

# Integration von Kontextwissen zur videobasierten Szenenanalyse

Jörg Illmann, Christian Köhler  
Forschungsinstitut für anwendungsorientierte Wissensverarbeitung  
FAW Ulm, Helmholtzstr. 16, D-89081 Ulm  
{illmann, koehler}@faw.uni-ulm.de

## Abstract

Heutzutage stehen Entwicklern umfangreiche Bildverarbeitungssysteme mit einer Vielzahl leistungsfähiger Verfahren als Basis neuer Entwicklungen für Navigations- und Erkennungsaufgaben zur Verfügung. Dennoch ist die Integration dieser Verfahren in eine komplexe Steuerungsarchitektur mobiler Robotersysteme immer noch eine offene Frage. Anwendungen im Bereich Mensch-Roboter-Interaktion und autonomer Robotersteuerung erfordern einen neuen Grad von Laufzeitflexibilität: dynamische Konfiguration visueller Methoden zur Berücksichtigung des aktuellen Kontext von Aufgabe und Umgebung und Vorabwissen. In diesem Artikel beschreiben wir die Integration videobasierter Verhaltensweisen in eine Architektur für sensomotorische Systeme. Am Beispiel der Analyse eines Tischdeckenszenarios werden videobasierte Verfahren, die Struktur des Systems und die Kopplung der videobasierten Module mit der Planungs- und Ausführungsebene vorgestellt.

## 1 Einleitung

Die Realisierung visueller Verhaltensweisen auf einem autonomen mobilen Roboter geht über die Entwicklung von Verfahren zur Auswertung von Bilddaten hinaus. Die hohen Anforderungen visueller Sensoren an Übertragungsbandbreite erfordern eine aufgabenabhängige Konfiguration visueller Verhaltensweisen [1]. So ist es notwendig, bestimmte Module zur Ausführungszeit zu aktivieren bzw. zu deaktivieren, wenn sie nicht mehr benötigt werden.

Echtzeitanforderungen, die sich aus dem Einsatz mobiler Roboter in Alltagsumgebungen ergeben erfordern die dynamische Anpassung der Verfahren an den aktuellen Aufgaben- und Umgebungskontext und die Nutzung von Kontextinformation. Durch Nutzung dieser Information können visuelle Sensoren geeignet parametrisiert werden, Bearbeitungsbereiche eingeschränkt und die Zahl möglicher Hypothesen reduziert werden. Die kontextabhängige Konfiguration der Algorithmen erlaubt es somit den Berechnungsaufwand auf das zu minimieren, was notwendig ist, um die aktuellen Ziele zu erreichen. Dabei soll soviel Wissen wie möglich genutzt werden: Was ist die Aufgabe, die der Roboter durchzuführen hat? Wie ist der Zustand des Weltmodells? Was erwartet der Roboter zu sehen? Welche Informationen sind über wahrzunehmende Objekte und Szenenbestandteile bekannt? Über welche Fähigkeiten verfügen die verwendeten Sensoren und wie können diese eingesetzt werden, um den Erkennungsprozeß möglichst einfach, exakt und schnell durchführen zu können?

In diesem Artikel wird neben den eingesetzten Verfahren zur Erkennung von Objekten die Integration in die Robotersteuerungsarchitektur dargestellt. Es zeigt sich, daß die kontextabhängige Konfiguration für die robuste Ausführung erforderlich ist. Interessante Regionen im Bild, die mit komplexen Operatoren analysiert werden werden durch Verknüpfung verschiedener Hinweise, die sich aus der Anwendung von niederen Bildverarbeitungsoperatoren ergeben bestimmt. Die Operatoren werden aufgrund von Kontextwissen und Informationen, die aus den Bildern selbst gewonnen werden, geeignet kombiniert und parametrisiert. So geht Wissen über Farbinformation oder die erwartete Zielposition

eines Objekts ein, um den Suchbereich auf bestimmte Regionen einzuschränken. Die Kenntnis räumlicher Relationen zwischen Szenenbestandteilen wird genutzt, um interessante Regionen für die Suche zu bestimmen. Ist beispielsweise bekannt, daß sich ein gesuchtes Objekt auf einem Tisch befindet, so wird die Suche nur in diesem Bereich stattfinden. Das Vorwissen über Relationen wird zusätzlich genutzt, um bestimmte Objekthypothesen auszuschliessen.

Auch die Auswahl der gewählten Verfahren hängt vom aktuellen Kontext ab: fordert eine Aufgabe strenge Echtzeitanforderungen, wie z.B. das Verfolgen eines Objekts müssen diese unbedingt erfüllt werden. Dies geschieht durch Reduktion der Bildauflösung und die Nutzung von Wissen über die erwartete Position und Erscheinung des Objekts. Beim Verfolgen ist ein Genauigkeitsverlust zulässig, da die Objektposition kurz zuvor genau bestimmt wurde oder für das Folgen hohe Genauigkeit nicht notwendig ist. Sollen andererseits statische Objekte manipuliert werden, so ist die Genauigkeit der Positionsbestimmung essentiell, die Ausführungszeit aber weniger bedeutsam, da das Objekt seine Position nicht verändern wird. Auch Vorwissen über die Umgebung ist für die Verfahrensauswahl sinnvoll nutzbar. Ist bekannt, daß bestimmte Objekteigenschaften in der Umgebung selten auftreten, so werden diese zur Hypothesengenerierung genutzt. So ist die Bewegungsanalyse einer Szene sinnvoll, wenn der Szenenhintergrund und der Beobachter statisch ist, sich das Objekt aber bewegt. Ansonsten wird ein Modell der Erscheinung (Farb- oder Texturverteilung) oder Entfernungsinformation zur Vordergrund-Hintergrund-Trennung verwendet.

Ein anderer Punkt, der eine weitreichende Konfiguration der visuellen Verfahren sinnvoll macht ist die Begrenzung der verfügbaren Ressourcen. Müssen zwei Aufgaben gleichzeitig erledigt werden (zB. Freiraumüberwachung und Verfolgen eines Objekts), so kann die Genauigkeit beider Verfahren reduziert werden, um eine schritthaltende Ausführung zu ermöglichen.

In Abschnitt 2 wird das Videosystem vorgestellt, die Möglichkeiten der Sensoren und das Kameramodell beschrieben. Abschnitt 3 stellt eine modellbasierte Objekterkennung und die Repräsentationen von Modellen, Objekten und Umwelt vor. Anschließend werden kurz vorhandene videobasierte Verhaltensweisen und schliesslich die Integration in das Gesamtsystem beschrieben.

## 2 Videosystem

Zur Objekterkennung und Szeneninterpretation wird ein monokulares Farbkamerasystem eingesetzt, das auf einer Schwenk-Neige-Einheit montiert ist. Die optischen Eigenschaften und die Blickrichtung der Kamera können über serielle Schnittstellen parametrisiert werden. Dies erlaubt die aufgabenabhängige Konfiguration des Sensorsystems. Über die Steuerung der Kamerabrennweite wird der zu analysierende Bildausschnitt festgelegt. Ist die Aufgabe das Suchen von Objekten in der Umgebung oder die Objektverfolgung, so wird eine Weitwinkelseinstellung gewählt, die die Überwachung eines großen Umgebungsbereiches ermöglicht. Zur exakten Positionsbestimmung und Objekterkennung wird auf kleinen Umgebungsbereich fokussiert, der dann analysiert wird. Die Schwenk-Neige-Einheit erlaubt die Mitführung der Kamera beim Verfolgen von Objekten und die Fokussierung auf interessante Bereiche.

### 2.1 Kameramodell

Zur Bestimmung der optischen Parameter der Kamera wird ein Kameramodell nach [6] verwendet. Dieses modelliert die Kamera durch 11 externe und interne Parameter als Kamera mit radialer Linienverzerrung.

Genutzt wird das Kameramodell zur

- Ausrichtung der Kamera auf einen interessanten Umgebungsbereich  
Ist die ungefähre Position eines Objekts a-priori aus der Wissensbasis oder einer vorhergehenden

Szenenanalyse bekannt, so wird die Kamera auf diesen Bereich ausgerichtet, sofern er sich in der Nähe des Roboters befindet und die Erkennung von dieser Ansicht durchgeführt.

- Bestimmung der Weltposition ausgezeichneter Punkte in der Szene  
Zur Objektklassifikation und Positionsbestimmung werden ausgezeichnete Punkte an Objekten bestimmt. Die Position dieser ausgezeichneten Punkte in der Welt wird durch eine inverse Projektion bestimmt.
- Generierung von virtuellen Ansichten  
Unser Ansatz zur Erkennung von Objekten basiert auf der Verwendung virtueller Ansichten der Szene. Abhängig von der Kameraposition und -orientierung wird das Bild der Szene in eine virtuelle Ansicht transformiert. Ist bekannt, daß das zu erkennende Objekt planar ist (Teller) und auf dem Tisch steht, so wird eine Aufsicht auf die Szene generiert werden. Die Bestimmung von Szenenbestandteilen erfolgt dann in dieser Ansicht (siehe Abbildung 1).

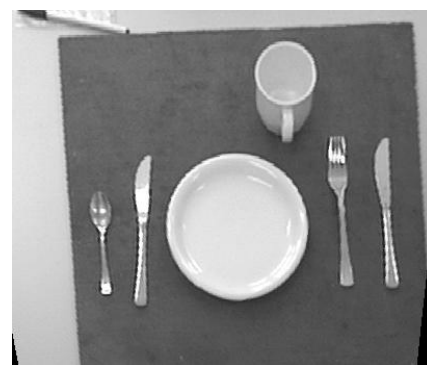


Abbildung 1: Top-View Ansicht einer Tischszene

### 3 Modellbasierte Objekterkennung

Zur videobasierten Objekterkennung und -lokalisierung werden zweidimensionale Modellansichten von Objekten verwendet. Die Erkennung basiert auf dem Vergleich der Modelle mit Bildbestandteilen anhand von Konturen und Grauwertmustern.

#### 3.1 Modellrepräsentation

Alle bekannten, zu erkennenden Objekte sind in einer Modelldatenbasis gespeichert. Diese enthält spezifische Modelle für charakteristische Modellansichten. Jede dieser Ansichten wird durch verschiedene Merkmale beschrieben. Neben einfachen formbeschreibenden Merkmalen wie Größe der sichtbaren Oberfläche, Exzentrizität und Orientierung werden Objekte durch Varianten von Momenten-Modellen, Konturmodellen [2] und Eigenraummodellen [?] beschrieben. Während die einfachen formbeschreibenden Merkmale zur Hypothesengenerierung verwendet werden, dienen Kontur- und Eigenmodelle zur Klassifikation.

#### 3.2 Objektdatenbasis

Die Objektdatenbasis spielt eine wichtige Rolle in der Interaktion mit anderen Modulen des Robotersystems. Sie enthält alle bereits gefundenen oder vorab bekannten Objekte mit ihren Eigenschaften.

Sie wird genutzt, um Erwartungen über ein bestimmtes Objekt zu generieren: Wie sieht es aus, wo ist es zu erwarten oder welche Methoden werden zur Erkennung einer spezifischen Instanz verwendet.

Die Bezeichnung von Objektinstanzen erfolgt über eindeutige Identifikatoren. Über diese können Attribute einer Instanz erfragt werden oder Verhaltensweisen mit dieser Instanz ausgeführt werden. Die Objektdatenbasis dient auch dazu festzustellen, ob ein gefundenes Objekt neu oder bereits bekannt ist. Hierzu werden Attribute eines gefundenen Objekts mit allen bekannten Instanzen der Datenbasis verglichen und falls sich eine Übereinstimmung, zB. in Position, Orientierung, Typ und Farbmuster ergibt ein Abgleich durchgeführt.

### 3.3 Schnittstelle des Objekterkennungsmoduls

Das Modul zur Erkennung und Lokalisierung von Objekten nutzt als Klient die Funktionalitäten anderer Module des Systems (beispielsweise eines Image-Servers, der Bilddaten zur Verfügung stellt oder eines Kamera-Servers, der Informationen über das Kameramodell liefert). Als Server stellt es anderen Modulen Funktionalitäten zur Objekterkennung und -lokalisierung bereit.

- **Aktivierung von Verhaltensweisen:** Suchen, Verifizieren oder Verfolgen von Objekten
- **Definition von Zielobjekten:** Über die Schnittstelle zur Definition von Zielobjekten lassen sich die Eigenschaften von einem oder mehreren Objekten festlegen, die aufzufinden sind. Die wichtigsten sind:

Objektyp	legt die Klasse des gesuchten Objekts fest. Dies dient zur Auswahl der entsprechenden Modelle aus der Modelldatenbasis.
Objektfarbe	bezeichnet symbolisch die erwartete Farbe des Objekts. Dies können symbolische Bezeichner wie zB. <i>rot</i> oder Farbverteilungen in Form von Histogrammen sein.
Objektposition	bezeichnet symbolisch (z.B. <i>auf dem Boden</i> ) oder numerisch die erwartete Zielposition eines Objekts.
Objekteigenschaften	gibt spezifische Eigenschaften des Objekts an, die zur Erkennung genutzt werden können (z.B. <i>Objekt bewegt sich</i> ).
Relationen	dienen zur Definition räumlicher Beziehungen zwischen Objekten, die zur Erkennung verwendet werden können (z.B. <i>Objekt Messer befindet sich rechts von Objekt Teller</i> ). Dies dient zur Einschränkung des Suchbereichs und zur Verifikation.
Detektionsmethoden	zur Erkennung des Objekts können spezielle Detektionsmethoden definiert werden, die Default-Methoden überschreiben.
- **Bereitstellung von Information über Zielobjekte**  
liefert auf Anfrage Eigenschaften einer spezifizierten Objektinstanz, wie Objektklasse, Farbdefinition, Position im Weltkoordinatensystem, Relationen zu anderen gefundenen Objekten.

## 4 Videobasierte Verhaltensweisen

Das vorgestellte Modul zur Objekterkennung wird verwendet, um in Kooperation mit anderen Verhaltensmodulen auf einfache Weise komplexere Verhaltensweisen zu realisieren.

- **Suchen von Objekten:** (`search ((type ball) (color red))`)  
durchsucht die Umgebung bis ein spezifiziertes Zielobjekt gefunden wurde oder die Operation abgebrochen wird. Gefundene Objekte werden benannt, lokalisiert und in die Objektdatenbasis eingetragen.

- Tracking von Objekten: (track (objectid))  
Zentrieren der Kamera auf das Zielobjekt, wobei die Objekteigenschaften werden kontinuierlich aktualisiert werden.
- Annähern an Objekte: (follow (objectid))  
Zusätzlich zum Tracking wird die Position des Zielobjektes an ein Modul zur Bewegungssteuerung geschickt, das unter Hinterhindernisvermeidung ein Folgeverhalten realisiert [3].
- Analyse einer Szene: (analyse)  
durchsucht die Umgebung vollständig und benennt alle gefundenen Objekte eindeutig.
- Erkennung statischer Handgesten und Zeigerichtungsbestimmung: (recognize\_gesture) [5]

## 5 Architektur

Die zugrundeliegende Systemarchitektur wird in [4] dargestellt. Auf Implementierungsebene setzt sie sich aus drei Ebenen zusammen: die Module der *Skill-Ebene* realisieren robuste, reaktive Methoden zur Bewegungsführung, Umgebungserfassung, Objekt- und Gestenerkennung. Diese Ebene stellt außerdem Methoden zur Ansteuerung der Sensorik und Aktorik bereit. Die *Planausführungsebene* übernimmt die zentrale Kontrolle, indem sie auf der Agenda anstehende Aufgaben geordnet zur Ausführung bringt. Sie erlaubt die Realisierung unterschiedlicher Verhaltensweisen durch entsprechende Parametrierung der Module auf der Skill-Ebene. Situationsabhängig werden geeignete Verhaltensmodule ausgewählt, die zum Erreichen des Zielzustandes geeignet sind. Die kommandierten Module melden Ereignisse zurück, anhand derer der Weltzustand im RAP-Memory aktualisiert wird.

Die *deliberative Ebene* setzt sich aus Wissensbasis und symbolischen Planer zusammen. Der Agendainterpreter nutzt den symbolischen Planer, indem er ihn für spezielle vordefinierte Probleme zusammen mit dem dafür notwendigen Teil der Wissensbasis aufruft. Derzeit wird als Wissensbasis das RAP-Memory genutzt. Dieser Teil des Systems arbeitet ausschließlich mit propositionaler Logik. Die dort modellierten Fakten beschreiben fahrbare Wege des Roboters, Orte und Objekte mit ihren Eigenschaften, die für die Generierung eines zielgerichteten Handlungsplanes relevant sind. Die Abbildung 2 zeigt einen vereinfachten Ausschnitt der Gesamtarchitektur. Pfeile, die mit durchgezogenen Linien dargestellt wurden zeigen den Datenfluß, Pfeile mit gestrichelten Linien Signalfluß zur Laufzeit.

## 6 Zusammenfassung und Ausblick

In der vorgestellten Arbeit ist die Kopplung zwischen deliberativer und bildverarbeitender Ebene ausschließlich auf den Handlungskontext des Roboters ausgerichtet. Hiermit ist es möglich, den Berechnungsaufwand auf die im Handlungskontext notwendigen Schritte zu reduzieren. Der eigentliche Erkennungsprozess profitiert noch wenig von den konzeptnahen Beschreibungen.

Erfolgsversprechend scheint die Modellierung der Abhängigkeiten von Merkmalen einzelner Objekte, die zur Erkennung herangezogen werden. Aus dieser Feststellung ergibt sich eine Fülle von Verbesserungsmöglichkeiten. Es bleibt zu prüfen, welche Abhängigkeiten sich als geeignet erweisen, um die Erkennungsleistung zu verbessern.

### Förderung

Diese Forschungsarbeiten wurden im Rahmen des BMBF-Leitprogramms *Mensch-Technik-Interaktion in der Wissensgesellschaft* im Projekt *Intelligente anthropomorphe Assistenzsysteme - MORPHA* durchgeführt (Fördernummer 01 IL 902 F6.)

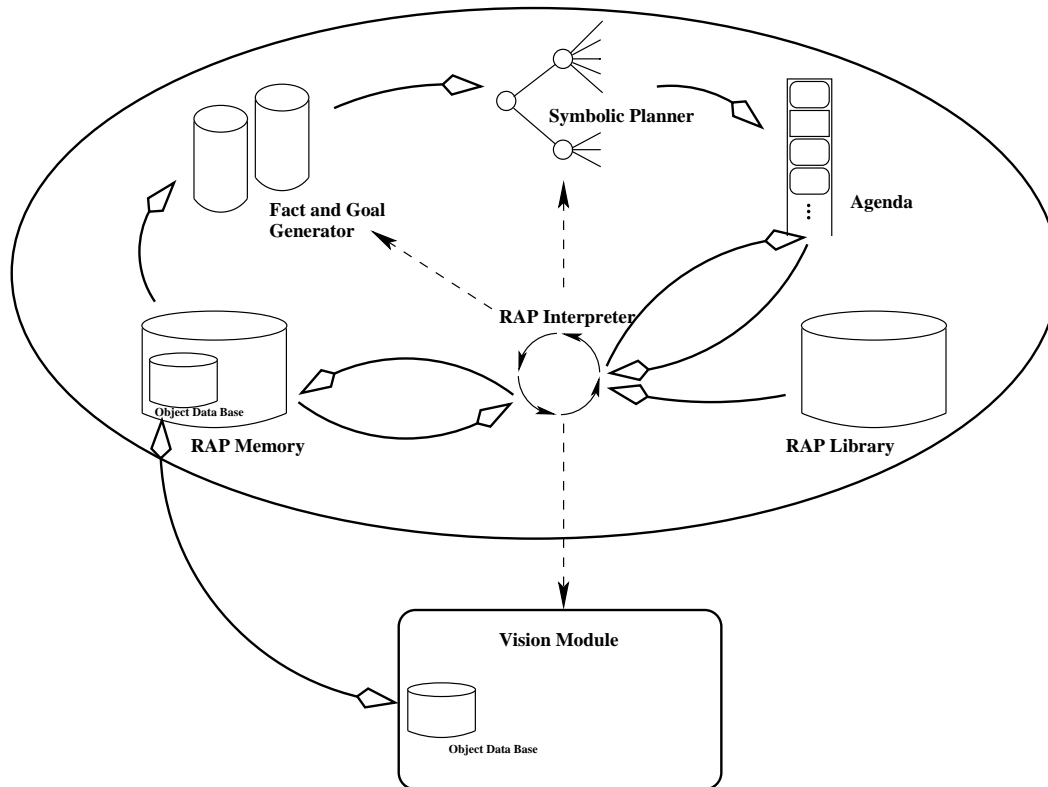


Abbildung 2: Vereinfachter Ausschnitt der Gesamtarchitektur

## Literatur

- [1] J.R. Firby, P.N. Prokopowicz, M.J. Swain, and R.E. Kahn. Gargoyle: An environment for real-time, context-sensitive, active vision. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI-96, Portland OR, August 1996*, pages 930–937, 1996.
- [2] D. Huttenlocher, D. Klanderman, and A. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.
- [3] C. Schlegel, J. Illmann, H. Jaberg, M. Schuster, and R. Wörz. Integrating vision based behaviors with an autonomous robot. In *International Conference on Vision Systems, ICVS '99, Las Palmas de Gran Canaria, Canary Islands, Spain, 1999*.
- [4] C. Schlegel and R. Wörz. The software framework smartsoft for implementing sensorimotor systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyongju, Korea, volume 3*, pages 1610–1616, 1999.
- [5] M. Strobel, J. Illmann, and E. Prassler. Intuitive programming of a mobile manipulator system designed for cleaning tasks in home environments. In *Proceedings of the 2001 IEEE Int. Conf. On Field and Service Robotics (FSR 2001), Helsinki, Finland, 2000*.
- [6] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4), 1987.