[Lanser96] S. Lanser, C. Zierl: *MORAL: Ein System zur videobasierten Objekterkennung im Kontext autonomer mobiler Systeme*, in G. Schmidt, F. Freyberger (Hrsg.): Autonome Mobile Systeme 1996, Informatik aktuell, Springer, 1996

[Murray94] D. Murray, A. Basu: *Motion Tracking with an Active Camera*, IEEE Transactions on Pattern Recognition and Machine Intelligence, Bd. 16, Nr. 5, pp. 449-459, 1994

[Pomierski95] T. Pomierski, H. M. Groß: *Verfahren zur empfindungsgemäßen Farbumstimmung*, in G. Sagerer, S. Posch, F. Kummert (Hrsg.): Mustererkennung 1995, Informatik aktuell, Springer, 1995

[Schmidt94] M. Schmidt: *3D-Erkennung von Fahrzeugen in Echtzeit aus monokularen Bildfolgen*, Dissertation, Universität der Bundeswehr, München, 1994

[Schraft96] R. D. Schraft, H. Volz: *Serviceroboter*, Springer, 1996

[Sommerau95] M. Sommerau, G. Mamier, A, Zell, M. Vogt, P. Levi: *Fast Face Localization and Tracking with Model-Based Time Synchronization of a Head System*, in G. Sagerer, S. Posch, F. Kummert (Hrsg.): Mustererkennung 1995, Informatik aktuell, Springer, 1995

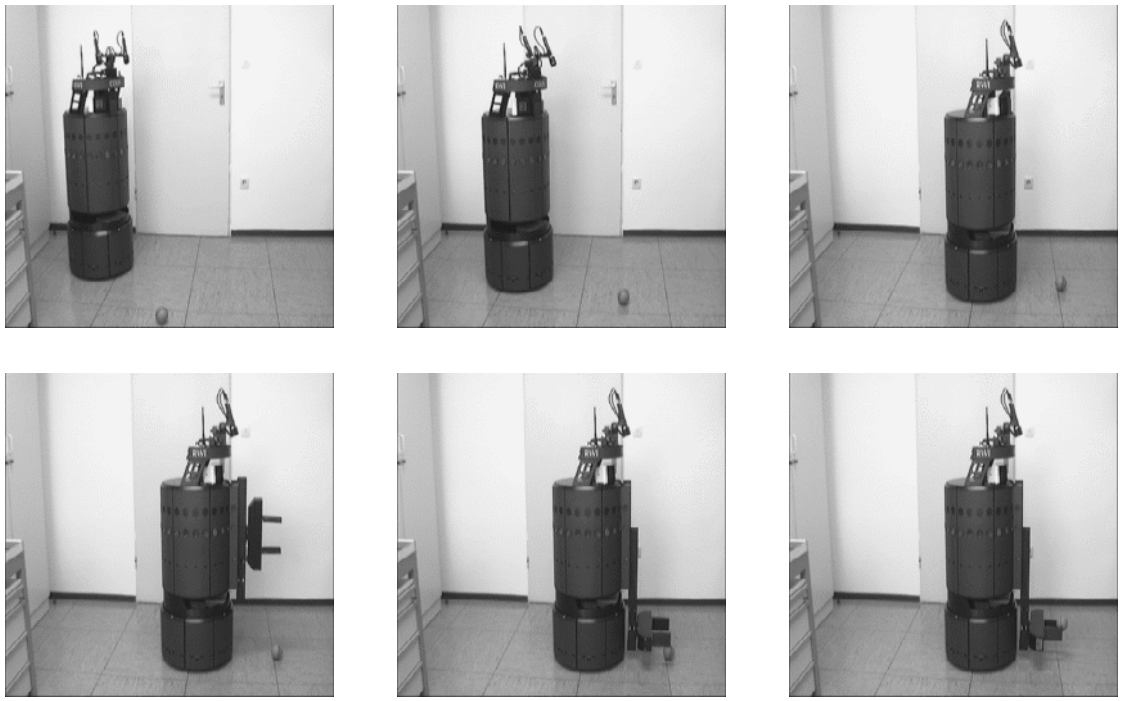

Abbildung 6: Verfolgung und Greifen eines Objekts

Die Einschränkung der bisherigen Implementierung der Fahrsteuerung auf Objekte, die sich auf dem Boden befinden, könnte durch eine Erhöhung des Modellwissens und eine Verfeinerung der Steueralgorithmen beseitigt werden.

Das System arbeitet mit einer Geschwindigkeit von 10 Bildern pro Sekunde und kann somit sich mäßig schnell bewegende Objekte sicher und schritthaltend verfolgen. Eine weitere Geschwindigkeitssteigerung wäre durch eine Parallelisierung der Extraktions- und Steuerungsalgorithmen möglich.

Literatur

[Bollmann96] M. Bollmann, T. Hempel, B. Mertsching: *Improved Edge Detection by the Evaluation of Color Contrast Information*, 2. Workshop Farbbildverarbeitung, Schriftenreihe des Zentrum für Bild- und Signalverarbeitung Ilmenau, Report 1/96, pp. 1-6, 1996

[Daniilidis96] K. Daniilidis, M. Hansen, G. Sommer: *Real Time Pursuit and Vergence Control with an Active Binocular Head*, in G. Schmidt, F. Freyberger (Hrsg.): Autonome Mobile Systeme 1996, Informatik aktuell, Springer, 1996

[Daxwanger96] W. Daxwanger, E. Ettelt, C. Fischer, F. Freyberger, U. Hanebeck, G. Schmidt: *ROMAN: Ein mobiler Serviceroboter als persönlicher Assistent in belebten Innenräumen*, in G. Schmidt, F. Freyberger (Hrsg.): Autonome Mobile Systeme 1996, Informatik aktuell, Springer, 1996

[Dose94] M. Dose, S. Fuhrmann, E. Schulze-Krüger, W. M. Theimer: *An Autonomous Mobile Robot: A Development Tool for Operation in a Natural Environment*, Internal Report IR-INI 94-02, Institut für Neuroinformatik, Ruhr-Universität Bochum, 1994

[Kass87] M. Kass, A. Witkin, D. Terzopoulos: *Snakes: Active contour models*, International Journal of Computer Vision, Bd. 1, Nr. 4, pp. 321-331, 1987

bestimmt. Bp_x und Bp_y bezeichnen dabei die Objektposition in Bildkoordinaten. Die beiden Werte d und φ bilden die Grundlage für die Regelung der Fahrgeschwindigkeit und -richtung. Die Regelung wird ähnlich zur Regelung der Blickrichtung der Kamera über einen Proportionalregler mit Differentialglied realisiert.

5 Einsatz auf dem mobilen Roboter ROBIN

Das beschriebene System wurde auf dem mobilen Roboter ROBIN vom Typ RWI B21 (Abbildung 5) implementiert. ROBIN besitzt zwei Doppelprozessor-Pentium-PCs, die unter Linux betrieben werden. Der rechte PC ist mit zwei Framegrabbern ausgestattet und führt vorwiegend die Bildverarbeitung durch, der linke ist für die Fahrsteuerung, Navigation und Planung zuständig. Die Basis selbst besitzt einen MC68332 Mikrocontroller für die Low-Level-Regelung der Fahrkommandos. Der Roboter ist in der Lage, sich aus dem Stand in jede Richtung zu bewegen, wobei der Synchronantrieb alle vier Räder gleichzeitig in die gleiche Richtung dreht. Synchron mit den Rädern wird auch das Oberteil gedreht, während die Basis immer die gleiche Orientierung behält. Der Roboter besitzt zwei Sony CCD-Farbkameras, die auf einer Schwenk-Neige-Einheit montiert sind. Der integrierte 4-DOF-Greifarm läßt sich bei Bedarf aktivieren und kann im normalen Fahrbetrieb vollständig eingefahren werden.

Die Verarbeitungsgeschwindigkeit des Roboters reicht aus, um den beschriebenen Detektionsalgorithmus mit einer Frequenz von 10 Hz ablaufen zu lassen. Dies genügt, um ohne Einsatz von Spezialhardware für die Bildverarbeitung eine schritthaltende Verfolgung von Objekten zu gewährleisten, wie sie in Indoor-Anwendungen vorkommen. Der Roboter kann sich dabei mit einer Geschwindigkeit von 0,3 m/s bewegen. Die maximale Geschwindigkeit der zu verfolgenden Objekte hängt von deren Abstand zum Roboter ab. Das System wurde im praktischen Einsatz auf der CeBIT ’97 demonstriert. Eine Beispielsequenz für das Verfolgen und Greifen eines Objekts ist in Abbildung 6 dargestellt.

Abbildung 5: Der mobile Roboter ROBIN

6 Diskussion und Ausblick

Es wurde ein System beschrieben, das eine farbbasierte Verfolgung von Objekten mittels Kamera- und Fahrsteuerung mit einem mobilen Roboter ermöglicht. Wesentlicher Bestandteil des Systems ist ein adaptives Verfahren, das neben einer Anpassung an globale Beleuchtungsveränderungen durch eine einfache Farbkorrektur zusätzlich auf lokale Luminanz- und Chrominanzschwankungen lernfähig reagieren kann. Das Verfahren lehnt sich dabei insofern an das menschliche Sehsystem an, als es sich nicht auf die Filterung bestimmter vorgegebener Farbwerte beschränkt, sondern unter Zuhilfenahme von Farbkontrasten eine ständige Aktualisierung der Detektionsintervalle vornimmt.

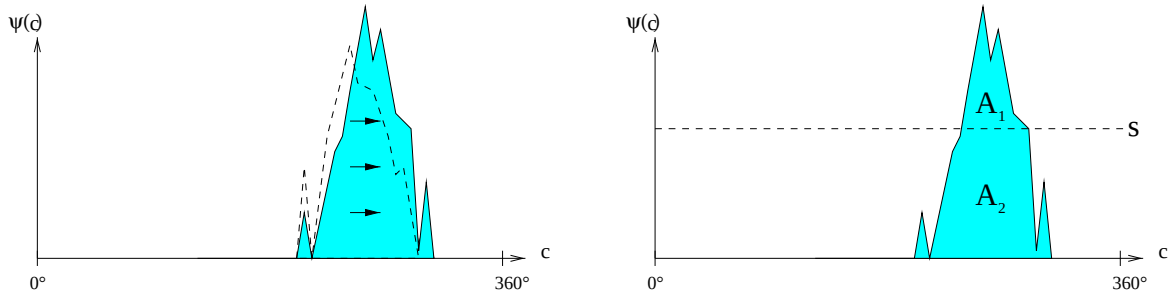

Abbildung 4: links: Anpassung der gelernten Farbwerte, rechts: Schwellwertdefinition anhand des Flächenverhältnisses

4 Kamera- und Fahrsteuerung

Das Ziel der Kamerasteuerung ist, mit Hilfe von Schwenk- und Neigebewegungen dem erkannten Objekt so nachzufahren, daß dieses nicht aus dem Sichtbereich wandert (Blickstabilisierung). Dabei soll eine möglichst weiche Kameraführung erreicht werden. In der vorliegenden Arbeit wird dazu ein erweiterter PD-Regler der folgenden Form verwendet:

$$y_t = K_P x_{w,t} + K_D \frac{x_{w,t} - x_{w,t-1}}{T} \tag{16}$$

$x_{w,t}$ bezeichnet die Regelabweichung zwischen dem Istwert x und dem Sollwert W zum Zeitpunkt t, K_P und K_D sind konstante Regelfaktoren, die das Ausmaß der Reaktion bestimmen. T ist das Zeitintervall zwischen zwei Messungen. Die Einführung des Differentialglieds, in das Informationen über die momentane Geschwindigkeit und Beschleunigung des zu verfolgenden Objekts einfließen, bewirkt ein schnelleres Reagieren auf plötzliche Änderungen. Um ein Schwingen der Kamerabewegung zu unterdrücken, wurde zusätzlich ein ε-Bereich festgelegt, in dem keine Kamerakorrektur vorgenommen wird.

Die Fahrsteuerung des Roboters hat die Aufgabe, einem erkannten Objekt nachzufahren und bei genügend kleiner Distanz das Objekt mit dem Greifarm aufzunehmen. In der hier vorgestellten einfachen Version wird von der Annahme ausgegangen, daß sich das Objekt auf dem Boden befindet. Für die Berechnung der Robotergeschwindigkeit und -richtung ist es wichtig, die Entfernung des erfaßten Objekts möglichst genau zu ermitteln. Um schließlich das Objekt zu greifen, ist es sogar notwendig, die Position auf circa einen Zentimeter genau zu kennen. Die verwendete Formel zur Abstandsberechnung liefert im Nahbereich Ergebnisse mit hoher Genauigkeit, in zunehmender Entfernung mit abnehmender Genauigkeit:

$$d = \frac{h_c(z_0 + Bp_y \cdot \tan\theta)}{z_0 \cdot \tan\theta - Bp_y} + c \tag{17}$$

h_c bezeichnet die Höhe und θ den Neigewinkel der Kamera, z_0 ist die Brennweite und c der horizontale Abstand zwischen Kamera und Robotermittelachse. Die Richtung des erfaßten Objekts in Roboterkoordinaten wird durch

$$\varphi = \operatorname{atan}\left(\frac{Bp_x}{Bp_y}\right) \tag{18}$$

H und *B* bezeichnen dabei die Höhe bzw. die Breite des Objekts. Die Größenangabe in der ersten Bedingung hängt von der verwendeten Auflösung ab und muß bei anderen Auflösungsstufen entsprechend angepaßt werden.

Das Lernen neuer Farben ist der letzte und wichtigste Schritt des Verfahrens. Dazu ist es notwendig, neue Farbwerte, die das aktuell detektierte Objekt umfaßt, aus dem Bild zu ermitteln. Allerdings darf dabei die zuvor gelernte Information nicht verlorengehen, sondern muß allmählich an die neuen Licht- und Farbverhältnisse angepaßt werden. Aus diesem neu angepaßten Wissen wird schließlich ein neues intervall *I* ermittelt, das in der folgenden Iteration zur Erkennung des Objekts herangezogen wird.

Die oben beschriebene Gradientenbildung trägt dazu bei, auch neue Objektfarben des Ähnlichkeitsbereichs (Intervall I^{ε}) im Bild mit einzuschließen. Es wird somit möglich, bisher untrainierte Farben zu erhalten, die den erlernten Farbbereich erweitern bzw. verschieben. Das Lernen wird dadurch realisiert, daß zu jedem Farbwert *c* ein aktueller Lernwert $\psi(c)$ mitgeführt wird, der angibt, wie häufig der entsprechende Farbwert in der Vergangenheit dem Objekt zugeordnet werden konnte. Da der momentane Lernprozeß stärker ins Gewicht fallen soll als ältere Lernvorgänge, wird folgende Lernfunktion verwendet:

$$\psi_{n+1}(c) = \psi_n(c) \cdot (1 - \alpha) + \zeta(c) \cdot k \cdot \alpha \tag{11}$$

$\zeta(c)$ ist die Fläche (Anzahl der Pixel) mit Farbwert *c* des neu erkannten Objekts, α repräsentiert den Lernfaktor. *k* ist eine feste Konstante, die unter Verwendung von diskreten Werten $\psi(c)$ als Skalierungsfaktor dient. Dabei stellt sich für den diskreten Fall eine Auflösungsgenauigkeit von $\tau = 1/k$ ein. Die Lernfunktion fungiert gleichzeitig als Tiefpaß-Filter, d.h. Ausreißer fallen nicht allzu stark ins Gewicht und werden ausgefiltert.

Um aus der Lernfunktion das neue Intervall $I = [l_{min}, l_{max}]$ zu bestimmen, wird eine Schwelle *s* eingeführt:

$$l_{min} = \{c \mid (\psi(c) > s) \wedge \forall c' < c : \psi(c') \leq s\} \tag{12}$$

$$l_{max} = \{c \mid (\psi(c) > s) \wedge \forall c' > c : \psi(c') \leq s\} \tag{13}$$

Die Schwelle *s* muß so gewählt werden, daß das Intervall *I* eindeutig bestimmt ist. Es muß außerdem die Bedingung

$$\exists c : \psi(c) > s \tag{14}$$

erfüllt sein, so daß sich mindestens ein Intervall *I* ergibt. Die Schwelle darf aber auch nicht zu klein gewählt werden, damit einzelne Peaks, die zum Beispiel von fehlerhaft ermittelten und gelernten Farbwerten herrühren, nicht ein eigenes zusätzliches Intervall bilden. Wählt man *s* dynamisch, so daß gilt

$$A_1 = \kappa \cdot A_2 , \tag{15}$$

wobei A_1 die Fläche überhalb und A_2 die Fläche unterhalb der Schwellwertlinie bezeichnet, so ist sichergestellt, daß man mindestens ein Intervall erhält. κ ist dabei ein fester Faktor, der im Beispiel in Abbildung 4 rechts auf $\kappa = 2/3$ gesetzt ist.

Sobald das neue Intervall *I* bestimmt ist, kann mit dem nächsten Iterationsschritt begonnen werden. In Abbildung 4 links wird die dynamische Veränderung der Lernwerte während eines Iterationsschritts graphisch dargestellt.

oder nicht. Beispielsweise ergibt sich für die Farbwerte H_1 und H_2 aus Abbildung 3a keine eindeutige Entscheidung:

$$d_1^{bias'} = d_2^{bias'} = 0\text{, aber } H_1 \notin I^{\varepsilon} \text{ und } H_2 \in I^{\varepsilon} \tag{9}$$

Aus diesem Grund wird ein modifizierter Schwellwertoperator eingesetzt, der eine Inversion der ursprünglichen Abstandfunktion vornimmt (Abbildung 3b) und anschließend die Funktionswerte um $\max(H) - \varepsilon$ absenkt (Abbildung 3c):

$$d_{xy}^{bias_1} = \begin{cases} -d_{xy} + \varepsilon & d_{xy} < \varepsilon \\ 0 & \text{sonst} \end{cases} \tag{10}$$

Als Ergebnis erhält man ein Bild im Hue-Differenz-Raum, in dem nur noch Punkte gesetzt sind, deren ursprüngliche Differenzwerte innerhalb von I^{ε} liegen.

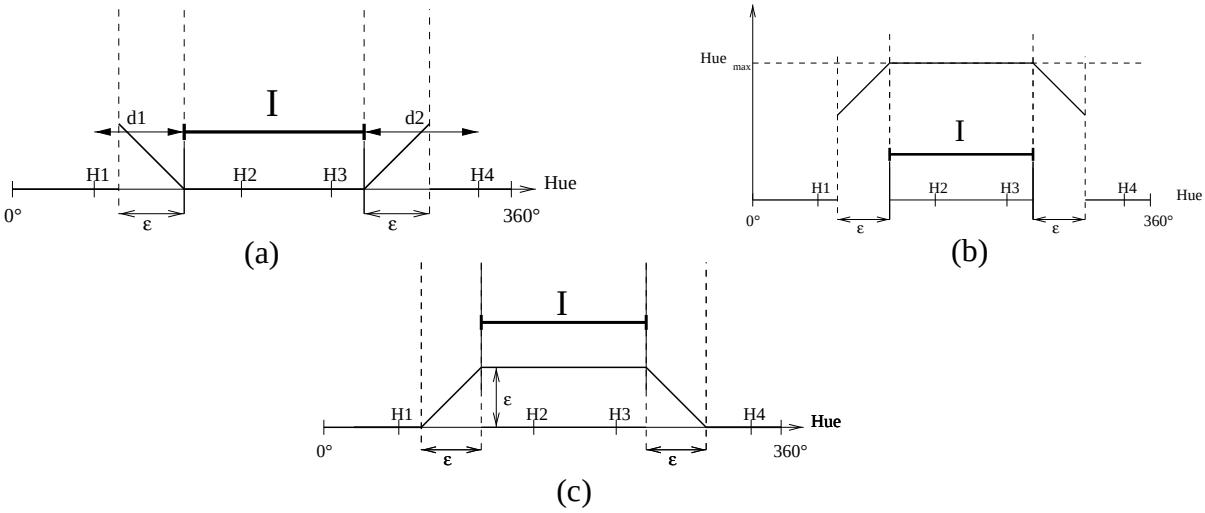

Abbildung 3: (a) einfache Schwellwertoperation, (b) Schwellwertoperation mit Invertierung, (c) modifizierter Schwellwertoperator

Der nächste Schritt des Verfahrens besteht darin, Kanten im Hue-Differenz-Raum zu bestimmen. Dazu wird eine Gradientenbildung mit anschließendem Schwellwertoperator verwendet. Der Gradient kann nur dann größer als Null sein, wenn mindestens eine beteiligte Kantenfarbe innerhalb des ε-Bereichs liegt. Somit ist gewährleistet, daß nur Kanten detektiert werden, die im oder nahe am Objektfarbenbereich liegen.

Um die im vorhergehenden Schritt erhaltenen Kanten zu schließen, wird ein Closing-Operator bestehend aus einer Dilatation und einer Erosion angewendet. Anschließend kommen ein Füll- und ein Segmentierungsalgorithmus zum Einsatz, die als Ergebnis eine Liste von detektierten Objekten liefern. Dabei stehen zu jedem Element dieser Liste eine Reihe von Informationen wie Schwerpunkt, horizontale und vertikale Ausdehnung, Flächengröße usw. zur Verfügung.

Zur Auswahl eines Objekts aus der erstellten Objektliste fließen an dieser Stelle Informationen über die Form der zu detektierenden Objekte ein. Für die Erkennung von farbigen Squiggle-Bällen wurden beispielsweise folgende Bedingungen verwendet:

- 60 Pixel > H > 5 Pixel und 60 Pixel > B > 5 Pixel
- Fläche > $H \cdot B \cdot 3/4$
- $0{,}9 < H/B < 1{,}1$

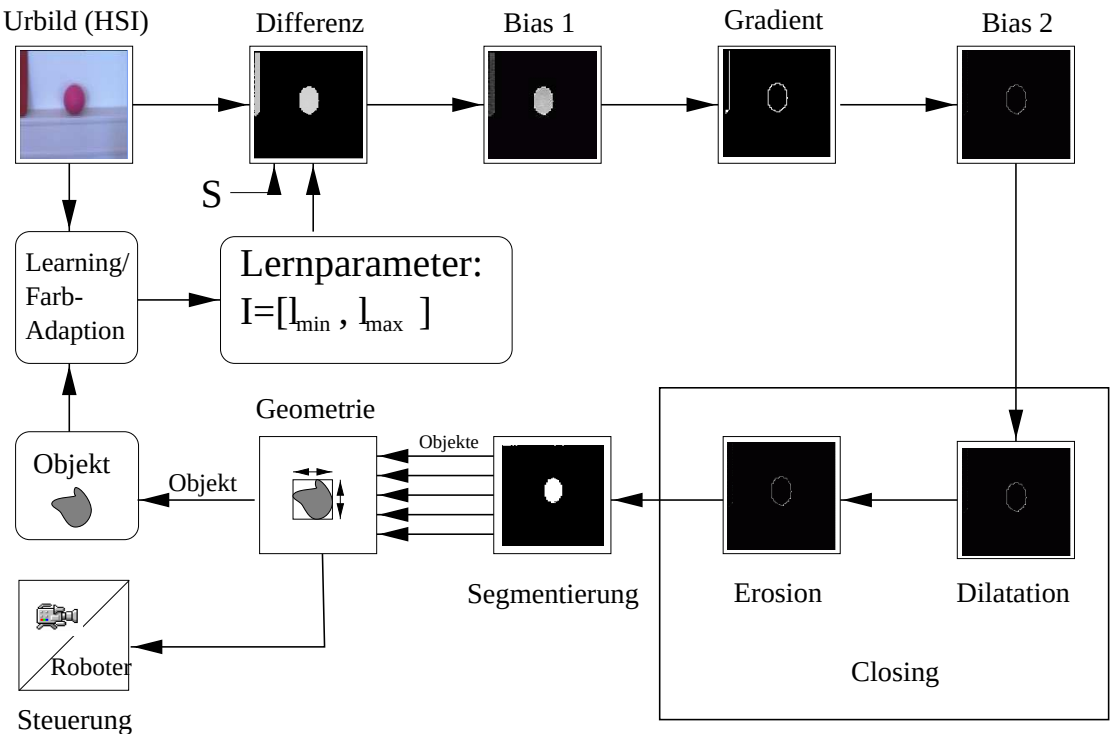

Abbildung 2: Ablaufschema des adaptiven Verfahrens

$$d_{xy} = \begin{cases} 0 & h_{xy} \in I \\ \min(d_h(h_{xy}, l_{min}), d_h(h_{xy}, l_{max})) & \text{sonst} \end{cases} \tag{5}$$

Hierbei bezeichnet h_{xy} den Hue-Wert des Bildpunkts an der Position (x, y); die Abstandsfunktion d_h ist defniniert durch

$$d_h(a, b) = \min((a-b)\bmod(\max(H)), (b-a)\bmod(\max(H))), \tag{6}$$

wobei max(*H*) den maximal möglichen Hue-Wert bezeichnet.

Durch Einführung mehrerer Intervalle I_n, $0 \le n \le L$, kann der Algorithmus leicht für die gleichzeitige Detektion von mehreren unterschiedlichen Farben erweitert werden, indem das Minimum über die Abstände zu allen Intervallen berechnet wird:

$$d_{xy} = min\{d_{xy,n} \mid 0 \le n \le L\} \tag{7}$$

Es ist sinnvoll, alle Farben, die vom Intervall *I* mehr als um einen bestimmten Betrag ε abweichen, zu unterdrücken, da von diesen Farbwerten angenommen werden kann, daß es sich auch bei veränderten Lichtverhältnissen und einer leichten Farbverschiebung nicht um die gesuchten Objektfarben handelt. Wendet man auf das in Schritt 1 erhaltene Ergebnis einen Schwellwertoperator

$$d_{xy}^{bias'_1} = \begin{cases} d_{xy} & d_{xy} < \varepsilon \\ 0 & \text{sonst} \end{cases} \tag{8}$$

an, so erhält man eine Abstandsfunktion, wie sie in Abbildung 3 links dargestellt ist. Das sich aus der Erweiterung von Intervall *I* um ε ergebende Intervall sei $I^{\varepsilon} = [I_{min} - \varepsilon, I_{max} + \varepsilon]$. Nachteil dieser neuen Abstandsfunktion ist, daß sie an den Grenzen von I^{ε} unstetig ist und sich die Funktionswerte nicht dazu verwenden lassen, eine Aussage darüber zu machen, ob ein bestimmter Farbwert in I^{ε} liegt

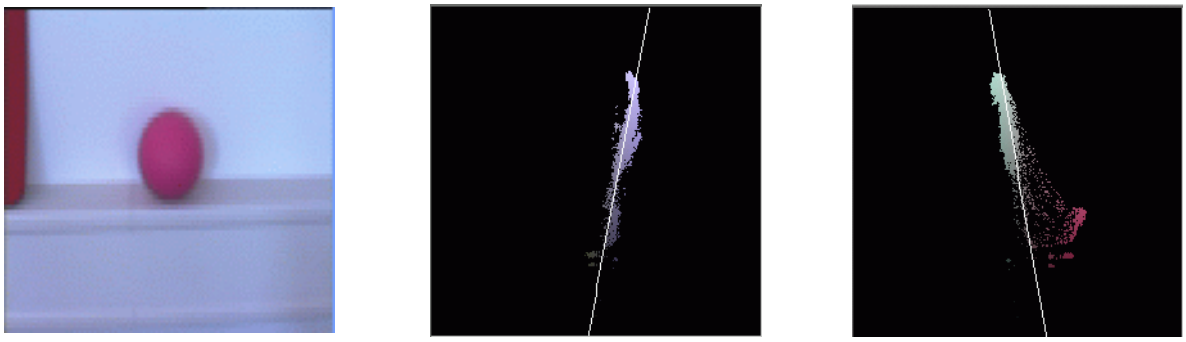

Abbildung 1: Beispielbild (links) mit dazugehöriger YU- (Mitte) und YV-Farbverteilung (rechts)

Einfache Ansätze, wie sie z.B. in [Sommerau95] verwendet werden, arbeiten mit einer statischen Unterteilung des Farbraums, d.h. es wird in einem Trainingsschritt vor der eigentlichen Verarbeitung der Bilddaten festgelegt, welche Farben zu *K*' und welche zu *K*' gehören sollen. Diese Unterteilung wird während der Verarbeitung nicht mehr verändert. Die Partitionierungen können mittels Intervalldefinitionen, Lookup-Matrizen oder durch ein trainiertes neuronales Netz realisiert werden. Vorteil dieser Verfahren ist ihre Einfachheit und die hohe Geschwindigkeit bei der Verarbeitung. Sie haben jedoch die nachteilige Eigenschaft, daß sie sich nicht an sich ändernde Lichtverhältnisse anpassen können. Wechseln in einer Szene die Beleuchtungseigenschaften bezüglich Intensität oder gar bezüglich der Farbe, so kann es sein, daß Farben von gelernten Objekten sich in Luminanz und Chrominanz so stark verändern, daß sie nicht mehr dem festgelegten Farbraum *K*' angehören. Dies gilt umso mehr, je genauer die Farbunterscheidung durch möglichst präzise und eng eingegrenzte Farbräume gewünscht wird. Solche Lichtänderungen treten nicht nur bei einer Änderung der Szene auf (zum Beispiel zwei Räume mit unterschiedlichen Lampen), sondern werden auch schon bei Veränderungen der Kamerasicht bzw. -position hervorgerufen. Die Folge ist, daß die zu detektierenden Objekte nicht mehr erkannt werden.

Der im folgenden vorgestellte adaptive Ansatz orientiert sich am menschlichen Sehsystem, das Farben nicht alleine an der spektralen Zusammensetzung, sondern auch anhand von Kontrasten zu anderen Farben identifiziert [Bollmann96]. So erscheint dem Gehirn ein bestimmtes Rot auch dann als dieses Rot, wenn es von einem Schatten überdeckt oder von einem andersfarbigen Licht bestrahlt wird. Das adaptive Verfahren wertet neben der reinen Farbinformation auch Farbkonstraste und Farbkanten mit aus und verwendet diese zusätzlichen Informationen dazu, die Farbraumpartitionierung dynamisch anzupassen. Das Ergebnis ist ein stabiler und lernfähiger Algorithmus, der Änderungen der Objektfarben im zeitlichen Verlauf erkennen kann und diese entsprechend hinzulernt. Ferner ist das Verfahren imstande, unvollständig gelernte Farbinformationen zu komplettieren.

Der prinzipielle Ablauf des Verfahrens ist in Abbildung 2 dargestellt. Ausgangspunkt ist ein Bild im helligkeitsinvarianten Farbraum HSI. Dieses wird aus dem mittels des oben beschriebenen einfachen Farbanpassungsverfahrens korrigierten YUV-Bild berechnet. Alle weiteren Betrachtungen basieren lediglich auf dem Hue-Anteil des HSI-Bildes. Des weiteren ist zu Anfang ein Intervall $I = [l_{min}, l_{max}]$ vorgegeben, wobei l_{min} den kleinsten und l_{max} den größten Hue-Wert der zu detektierenden Farbe bezeichnet.

Im ersten Schritt wird für alle im Bild befindlichen Pixel ein Ähnlichkeitsmaß zu den Farbwerten des Intervalls *I* bestimmt:

2 Farbkorrektur

Ein Problem farbbasierter Verfahren, das insbesonere bei mobilen Fahrzeugen auftritt, ist die Veränderung der Farbinformation durch sich verändernde Beleuchtungseinflüsse. Allein die Transformation der Bilddaten in einen helligkeitsinvarianten Farbraum (z.B. HSV, HSI oder YUV) und die Vernachlässigung des Helligkeitsanteils kann dieses Problem nicht lösen. Dazu müßte reines Weißlicht die Objekte der beobachteten Szene beleuchten, was in der Realität so gut wie nie der Fall ist. Das Tageslicht besitzt ein deutlich gelblich gefärbtes Spektrum, das sich zudem je nach Tageszeit stark ändert. In Innenräumen kommen dazu noch zusätzlich Veränderungen des einfallenden Lichts durch Fensterscheiben, die eventuell getönt, verspiegelt oder absorbierend sind. Vor allem aber strahlen Glühbirnen und Leuchtstoffröhren im Regelfall kein reines weißes Licht ab, sondern haben einen roten oder bläulichen Farbcharakter.

Selbst unter idealen Bedingungen, d.h. wenn ausschließlich weißes Licht einen Raum ausstrahlen würde, würde sich auch hier im Regelfall kein idealer Zustand einstellen. Im Raum existierende Gegenstände reflektieren Licht und erzeugen eine diffuse Lichtabstrahlung in ihrer Eigenfarbe.

Als erster Vorverarbeitungsschritt kommt daher in dem hier vorgestellten System eine einfache Farbkorrektur zum Einsatz, die direkt auf das vom Vision-System gelieferte YUV-Bild angewendet wird. Sie stellt im wesentlichen eine vereinfachte Form der in [Pomierski95] vorgeschlagenen Farbumstimmung dar. Für jeden Bildpunkt $P(x, y) = f(x, y, (Y, U, V))$ werden zunächst die Quotienten U/Y und V/Y ermittelt. Über alle Quotienten der Pixel werden die beiden Mittelwerte m_u und m_v bestimmt, die die Steigung der in Abbildung 1 eingezeichneten Geraden g_u und g_v mit

$$g_u(Y) = Y \cdot m_u + c_u \quad \text{und} \quad g_v(Y) = Y \cdot m_v + c_v \tag{1}$$

darstellen. Geht man von der Annahme aus, daß alle Originalfarben der betrachteten Szene stochastisch normalverteilt sind, so bedeutet dies, daß zur Farbkorrektur die Geraden g_u und g_v auf die Geraden $g'_u \equiv 0$ und $g'_v \equiv 0$ abgebildet werden müssen. Es ergibt sich somit für die korrigierten *U*- und *V*-Werte

$$U'_{xy} = U_{xy} - m_u \cdot Y_{xy} - c_u \quad \text{und} \quad V'_{xy} = V_{xy} - m_v \cdot Y_{xy} - c_v \tag{3}$$

mit

$$m_u = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \frac{U_{xy}}{Y_{xy}} \quad \text{und} \quad m_v = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \frac{V_{xy}}{Y_{xy}} \,. \tag{4}$$

Eine Farbkorrektur dieser Art kann zwar globale Beleuchtungsveränderungen ausgleichen, vermag jedoch nicht auf lokale Farb- und Helligkeitsschwankungen adäquat zu reagieren. Aus diesem Grund wird das im folgenden beschriebene Verfahren zur adaptiven Farbdetektion eingesetzt.

3 Ein adaptives Verfahren zur farbbasierten Objekterkennung

Um farbbasiert Objekte detektieren zu können, muß der verwendete Farbraum *K* in zwei Komplementärräume K' und K' partitioniert werden. Farbwerte, die als Objektfarbe identifiziert werden sollen, sind in K' enthalten, alle anderen in K'. Die Schwierigkeit besteht darin, eine geeignete Wahl und Darstellung dieser Unterräume zu finden, bei der folgende Eigenschaften erfüllt sind:

Die Codierung und Berechnung sollte einfach und effizient sein, allgemein genug, um Veränderungen der Luminanz zu tolerieren und speziell genug, um ähnliche, aber unterschiedliche Farben bis zu einem gewissen Grad noch trennen zu können.

Adaptive farbbasierte Objektverfolgung mit dem mobilen Roboter ROBIN

Stefan Feyrer, Oliver Schimmel, Andreas Zell
Wilhelm-Schickard-Institut für Informatik, Abt. Rechnerarchitektur
Universität Tübingen
Köstlinstraße 6, D-72074 Tübingen
{feyrer, schimmel, zell}@informatik.uni-tuebingen.de

Zusammenfassung

Diese Arbeit beschreibt ein System, mit Hilfe dessen ein mobiler Roboter in die Lage versetzt wird, farbige Objekte in einer dynamischen Umgebung visuell zu detektieren, mittels Blick- und Fahrsteuerung zu verfolgen und mit Hilfe eines Greifarms vom Boden aufzunehmen. Die Schwierigkeit bei der Extraktion farbiger Objekte aus Bildfolgen ist, einerseits den Einfluß von sich verändernden Lichtverhältnissen zu minimieren und andererseits gleichzeitig die Spezifität der zu detektierenden Farbbereiche zu maximieren. Dieses Problem wird in der vorliegenden Arbeit durch den Einsatz eines adaptiven Verfahrens gelöst, das basierend auf der Berechnung des Farbgradienten im HSI-Raum Farbkanten zur Objektsegmentierung heranzieht. Zusammen mit geometrischen Informationen über das Objekt bildet dies die Grundlage für ein Nachlernen von Farben und damit für ein adaptives Verhalten des Algorithmus. Der Einsatz des Systems auf dem mobilen Roboter ROBIN wird beschrieben und diskutiert.

1 Einleitung

Roboter sind heutzutage in vielen Gebieten im Einsatz. Während die meisten Roboter mit einer speziell an ihr Aufgabengebiet angepaßten Ausstattung ausgerüstet sind, wird immer häufiger versucht, mobile Roboter zu entwickeln, die sich lernfähig und intelligent in ihrer Umgebung zurechtfinden und somit für den Einsatz ein wesentlich höheres Maß an Flexibilität bieten [Schraft96, Daxwanger96, Dose94]. Basale Fähigkeiten solcher Systeme sind eine sichere Detektion und eine zuverlässige Manipulation von Objekten in einer dynamischen Umgebung. Heutige mobile Robotersysteme sind in ihrer Rechenkapazität so leistungsfähig, daß die Detektion von Objekten meist visuell vorgenommen wird.

Es gibt im wesentlichen drei Verfahrensklassen zur visuellen Objektdetektion. Bewegungsorientierte Verfahren, wie sie z.B. in [Daniilidis96], [Murray94] und [Kass87] verwendet werden, haben den Nachteil, daß beim Einsatz auf mobilen Systemen eine aufwendige Kompensation der Eigenbewegung vorgenommen werden muß. Formbasierte Verfahren ([Lanser96, Schmidt94]) benötigen Modellinformation über die zu erkennenden Objekte. Rein farbbasierte Verfahren sind zwar weniger rechenzeitaufwendig, erreichen aber keinen hohen Grad an Robustheit bezüglich der Diskriminierung von Objekten. In konkreten Anwendungen kommt deshalb meist eine Kombination verschiedener Verfahren zum Einsatz. In dieser Arbeit wurde der Schwerpunkt auf eine farbbasierte Detektion gelegt, unterstützt durch einen formbasierten Ansatz.