

COBOLT: Ein System zur datengetriebenen Verfolgung bewegter Objekte in Echtzeit

J. Denzler, H. Niemann

Lehrstuhl für Mustererkennung (Informatik 5), Universität Erlangen–Nürnberg,
Martensstr. 3, 91058 Erlangen, Germany

Kurzfassung Dieser Beitrag stellt das Echtzeit-Objektverfolgungssystem COBOLT vor, das eine datengetriebene Verfolgung bewegter Objekte in Echtzeit ermöglicht. Das System besteht aus den Modulen Objektdetektion, Objektverfolgung, Prädiktion und Kamerasteuerung. Ein spezielles Modul zur Kontrolle des Ablaufs (Aufmerksamkeitskomponente) überwacht den Gesamtlauf des Systems und detektiert und behebt Fehler in der Verarbeitung. Damit gelingt eine robuste Verfolgung in Echtzeit, die keinerlei Spezialhardware oder problemabhängiges Wissen benötigt und auf Standard-Arbeitsplatzrechnern zum Einsatz kommt.

Die Robustheit und Qualität des Systems zeigt der Einsatz in einer natürlichen Umgebung. Wir stellen Ergebnisse von über fünf Stunden Verfolgung von Fußgängern für unterschiedliche Witterungsbedingungen vor (Schneematsch, Sonnenschein, bewölkt). Die Verfolgung gelingt in bis zu 92 % der Fälle trotz wechselnder Beleuchtung, natürlicher Umgebung und mehreren bewegten Objekten.

1 Einleitung

In den vergangenen Jahren wurden im Bereich des Rechnersehens zunehmend Verfahren entwickelt, deren reduzierte Komplexität einen Echtzeiteinsatz ermöglicht. Dazu zählen z.B. Algorithmen zur Detektion und Verfolgung bewegter Objekte [7, 10]. Zweifellos stellen solche Ansätze auf dem Gebiet des signalnahen Rechnersehens für die angestrebte Konstruktion autonomer mobiler Systeme, sowohl im industriellen Bereich als auch im Dienstleistungssektor neben den generellen Problemen, die in der Robotik behandelt werden, einen wichtigen Forschungsgegenstand dar.

Ein wichtiger Aspekt von echtzeitfähigen Verfahren ist der Einsatz in einem Gesamtsystem, das im Falle der Verfolgung bewegter Objekte, in einem geschlossenen Kreislauf aus Wahrnehmung (Bildaufnahme und Verarbeitung) und Handlung (Kamerasteuerung) arbeiten muß. Erst im sogenannten Echtzeitbetrieb stellen sich Probleme heraus, die bei der "offline" Verarbeitung von vorab aufgenommenen Bildfolgen nur teilweise erkannt werden können. Als Beispiel dafür kann die Erkennung und Behebung von Verarbeitungsfehlern genannt werden, die in einem autonomen mobilen System die erforderliche Robustheit und Verfügbarkeit über einen langen Zeitraum (Stunden, Tage) sicherstellt.

Auf dem Gebiet der Bewegungserkennung und -verfolgung gibt es nur wenige Arbeiten [11], die eine Integration in einem Gesamtsystem betrieben haben. Das bekannteste Beispiel ist VaMP, das den Einsatz in einem fahrerunterstützenden System in einem Fahrzeug auf Autobahnen ermöglicht [5].

Unser Beitrag unterscheidet sich von der Konzeption von bisherigen Arbeiten dahingehend, daß weder Spezialhardware (z.B. DataCube [1], Transputer

[5]) zum Einsatz kommt noch das System spezielles anwendungsspezifisches Wissen über den Problemkreis benötigt. Damit kann das System auf Standard-Arbeitsplatzrechnern und in unterschiedlichen Szenarien ohne Modifikation eingesetzt werden. Die Entwicklung erfolgte in der objektorientierten Programmiersprache C++, die zum einen eine leichte Portierbarkeit sicherstellt, zum anderen eine mögliche Erweiterung um wissenbasierte, high-level Komponenten eines Musteranalyse-System einfach ermöglicht. Erste Beispiele dazu werden wir im folgenden in Form einer automatischen 3D-Modellbildung zur Prädiktion geben.

Der Beitrag gliedert sich in folgende Abschnitte. In Abschnitt 2 stellen wir die Struktur des Gesamtsystems COBOLT vor und gehen auf die wichtigsten Module genauer ein. Die Robustheit des System wird auf Grundlage von fünf Stunden Echtzeitverfolgung in einer natürlichen Umgebung (Verfolgung von Fußgängern) bei unterschiedlichen Witterungsbedingungen in Abschnitt 3 demonstriert. Der Beitrag schließt mit einer Bewertung des Ansatzes und einer Zusammenfassung.

2 Das System COBOLT

Im folgenden werden die einzelnen Module des Systems COBOLT vorgestellt. Ein Überblick über das System befindet sich in Bild 1. Auf eine ausführliche Beschreibung muß aus Platzgründen verzichtet und dazu auf [2] verwiesen werden.

Die beiden Hardwarekomponenten des Systems (Bildeinzug, Kamera-Regler) sind in einer objektorientierten Klassenhierarchie verborgen. Für neue Komponenten, z.B. neue Bildeinzugs hardware, muß lediglich die Schnittstelle in den Klassen implementiert werden. Die Verbindungen zwischen den Modulen stellen Datenübertragungs- oder Kommunikationspfade dar. Zur Erhöhung der Systemleistung wurde das Modul Kamerasteuerung mittels der parallelen virtuellen Maschine [6] auf einen zweiten Rechner des Typs HP 735/99 verteilt. Die restlichen Module laufen auf einer SGI Onyx, mit zwei R10000 Prozessoren. An dieser Stelle sei ausdrücklich darauf hingewiesen, daß die Bildeinzugsrate (25 Bilder pro Sekunde) der Onyx im Vordergrund steht, deren Rechenleistung im Vergleich zur HP 735/99 keine Rolle spielt.

Das System arbeitet in zwei Phasen. In der Initialisierungsphase (IP) detektiert das Modul Bewegungsdetektion bei statischer Kamera Bewegung in der Szene und initialisiert gegebenenfalls die Verfolgung. Dazu wird in die Verfolgungsphase (VP) gewechselt, in der das Modul Objektverfolgung die Kontur des bewegten Objekts verfolgt und dessen Position im Bild an das Modul Kamerasteuerung sendet. Bei vorhandenem 2D- oder 3D-Wissen führt das Modul Vorhersage eine Prädiktion der Konturveränderung durch. Über den Ablauf während der VP wacht das Modul Aufmerksamkeitskomponente.

Das Modul Bewegungsdetektion. Neben der Verfolgung selbst, muß auch die automatische Initialisierung der Verfolgung, d.h. die Bewegungsdetektion, Echtzeitanforderungen genügen. Die Detektion von Bewegung geschieht bei statischer Kamera. Dadurch kann ein sehr effizientes Differenzbildverfahren zum Einsatz kommen, das als Ergebnis nach einer Binärisierung und Verschmelzung von Pixeln im Bild, an denen Grauwertänderungen stattgefunden haben, eine signifikante Region im Bild liefert. Von dieser Region wird angenommen, daß sie einem bewegten Objekt entspricht. Bei der Auswahl unter mehreren Regionen wird die erwartete Größe eines bewegten Objekts berücksichtigt und die größte ausgewählt.

Der Schwellwert für die Binärisierung paßt sich automatisch der Umgebung an, indem die Anzahl von "bewegten" Regionen im Bild minimiert wird. Der Einfluß

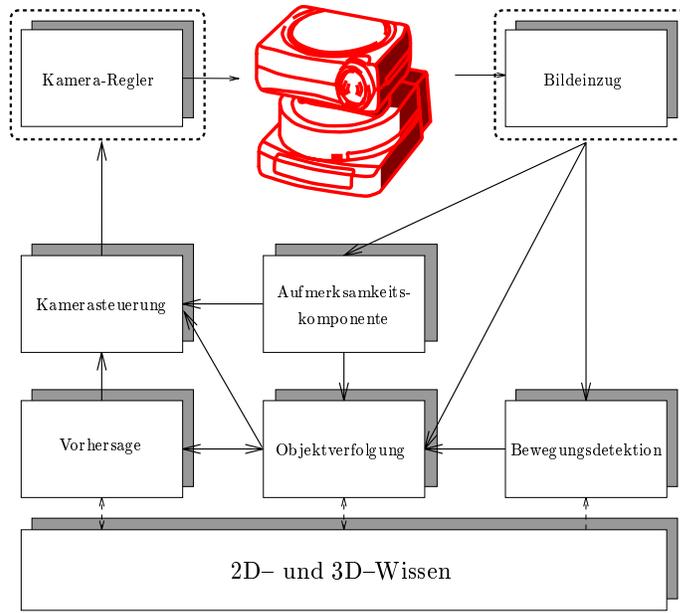


Bild 1. Überblick über das System COBOLT zur Bewegungsverfolgung

von Kameraräuschen, plötzliche Beleuchtungsänderungen oder Hintergrundbewegungen (Bäume, Sträucher) wird somit automatisch reduziert. Bei mehreren bewegten Objekten in der Szene (im aktuellen System: mehr als 8 bewegte Objekte) führt dies ebenfalls zu einer Erhöhung des Schwellwertes. Wird über eine längere Zeit hinweg (im aktuellen System: eine Sekunde) keine Grauwertänderung im Bild detektiert, so wird der Schwellwert sukzessive herabgesetzt. Als Vorteil dieser automatischen Anpassung des Schwellwertes ergibt sich eine Unabhängigkeit des Moduls von der zu bearbeitenden Szene. Während der ersten, typischerweise zwei bis drei Sekunden muß sich die Bewegungsdetektion an die Umgebung adaptieren, mit der Folge, daß in dieser Zeit keine Objekte lokalisiert werden können.

Das Modul Objektverfolgung. Das Modul Objektverfolgung erhält von der Bewegungsdetektion eine Region im Grauwertbild. Dessen Schwerpunkt dient als initialer Referenzpunkt $\mathbf{m} = (x_m, y_m)^T$ für die sogenannten aktiven Strahlen zur Konturverfolgung [4]. Das Verfahren der aktiven Strahlen baut auf den aktiven Konturen auf [8], verwendet jedoch eine radiale Repräsentation der Konturpunkte. Damit wird die Konturextraktion im 2D auf 1D-Suchprobleme reduziert. Dies wirkt sich in einer deutlich geringeren Komplexität der Konturextraktion aus. Ein aktiver Strahl $\varrho_{\mathbf{m}}(\phi, \lambda)$ ist definiert als

$$\varrho_{\mathbf{m}}(\phi, \lambda) = f(x_m + \lambda \cos(\phi), y_m + \lambda \sin(\phi)), \quad 0 \leq \lambda \leq n_\phi, \quad (1)$$

d.h. diejenigen Grauwerte, die ausgehend vom Referenzpunkt \mathbf{m} in Richtung ϕ liegen. Die Repräsentation der Kontur eines Objekts erfolgt über den Referenzpunkt \mathbf{m} und der Funktion $\lambda(\phi)$, die anschaulich den Abstand eines Konturpunktes zum Referenzpunkt beschreibt. Für die Gesamtkontur definiert man eine

sogenannte interne und externe Energie. Die interne Energie E_i beschreibt die Verformungsmöglichkeit und Kohärenz der Kontur, während in der externen Energie E_e die zu extrahierenden Bildmerkmale enthalten sind. Ausgehend von der Energiedefinitionen aktiver Konturen [8] können für aktive Strahlen folgende Energien abgeleitet werden [4]:

$$E_i = E_i(\lambda(\phi)) = \frac{\alpha(\phi) \left| \frac{\partial}{\partial \phi} \lambda(\phi) \right|^2 + \beta(\phi) \left| \frac{\partial^2}{\partial \phi^2} \lambda(\phi) \right|^2}{2}. \quad (2)$$

$$E_e = H(\varrho_{\mathbf{m}}(\phi, \lambda)) = - \left| \frac{\partial}{\partial \lambda} \varrho_{\mathbf{m}}(\phi, \lambda) \right|^2. \quad (3)$$

Die Konturextraktion kann nun als ein Optimierungsproblem definiert werden: Gesucht ist eine Funktion $\lambda^*(\phi)$, die das Funktional E

$$E = \int_0^{2\pi} [E_i(\mathbf{c}_{\mathbf{m}}(\phi)) + E_e(\mathbf{c}_{\mathbf{m}}(\phi))] d\phi \quad (4)$$

$$(5)$$

minimiert. Dazu wird vergleichbar zu aktiven Konturen die zugehörige Euler-Lagrange Differentialgleichung im Diskreten iterativ gelöst. Aus der so ermittelten Funktion $\lambda^*(\phi)$ kann dann die Repräsentation der extrahierten Kontur über deren Konturpunkte $\mathbf{c}_{\mathbf{m}}(\phi)$ in der Bildebene bestimmt werden:

$$\mathbf{c}_{\mathbf{m}}(\phi) = (x_m + \lambda^*(\phi) \cos(\phi), y_m + \lambda^*(\phi) \sin(\phi))^T, \quad 0 \leq \phi < 2\pi. \quad (6)$$

Einer der wichtigsten Vorteile der aktiven Strahlen gegenüber den aktiven Konturen liegt in der Reduktion der Energieminimierung vom 2D auf ein 1D-Suchproblem entlang des aktiven Strahls $\varrho_{\mathbf{m}}(\phi, \lambda)$. Desweiteren kann ein Any-Time Algorithmus zur Konturextraktion angegeben werden, der eine dynamische Anpassung der Genauigkeit der Konturrepräsentation an die zur Verfügung stehende Rechenzeit ermöglicht. Nähere Einzelheiten zu dem Verfahren befinden sich in [2, 4].

Das Modul Aufmerksamkeitskomponente. Ein wichtiger Aspekt eines Echtzeitsystems ist dessen Fähigkeit, auch fehlerhafte Ergebnisse, wie sie zwangsläufig bei einem Einsatz in natürlicher Umgebung auftreten, zu handhaben. Dabei soll das System nicht abbrechen, sondern die Fehler erkennen, beheben und somit über einen längeren Zeitraum (Stunden, Tage) einsatzfähig bleiben. Im System CO-BOLT sorgt das Modul Aufmerksamkeitskomponente für eine robuste Verarbeitung. Fehler treten meistens in der automatischen Initialisierung der Verfolgung bzw. in Konturextraktion selber auf. Beispielsweise können starke Hintergrundkanten in der Nähe des bewegten Objekts innerhalb des Minimierungsschrittes (5) ein besseres Minimum als die Objektkontur darstellen. Als Ergebnis wird statt der Objektkontur eine Hintergrundkontur extrahiert. Um solche Fehler während der Verfolgung zu erkennen, werden Formmerkmale der Kontur (x - und y -Moment) berechnet. Ändern sich diese Momente von einem Bild zum nächsten stark, so liegt ein Fehler in der Konturextraktion vor. Dies erkennt die Aufmerksamkeitskomponente, hält die Verfolgung an und schaltet das System zurück in die IP. Zusätzlich erkennt man bei einer Prädiktion Fehler in der Verfolgung an einer großen Abweichung zwischen vorhergesagter und extrahierter Objektkontur.

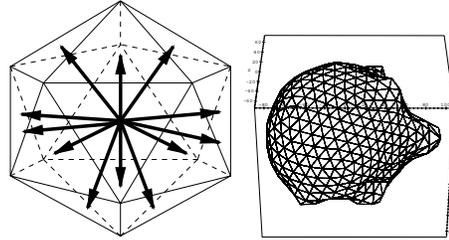


Bild 2. Oben: Bilder aus 36 Drehtelleraufnahmen mit einer Drehung von jeweils 10 Grad. Unten links: 3D-Strahlenrepräsentation. Unten rechts: Automatisch, aus der Konturinformation generiertes Modell

Das Modul Kamerasteuerung. Der Schwerpunkt der Kontur des bewegten Objekts wird an das Modul Kamerasteuerung gesendet. Mittels eines PID-Reglers werden Steuersignale für eine Multimedia-Kamera (Canon VCC1) berechnet, mittels der die Regelabweichung, d.h. die Entfernung von Objektschwerpunkt und Bildmitte, eliminiert wird. Neben linearen Reglern wurden auch Fuzzy-Regler getestet. Die bei linearen Reglern theoretisch nachweisbare Stabilität gab den Ausschlag für den Einsatz eines PID-Reglers.

Die 3D-Modellbildung und -Prädiktion. Für reine 2D-Bewegungen des Objekts genügt die implizite Prädiktion über den I-Anteil des Kamerareglers, um Teilverdeckungen erfolgreich bearbeiten zu können. Bei Konturänderungen, die auf Bewegungen im 3D zurückzuführen sind (z.B. Drehung mit Erscheinen neuer Kanten) wird eine 3D-Prädiktion benötigt.

Die Repräsentation aktiver Strahlen im 2D läßt sich auf auf eine 3D-Strahlenrepräsentation erweitern (siehe Bild 2, unten links). Liegen typische Vertreter der zu verfolgenden Objekte vor, so kann man in einer Trainingsphase automatisch aufgrund der Konturinformation ein 3D-Strahlenmodell aufbauen (siehe Bild 2). Dazu werden Verfahren aus dem Bereich "shape-from-contour" verwendet [9]. Das dann vorliegende 3D-Modell des Objekt ermöglicht eine sehr effiziente Berechnung der Kontur bei bekannter Lage des Objekts. Während der Verfolgung kann die Lage aufgrund der Konturinformation geschätzt und somit die Bewegung im 3D mittels eines Kalman-Filters vorhergesagt werden. Ein Ansatz, der für aktive Konturen eine solche 3D-Vorhersage durchführt, befindet sich in [3]. Als Ergebnis erhält man im 2D in jede Richtung ϕ einen Suchbereich für λ für den zugehörigen Konturpunkt auf dem Strahl $q_m(\phi, \lambda)$. Der Ablauf ist in Bild 3 verdeutlicht. Die Rechenzeit für eine 2D-Konturberechnung aus einem gegebenen 3D-Modell aus 960 3D-Strahlen liegen in Abhängigkeit von der gewünschten Genauigkeit der 2D-Kontur zwischen 8 und 26 msec auf einer SGI Onyx mit R10000 Prozessor und somit im Echtzeitbereich.

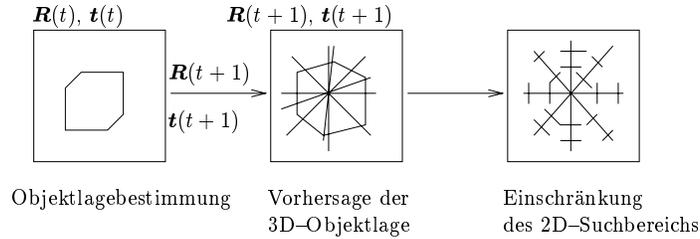


Bild 3. Vorhersage bei Verwendung des 3D-Strahlenmodells und Einschränkung des Suchbereichs für aktive Strahlen.

3 Verfolgung von Fußgängern

Die folgende Auswertung zeigt die Leistungsfähigkeit bei der Verfolgung von Fußgängern in einer natürlichen Umgebung (Platz vor dem Institutsgebäude). Eine Multimedia-Kamera (Schwenk-/Neigefreiheitsgrade) ist aus dem ersten Stock auf den Platz gerichtet. Kein spezifisches Wissen über den Problemkreis (z.B. Form der Kontur von Fußgängern, Bewegungsmodell) kommt zum Einsatz; ohne Modifikationen lassen sich ebenfalls Köpfe von Personen mit dem Anwendungsgebiet Videokonferenz oder innerhalb einer Laborszene eine fahrende Spielzeugsisenbahn verfolgen [2].

Zur Auswertung wurde das Videosignal, das mit einer Markierung des verfolgten Objekts überlagert wurde, synchron auf Videoband geschrieben. Die visuelle Auswertung des Bandes (insgesamt fünf Stunden) führte zu den Ergebnissen, die in Tabelle 1 zusammengefaßt sind. Zusätzlich wurde die Güte der Konturextraktion anhand von 3000 Grauwertbildern (siehe auch Bild 4) manuell bewertet. Vier unterschiedliche Witterungsbedingungen wurden getestet: Schnee(-matsch), Sonnenschein, bewölter Himmel und Regen. Beispielbildfolgen, die auch den datenbasierten Ansatz unterstreichen (Schneepflug) sind in Bild 4 zu finden. Für Regen konnten keine sinnvollen Ergebnisse erzielt werden, da die Bewegungsdetektion in vielen Fällen — aufgrund der Spiegelungen in Pfützen — fehlerhafte Initialisierungen liefert. Dies konnte aufgrund des einfachen Verfahrens zur Bewegungsdetektion erwartet werden.

Wetter	Gesamtzeit	# Personen	Zeit in VP	# verfolgte Personen (Zeit)
Schnee	120 min	226	21 min	101 (11 min)
Bewölkt	90 min	128	14 min	93 (11 min)
Sonne	90 min	152	13 min	102 (12 min)
Gesamt	300 min	506	48 min	296 (34 min)

Tabelle 1. Ergebnis der Verfolgung bei unterschiedlichen Witterungsbedingungen. Ausgewertet ist die Gesamtzeit des Versuchs, die Anzahl von Personen, die sich während dieser Zeit über den Platz bewegt hat, sowie die Zeit, während der sich das System in der Verfolgungsphase befand, und die Anzahl und Zeit der korrekten Verfolgungen.



Bild 4. Beispiele der Verfolgung. Die mittlere Zeile verdeutlicht die Auswirkungen des datenbasierten Ansatzes. Auch Personen aus einer Personengruppe können detektiert und verfolgt werden.

Die Ergebnisse in Tabelle 1 zeigen, daß die Bewegungsdetektion in natürlicher Umgebung im Mittel mit einer Erfolgsrate von 58 % gelingt. Bei Schnee liegt diese Rate bei 44.7 %, bei bewölktem Himmel bei 72.6 %. Während sich das System in der VP befindet, werden im Mittel in 71 % der Fälle (34 min von 48 min) bewegte Personen verfolgt. Während der restlichen Zeit extrahiert das System Hintergrundobjekte. Auch hier liegt mit 52.4 % das schlechteste Ergebnis bei Schnee; bei Sonne oder "bewölkt" liegt die Erfolgsrate bei 92.3 % bzw. 78.6 %, d.h. in 92.3 % der Zeit in der VP wurde auch tatsächlich ein Fußgänger verfolgt. Dies unterstreicht die Leistungsfähigkeit sowohl der Aufmerksamkeitskomponente, als auch des Moduls Objektverfolgung. Die geringere Erfolgsrate bei Schnee(-matsch) läßt sich bei Betrachtung der zugehörigen Grauwertbilder erklären. Im Grauwertbild heben sich z.B. rote Jacken nur sehr schlecht von braunem Schneematsch ab. Da die Verfolgung auf der Extraktion von Konturen basiert (siehe (3)), bereiten schwache Objektkonturen in der Nähe von starken Hintergrundkonturen noch Probleme. Eine Möglichkeit, dieses Problem zu beheben, besteht in einer Erweiterung der Energie in (3) um Texturenergien, wie z.B. die Grauwertvarianzänderung [2].

4 Zusammenfassung und Ausblick

Das in diesem Beitrag vorgestellte System COBOLT ermöglicht eine datenbasierte Verfolgung bewegter Objekte über deren Kontur ohne problemabhängiges Wissen

in Echtzeit. Dabei wird auf keinerlei Spezialhardware zurückgegriffen, so daß eine schrittweise Erweiterung um wissensbasierte Komponenten ermöglicht wird. Ein Beispiel in Form einer 3D-Modellbildung und Vorhersage wurde gegeben.

Die Verfolgung von Fußgängern in natürlicher Umgebung gelingt bei korrekter Initialisierung in bis zu 92 % der Fälle. Die Qualität hängt dabei stark von den Witterungsbedingungen ab. Dies liegt vor allem an der automatischen Initialisierung der Verfolgung, die mit dem Differenzbildverfahren in natürlicher Umgebung erwartungsgemäß schlechter arbeitet als in einer Laborszene.

In der Zukunft werden wir die wissensbasierte Komponente des Systems erweitern, um mit problemabhängigen Wissen eine weitere Steigerung der Güte zu erreichen. Ebenfalls werden wir die angesprochenen "Texturenergien" für aktive Strahlen in das Echtzeitsystem integrieren, die auch mit dem Gradienten kombiniert werden können und somit eine Steigerung der Güte der Verfolgung im Falle von schwachen Objektkonturen erwarten lassen.

Literaturverzeichnis

1. Daniilidis, K.; Hansen, M.; Krauss, C.; Sommer, G.: *Auf dem Weg zum künstlichen aktiven Sehen: Modellfreie Bewegungsverfolgung durch Kameranachführung*, in *DAGM 1995, Bielefeld*, 1995, S. 277–284.
2. Denzler, J.: *Aktives Sehen zur Echtzeitobjektverfolgung*, Dissertation, Technische Fakultät, Universität Erlangen–Nürnberg, Erlangen, Juni, 1997.
3. Denzler, J.; Niemann, H.: *3D Data Driven Prediction for Active Contour Models based on Geometric Bounding Volumes*, *Pattern Recognition Letters*, Bd. 17, Nr. 11, 1996, S. 1171–1178.
4. Denzler, J.; Niemann, H.: *Echtzeitobjektverfolgung mit aktiven Strahlen*, in *Mustererkennung, 1996*, Heidelberg, 1996, S. 84–91.
5. Dickmanns, E.; Behringer, R.; Brüdigam, C.; Dickmanns, D.; Thomanek, F.; Holt, V.: *An All-Transputer Visual Autobahn-Autopilot/Copilot*, in *International Conference on Computer Vision*, Berlin, 1993, S. 608–615.
6. Geist, A.; Beguelin, A.; Dongarra, J.; Jiang, W.; Manchek, R.; Sunderam, V.: *PVM 3.0 User's Guide and Reference Manual*, ORNL/TM-12187, Engineering Physics and Mathematics Division, Oak Ridge National Laboratory, Tennessee, 1993.
7. Hager, G.; Toyama, K.: *X Vision: Combining Image Warping and Geometric Constraints for Fast Visual Tracking*, in Blake, A. (Hrsg.): *Computer Vision - ECCV 96*, Berlin, Heidelberg, New York, London, 1996, S. 507–517, Lecture Notes in Computer Science.
8. Kass, M.; Wittkin, A.; Terzopoulos, D.: *Snakes: Active Contour Models*, *International Journal of Computer Vision*, Bd. 2, Nr. 3, 1988, S. 321–331.
9. Laurentini, A.: *How Far 3D Shapes Can Be Understood from 2D Silhouettes*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 17, Nr. 2, 1995, S. 188–195.
10. Smith, S.; Brady, J.: *ASSET-2: Real-Time motion segmentation and shape tracking*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 17, Nr. 8, 1995, S. 814–820.
11. Uhlin, T.; Nordlund, P.; Maki, A.; Eklundh, J.: *Towards an Active Visual Observer*, in *International Conference on Computer Vision*, Cambridge, Massachusetts, 1995, S. 679–686.