuführen auf eine kritische Kugelkonstellation (sehr enge Durchfahrt) und parallel dazu vergleichsweise große Abweichungen der Bilderkennung und der Fahrzeugmotorik vom Idealverhalten.

Bisweilen kommt es zu Endlosschleifen bei der Erfassung der Fahrzeugposition, die in der teilweisen Verdeckung der Markierungen durch die Steuerleitungen begründet liegen. Solche Situationen sind nur durch eine kurze manuelle Hilfeleistung von außen zu bewältigen, indem die Lage der Kabel leicht verändert wird.

7.2 Treffergenauigkeit

7.2.1 "Benchmark"-Test

Um die Trefferquote des Systems bei der Durchführung von Billardstößen einschätzen zu können, wurde eine einfache Testreihe absolviert.

Dazu wurden jeweils zwei Kugeln (weiße Spielkugel und eine gelbe, zu versenkende Kugel) gezielt an bestimmten Positionen der Spielfläche plaziert. Dabei wurden verschiedene Kombinationen in den Entfernungen der Kugeln voneinander und von der anzuvisierenden Tasche sowie unterschiedliche Winkel der Verbindungsgeraden zwischen den beiden Kugeln bzw. der gelben Kugel und der Tasche ausgewählt.

Abb. 7.1 soll die untersuchten Parameter veranschaulichen. Durchgeführt wurden sowohl Stöße auf eine Seitentasche als auch auf eine Ecktasche.

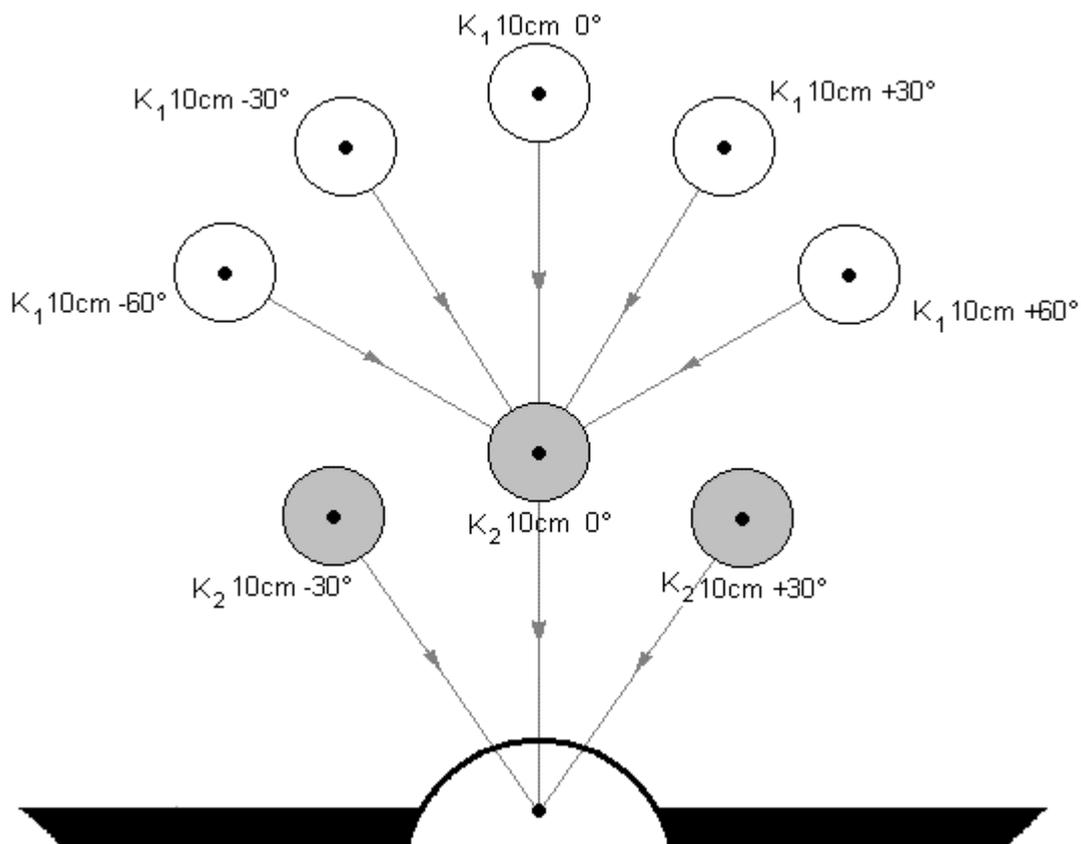


Abb. 7.1 Vorgehensweise beim Test der Trefferquote

Das Schema zeigt das Beispiel der Seitentasche. Die Spielkugel ist mit K_1 , die zu versenkende Kugel mit K_2 bezeichnet. Die Entfernungen von K_2 sind relativ zum **Zielpunkt** in der Mitte der Tasche angegeben, die Winkel beziehen sich auf die durch den Zielpunkt verlaufende Senkrechte zur Bande.

Die Parameter von K_1 ergeben sich jeweils relativ zur Position von K_2 , insbesondere beziehen sich die Winkel auf die Abweichung der Verbindungsgeraden K_1K_2 von der Geraden, welche von K_2 zum Zielpunkt führt.

Beim Stoß auf die Seitentasche entspricht die Bezugsgerade für die Winkelmessung von K_2 der Winkelhalbierenden des rechten Winkels, welcher von den beiden angrenzenden Banden gebildet wird.

Die Ergebnisse der Testreihe sind in den folgenden Tabellen dargestellt. Dabei wurde jeder Versuch nur einmal ausgeführt. Die Resultate tragen also nicht den Charakter von echten Meßwerten, sondern stellen eher eine Art Bestandsaufnahme dar.

K ₂ (gelb)	10cm 0°	20cm 0°	10cm +30°	10cm -30°
K ₁ (weiß)				
10cm 0°	•	•	•	•
10cm +30°	•	•	! 1cm	•
10cm -30°	•	•	•	! 1cm
10cm +60°	! 2cm	! 8cm	-	-
10cm -60°	! 2cm	! 2cm	-	-
20cm 0°	•	•	•	•
20cm +30°	•	! 12cm	! 1cm	•
20cm -30°	! 5cm	•	•	! 1cm
20cm +60°	! 3cm	! 3cm	-	-
20cm -60°	! K ₂ nicht getroffen	! 8cm	-	-
30cm 0°	•	-	-	-
30cm +30°	! 2cm	-	-	-
30cm -30°	•	-	-	-
30cm +60°	! 4cm	-	-	-
30cm -60°	! K ₂ nicht getroffen	-	-	-

Tab. 7.1 Ergebnisse der Stöße auf die Seitentasche

Die Symbole bedeuten im Einzelnen:

- K₂ wurde versenkt
- ! K₂ wurde nicht versenkt, der angegebene Wert entspricht der Distanz, um welche die Tasche verfehlt wurde (Schätzwert)
- Ein Stoß mit den betreffenden Parametern wurde nicht durchgeführt.

In den zwei gesondert ausgewiesenen Fällen hat die Spielkugel die zu versenkende Kugel verfehlt.

K ₂ (gelb)	10cm 0°	20cm 0°
K ₁ (weiß)		
10cm 0°	•	•
10cm +30°	!	•
10cm -30°	•	•
10cm +45°	•	!
10cm -45°	•	!
20cm 0°	•	•
20cm +30°	•	•
20cm -30°	•	•
20cm +45°	!	!
20cm -45°	•	!

Tab. 7.2 Ergebnisse der Stöße auf die Ecktasche

Als Resultat dieser Versuche kann eingeschätzt werden, daß die Trefferquote des Billardsystems sich in den entsprechenden Fällen um ca. 50% bewegt und als durchaus akzeptabel zu bezeichnen ist.

7.2.2 Fehlerquellen bei der Fahrzeugausrichtung

Die folgende Übersicht zählt verschiedene Gründe für auftretende Fehler bei der Ausrichtung des Fahrzeuges auf. Eine quantitative Ermittlung der Einflüsse wurde nicht durchgeführt, da sich die Effekte im Allgemeinen einer meßtechnischen Erfassung mit den zur Verfügung stehenden Mitteln entziehen.

In Fällen, in denen eine solche Erfassung möglich war, wurde diese durchgeführt und die Ergebnisse zur Korrektur von Technik und Algorithmen eingesetzt. Die Übersicht enthält somit nur die Probleme, deren Kompensation mit vertretbarem Aufwand nicht gelang.

- Montagetoleranzen des Billardtisches
 - Es kann nicht davon ausgegangen werden, daß der verwendete Billardtisch in seinen Fertigungsparametern den Anforderungen an die Maßhaltigkeit eines echten Billardtisches genügt.
- Montagetoleranzen der Kamera-Schiene
 - Die exakte Lage der Schiene über der Längsachse des Spielfeldes ist aufgrund der relativ einfachen Konstruktion des Aufbaus nicht garantiert.
 - Die Schiene selbst ist nicht ideal gerade, insbesondere ist sie sogar in sich leicht verdreht (Torsion).
- Montagetoleranzen des Kamera-Schlittens
 - Zu nennen ist vor allem die nicht triviale Justierung der Kamera zwischen den beiden fixierenden Platten. Auf die an der Kamera vorhandene Standard-Stativbefestigung wurde bewußt nicht zurückgegriffen, da sie praktisch nur einen einzigen Punkt zur Fixierung bereitstellen würde.
- Fehler bei der Kamera-Positionierung
 - Die Positionen der Magnete, auf welche das im Kameraschlitten integrierte Reed-Relais anspricht (vgl. Kap. "Technische Details") wurden experimentell ermittelt und sind deshalb nicht ideal. Sichtbar wird dieser Fakt u.a. in verschiedenen Abbildungen der vorliegenden Arbeit. Dort erscheinen vereinzelt Kugeln, welche sich an den Grenzen zwischen zwei Teilbildern ("Streifen") befinden, teilweise unvollständig.
- Montagetoleranzen des Fahrzeuges
 - Die Hochachse des Fahrzeuges, um welche die Drehung erfolgt, verläuft mit Sicherheit nicht exakt durch den von den Markierungen bestimmten Fahrzeugmittelpunkt.
- Beschränkung der Erkennungsgenauigkeit der Fahrzeug-Markierungen
- beschränkte Genauigkeit der Kugel-Feinerkennung

(Die beiden letzten Punkte wurden bereits an den entsprechenden Stellen erläutert.)

8 Ausblick

In seiner vorliegenden Form bietet das Billardroboter-System eine Vielzahl von Ansatzpunkten, um die eingesetzten Verfahren und Komponenten zu verbessern und zu erweitern.

Von großem Vorteil wäre der Einsatz einer qualitativ höherwertigen Kamera, insbesondere einer solchen mit einer höheren Bildpunktzahl. Bei Auflösungen ab ca. 800 Px in jeder Dimension wäre es möglich, die Kamera stationär über dem Feld zu montieren und somit alle Probleme zu umgehen, welche mit der Kameramechanik und den acht Teilbildern verbunden sind. Desweiteren wäre eine schnellere Kamera-Rechner-Schnittstelle vorteilhaft für die praktische Arbeit und Demonstration des Systems.

Der Einsatz einer solchen, leistungsfähigeren Kamera ist auch allen Bemühungen vorzuziehen, die Kameramechanik zu verbessern, da der diesbezügliche Aufwand sicher bedeutend höher wäre als die Kosten der Kamera. Die absehbare technische Entwicklung deutet darauf hin, daß das Angebotsspektrum für Bilderkennungs-Hardware sich stark erweitern wird und deren Kosten kontinuierlich sinken werden.

Verbesserungen der Eigenschaften des Roboters selbst (z.B. Fähigkeit zu dosierten Stößen) sind ebenfalls wünschenswert, erfordern aber professionelle Fertigungstechnik. Insbesondere eine kabellose Steuerung wäre von Vorteil, wirft aber das grundsätzliche Problem der Energiezufuhr für den Stößel auf, der dann wahrscheinlich grundsätzlich anders zu konstruieren wäre. Wie eine solche Alternative aussehen könnte ist jedoch nicht bekannt.

Reizvoll wäre der Versuch, das System auf einen echten Billardtisch zu portieren. Die Hauptprobleme bestünden hierbei wiederum in der zu verwendenden Kamera und der aufzubringenden Stoßkraft.

Die aus wissenschaftlicher Sicht interessantesten Ansatzpunkte bieten die Stoßwahlalgorithmen und die Umsetzung des Konzeptes "Lernen" in vielfältiger Form.

Mit dem Ziel der Annäherung des Stoßrepertoires des Roboters an die Fähigkeiten menschlicher Billardspieler bietet sich die Umsetzung zahlreicher technischer und taktischer Varianten der Stoßwahl an. Beispiele sind die Einbeziehung der Banden sowie das vorausschauende Spiel, welches auch nachfolgende Stöße in Planung einbezieht. Für die Umsetzung dieser Ziele stehen u.a. sowohl heuristische Verfahren als auch neuere Algorithmen der nichtklassischen Optimierung bzw. der Simulation zu Auswahl.

Voraussetzung dazu wäre eine hohe Verlässlichkeit der mechanischen Komponenten.

Um die Fähigkeiten des Systems zu entwickeln bzw. zu verbessern existieren verschiedene Möglichkeiten des Einsatzes von Lernalgorithmen. Dies beginnt bei der Bilderkennung (Kugelerfassung, Farbzuoordnung) und reicht über die Stoßwahl und Wegplanung bis zur Kamera-Fahrzeug-Koordination.

Es ist z.B. denkbar, die gesamten Aufgaben der Parameterermittlung und Bewertung von Stößen sowie eine Auswertung von durchgeführten Stößen zu koppeln und mit Hilfe geeigneter Algorithmen (Kohonen-Karten, genetische Algorithmen) dem System selbst zu übertragen. Dadurch hätte der Billardroboter die Möglichkeit, seine Fertigkeiten mit fortlaufender Anzahl von Versuchen ("Spielpraxis") selbständig zu erwerben und zu verbessern.

Literaturverzeichnis

- [Bäss93] Bässmann, Henning u. Besslich, Philipp W.
Bildverarbeitung AdOculus
Springer Verlag, Berlin, 1993
- [Brai93] Braitenberg, Valentin
Vehikel: Experimente mit kybernetischen Wesen
Rowohlt Verlag, Reinbeck bei Hamburg 1993
- [Briggs90] Briggs, John und Peat, F. David
Die Entdeckung des Chaos
Carl Hanser Verlag, München, 1990
- [Brooks91] Brooks, Rodney A.
"New Approaches to Robotics"
in "Science", 13. September 1991
- [Conn96] Connectix Corporation, San Mateo, USA
ColorQuickCam für Windows Anwenderhandbuch, 1995/96
- [Drey85] Dreyfus, Hubert L.
Was Computer nicht können: die Grenzen künstlicher Intelligenz
Athenäum Verlag, Frankfurt/M., 1985
- [Ever95] Everett, H.R.
Sensors for Mobile Robots
AK Peters, 1995
- [Haber95] Haberächer, Peter
Praxis der digitalen Bildverarbeitung und Mustererkennung
Hanser Verlag, München/Wien, 1995
- [Jones96] Jones, Joseph L. u. Flynn, Anita M.
Mobile Roboter - Von der Idee zur Implementierung
Addison-Wesley (Deutschland), 1996
- [Jörg94] Jörg, Klaus-Werner
Echtzeitfähige Multisensorintegration für autonome mobile Roboter
BI-Wissenschafts-Verlag, Mannheim, 1994
- [Klette95] Klette, Reinhard u. Zamperoni, Piero
Handbuch der Operatoren für die Bildverarbeitung
Vieweg Verlag, Braunschweig/Wiesbaden, 1995
- [Levy93] Levy, Steven
KL-Künstliches Leben aus dem Computer
Droemer Knauer, München 1993

- [Maho91] Mahowald, Misha A. u. Mead, Carver
Die Silicium-Netzhaut
in "Spektrum der Wissenschaft", Heft 7/1991
- [Mess97] Messmer, Hans-Peter
PC-Hardwarebuch
Addison-Wesley Longman, Bonn, 1997
- [Miel85] Miel, Günter u. Fiebich, Rolf
Steuern und Regeln selbst erlebt
Urania-Verlag, Leipzig/Jena/Berlin, 1985
- [MiSo92] Microsoft Corporation
Video for Windows 1.1 Programmer's Guide
1992-93
- [Paus95] Paus, Hans J.
Physik in Experimenten und Beispielen
Hanser Verlag, München/Wien 1995
- [Pinz94] Pinz, Axel
Bildverstehen
Springer Verlag, Wien/New York, 1994
- [Prau96] Prautzsch, Friedrich
Schrittmotor-Antriebe: präzise positionieren in computergerechter Technik
Franzis Verlag, Feldkirchen, 1996
- [Rand97] Randow, Gero von
Roboter - Unsere nächsten Verwandten
Rowohlt Verlag, Reinbek bei Hamburg, 1997
- [Ritter91] Ritter, Helge / Schulten, Klaus / Martinez, Thomas
Neuronale Netze:
Eine Einführung in die Neuroinformatik selbstorganisierender Netzwerke
Addison-Wesley (Deutschland), 1991
- [Schiff94] Schiffer, Franz
Billard: Pool, Snooker, Karambol; Regeln, Technik, Geschichte
Hugendubel Verlag, München, 1994
- [Sedg92] Sedgewick, Robert
Algorithmen
Addison-Wesley (Deutschland), 1992
- [Thom93] SGS Thomson Microelectronics
Datenblatt Motortreiber-ICs L293B/E
1993

- [Thro94] Thro, Ellen
Künstliches Leben - Eine spielerische Entdeckungsreise:
Einführung in Theorie und Praxis einer neuen Wissenschaft
Addison-Wesley (Deutschland), 1994
- [Var90] Varela, Francisco J.
Kognitionswissenschaft, Kognitionstechnik: eine Skizze aktueller Perspektiven
Suhrkamp Verlag Frankfurt/M., 1990
- [Villa97] Villasenor, John u. Mangione-Smith, William H.
Wandlungsfähige Mikrochips
in "Spektrum der Wissenschaft", Heft 8/1997
- [Voss91] Voss, Klaus u. Süße, Herbert
Praktische Bildverarbeitung
Hanser Verlag München/Wien, 1991

Verzeichnis von Internet-Links

(Stand April 1998)

- [WWW1] www.cs.tu-bs.de/rob/projects/monamove/welcome.html
Projekt MONAMOVE an der TU Braunschweig
- [WWW2] www.connectix.com
Homepage der Firma Connectix
- [WWW3] www.interplay.com
Homepage der Firma Interplay,
Hersteller der Billard-Simulation "Virtual Pool"

Anhang Farbbilder

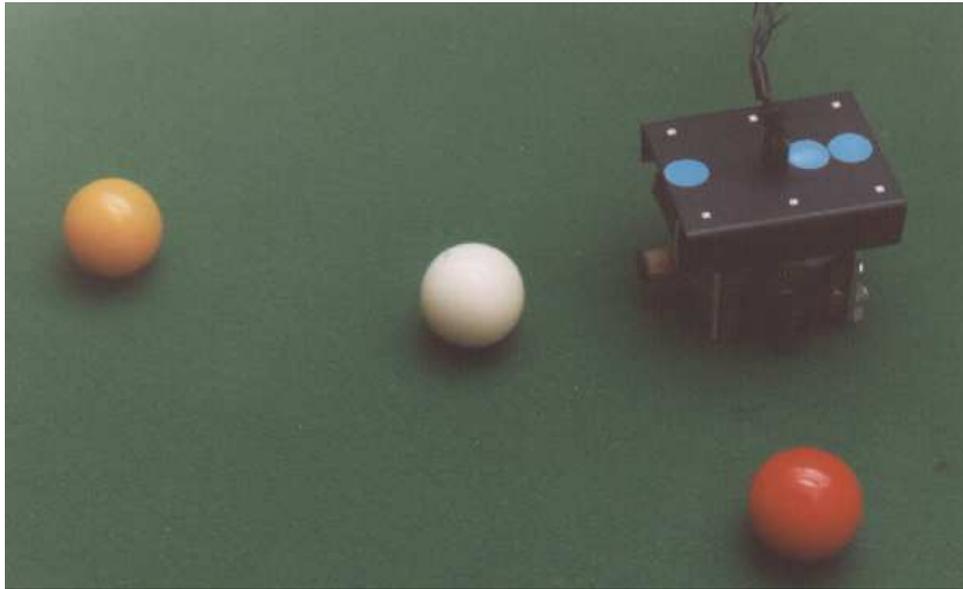


Abb. F 1.1 Der Billard-Roboter

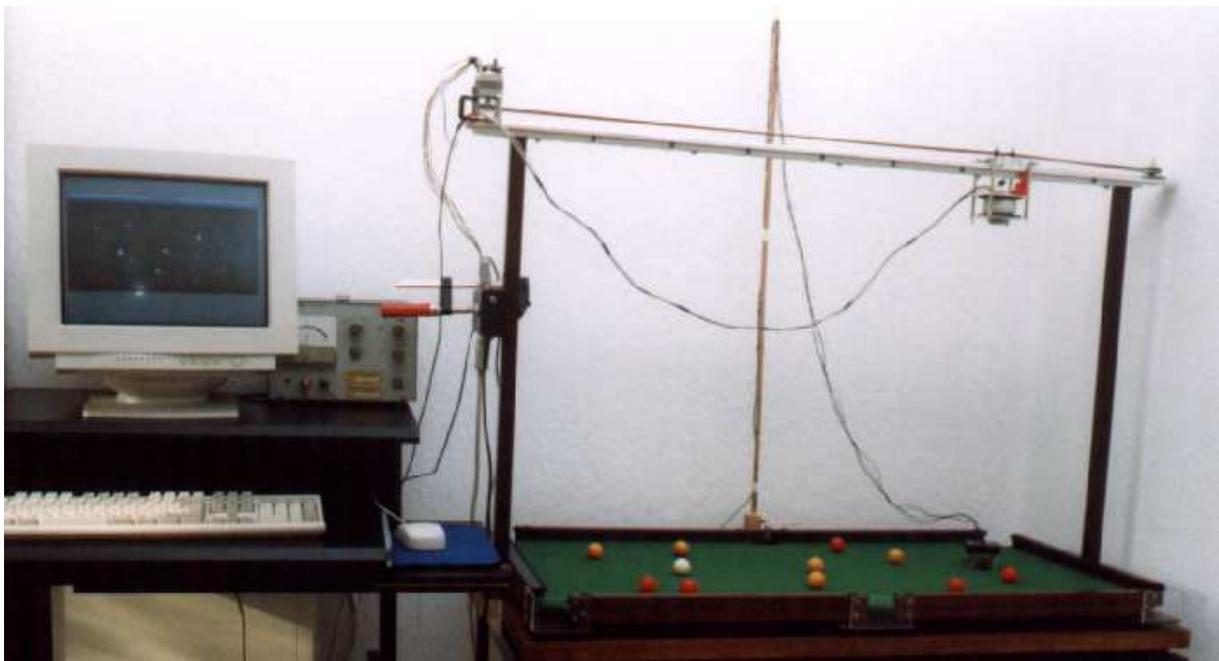


Abb. F 1.2 Gesamtansicht des Billardsystems

Zu sehen sind der Billardtisch mit Kugeln und Roboterfahrzeug, darüber montiert die Schiene, auf welcher sich der Kameraschlitten bewegt sowie der Steuercomputer und ein Netzteil. Die senkrechte Stange im Hintergrund dient der Befestigung und Führung der Steuerleitungen des Roboters. Das Foto wurde mit einem Weitwinkelobjektiv aufgenommen und wirkt deshalb perspektivisch leicht verzerrt.



Abb. F 2.3 "Christmas-Lights" durch geringe Umgebungshelligkeit

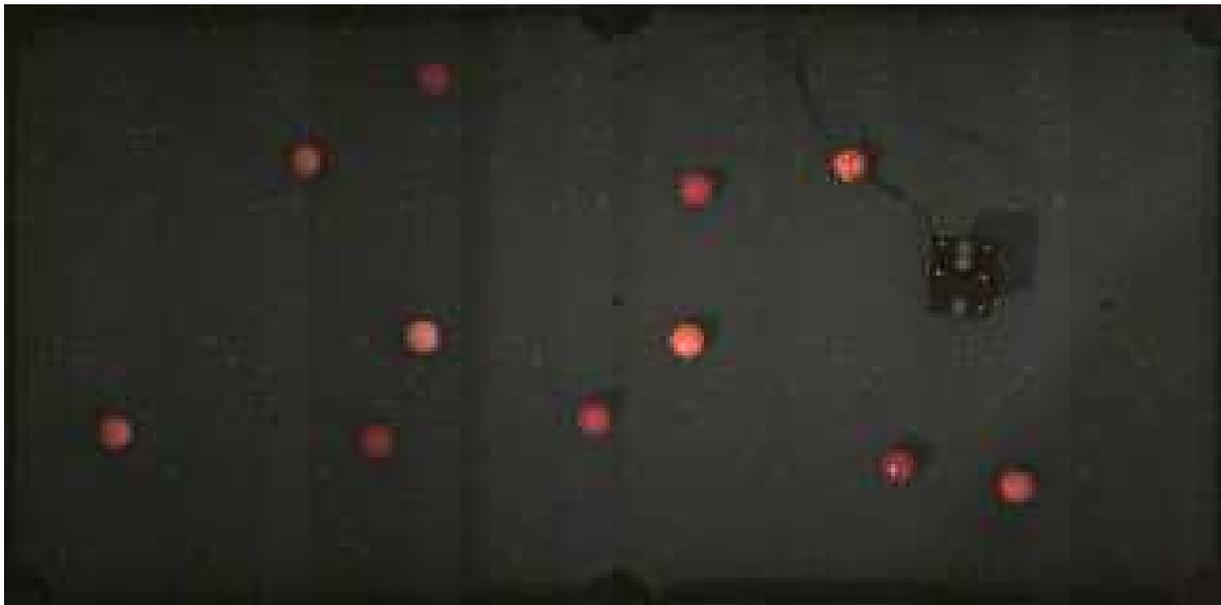


Abb. F 2.4 Beispiel eines Rohbildes der Kugel-Groberkennung

Die Umgebungsbedingungen (Grundhelligkeit, Schatten) wurden bewußt verschlechtert, um die Funktionsweise der Algorithmen zu verdeutlichen. Der großflächige Schatten in der linken Bildhälfte und der dreieckige in der rechten, unteren Ecke wurden "künstlich" erzeugt. Man beachte auch die teilweise Verdeckung der dem Roboterfahrzeug nächsten gelben Kugel durch dessen Steuerkabel.

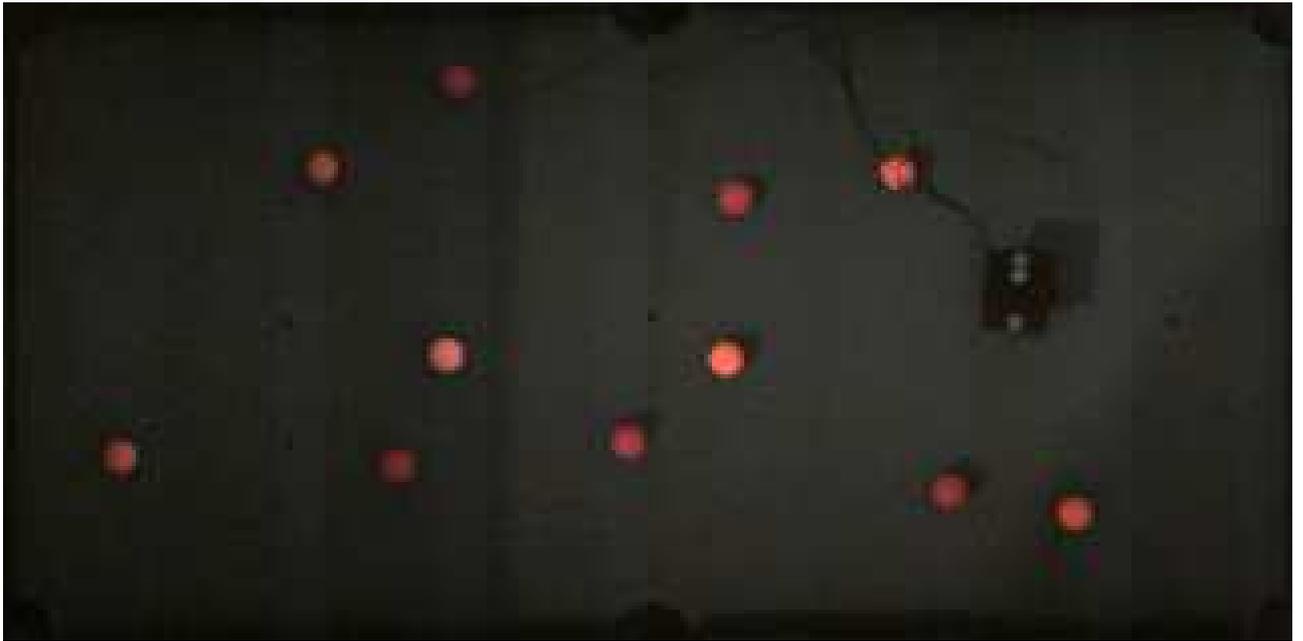


Abb. F 2.5 Ausgangsbild der Kugel-Groberkennung (nach "Christmas-Filter")

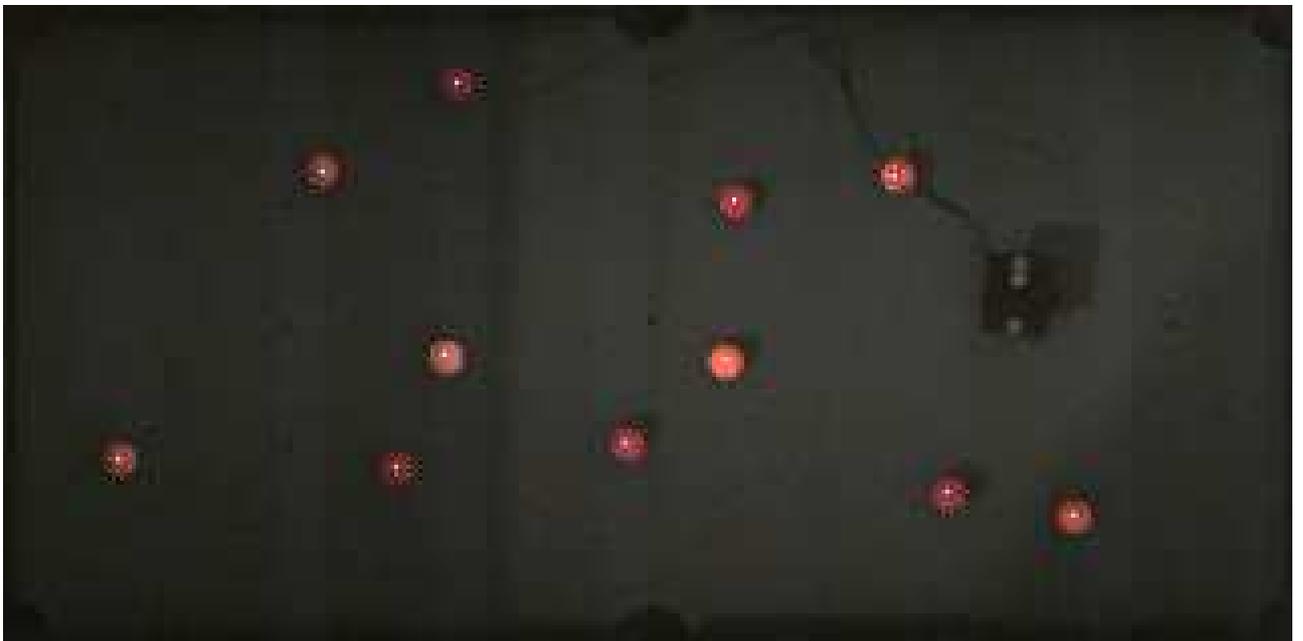


Abb. F 2.17 Ergebnis der Kugel-Groberkennung

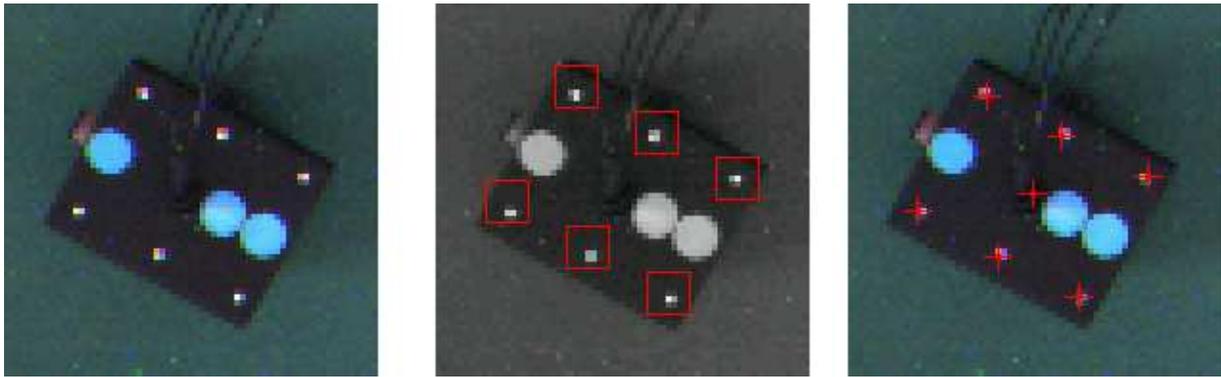


Abb. F 2. 23 Auffinden der Markierungen bei der exakteren Variante der Fahrzeugerkennung

Das linke Bild zeigt den aufgenommenen Bildausschnitt (65×65 Px aus Gesamtbild von 160×320 Px), dessen Position auf den Ergebnissen der Fahrzeug-Groberkennung beruht. In der Mitte ist das nach der Formel $(R+G+B)/3$ errechnete Grauwertbild zu sehen, in welchem innerhalb der hervorgehobenen Bereiche die Suche nach den sechs weißen Markierungen erfolgt. Im rechten Bild sind die als Markierungen identifizierte Pixel und der aus diesen errechnete Fahrzeug-Mittelpunkt eingezeichnet (Darstellung ohne perspektivische Korrektur).

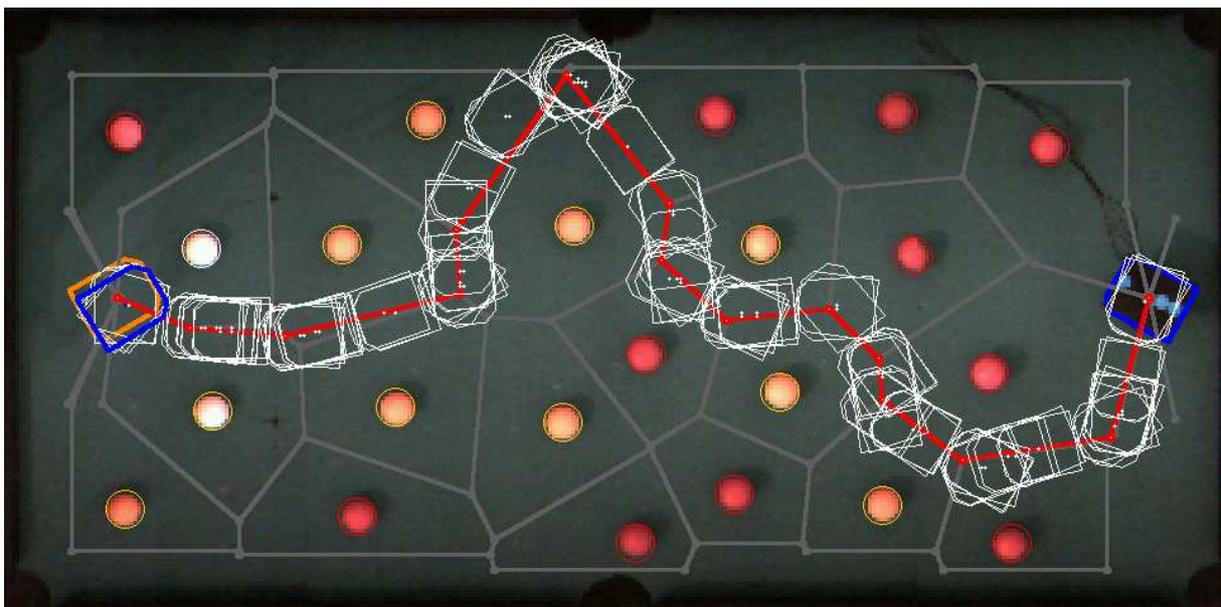


Abb. F 5.2 Beispiel einer vom Roboter absolvierten Fahrt

Die dargestellte Abbildung ist Teil eines Screenshots des Steuerprogrammes.

Zur Schaffung eines besonders anspruchsvollen Parcours wurden alle verfügbaren Kugeln auf dem Spielfeld platziert. Die Fahrt erfolgte vom Startpunkt auf der rechten Seite zum Zielpunkt links. Der ausgewählte Stoß sieht vor, die im Bild oberste gelbe Kugel in der mittleren Tasche der oberen Bande zu versenken. Die um die Kugeln gezeichneten Kreise symbolisieren die erkannte Position der Kugeln sowie die jeweils ermittelte Kugelfarbe. In diesem Ausnahmefall befinden sich zwei weiße Kugeln auf der Spielfläche, von denen die obere die Spielkugel ist. Auch die links oben befindliche rosa Kugel kommt im Normalfall nicht zum Einsatz.

Die berechnete Zielposition der Fahrt ist orange dargestellt, die tatsächlich erreichte blau. Die weiß dargestellten Fahrzeugumrisse markieren jeweils eine Stelle, an der ein Bild aufgenommen und eine neue Bewegung berechnet wurde. Besonders am Beispiel des obersten Punktes der Strecke ist gut zu erkennen, in welcher Weise Drehungen ausgeführt werden. Zu erkennen ist weiterhin, daß die Fahrt nur selten exakt der vorgegebenen Strecke folgt. Die Abweichungen sind aber, wie in den entsprechenden Abschnitten erläutert wird, in jedem Fall akzeptabel.

Der aus Sicht des Betrachters scheinbare Umweg in der Fahrtroute resultiert aus der Bewertungsfunktion der Wegplanung, welche den sichersten Weg ermittelt. Die grau dargestellten Linien veranschaulichen einen Zwischenschritt der Wegplanung. Erläuterungen zu diesen beiden Fakten finden sich im Kapitel "Wegplanung".

Das Absolvieren dieser Route nahm ca. 2min in Anspruch, wobei etwa die Hälfte der Zeit auf das Einlesen der Bilder von der Kamera in den Rechner entfiel. Die Kugel wurde im übrigen erfolgreich versenkt.



Abb. F 6.2 Detailansicht des Kameraschlittens

Zu erkennen sind der Seilzug, die als Räder verwendeten Kugellager, die Profilschiene und die Einfassung der Kamera. In den auf die obere Fixierungs-Platte aufgesetzten Teil ist der Reedkontakt integriert. Zwei der dazugehörigen Magnete befinden sich rechts bzw. links vom Schlitten an der Schiene. (Erläuterungen im entsprechenden Text)

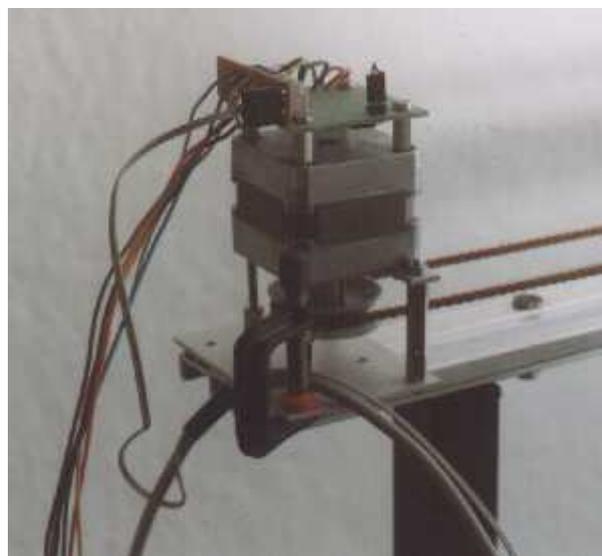


Abb. F 6.3 Detailansicht des Schrittmotors der Kameramechanik

Von der oben aufgesetzten Steuerelektronik führen Kabel zum PC-Interface. Das bogenförmig von einer kleinen Schraubzwinde gehaltene Kabel verbindet Kamera und Computer.

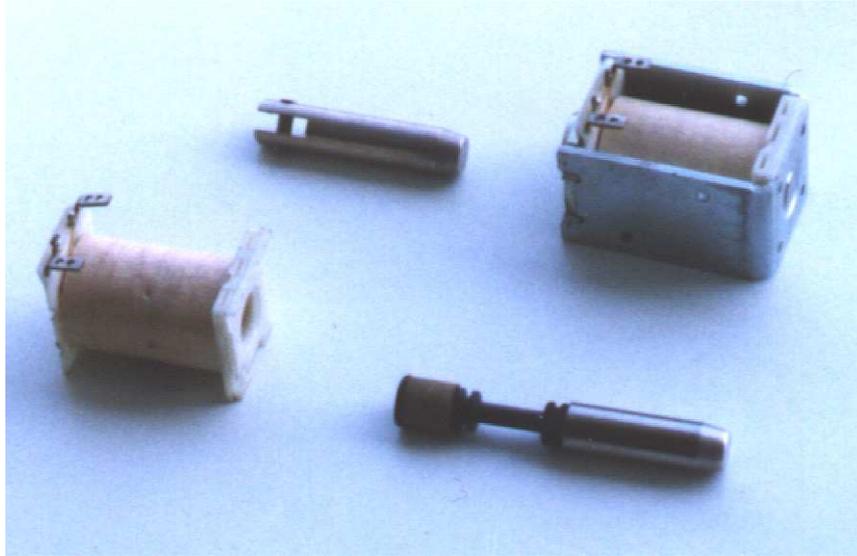


Abb. F 6.4 Stößelspule und -kern

Abgebildet sind ein im Handel erworbener Zugmagnet und der dazugehörige Kern in ihrer ursprünglichen Form (oben) sowie eine einzelne Spule und der im Stößel integrierte spezielle Kern (unten). Im Mittelabschnitt des Stößelkerns sind zwei Gummimuffen zu erkennen, welche die Auswirkungen der Stöße auf das Fahrzeug dämpfen.

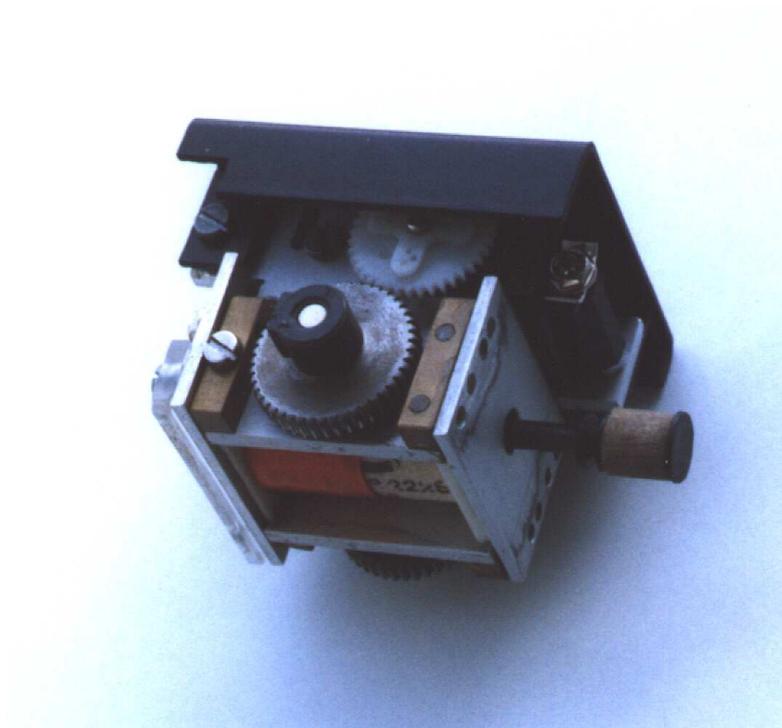


Abb. F 6.5 Nahaufnahme des Roboterfahrzeuges in Seitenlage

Zu erkennen sind (von oben nach unten): das „Dach“, das Zahnrad des rechten Motors (weiß), das rechte Antriebs(zahn)rad und die zwischen den tragenden Aluminiumplatten eingesetzte Spule. Rechts befindet sich der Stößel in seiner vordersten Position (wie während des Stoßes). Am Heck (links) befindet sich die Acrylglasfläche, auf welcher das Fahrzeug abgestützt wird. Das schwarze Bauteil, von dem ein Großteil des Rades verdeckt wird, ist ein Stelling, welcher das Rad auf der Achse fixiert.

Anhang Kostenübersicht

Diese Aufstellung gibt einen Überblick über die Kosten der Hauptkomponenten des Billardsystems.

Alle Kosten sind in DM angegeben und als gerundete Werte zu betrachten.

Billardtisch incl. Kugeln		170
Connectix ColorQuickCam		420
Schrittmotor incl. Elektronik		60
Servomotoren des Fahrzeuges	je	60
Mechanik des Fahrzeuges (Zahnräder, Chassis)		40
Bauteile der Interface-Elektronik (Motortreiber, elektron. Relais)		50

Dazu kommt eine Vielzahl von mechanischen Einzelteilen. Inclusive dieser und zuzüglich vieler Bauteile, welche nur während der Entwicklungsphase zum Einsatz kamen bzw. bei Experimenten beschädigt wurden, betragen die Gesamtkosten des Projektes schätzungsweise 1300 DM.

In die Rechnung nicht einbezogen sind der Steuer-PC und sämtliche Werkzeuge.

Selbständigkeitserklärung

Ich versichere hiermit, daß ich die vorliegende Arbeit selbständig und unter ausschließlicher Verwendung der angegebenen Hilfsmittel und Quellen erstellt habe.

Leipzig, 30. April 1998

Arnd Müller