Antragskurzfassung

Schwerpunktprogramm „Kooperierende Teams mobiler Roboter in dynamischen Umgebungen“ (Geschäftszeichen: SPP 1125)

Antragsteller: Dr. Ubbo Visser
Bereich „Intelligente Systeme“

Dr. Thomas Röfer
Bremer Institut für Sichere Systeme, Bereich „Kognitive Robotik“

Technologie-Zentrum Informatik (TZI)
Universität Bremen

Forschungsvorhaben: Automatische Plan- und Intentionserkennung fremder mobiler Roboter in kooperierenden und konkurrierenden dynamischen Umgebungen

Dauer: 6 Jahre
hier: 3. und 4. Jahr

Antrag: Gewährung einer Sachbeihilfe und zwar für:

Personal:	2 wiss. Mitarbeiter BAT IIa für 24 Monate
	4 stud. Hilfskräfte, 19 h/Woche für 24 Monate
Kleingeräte:	2000 €
Verbrauchsmaterial:	1440 €
Reisekosten:	25.000 €

Zusammenfassung des Forschungsvorhabens:

Im Rahmen des Schwerpunktprogramms soll die Technologie autonomer mobiler Roboter in kooperativen und konkurrierenden dynamischen Umgebungen weiterentwickelt werden. Ein noch nicht ausreichend untersuchtes Problem innerhalb vieler Anwendungen autonomer Agenten ist eine adäquate Betrachtung des Umfelds. So ist es z.B. im Rahmen mobiler Rollstühle wichtig, nicht nur Objekte zu erkennen, sondern das Verhalten und die Aktionen anderer autonomer Agenten zu analysieren und ihre künftigen Aktionen zu antizipieren.

Anhand des RoboCup-Szenarios sollen Methoden entwickelt werden, die es ermöglichen, sowohl primitive Aktionen als auch komplexe Verhaltensweisen anderer autonomer Agenten zu klassifizieren, Aktionsfolgen und Strategien zu erkennen und selbst angemessen zu agieren und reagieren.

Die gesammelten Erkenntnisse sollen anschließend sowohl zur Steuerung eines autonomen Rollstuhls als auch zur Weiterentwicklung der Robustheit und Sicherheit in anderen relevanten Bereichen, wie z.B. in elektronischen Märkten, eingesetzt und evaluiert werden.

Automatische Plan- und Intentionserkennung fremder mobiler Roboter in kooperierenden und konkurrierenden, dynamischen Umgebungen

Folge-Antrag an die Deutsche Forschungsgemeinschaft
auf Förderung im Rahmen des Schwerpunktprogramms
„Kooperierende Teams mobiler Roboter
in dynamischen Umgebungen"

an der

Universität Bremen

November 2002

1 Allgemeine Angaben

Antrag auf Gewährung einer Sachbeihilfe im Rahmen des DFG-Schwerpunktprogramms „Kooperierende Teams mobiler Roboter in dynamischen Umgebungen“ (Geschäftszeichen SPP 1125)

1.1 Antragsteller

Dr. Ubbo Visser, Wissenschaftlicher Assistent (C1)
Geburtsdatum: 23. 6. 1964
Nationalität: deutsch
FB3 Mathematik und Informatik
Universität Bremen
Postfach 33 04 40
D-28334 Bremen
Tel. (0421) 218 – 78 40
Fax (0421) 218 – 71 96

Private Anschrift:
Gustav-Heinemann-Str. 22
D-28215 Bremen
Tel. (0421) 43 40 860
Email: visser@informatik.uni-bremen.de

Dr. Thomas Röfer, Wissenschaftlicher Assistent (C1)
Geburtsdatum: 1. 3. 1967
Nationalität: deutsch
FB3 Mathematik und Informatik
Universität Bremen
Postfach 33 04 40
D-28334 Bremen
Tel. (0421) 218 – 46 59
Fax (0421) 218 – 30 54

Private Anschrift:
Rudolf-Alexander-Schröder-Str. 96
D-28215 Bremen
Tel. (0421) 35 06 361
Email: roefer@informatik.uni-bremen.de

1.2 Thema

Automatische Plan- und Intentionserkennung fremder mobiler Roboter in kooperierenden und konkurrierenden, dynamischen Umgebungen

1.3 Kennwort

Strategiediagnose

1.4 Fachgebiet und Arbeitsrichtung

Künstliche Intelligenz, Kognitive Robotik

1.5 Voraussichtliche Gesamtdauer

6 Jahre

1.6 Antragszeitraum

2 Jahre

1.7 Gewünschter Beginn der Förderung

Gewünschter Beginn der Förderung : 01.11.2003

Datum der bisherigen Bewilligung: 01.06.2001 – 31.05.2003

Personalmittel reichen voraussichtlich bis: 31.10.2003 (Beginn 1. Projektphase 01.11.2001)

Sachmittel reichen voraussichtlich: 31.10.2003

1.8 Zusammenfassung

Im Rahmen des Schwerpunktprogramms soll die Technologie autonomer mobiler Roboter in kooperativen und konkurrierenden dynamischen Umgebungen weiterentwickelt werden. Ein noch nicht ausreichend untersuchtes Problem innerhalb vieler Anwendungen autonomer Agenten ist eine adäquate Betrachtung des Umfelds. So ist es z.B. im Rahmen mobiler Rollstühle wichtig, nicht nur Objekte zu erkennen, sondern das Verhalten und die Aktionen anderer autonomer Agenten zu analysieren und ihre künftigen Aktionen zu antizipieren.

Anhand des RoboCup-Szenarios sollen Methoden entwickelt werden, die es ermöglichen, sowohl primitive Aktionen als auch komplexe Verhaltensweisen anderer autonomer Agenten zu klassifizieren, Aktionsfolgen und Strategien zu erkennen und selbst angemessen zu agieren und reagieren.

Die gesammelten Erkenntnisse sollen anschließend sowohl zur Steuerung eines autonomen Rollstuhls als auch zur Weiterentwicklung der Robustheit und Sicherheit in anderen relevanten Bereichen, wie z.B. in elektronischen Märkten, eingesetzt und evaluiert werden.

2 Stand der Forschung, eigene Vorarbeiten

2.1 Stand der Forschung (siehe Projektbericht im Anhang)

2.2 Eigene Vorarbeiten (siehe Projektbericht im Anhang)

2.3 Veröffentlichungen der letzten fünf Jahre (1998-2002)

Bousonville, T.; Knirsch, P.; Timm, I.J. (1999). Einsatz von Graphregel-basierten Agenten zur flexiblen Modellierung von Tourenplanungsproblemen. In: *Proceedings des Workshops „Agententechnologie“ im Rahmen der 23. Jahrestagung für Künstliche Intelligenz*, TZI Bericht 16, Bremen.

Brunn, R., Düffert, U., Jüngel, M., Laue, T., Lötzsch, M., Petters, S., Risler, M., Röfer, T., Spiess, K., Sztybryc, A. (2002). GermanTeam 2001. In: *RoboCup 2001*. Lecture Notes in Artificial Intelligence 2377. Springer. 705-708.

Drücker, C.; Hübner, S.; Schmidt, E.; Visser, U.; Weland, H.-G. (2000). Virtual Werder: Using the Online-Couch to Team Formations. In: Balch, T.; Stone, P.; Kraetschmar, G. (Eds.): *4th International Workshop on RoboCup.* Carnegie Mellon University Press, Melbourne, Australia, 2000. 217-222.

Drücker, C.; Hübner, S.; Schmidt, E.; Visser, U.; Weland, H.-G. (2000). „As time goes by” – Using time series based decision tree induction to analyze the behavior of opponent players. In: RoboCup 2001: RoboCup Soccer World Cup V.

Drücker, C.; Hübner, S.; Visser, U.; Weland, H.-G. (2002). “As time goes by” - Using time series based decision tree induction to analyze the behaviour of opponent players; In A. Birk & S. Coradeschi & S. Tadokoro (Eds.), *RoboCup 2001: Robot Soccer World Cup V* (Vol. 2377, pp. 325-330). Seattle, WA: Springer Verlag.

Düffert, U.; Jüngel, M.; Laue, T.; Lötzsch, M.; Risler, M.; Röfer, T. (2003). GermanTeam 2002. In: *RoboCup 2002*, Lecture Notes in Artificial Intelligence, Springer, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings).

Eschenbächer, J.; Knirsch, P.; Timm, I.J. (2000). Demand Chain Optimization by Using Agent Technology. In: *Proceedings of IFIP WG 5.7 Conference*. Tromsö, Norway.

Haugeneder, H.; Kraetzschmar, G.; Müller, H.J.; Weiß, G.; Wrobel, S. (Hrsg.)(1996). Lernen, Adaption und Selbstorganisation in verteilten intelligenten Systemen. *Forschungsbericht FKI-217-96*, Institut für Informatik, Technische Universität München.

Herzog, O.; Moini, A.; Hewitt, C.; Hofkin, R. (1998). Negotiation-Based Cooperation In Multi-Agent Systems. In: *Software Productivity Consortium, Herndon*, Va., SPC-98050-CMC.

Knirsch, P.; Timm, I. J. (1999a). Adaptive Multiagent Systems Applied on Temporal Logistics Networks. In: *Proceedings 4th International Symposium on Logistics (ISL-99)*. Florence, Italy.

Knirsch, P.; Timm, I. J. (1999b). Multi-Agentensysteme zur Unterstützung temporärer Logistiknetzwerke. In: Kopfer, H.; Bierwirth, Chr. (Hrsg.): *Logistik Management - Intelligente I+K Technologien*. Springer. Berlin.

Kollmann, J.; Röfer, T. (2000). Echtzeitkartenaufbau mit einem 180°-Laser-Entfernungssensor. In: *Autonome Mobile Systeme 2000*. Informatik aktuell. Springer (Im Erscheinen).

Krieg-Brückner, B. (1998). A Taxonomy of Spatial Knowledge for Navigation. In: Schmid, U.; Wysotzki, F. (Eds.). *Qualitative and Quantitative Approaches to Spatial Inference and the Analysis of Movements*. Technical Report, 98-2, Technische Universität Berlin, Computer Science Department.

Krieg-Brückner, B.; Gräser, A.; Lohmann, B. (2000). Autonomiegewinn für behinderte Menschen durch Rehabilitations-Roboter. In: *Impulse aus der Forschung* 1/2000. Universität Bremen. 6-10.

Krieg-Brückner, B.; Röfer, T.; Carmesin, H.-O.; Müller, R. (1998). A Taxonomy of Spatial Knowledge for Navigation and its Application to the Bremen Autonomous Wheelchair. Freksa, Ch.; Habel, Ch.; Wender, K. F. (Eds.): *Spatial Cognition*. Lecture Notes in Artificial Intelligence 1404. Springer. 373-397.

Lankenau, A.; Meyer, O. (1998). Safety in Robotics: The Bremen Autonomous Wheelchair. In: *Proceedings of AMC'98, 4th Int. Workshop on Advanced Motion Control*. Coimbra, Portugal. 524-529.

Lankenau, A.; Meyer, O. (1999). Formal methods in robotics: Fault tree based verification. In: *Proc. of Quality Week Europe*. Brüssel, Belgien.

Lankenau, A.; Röfer, T. (1998). Architecture of the Bremen Autonomous Wheelchair. In: Hildebrand, B.; Moratz, R.; Scheering, Ch. (Hrsg.): *Architectures in Cognitive Robotics*. Technischer Bericht 98/13. SFB 360 „Situierte Künstliche Kommunikatoren". Universität Bielefeld. 19-24.

Lankenau, A.; Röfer, T. (2000a). Rollstuhl „Rolland" unterstützt ältere und behinderte Menschen. In: FIfF-Kommunikation 2/2000, *Informationstechnik und Behinderung*. Forum InformatikerInnen für Frieden und gesellschaftliche Verantwortung (FIfF) e.V. 48-50.

Lankenau, A.; Röfer, T. (2000b). Smart Wheelchairs - State of the Art in an Emerging Market. In: *Zeitschrift Künstliche Intelligenz*. Schwerpunkt Autonome Mobile Systeme. Fachbereich 1 der Gesellschaft für Informatik e.V., arenDTaP. 37-39.

Lankenau, A., Röfer, T. (2001). A Safe and Versatile Mobility Assistant. In: *IEEE Robotics and Automation Magazine* 7, No. 1, March 2001. 29-37.

Lankenau, A.; Röfer, T. (2000). The Role of Shared Control in Service Robots - The Bremen Autonomous Wheelchair as an Example. In: Röfer, T.; Lankenau, A.; Moratz, R. (Hrsg.): *Service Robotics - Applications and Safety Issues in an Emerging Market*. Workshop Notes. European Conference on Artificial Intelligence 2000 (ECAI 2000). 27-31.

Lankenau, A., Röfer, T. (2001). Selbstlokalisation in Routengraphen. In: Levi, P., Schanz, M. (Hrsg.): *Autonome Mobile Systeme 2001*. Informatik aktuell. Springer. 157-163.

Lankenau, A., Röfer, T. (2002). Mobile Robot Self-Localization in Large-Scale Environments. In: *Proceedings of the IEEE International Conference on Robotics and Automation 2002 (ICRA-2002)*. IEEE. 1359-1364.

Lankenau, A., Röfer, T., Krieg-Brückner, B. (2002). Self-Localization in Large-Scale Environments for the Bremen Autonomous Wheelchair. In: *Spatial Cognition III*. Lecture Notes in Artificial Intelligence. Springer, im Erscheinen.

Malsch, Th.; Müller, H. J. (Hrsg.) (1998). Sozionik: Wie VKI und Sozionik von einander lernen können. Harburger Berichte zur Sozionik. 1, Technische Universität Hamburg-Harburg.

Meyer, J.; Adolph, R.; Stephan, D.; Daniel, A.; Seekamp, M.; Weinert, V.; Visser, U. (2002). *Decision-making and Tactical Behavior with Potential Fields;* In: Proceedings of the Proceedings of the RoboCup-2002: Robot Soccer World Cup VI, Fukuoka, Japan, pp.300-307.

Miene, A.; Visser, U.;2002; Interpretation of spatio-temporal relations in real-time and dynamic environments; In A. Birk & S. Coradeschi & S. Tadokoro (Eds.), *RoboCup 2001: Robot Soccer World Cup V* (Vol. 2377, pp. 441-446). Seattle, WA: Springer.

Müller, R. Röfer, T.; Lankenau, A.; Musto, A.; Stein, K.; Eisenkolb, A. (2000). Coarse Qualitative Descriptions in Robot Navigation. In: Freksa, Ch.; Brauer, W.; Habel, Ch.; Wender, K. F. (Eds.): *Spatial Cognition II*. Lecture Notes in Artificial Intelligence 1849. Springer. 265-276.

Musto, A.; Stein, K.; Eisenkolb, A.; Röfer, T. (1999). Qualitative and Quantitative Representations of Locomotion and their Application in Robot Navigation. In: *Proc. of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*. Morgan Kaufman Publishers, Inc. San Francisco, CA. 1067-1073.

Musto, A.; Stein, K.; Eisenkolb, A.; Röfer, T.; Brauer, W.; Schill, K. (2000). From Motion Observation to Qualitative Motion Representation. In: Freksa, Ch.; Brauer, W.; Habel, Ch.; Wender, K. F. (Eds.): *Spatial Cognition II*. Lecture Notes in Artificial Intelligence 1849. Springer. 115-126.

Ranze, K. C.; Hollmann, O.; Müller, H. J.; Herzog, O. (1998). Handling Conflicts in Distributed Assessment Situations.- In: *Workshop 'Conflicts among agents: avoid or use them?'*. Workshop 16, ECAI'98, Brighton, UK.

Röfer, T. (1998a). Panoramic Image Processing and Route Navigation. Dissertation. BISS Monographs 7. Shaker-Verlag.

Röfer, T. (1998b). Strategies for Using a Simulation in the Development of the Bremen Autonomous Wheelchair. In: Zobel, R.; Moeller, D. (Eds.): *Simulation-Past, Present and Future*. Society for Computer Simulation International. 460-464.

Röfer, T. (1998c). Routenbeschreibung durch Odometrie-Scans. In: Wörn, H.; Dillmann, R.; Henrich, D. (Hrsg.): *Autonome Mobile Systeme 1998*. Informatik aktuell. Springer. 122-129.

Röfer, T. (1999). Route Navigation and Panoramic Image Processing. In: *Ausgezeichnete Informatikdissertationen 1998*. B. G. Teubner Stuttgart, Leipzig. 132-141.

Röfer, T. (1999). Route Navigation Using Motion Analysis. In: Freksa, C.; Mark, D. M. (Eds.): *Spatial Information Theory, Proc. COSIT '99*. Lecture Notes in Computer Science 1661. Springer. 21-36.

Röfer, T. (2001). Building Consistent Laser Scan Maps. In: *Proc. of the 4th European Workshop on Advanced Mobile Robots (Eurobot 2001)*. Lund University Cognitive Studies, Vol. 86, 83-90.

Röfer, T. (2001). Konsistente Karten aus Laser Scans. In: Levi, P., Schanz, M. (Hrsg.): *Autonome Mobile Systeme 2001*. Informatik aktuell. Springer. 171-177.

Röfer, T. (2002). Using Histogram Correlation to Create Consistent Laser Scan Maps. In: *Proceedings of the IEEE International Conference on Robotics Systems (IROS-2002)*.

Röfer T. (2003). An Architecture for a National RoboCup Team. In: *RoboCup 2002*. Lecture Notes in Artificial Intelligence. Springer, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings, 388-395).

Röfer, T.; Lankenau, A. (1998). Architecture and Applications of the Bremen Autonomous Wheelchair. In: Wang, P. P. (Ed.): *Proc. of the 4th Joint Conference on Information Systems* 1. Association for Intelligent Machinery. 365-368.

Röfer, T.; Lankenau, A. (1999a). Ensuring Safe Obstacle Avoidance in a Shared-Control System. In: J. M. Fuertes (Hrsg.): *Proc. of the 7th International Conference on Emergent Technologies and Factory Automation*. 1405-1414.

Röfer, T.; Lankenau, A. (1999b). Ein Fahrassistent für ältere und behinderte Menschen. In: Schmidt, G.; Hanebeck, U.; Freyberger, F. (Hrsg.): *Autonome Mobile Systeme 1999*. Informatik aktuell. Springer. 334-343.

Röfer, T.; Lankenau, A. (2000). Architecture and Applications of the Bremen Autonomous Wheelchair. In Wang, P. (Ed.): *Information Sciences* 126:1-4. Elsevier Science BV. 1-20.

Röfer, T., Lankenau, A. (2002). Route-Based Robot Navigation. In: Freksa, C. (Hrsg.): *Künstliche Intelligenz - Themenheft Spatial Cognition*. Fachbereich 1 der Gesellschaft für Informatik e.V., arenDTaP, im Erscheinen.

Röfer, T.; Lankenau, A.; Moratz, R. (Hrsg.) (2000). Service Robotics - Applications and Safety Issues in an Emerging Market. Workshop Notes. European Conference on Artificial Intelligence 2000 (ECAI 2000).

Röfer, T.; Müller, R. (1998). Navigation and Routemark Detection of the Bremen Autonomous Wheelchair. In: Lüth, T.; Dillmann, R.; Dario, P.; Wörn, H. (Eds.): *Distributed Autonomous Robotics Systems*. Springer. 183-192.

Timm, I. J. (1998). Multi-Agentensysteme zur Unterstützung ökologischer Transportlogistik. In: Haasis, H.-D.; Ranze, K.C. (Hrsg.): *Umweltinformatik-98 -Vernetzte Strukturen in Informatik, Umwelt und Wirtschaft* 12. Internationales Symposium „Informatik für den Umweltschutz" der Gesellschaft für Informatik (GI). Bremen.

Timm, I. J. (2000). Multiagent Architecture for D-Sifter – A modern approach to flexible information filtering in dynamic environments. *TZI-Technical Report* 21. Bremen.

Timm, I. J.; Knirsch, P. (1999). Ökologische Optimierung in der verteilten Tourenplanung durch Multi-Agentensysteme. In: *Proceedings des Workshops „Agententechnologie"* im Rahmen der 23. Jahrestagung für Künstliche Intelligenz, TZI-Bericht Nr. 16-99, Bremen, 1999.

Timm, I. J.; Knirsch, P.; Blome, A.; Schröter, M. (1998). Perspektiven von Multi-Agentensystemen in Logistikketten verteilter Produktion. In: Hellingrath, B. (Hrsg.): *Anwenderforum Logistik, KI-98: Entwicklungsrichtungen der Logistik - Anwendungsmöglichkeiten der Künstlichen Intelligenz*, Fraunhofer Institut Materialfluß und Logistik, Dortmund.

Timm, I. J.; Knirsch, P.; Petsch, M.; Visser, U.; Fischer, K.; Herzog, O.; Kirn, S.; Zelewski, S. (Eds.) (1999b). Proceedings des Workshops „Agententechnologie" auf der KI'99: Agententechnologie – Multiagentensysteme in der Informationslogistik und wirtschaftswissenschaftliche Perspektiven der Agenten-Konzeptionalisierung. TZI-Bericht Nr. 16, Bremen.

Timm, I.; Herzog, O.; Woelk, P.-O.; Siebert, K.; Tönshoff, H. K. (1999a). Integrierte agentenunterstützte Arbeits-Planung und Fertigungs-Steuerung (IntaPS). In: Kirn, S.; Petsch, M. (Hrsg.): *Workshop „Intelligente Softwareagenten und betriebswirtschaftliche Anwendungsszenarien"*, Arbeitsbericht Nr. 14, Juli 1999, Technische Universität Ilmenau.

Timm, I.J.; Knirsch, P.; Müller, H.-J.; Petsch, M.; Abchiche, N. , Davidsson, P.; Demazeau, Y.; Garijo, F. J.; Herzog, O.; Kirn, St.; Petrie, C.; Tessier, C. (Eds.) (2000). Agent Technologies and Their Application Scenarios in Logistics. 14^{th} ECAI Workshop Notes (13), Berlin.

Visser, U.; Drücker, C.; Hübner, S.; Schmidt, E.; Weland, H.-G. (2001). *Recognizing Formations in Opponent Teams;* In: Proceedings of the RoboCup 2000, Robot Soccer World Cup IV, Melbourne, Australia, pp.391 - 396

Visser, U.; Weland, H.-G. (2002). *Using online learning to analyze the opponents behavior;* In: Proceedings of the Proceedings of the RoboCup-2002: Robot Soccer World Cup VI, Fukuoka, Japan, pp.72-81.

Werner, S.; Krieg-Brückner, B.; Herrmann, T. (2000). Modelling navigational knowledge by route graphs. In: Freksa, Ch.; Brauer, W.; Habel, Ch.; Wender, K. F. (Eds.): *Spatial Cognition II*. Lecture Notes in Artificial Intelligence 1849. Springer. 295-317.

3 Ziele und Arbeitsprogramm

3.1 Ziele

Die Arbeiten, die in diesem Projekt durchgeführt werden sollen, verfolgen drei Ziele: die Erkennung der Strategie von Opponenten, den Aufbau eines RoboCup-Teams und die Nutzung der Ergebnisse für den Bremer Autonomen Rollstuhl (Roboter) *Rolland* und andere relevante Bereiche, z. B. für Electronic Commerce. Mit diesen Zielen wird ein Beitrag zur Integration von reaktiven und deliberativen Robotersteuerungen geleistet. Die Möglichkeit, Strategien in Echtzeit zu diagnostizieren (z. B. die Erkennung von Handlungen und / oder Handlungsfolgen), ist eine wichtige Unterstützung bei Handlungsentscheidungen. Diese Hypothese soll am Beispiel eines RoboCup-Teams gezeigt werden. Gleichzeitig sind wir überzeugt, dass die zu entwickelnden Methoden und Techniken in anderen Bereichen genutzt werden können.

3.1.1 Strategieerkennung

Dieses Forschungsvorhaben befasst sich mit der automatischen Erkennung von Aktionen, Aktionsfolgen, Taktiken bis hin zu komplexen Strategien in der Umgebung befindlicher Agenten. Eine neue Herausforderung bei der Entwicklung von autonomen physikalischen Agenten besteht darin, die Umgebung adäquat zu modellieren. In diesem Rahmen kommt der Modellierung anderer aktiver Entitäten in der Umgebung eine entscheidende Bedeutung zu. Hier soll untersucht werden, wie Aktionen anderer mobiler Agenten identifiziert und klassifiziert werden können. Da Verhalten im RoboCup als eine über die Zeit geordnete Veränderung (Historie) räumlicher Relationen interpretiert werden kann (auf Basis sensorischer Informationen), ist zu untersuchen, wie Verhalten auf Basis dieser Daten adäquat beschrieben werden kann. Zudem soll eine flexible und möglichst universelle, räumliche Schnittstelle zur Sensorik spezifiziert werden, um die zukünftigen Planerkennungskomponenten flexibel in allen Ligen einsetzen zu können[1].

In einem nächsten Schritt sollen aus generischen Aktionen und hieraus resultierenden Aktionsfolgen Verhaltensmuster bzw. Taktiken erkannt werden. Aus diesen Mustern lassen sich zukünftige Aktionen vorhersagen und erklären – somit ist ein Agieren in der Umgebung unter Berücksichtigung der Aktionen anderer autonomer Agenten möglich. Innerhalb dieses Bereichs sollen die anderen physikalischen Agenten nicht isoliert betrachtet werden. Vielmehr soll davon ausgegangen werden, dass es sich hierbei um eine selbstorganisierende Gruppe mit einem gemeinsamen, dem eigenen widersprechenden Ziel handelt. Somit ist eine Aktion der Gruppe fremder Agenten auch eine Bedrohung der eigenen Pläne, Ziele und präferierten Aktionen und muss entsprechend berücksichtigt werden. Das Agieren unter der Berücksichtigung der Handlungen Anderer setzt einen hohen Grad an Adaptivität und Lernfähigkeit zur Steuerung des Agenten voraus. Somit bilden im Besonderen auch diese Forschungsbereiche einen Schwerpunkt des Projekts.

Die Technologie der Strategieerkennung ist in zwei wichtigen Forschungsgebieten von großem Interesse. Auf der einen Seite kann die Vorhersagequalität von Aktionen physikalischer Agenten gesteigert werden, was im Rahmen der Steuerung autonomer Agenten eine große Rolle spielt, auf der anderen Seite ist im Bereich des Electronic Commerce die Steigerung der Robustheit und Sicherheit von elektronischen Märkten wichtig. Darüber hinaus hilft Plan- und Intentionserkennung in beiden Szenarien beobachtete Handlungen zu erklären und ermöglicht damit eine bessere Selektion eigener Handlungen (höhere Adaptivität).

---

[1] Diese Aufgabe wurde im Rahmen der „*AG3-Architektur*“ des Schwerpunktprogramms übernommen.

3.1.2 *Weiterführung der aufgebauten realen RoboCup-Teams*

Alle im Laufe der ersten Projektphase aufgebauten Teams (Sony-Legged-League, Simulationsliga) sollen in der zweiten Phase weitergeführt werden. Die bisherigen Erfahrungen haben gezeigt, dass die in der ersten Projektphase implementierte Weltmodellierung noch nicht ausreichend robust ist und die vorhandenen Informationen nicht optimal ausnutzt. Zwar erfüllt die realisierte Selbstlokalisierung die in sie gesetzten Erwartungen, die Modellierung der Ball- und Spielerpositionen sind aber noch unzureichend. Zudem werden die 2002 in der Sony-Legged-League eingeführten Möglichkeiten zur Kommunikation zwischen den Agenten noch nicht optimal genutzt. Daher soll in der zweiten Phase eine integrierte probabilistische Weltmodellierung realisiert werden, die alle Unsicherheiten der Perzeption modelliert und die Inter-Roboter-Kommunikation nutzt. Zudem soll das Weltmodell zwei Ebenen haben, zum einen eine lokale Sicht, die keine Messungen anderer Roboter enthält und für Nahbereichsaktionen genutzt werden kann, z.B. Dribbeln, Schießen oder Greifen nach dem Ball. Zum anderen gibt es eine globale Ebene, in der die Informationen aller Roboter zusammenkommen. Diese ist eher für komplexe Aktionen, wie z.B. das Passspiel interessant und stellt auch die Basis der angestrebten Strategieerkennung in einer physikalischen Roboterliga dar.

Zudem strebt die Entwicklung im RoboCup immer höheren Anforderungen entgegen, um sich dem für 2050 gesetzten Ziel schrittweise anzunähern. Jedes Jahr werden gerade in den physikalischen Roboterligen neue Regelwerke für die Wettkämpfe entwickelt, die immer neue Herausforderungen stellen. Für notwendige Aufgaben wie Lokalisation oder Ball- und Gegnererkennung werden verschiedene Sensoren (Ultraschall, Laser, Kamera) verwendet und neue Lösungen entwickelt, die im nächsten Jahr vielleicht schon wieder umstrukturiert werden müssen. Eine Möglichkeit, dieses Problem anzugehen, besteht in der Verarbeitung von Bildinformationen. Fest angebrachte Distanzsensoren ermöglichen lediglich eine 2-dimensionale Kollisionsvermeidung und Lokalisation (solange es Banden gibt), aber z.B. keine 3-dimensionale Erfassung der Umwelt oder automatische Klassifikation von Objekten. Durch Bildverarbeitung sind theoretisch alle Probleme zugänglich; Lokalisation, Entfernungsmessung und Objektklassifikation können unter Verwendung der Daten eines Sensors behandelt werden. Zudem ist die Bildverarbeitung die einzige Komponente, die in allen physikalischen Roboterligen verwendet werden kann. Daher werden wir uns in der zweiten Phase verstärkt mit der Bildverarbeitung beschäftigen, zum einen, weil der zentrale Sensor in der Sony-Legged-League eine Kamera ist, zum anderen, weil die 3-dimensionale Erfassung der Umwelt in unserem Anwendungsszenario außerhalb des RoboCups, dem Bremer Autonomen Rollstuhl, ein tatsächliches Mehr an Information bringt.

3.1.3 *Nutzung für den Bremer Autonomen Rollstuhl*

Für den Rollstuhl ist die visuelle Erfassung der Umgebung besonders wichtig, da dieser eine natürliche Umgebung zusätzlich zur Erfassung durch einen Laserscanner auch durch Bildverarbeitungsmethoden 3-dimensional analysieren soll. Daher muss entsprechend zum Grundsatz der Strategieerkennung, die Ligen-übergreifend anwendbar sein soll, ein allgemeiner Ansatz zur Nutzung natürlicher Bildmerkmale entwickelt werden. Merkmale wie Ecken und Kanten oder Farbe sollen ausgenutzt und in Kombination mit neu einzubringenden kognitiven Bildmerkmalen wie z.B. Symmetrie erprobt werden.

Auch die zentrale Zielsetzung der Strategieerkennung autonomer Agenten soll für den Bremer Autonomen Rollstuhl nutzbar gemacht werden. Hierbei lässt sich der Ansatz der Strategieerkennung und -planung aus einer anderen Perspektive betrachten: Die Erkennung der Bewegung der den Rollstuhl umgebenden Objekte, d.h. Personen, kann sowohl die Geschwindigkeit als auch die Sicherheit des Systems erhöhen, da ermittelt werden kann, welche Passanten die Bahn des Systems aufgrund der vorhergesagten wahrscheinlichen Bewegung kreuzen oder beeinflussen könnten (Bewegungsprädiktion). Im übertragenen Sinn können die Personen im Umfeld des Rollstuhls als „gegnerisches Team“ interpretiert werden. Dieses Umfeld wechselt zwar ständig, zeigt aber ähnliche, von der Situation (leerer Fußweg, Marktplatz) abhängige Verhaltensweisen bzw. Strategien. Lassen sich hieraus Eigenschaften ableiten, die die Ziele des Systems gefährden könnten, z.B. die Kreuzung der Fahrbahn des Rollstuhls, so soll dem durch die Planung eigener Strategien entgegengewirkt werden. Das „eigene Team“ besteht im Normalfall nur aus dem Rollstuhl bzw. aus seinen Sensoren und Aktoren. Es wäre aber durchaus vorstellbar und wünschenswert, sich z.B. an einer Begleitperson zu orientieren, der gefolgt werden soll.

Auf der Basis dieser Informationen soll dann längerfristig ein Navigationsverhalten entwickelt werden, das ein „Mitschwimmen“ des Rollstuhls in einer sich bewegenden Menschenmenge ermöglicht. Dadurch kommt es zu einer natürlichen Kooperation des Fahrzeugs mit seiner Umgebung, ähnlich wie Menschen in solchen Situationen ihre Bewegungen mit denen anderer koordinieren. Für ein solches Verhalten existieren eine ganze Reihe von Ansätzen (z.B. Mataric, 1997; Hong und Loo, 1998), von denen bisher aber keines auf einem realen, kinematisch beschränkten System umgesetzt wurde.

3.1.4 Relevanz für den Elektronischen Handel

Die Bedeutung der Multiagentensysteme für die Betriebswirtschaft und den Einsatz im elektronischen Handel nimmt immer mehr zu. Hier konnten sich bisher Multiagentensysteme jedoch nicht durchsetzen, da sie u.a. im Einsatz in elektronischen Märkten durch das Einschleusen „feindseliger“ Agenten zu einem unkalkulierbaren Risiko werden können. Sollen Kooperationen durch Multiagentensysteme nicht nur in geschlossenen Märkten unterstützt werden, wo die Kooperationspartner bekannt sind und externe (außerhalb des Agentensystems zu vereinbarende) Verträge das Verhalten stark einschränken, ist es für die Agenten notwendig, fremde Strategien zu erkennen und entsprechend zu reagieren. Es ist auch denkbar, Agenten mit speziellen Funktionen, wie der Strategieerkennung, zum Schutze des Systems einzusetzen, die destruktive oder sabotierende Agenten aus dem Markt ausschließen können. Hierbei reicht das Ziel dieser Diagnostik „gegnerischen“ Verhaltens von sicherheitspolitischen Aspekten, wie der Vermeidung einer Sabotage des Marktes, bis hin zur Optimierung des eigenen Ertrags. Die Anpassung von Strategien an gegnerische Verhaltensweisen wird üblicherweise durch eine Analyse bestimmter Indikatoren vollzogen. Hierbei beziehen sich die Indikatoren auf eigene und somit „objektiv“ messbare Parameter. Neu an diesem Ansatz ist der Versuch, das gegnerische Verhalten in Form einer Strategie zu erkennen und so gezielt nicht nur auf Basis des aktuellen Zustands, sondern auch unter Hinzunahme des voraussichtlichen Verhaltens des Gegners eine Strategie bzw. Aktion auszuwählen.

3.1.5 Einordnung in das Schwerpunktprogramm

Das Arbeitsprogramm des Schwerpunktprogramms umfasst Forschungsbereiche, die „für schnelle, mobile Roboter relevant sind“ (Auszug aus dem SPP-Antrag, S. 7). Dabei ist das wesentliche Forschungsziel eine Kombination von schneller Reaktion, Autonomie und Kooperation. Autonome, mobile Roboter sollen damit in dynamischen Umgebungen und auf unvorhersehbare Ereignisse schnell und vor allem im Team reagieren können.

Unser Ansatz liefert mit der Erkennung von gegnerischen Strategien einen wichtigen Beitrag zur Entscheidungsfindung der Roboter in Echtzeit. Die Arbeiten für das Simulationsteam Virtual Werder (vgl. Zwischenbericht) haben gezeigt, dass sich die Spielergebnisse signifikant verbessern. Die zu leistenden Arbeiten in ordnen sich in die Teildisziplin Lernverfahren, genauer *Lernen komple-*

plexer Aufgaben und Handlungsverläufe in Multiagentensystemen ein (vgl. SPP-Antrag, S. 8). Dabei ist vorgesehen, nicht nur neuronale Lernverfahren zu verwenden oder zu modifizieren, sondern ebenso symbolische Lernverfahren zu betrachten.

Durch das Erkennen von Handlungen bzw. Handlungsfolgen können Entscheidungen schneller und effektiver getroffen werden. Deswegen werden Reaktionen und eigene Aktionen in Roboter-Teams in dynamischen Umgebungen wirksamer. Hiermit und mit der Weiterentwicklung des GermanTeam in der Sony-Legged-League, gemeinsam mit der HU Berlin und Universitäten außerhalb des Schwerpunktes, ordnen wir uns weiterhin in die Teildisziplin „Kooperation und Multiagentensysteme" ein.

3.1.6 Rahmenbedingungen zur Plan- und Intentionserkennung in hochdynamischen Multiagentensystemen (- zur Strukturierung der Arbeitspakete)

Bei der Konzeptionalisierung und Strukturierung der Arbeitspakete stehen vier Anforderungen im Vordergrund:

- **Anwendbarkeit:** Die in diesem Projekt entwickelten Ergebnisse sollen für verschiedene Mannschaften und verschiedene Ligen anwendbar sein (und damit z.B. auch für hetrogene Mannschaften). Zu diesem Zweck werden frühzeitig entsprechende Schnittstellen definiert und in den verschiedenen Ligen und am Rollstuhl validiert. Um die Anwendbarkeit der zu entwickelnden Konzepte zu belegen, werden zudem alle Konzepte praktisch für den *Bremer Autonomen Rollstuhl* implementiert und getestet.

- **Granularität:** Planererkennung wird auf verschiedenen Granularitäts-Ebenen und unter unterschiedlichen konzeptionellen Aspekten untersucht. Dies spiegelt sich in den jeweiligen Arbeitspaketen wider. So wird Planerkennung sowohl für einzelne Spieler, für Mannschaftsteile (wie Angriff, Abwehr, etc.), sowie für ganze Teams untersucht. Dabei müssen die Abhängigkeiten zwischen der Strategie einer Mannschaft, der Taktik eines Mannschaftsteils und der Aktion eines einzelnen Spielers berücksichtigt werden, um zu verlässlichen Ergebnissen zu kommen. Zudem werden auf jeder Granularitätsebene verschiedene Ausrichtungen betrachtet. In manchen Kontexten ist eine exakte Vorhersage eines bestimmten Verhaltensmusters erforderlich, in anderen eher die Erklärung (und in der Folge die indirekte Vorhersage von zukünftigem Verhalten).

- **Hinreichende Effizienz:** Ein charakteristisches Merkmal des RoboCup-Szenarios ist, dass es in einer physikalischen Welt auf physikalischen Robotern (echten und simulierten) mit Aktorik, Sensorik und mit beschränkten Ressourcen stattfindet. Planerkennung sollte eine effizientere Verwendung dieser Ressourcen unterstützen (z. B. durch Fokussierung der Ressourcen) und sich in bestehende Multiagentensystemen einbetten lassen.

- **Räumliche Modellierung:** Planerkennung im RoboCup heißt aus einer über der Zeit geordneten Sequenz von Aktionen einen Plan oder eine Intention zu erkennen. Dabei gilt, dass Aktionen und ihre Auswirkungen über Veränderungen von räumlichen Positionen und Relationen beschrieben werden. Daher ist es entscheidend, eine geeignete räumliche, möglichst qualitative Repräsentation von Raum zu finden, die es erlaubt, Planerkennung auf verschiedenen Ebenen von zu konkreten Aktionen bis hin zu abstrakteren Strategien anzuwenden.

Das Problem der Plan- und Intentionserkennung in einem kooperativen, konkurrierenden und hochgradig dynamischen Multi-Agenten-Szenario (MAS) kann nicht ausschließlich durch die Anwendung einer einzelnen Methode gelöst werden, sondern muss im Kontext der Gesamtarchitektur eines Agenten betrachtet werden. Die bisher untersuchten Einzelagenten-Planerkennungverfahren legen in fast allen Fällen eine Reihe für das RoboCup-Szenario unrealistische Annahmen zugrunde:

- **Verfügbarkeit von Informationen:** Einem Agenten stehen alle relevanten Informationen bezüglich der Welt einschließlich anderer Agenten zur Verfügung. Diese starke Annahme ist bereits in weitaus einfacheren Szenarien nicht zu halten. Sie gilt in besonderem Maße nicht für Agenten in einem RoboCup-Szenario, da der sensorische Input und seine Verarbeitung durch technische wie zeitliche Constraints fehlerbehaftet und durch prinzipielle Einschränkungen bezüglich Blickrichtung und Kommunikation unvollständig ist, so dass allgemein von keinem allozentrischen Weltbild ausgegangen werden kann.

- **Nicht-Determinismus:** Die geplante/intendierte Handlung eines Agenten steht nicht immer im Einklang mit der resultierenden Handlung in der physikalischen (oder simulierten) Welt. Dies ist nicht nur bedingt durch eine nicht immer perfekte Aktorik, sondern vor allem durch die hohe Dynamik und der nur eingeschränkten Vorhersagbarkeit in einem Agentenszenario. D.h. dass eine beobachtete Handlung nicht notwendigerweise im Einklang mit der Intention des agierenden Agenten stehen muss und dass ein plan-erkennender Agent die Nichtübereinstimmung zwischen Intention und Handlung berücksichtigen muss.

- **Flexibilität:** Ebenso wie für die beobachtete Handlung gilt auch für das Ziel, dass Intention und Ergebnis eines handelnden Agenten nicht immer übereinstimmen. Eine Handlung kann verschiedenen Zielen dienen (in machen Fällen sogar gleichzeitig) und das erschlossene Ziel stimmt nicht notwendigerweise mit dem intendierten überein. Dennoch sollte ein plan-erkennender Agent auch über unerwartete Ergebnisse Schlüsse über die zugrunde liegenden Ziele eines Agenten ziehen können.

- **Dynamische Umgebung:** Eine weitere gravierende Schwierigkeit der Planerkennung in MAS steht in Zusammenhang mit der Dynamik der Welt. Während der Planung und Durchführung einer Handlung kann ein Agent sowohl die Art der Ausführung (wie), als auch die zugrundeliegende Intention (was) ändern, so das nicht immer ein linearer Zusammenhang zwischen der beobachteten Handlungen erkennbar sein muss.

Ansätze, die diese für MAS-Szenarien unrealistischen Annahmen umgehen, werden z.B. im Rahmen von Plan-Managementsystemen/MAS-Planning untersucht. Es ergibt sich daher als Anforderung an die Arbeitspakete, Planerkennung auch für diese aktuellen Architekturen und Konzepte anwendbar zu machen. D.h. es wird in jeder Phase berücksichtigt, wie sich Planerkennung in die Gesamtarchitektur eines Agenten integrieren lässt.

3.2 Arbeitsprogramm

Das Arbeitsprogramm gliedert sich in drei übergeordnete Arbeitspakete die den Forschungsschwerpunkten entsprechen. Die Reihenfolge der Arbeitspakete spiegelt daher keine Bearbeitungsreihenfolge wider, sondern ist eine konzeptionelle Strukturierung der Aufgaben. Eine detaillierte Aufschlüsselung der Reihenfolge und der Meilensteine findet sich in Abschnitt 3.3.

3.2.1 Architektur

Die unter *Architektur* zusammengefassten Arbeitspakete schaffen die Grundlagen, um Plan- und Intentionserkennung im RoboCup praktisch anwenden zu können. Dabei geht es um die Konzeptionalisierung und Umsetzung einer Agentenarchitektur, die auf der einen Seite den aktuellen Stand der Forschung berücksichtigt und auf der anderen Seite die Grundlagen schafft, Planerkennung flexibel in verschiedenen Ligen und heterogenen Mannschaften anwenden zu können.

AP 1-1: Repräsentation von strategischem Wissen. Ziel der Planerkennung ist es, eine über die Zeit geordnete Folge von Aktionen einer abstrakteren, möglichst intentionalen Beschreibung zuzuordnen, die es erlaubt, alte Handlungen zu erklären und neue vorherzusagen. Primärer Gegenstand der Planerkennung ist daher ein zugrunde liegendes, möglichst deklaratives Verhaltensmodell, dem eine beobachtete Handlung zugeordnet werden kann. In diesem Arbeitspaket soll untersucht werden, welche bestehenden Ansätze zur Verhaltensmodellierung sich zur Planerkennung in einem RoboCup-Szenario eignen. Eine wesentliche Anforderung ist, dass das Verhaltensmodell geeignet sein muss, sowohl Planungs- wie Planerkennungsoperationen zu unterstützen. Darüber hinaus ist es erforderlich, Pläne/Verhalten auf verschiedenen Abstraktionsstufen beschreiben zu können, um einen inkrementellen Planungs- und Planerkennungsprozess zu unterstützen, der es erlaubt, sowohl Team-, als auch Gruppen- und Individualtaktiken zu erkennen. Auf Basis der Modellierungssprache wird in (AP 2-2) strategisches und taktisches Fußballwissen zur Plan- und Intentionserkennung modelliert. *(1 PM)*

AP 1-2: Entwicklung und Realisierung einer Plan-Management-Architektur zur Planerkennung für MAS. Aktuelle Ansätze zum Planen in Multiagenten-Szenarien haben zu einer Erweiterung der klassischen „*Single Agent Planning Architecture*“ geführt. Aufgrund der Dynamik eines Multi-Agenten-Szenarios, der vielfältigen Unsicherheiten beginnend mit der Sensorik über die Verarbeitung bis hin zur Aktorik, ist es notwendig, nicht nur kontextsensitiv differenzierte Pläne zu erzeugen, sondern eine Architektur zu verwenden, die es erlaubt, schnell und flexibel zu entscheiden, ob ein Plan noch ausgeführt werden kann, oder ob die Art der Ausführung eines Planes verworfen werden muss (was/wie). Außerdem helfen sie konstruktiv verschiedene Pläne miteinander zu verknüpfen und können dadurch den Mehrwert einer einzelnen Aktion maximieren. Da diese Architekturen den aktuellen Stand der Forschung im Bereich der Planung widerspiegeln und für den Planerkennungsprozess die gleichen schwierigen Rahmenbedingungen gelten, soll der Planerkennungsprozess mindestens konsistent mit den Anforderungen und Rahmenbedingungen einer solchen Architektur sein. Ziel in diesem Arbeitspaket ist es, eine Plan-Management-Architektur so zu erweitern, dass eine Plan- und Intentionserkennung auf Basis räumlicher Informationen sinnvoll integriert werden kann. Der Fokus liegt daher nicht auf der Entwicklung einer vollständig neuen Architektur, sondern auf die Adaptation und Erweiterung bestehender Ansätze. Für dieses Arbeitspaket ist eine Zusammenarbeit mit der RWTH Aachen und der HU Berlin geplant. *(5 PM)*

AP 1-3: Deliberative Komponente. Um Strategien zur Anwendung zu bringen, muss in einem ersten Schritt ein Konzept entwickelt und umgesetzt werden, dass (zunächst ohne Planerkennung) entscheidet, welche taktischen und strategischen Varianten in einer Situation konkret angewendet werden sollen/können. Diese erzeugten Verhaltensmuster dienen in den folgenden Arbeitspaketen als Testumgebung zur Validierung von Planerkennungskonzepten. In einem zweiten Schritt wird untersucht, inwieweit Planerkennung wiederum verwendet werden kann, um die *deliberative Komponente* zu verbessern. Es soll anhand spieltheoretischer Konzepte (formal) nachgewiesen werden, dass eine (funktionierende) Planerkennung entscheidenden Einfluss auf die Selektion von Zielen eines Agenten hat (im Rahmen dieser Projektphase kann dieser Aspekt sicher nicht abschließend betrachtet werden). *(1 PM)*

AP 1-4: Entwurf und Umsetzung einer räumlichen Inferenzmaschine. Nur ein Teil der zur Planerkennung (wie auch zur Planung) relevanten räumlichen Relationen kann durch Perzeption/Sensorik direkt ermittelt werden. Ein wesentlicher Teil der Informationen muss indirekt über räumliche Inferenzen bestimmt werden. An einem Beispiel verdeutlicht: wenn ein Agent *a* wissen will, ob sich ein Gegenspieler hinter ihm befindet, muss er seine aktuelle Aktion nicht abbrechen und sich umsehen, sondern er kann über die Information, dass sich links, (weit) hinter ihm ein Mitspieler *b* befindet und dem Wissen, dass sich weit rechts vor *b* ein Gegenspieler befindet, inferieren, dass sich dieser nah hinter ihm (Agent *a*) befinden muss. Um komplexere Pläne erkennen zu können, ohne Handlungen und Perzeption auf die Aufgabe der Planerkennung fokussieren zu müssen, ist es nützlich, möglichst viele Informationen flexibel indirekt ableiten zu können. In diesem Arbeitspaket werden basierend auf Erfahrungen in anderen Projekten bestehende Ansätze validiert, soweit erforderlich angepasst und umgesetzt. Im Vordergrund stehen hierbei zunächst nur metrische (Distanz) und ordinale (Ausrichtung) Inferenzen. In diesem Arbeitspaket ist eine enge Zusammenarbeit mit der RWTH Aachen und der HU Berlin vereinbart, um möglichst große Synergieeffekte zu erzielen. *(2 PM)*

AP 1-5: Integration Planerkennung aus Phase 1. Im Rahmen der ersten Projektphase sind u.a. Lernverfahren für die Onlineanalyse entwickelt worden. Dabei hat sich gezeigt, dass die Ergebnisse von hoher praktischer Bedeutung sind. Die Online-Analyse basiert auf einem zeitreihenbasierten Entscheidungsbaumverfahren, dass zunächst in der Simulationsliga getestet worden ist. Das Verfahren generiert propositionale Regeln, mit Hilfe derer spezielle Aspekte des Gegners analysiert werden können (z.B. Verhalten des Torwartes, Passverhalten). Die erzielten Ergebnisse müssen jetzt in einem weiteren Schritt so umgesetzt werden, dass sie sich für Anweisungen an eigene Spieler bzw. an das eigene Team verwenden lassen. Genau dieses soll in dem Arbeitpaket durchgeführt werden. *(1 PM)*

3.2.2 Weltmodellierung

Der Schwerpunkt der Weltmodellierung behandelt drei Aufgabengebiete. Für die Planung und Planerkennung ist eine angemessene *Repräsentation strategischen Wissens* und Verhaltens notwendig, die wiederum auf der *Repräsentation räumlichen Wissens* aufsetzt. Zur Umsetzung dieser Repräsentation in realen Umfeldern wie den physikalischen Roboterligen und dem Rollstuhl ist dazu die *sensorische Erfassung* der Umwelt und deren Umsetzung in ein Weltmodell nötig.

AP 2-1: Qualitativ-räumliches Wissen. Räumliches Wissen spielt in vieler Hinsicht eine zentrale Rolle:

- für die Wiederverwendbarkeit der entwickelten Konzepte in verschiedenen Ligen und verwandten Anwendungen wie den *Bremer Autonomen Rollstuhl*,
- für die Wartbarkeit und Erweiterbarkeit des entworfenen und modellierten Wissens und
- für Adäquatheit des modellierten Wissens.

Die in den verschiedenen Ligen ermittelten bzw. zur Verfügung stehenden räumlichen Informationen sind, bedingt durch unterschiedliche Rahmenbedingungen der verschiedenen Ligen, von sehr heterogenem Charakter. Während in der Small-Size-League über eine Deckenkamera ein sehr gutes allozentrisches, räumliches (Welt-) Modell generiert werden kann, sind die Voraussetzungen in der Sony-Legged-League und in der F2000-League weitaus schwieriger. Bei ersterer kann bedingt durch eine vorgegebene Sensorik und Restriktionen bezüglich der Ressourcen nur ein bedingt korrektes Modell der Welt erzeugt werden, welches zudem egozentrischen Charakter hat, während in der F2000-League versucht wird ein mindestens pseudo-allozentrisches und damit kommunizierbares Weltmodell zu erzeugen.

Damit Verhaltensmodelle zwischen verschiedenen Ligen ausgetauscht werden können, muss es nicht nur möglich sein, egozentrische und allozentrische räumliche Weltmodelle sowie deren Abbildungen untereinander zu beschreiben, sondern es muss auch eine abstraktere, qualitative Beschreibungsebene gefunden werden, die von den spezifischen absoluten Rahmenbedingungen abstrahiert. Diese Sichtweise deckt sich mit der räumlichen Beschreibung von (Trainingsmethoden) Taktiken und Strategien in der sportlichen Fachliteratur von verschiedensten (mindestens Ball-) Sportarten. Beispiele für solche Verhaltensmuster lassen sich auf der Ebene des Individual-, des Gruppen-, als auch des Teamverhaltens aufzeigen.

In diesem Arbeitspaket soll daher eine räumliche Beschreibungssprache mit Fokus auf die Verhaltensmodellierung entwickelt werden, die es erlaubt Verhalten nicht nur *quantitativ* und damit ligaspezifisch, sondern *qualitativ* zu beschreiben. Darüber hinaus sollte die Sprache auch die abstraktere Beschreibung von translations- und rotationsinvariantem Verhalten erlauben. Die mit dieser Sprache erzeugte Verhaltensbeschreibung ist nicht nur Grundlage für das Verhalten und die im Fokus dieses Antrages stehende Planerkennung, sondern nicht nur den flexiblen Einsatz von gleichen Verhaltensmustern in verschiedenen Ligen ermöglichen (abhängig von den zur Verfügung stehenden sensorischen Informationen und den Rahmenbedingungen der jeweiligen Aktorik), sondern auch die Planerkennungskomponente flexibel in verschiedenen (allen) Ligen einsetzbar machen. *(3 PM)*

AP 2-2: Modellierung von strategischem und taktischen Fußballwissen. Zur Validierung der Planerkennungskonzepte im RoboCup soll die im Arbeitspaket „*Repräsentation von strategischem Wissen (AP 1-1)*“ spezifizierte Verhaltensmodellierungssprache dazu genutzt werden, strategisches und taktisches Wissen konkret umzusetzen und als Verhaltensmodell den Bremer RoboCup-Mannschaften (zunächst Simulationsliga und dann Sony-Legged-League) zur Verfügung gestellt werden. Die Verhaltensmodelle werden auf Basis realer Verhaltens-Muster, wie sie sich in den Fußballlehrbüchern finden, nachmodelliert. Um Redundanzen zu vermeiden, ist bei diesem Arbeitspaket eine enge Zusammenarbeit mit der RWTH Aachen und der HU Berlin geplant. Von Interesse ist hier auch der Wissensakquisitionsaspekt: wie erwirbt man Wissen für Verhaltensmodelle physikalischer Roboter und wie setzt man diese Modelle universell verwendbar um. Das Ergebnis dieses Arbeitspaketes ist Grundlage vieler Arbeitspakete anderer Projekte. Nach einer ausführlichen Validierung wird die Modellierung jeder interessierten RoboCup-Mannschaft zur Verfügung gestellt. *(3 PM)*

AP 2-3: Komplexere räumliche Beschreibungen. Die in Arbeitspaket AP 2-1 zu spezifizierende räumliche Repräsentationssprache stellt die grundlegenden atomaren, räumlichen Beschreibungskomponenten zur Verfügung. Darauf basierend muss untersucht werden, welche komplexeren räumlichen Beschreibungen kompositionell, zum einen kontextunabhängig und zum anderen bezogen auf die RoboCup-Domäne, modelliert werden müssen.

In erster Instanz werden die entwickelten räumlichen Prädikate bei der Modellierung von taktischem und strategischem Fußball-Verhalten bezüglich ihrer Adäquatheit und Wartbarkeit getestet. Die Tragfähigkeit insbesondere der kontextunabhängigen Beschreibungen kann dabei sowohl in verschiedenen RoboCup-Ligen der Bremer Mannschaften sowie am Bremer Rollstuhl validiert werden.

Zudem muss für das RoboCup-Szenario untersucht werden, inwieweit synonyme räumliche Beschreibungen für eine adäquate Modellierung erforderlich sind. (Ein Beispiel: nah an der Bande (RoboCup) – nah an der Wand (Rollstuhl) vs. nah an Begrenzung (als universelle Beschreibung für beide Domänen)).

Erfahrungen in diesem Arbeitspaket können rückkoppelnd eine Erweiterung der in Arbeitspaket 2-1 spezifizierten räumlichen Sprache erfordern. *(2 PM)*

AP 2-4: Begrenzte Wahrnehmung eines Roboters. Als Basis zur Bildung eines globalen Weltmodells soll analysiert werden, wie die begrenzte Wahrnehmung eines Roboters modelliert werden kann. Dies ist insbesondere für die Sony-Legged-League wichtig, da die Roboter ihre Umwelt durch eine Kamera wahrnehmen, deren Bild nur einen kleinen Ausschnitt der Welt beschreibt. Bisher wird nur beachtet, was in diesem Ausschnitt gesehen wird, Informationen über Nichtgesehenes können aber genau so aufschlussreich sein. Durch die bessere Modellierung von Objekten und Raumrelationen unter Einbeziehung von explizitem Wissen über Spielerzahl, Verdeckungen oder Objekthöhe soll die Qualität der lokalen Weltmodelle verbessert werden. Bei diesem Arbeitspaket wird mit der Gruppe von der HU Berlin kooperiert, die sich ebenfalls mit der Wahrnehmung der Roboter beschäftigt. *(2 PM)*

AP 2-5: Globales Weltmodell und Kommunikation. Ein großes Problem, welches ebenfalls in der Sony-Legged-League am stärksten auftritt, ist die Kombination lokaler Modelle in ein Gesamtweltmodell. Für die Repräsentation räumlichen Wissens (siehe AP 2-1 und AP 2-3) ist ein möglichst aktuelles und umfassendes Weltmodell nötig, die einzelnen Roboter können aber aufgrund ihrer begrenzten Wahrnehmung nur einen Teil dieses Modells zu einem Zeitpunkt erfassen. Es stellt sich also die Frage, wie und wie gut sich die lokalen Modelle durch Kommunikation zu einem Gesamtmodell kombinieren lassen. Da die lokalen Modelle grundsätzlich probabilistischer Natur sind, ist zu erwarten, dass sich Unsicherheiten durch Kombination potenzieren, es muss also eine Möglichkeit gefunden werden, Beobachtungen einzelner Roboter möglichst gut zu bewerten. Dabei könnte es sinnvoll sein, eine Hierarchie der Modelle zu erstellen: Jeder Roboter besitzt zuerst sein lokales Modell. Ein kombiniertes Modell, in dem die eigenen Beobachtungen des einzelnen Roboters stärker einfließen, könnte in manchen Situationen nützlicher sein als ein globales Modell, in das die Beobachtungen aller Roboter gleichmäßig eingehen. Dieses Arbeitspaket stellt auch eine Zwischenschicht für die Arbeiten an der HU Berlin dar. Zum einen setzt die Modellierung auch auf die dort durchgeführten Arbeiten zur aktiven Wahrnehmung auf, zum anderen ist das Weltmodell sowohl die Basis für die in Bremen angestrebten Arbeiten zur Planerkennung, als auch für die Berliner Vorhaben zur Erweiterung ihrer Verhaltensarchitektur. *(4 PM)*

AP 2-6: Kognitive Bildverarbeitung. Für die Übertragung der Strategieerkennung in verschiedene Anwendungsfelder gilt es, in diesen Bereichen Objekte zu erkennen und in das Weltmodell zu übertragen. Die Anforderungen an die Robotik sind dabei jedoch sehr unterschiedlich. Methoden, die im RoboCup angewendet werden, können sehr schlecht auf den Rollstuhl übertragen werden, da die Umgebung im RoboCup sehr strukturiert und vereinfacht ist und ligenspezifisch verschiedene Sensortypen benutzt werden können. Die Bildverarbeitung das mächtigste Werkzeug der Wahrnehmung, es wird allerdings aufgrund seiner Komplexität im RoboCup nur in sehr einfacher Art und Weise genutzt. Es wäre wünschenswert, natürliche Bildmerkmale zu nutzen, die in allen Anwendungsfeldern verwendet werden können, um nicht an vorgegebene Strukturen bzw. künstliche Merkmale gebunden zu sein. Zwar würde das Problem im RoboCup erst mit strukturellen Regeländerungen der nächsten Jahre auftreten, im Rollstuhlumfeld besteht es jedoch schon jetzt, da die Bildverarbeitung hier nötig ist, um Objekte auch 3-dimensional zu erfassen (siehe AP 2-7). Daher ist die Bearbeitung dieses Problems notwendig für die Realisierung der Strategieerkennung auf dem Rollstuhl, zudem aber hilfreich und zukunftsweisend für den RoboCup. *(4 PM)*

AP 2-7: 3D-Informationsgewinnung für den Rollstuhl. Für die Erfassung von Personen, die sich in der Umgebung des Rollstuhls bewegen, muss der Rollstuhl jedoch mit Sensoren ausgestattet sein, die es ihm ermöglichen, sein Umfeld vollständig wahrzunehmen. Dies bedeutet zum einen, in möglichst vielen Richtungen Sensordaten zu sammeln, zum anderen, aber auch in unterschiedlichen Höhen, um z.B. auch Tische zu erfassen. Hierfür reichen die bisherigen Mittel – ein nach hinten gerichteter Laserscanner und ein Ring aus Ultraschallsensoren – nicht aus. Durch die Anwendung zweier omnidirektionaler Sichtsensoren ergeben sich diese Möglichkeiten. Setzt man eine symmetrische Anordnung der Systeme an den Seiten des Rollstuhls voraus, ist es in begrenztem Maße möglich, nach vorne und nach hinten Entfernungen auch in unterschiedlicher Höhe abzuschätzen. Jeder einzelne Spiegel kann auf seiner Seite den Nahbereich überwachen und wenn nötig durch zeitverzögerte Aufnahmen in diese Richtungen Distanzen abschätzen. Es stellt sich auch die Frage, ob sich Vorteile unterschiedlicher Sensorarten sinnvoll und gewinnbringend kombinieren lassen. Denkbar wäre hier zum Beispiel eine Aufmerksamkeitssteuerung der omnidirektionalen Bildverarbeitung, begründet auf den Daten des Laserscanners. *(2PM)*

3.2.3 Plan- und Intentionserkennung

In der ersten Phase soll/wurde die in der Literatur am besten untersuchte Variante der Planerkennung untersucht: die Erkennung und Vorhersage von Handlungen eines einzelnen kooperierenden oder konkurrierenden Agenten. Dabei werden die Arbeitspakete bezüglich verschiedener Problemstellungen differenziert. In vielen Anwendungskontexten ist ein Agent nicht an der Erkennung einer komplexen, eher abstrakten Intention eines anderen Agenten interessiert, sondern an der konkreten, möglichst präzisen Vorhersage des nächsten Handlungsschritts (z.B. beim Dribbling). In anderen Szenarien, z.B. bei der Plan- und Intentionserkennung durch den Trainer, ist man wiederum nicht an der kurzfristigen Vorhersage des nächsten Handlungsschritts, sondern an der längerfristigen taktischen und strategischen Ausrichtung interessiert. Diese Aspekte spiegeln sich in der Strukturierung der Arbeitspakete wider. In der dritten Phase wird, basierend auf den Erfahrungen und Ergebnissen der zweiten Phase, der komplexere Fall der Gruppen- und Team-Intentions- und Planerkennung untersucht. Hierbei müssen komplexe gruppendynamische Zusammenhängen berücksichtigt werden.

AP 3-1: Prädiktive Plan- und Intentionserkennung (eines einzelnen Agenten). Eine zentrale Anforderung an eine Plan- und Intentionserkennung ist die Unterstützung bei der unmittelbaren Vorhersage von zukünftigen Aktionen von kooperieren und konkurrierenden Agenten. (unter der Annahme, dass bei kooperativen Agenten eine differenzierte Kommunikation z.B. aus technischen Gründen (wie z.B. in der Simulationsliga) nur eingeschränkt möglich ist, oder aus zeitlichen Gründen nicht sinnvoll ist.

Eine zentrale Rahmenbedingung an einer prädiktiven Planerkennung ist, dass sie hinreichend schnell bezüglich des zeitlichen Horizonts der Prognose eine Vorhersage generiert. D.h. dass die Vorhersage des Verhaltens eines Agenten *a* bezüglich des Zeitpunkts t_k mindestens zum Zeitpunkt t_{k-1} zur Verfügung stehen muss, um in einem Online-Szenario anwendbar zu sein.
Von besonderem Interesse sind zum einen klassische, stochastische Ansätze zur Prädiktion wie statische und dynamische Bay'sche Netze, sowie aktuelle Erweiterungen wie *(4 PM)*

AP 3-2: Erklärungsbasierte Plan- und Intentionserkennung (eines einzelnen Agenten). Viele in der klassischen Planerkennungsliteratur untersuchte Ansätze fokussieren weniger auf eine zeitlich (kurzfristig) angelegte Vorhersage eines konkreten Verhaltensmusters, als vielmehr auf die Erklärung/Klassifikation von beobachteten Verhaltensmustern. Ein auf Erklärung fokussierender Ansatz ist prinzipiell universeller verwendbar z.B. zur Offline-Taktik- und Strategieanalyse. In diesen Szenarien steht nicht die möglichst präzise Vorhersage von Handlungen im Vordergrund, sondern abstraktere längerfristigere Intentionen[1]. Die am besten untersuchte Methode zur erklärungsbasierten Plan- und Intentionserkennung ist Abduktion. Basierend auf den aktuellen Resultaten soll untersucht werden, inwieweit Varianten dieser Verfahren auf die RoboCup- bzw. Rollstuhl-Domäne, sowohl in Online- wie Offline-Szenarien, anwendbar sind. *(4 PM)*

AP 3-3: Handlungsbasierte Plan- und Intentionserkennung (eines einzelnen Agenten). Bei der Plan- und Intentionserkennung im RoboCup gilt es zwei wichtige Szenarien zu unterscheiden: (1) Online- und (2) Offline-Analyse (bzw. zeitunkritische Analyse). Eine besondere Herausforderung stellt dabei das erste und im RoboCup sehr entscheidende Online-Szenario dar. Um Planerkennung auch in einem Online-RoboCup-Szenario anwenden zu können, mit mittel- und kurzfristigen möglichen Konsequenzen für das eigene Spiel, müssen hohe Anforderungen sowohl an die Qualität als auch an die Effizienz der Planerkennungs-Methodik gestellt werden. In diesem Arbeitspaket wird untersucht, wie sich Planerkennung mit Planung/Handlungskoordination Synergie bringend verknüpfen lässt, so dass sich für die Handlungssteuerung unmittelbar Vorteile aus einer Planerkennung ableiten. Dabei soll untersucht werden, ob und wie die Qualität und ggf. auch die Effizienz der Handlungsplanung verbessert werden kann, indem, heuristisch, unwahrscheinliche Verhaltensmuster des Gegners ausgeschlossen werden können. *(4 PM)*

AP 3-4: Strategie- und Taktikerkennung konkurrierender und kooperierender Gruppen von Agenten. Die Fußballdomäne zeichnet sich, wie auch z.B. elektronische Märkte, dadurch aus, dass Agenten nicht als vollständig unabhängige Entitäten agieren, sondern in Gruppen und Teams kooperierender Agenten eingebunden sind. Dabei stehen die Ziele eines einzelnen Agenten, wenn Kooperation gelingen soll, in direkter Beziehung zu den Zielen und Intentionen der Gruppe bzw. des Teams. In diesem Arbeitspaket soll untersucht werden, in welcher Beziehung die Intentionen und Absichten eines einzelnen Agenten zu denen seiner (taktischen) Gruppe stehen, d.h. inwieweit sie kohärieren bzw. divergieren. Dieses Wissen ist erforderlich, um zu bestimmen, wann einer Gruppe eine Intention zugeschrieben werden kann (Beispiel: (a) Bei wie vielen Agenten muss man erkennen, dass sie Raumdeckung spielen, um der gesamten Abwehr diese Intention zuschreiben zu können? (b) Wenn man die Intention/Taktik einer Gruppe erkannt hat, inwieweit kann man damit das Verhalten eines einzelnen Agenten vorhersagen?).

Darüber hinaus muss untersucht werden, wie sich Gruppen konstituieren bzw. definieren. Als Gruppe können nicht nur statische Konstrukte wie Abwehr, Angriff oder Mittelfeld verstanden werden, sondern auch dynamische, z.B. eine Gruppierung mit einem kurzfristigeren gemeinsamen Ziel (z.B. einen Konter zu spielen). Diese Ergebnisse der oben dargestellten Problemstellungen stellen die Grundlage dar, um Pläne und Intentionen (d.h. Taktiken und Strategien) von Gruppen von Agenten zu erkennen.

Das Erkennen von Gruppen-Intentionen kann rückkoppelnd wiederum verwendet werden, um die Qualität der Intentionserkennung einzelner Agenten zu verbessern, indem die möglichen Verhaltensmuster durch Kenntnis der Gruppen-Intention eingeschränkt werden können.

Dieses Arbeitspaket wird im Rahmen der zweiten Projektphase kaum vollständig in der gesamten Bandbreite bearbeitet werden können. Einige sich als besonders kompliziert erweisende Teilproblem werden in die dritte Projektphase umfassender behandelt werden müssen. *(6 PM)*

---

[1] Durch Koppelung mit Lernverfahren können jedoch die erzeugten Erklärungen wiederum als Grundlage zur Vorhersage von Gegner-Aktionen in zukünftigen Spielen dienen (Strategieanalyse).

3.3 Arbeits- und Zeitplan

Der zeitliche Aufbau des Arbeitsprogramms des Forschungsvorhabens ist in Abbildung 3 dargestellt. Die Laufzeit der einzelnen Arbeitspakete (Bezeichnung in der ersten Tabellenspalte) ist durch Balken über der Zeitachse dargestellt. Bei der Bezeichnung der Zeitachse wird von dem geplanten Beginn des Folge-Forschungsvorhabens im 4. Quartal 2003 (01.11.2003) ausgegangen.

Es wurden drei Meilensteine definiert: Zum ersten Meilenstein zu den German Open im April 2004 steht die Plan-Management-Architektur auf den Sony Robotern mit einer qualitativen, räumlichen Repräsentation zur Verfügung. Der zweite Meilenstein im September umfasst die prädiktive Planerkennung auf Basis differenzierter taktischer und strategischer Repräsentationen. Der dritte Meilenstein zu den German Open 2005 umfasst das kooperative, probabalistische Weltmodell und die vollständige prädiktive und erklärende Planerkennung eines Einzelagenten.

3.4 Weitere Perspektiven

Während in der ersten Teilphase die grundlegenden technischen und konzeptionellen Rahmenbedingungen geschaffen wurden, um Plan- und Intentionserkennung im RoboCup anwenden zu können, werden in der zweiten Phase die methodischen Grundlagen gelegt. Dabei steht die Entwicklung der Methodik neben der Schaffung und Aufrechterhaltung der technischen Rahmenbedingungen im Vordergrund (z.B. Anpassung an sich verändernde Spielregeln).

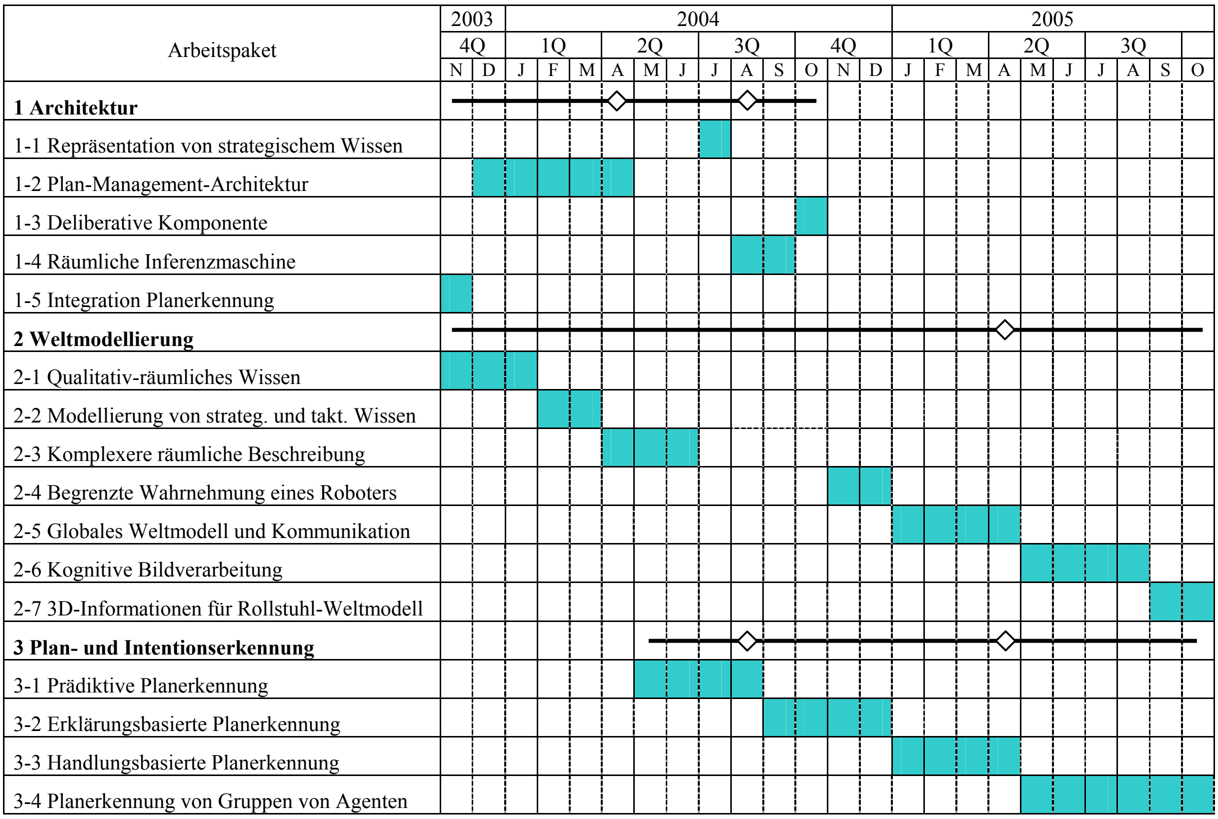


Abbildung 3: Der Arbeitsplan. Die Rauten repräsentieren die Meilensteine „PMS“,“Räumliches Wissen / prädiktive Planerkennung“ und „Weltmodell / Einzelagenten-Planerkennung“, die jeweils den Abschluss einer Projektphase bezeichnen. Als Termine wurden in Abhängigkeit zu den Arbeitspaketen jeweils die Termine der „German Open 2004“, „RoboCup 2004“ und „German Open 2005“ angesetzt.

In der dritten Phase des Schwerpunktprogramms sollen die Methoden zur Plan- und Intentionserkennung für Gruppen und Mannschaften abstrahiert und differenziert werden. D.h. zum einen soll der Ansatz soweit abstrahiert werden, dass er flexibel auch für nicht statisch definierte Gruppen angewendet werden kann. In vielen Multiagentensystemen sind die Kooperationen / Koalitionen sehr dynamisch und abhängig von den Intentionen und Überzeugungen der Agenten. In diesem Kontext heißt Plan- und Intentionserkennung auch Identifikation einer Gruppe mit gemeinsamen Interessen. Diese Gruppen können sich abhängig von den aktuellen Zielen und Überzeugungen flexibel umstrukturieren. Indem Planerkennung hilft Gruppen mit Intentionen zu identifizieren, kann sie auch helfen Gruppenverschiebungen schneller zu erkennen und die Art der Verschiebung sich veränderten Intentionen zuschreiben.

Auf Basis dieser Erweiterungen soll der Ansatz auch an elektronische Märkte angepasst und validiert werden. Ein interessantes Anwendungsfeld kann die Identifikation destruktiv handelnder Agenten in einem Markszenario sein. Eine frühzeitige Erkennung kann zu einer wichtigen, rechtzeitigen Regulierung und damit Optimierung des Marktes führen.

Auf dieser Basis soll dann eine Diagnose komplexer gegnerischer Verhalten und Strategien erfolgen, die sowohl das kurzfristige Handeln, also die reaktiven Anteile, als auch das langfristige Planen von kooperierenden Gegnern, also die deliberativen Aspekte, berücksichtigt. Stärken und Schwächen sollen analysiert werden; Ziel ist hier die Verhaltenssteuerung und Strategieauswahl für kooperierende Teams physikalischer Agenten auf Basis gegnerischer Strategien.

3.5 Untersuchungen am Menschen

Untersuchungen am Menschen werden nicht durchgeführt.

3.6 Untersuchungen am Tierversuche

Tierversuche werden nicht durchgeführt.

3.7 Gentechnologische Experimente

Gentechnologische Experimente werden nicht durchgeführt.

4 Beantragte Mittel

4.1 Personalbedarf

4.1.1 Zwei Wissenschaftliche Mitarbeiter

Wie bereits in Abschnitt 3.2 dargestellt, soll ein Wissenschaftlicher Mitarbeiter im Bereich „Intelligente Systeme“ mit besonderer Qualifikation auf dem Gebiet der Multiagentensysteme die Arbeitspakete im Bereich der Planerkennung bearbeiten. Ein zweiter Wissenschaftlicher Mitarbeiter im Bereich „Kognitive Robotik“ soll die Integrations-, Validierungs-Arbeitspakete (s.a. Anpassungen für Rollstuhl, Regeländerungen) angehen, wozu eine besondere Qualifikationen in Robotik und Bildverarbeitung benötigt wird. Die Architektur-Arbeitspakete werden von beiden Wissenschaftlern gemeinsam bearbeitet. In Anbetracht der Komplexität der Aufgabenstellungen und der aktuellen Arbeitsmarktlage für Informatiker werden für beide jeweils ganze Stellen beantragt.

zwei Stellen BAT IIa für 2 Jahre

4.1.2 Studentische Hilfskräfte

Die studentischen Hilfskräfte sollen die Einrichtung und Wartung der Experimentalroboter übernehmen sowie Programmierarbeiten im Bereich Künstliche Intelligenz durchführen. Jeder beantragte Wissenschaftliche Mitarbeiter soll durch zwei studentische Hilfskräfte unterstützt werden.

vier stud. Hilfskräfte (19 Std./Woche) für 2 Jahre

4.2 Wissenschaftliche Geräte

Für die Aibo-Roboter werden vier neue Zentraleinheiten (SuperCores, 400 MHz Takt statt bisher 200 MHz) benötigt. Im Handel befinden sich nur noch Roboter mit den neuen Zentraleinheiten, so dass neue Teilnehmer der Sony-Legged-League mit diesen antreten werden. Um konkurrenzfähig zu bleiben und verlässliche Daten bei der Validierung der entwickelten Konzepte zu bekommen, ist es dringend erforderlich die Zentraleinheiten der Sony-Roboter auf den aktuellen Stand der Technik zu bringen. Der vom Sony-Support angebotene Preis liegt bei 49,500 Yen (s. beiliegende E-Mail). Zuzüglich Steuern und Lieferung aus Japan ergibt das etwa 500,00 € (abhängig vom Wechselkurs).

Neue Zentraleinheiten für die Sony-Roboter	4 × 500,00 €
Summe	*2000,00 €*

4.3 Verbrauchsmaterial/Betriebskosten

Für den Bremer Autonomen Rollstuhl werden im Antragzeitraum neue Batterien benötigt (haben eine Lebensdauer von etwa 2 Jahren). Darüber hinaus müssen, um die Sony-Roboter instand zu halten, acht neue Batterien angeschafft werden (Lebensdauer ebenfalls 2 Jahre).

Batterien für den Bremer Autonomen Rollstuhl (s. Angebot)	1 × 360 €
Batterien für die Sony Aibo-Roboter (s. Preisliste)	8 × 135 €
Summe	*1440 €*

4.4 Reisen

Wie im Rahmenantrag für das Schwerpunktprogramm bereits motiviert wurde, besteht ein erhöhter Reisebedarf für die Mitarbeiter; dies schließt die Antragsteller mit ein. Daher wird die dort vorgesehene Summe von 7.500 DM (3750 €) pro Mitarbeiter und Jahr für Reisen zu Konferenzen zur Präsentation der Forschungsergebnisse und zur Teilnahme am Benchmark-Workshop und sonstigen Schwerpunkt-internen Treffen beantragt.

Darüber hinaus sind Koordinations-Treffen mit den Kooperationspartnern von der RWTH Aachen und der HU Berlin notwendig. Für die Arbeitspakete 1-1, 2-1 und 2-3 sind enge Kooperationen vorgesehen. Es wird angenommen, dass wir uns mit unseren Projektpartnern mindestens dreimal im Antragszeitraum treffen, d.h. mindestens zweimal an der HU Berlin oder der RWTH Aachen bei Kosten von ca. 250 € pro Treffen und Mitarbeiter (Hin- und Rückfahrt mit der Bahn + eine Übernachtung).

Das GermanTeam in der Sony-Legged-League trifft sich alle 2 Monate, also 12× in der Antragsphase. Dabei finden im Durchschnitt 3/4 aller Treffen (also 3× Dortmund, 3× Darmstadt, 3× Berlin) außerhalb von Bremen statt, an diesen Treffen nehmen zwei Mitarbeiter und zwei wissenschaftliche Hilfskräfte teil (9 × 4 × (Hin- und Rückfahrt mit der Bahn + eine Übernachtung)).

Reisemittel für 2 Wissenschaftliche Mitarbeiter × 2 Jahre (Konferenzen)	2 × 2 × 3750 €
Reisemittel für 2 Wissenschaftliche Mitarbeiter (AP 1-1, 2-1, 2-3)	2 × 2 × 250 €
Reisemittel für 2 Wissenschaftliche Mitarbeiter und zwei wissenschaftliche Hilfskräfte (als Mitglieder des GermanTeam)	9 × 4 × 250 €
Summe	*25.000 €*

4.5 Sonstige Kosten

Keine.

5 Voraussetzungen für die Durchführung des Vorhabens

5.1 Zusammensetzung der Arbeitsgruppe

Die Arbeitsgruppe setzt sich aus zwei kooperierenden Teilgruppen zusammen. Folgende Mitarbeiter werden, aus Mitteln der Universität bzw. des Landes Bremen finanziert, teilweise an dem Projekt mitarbeiten:

Dr. Ubbo Visser,

Dr. Thomas Röfer.

Im Übrigen tragen fünf Studierende aus dem studentischen Projekt „RoboCup", das im September 2002 endete, in freien Arbeitsgruppen zu dem Projekt bei. Seit Oktober 2002 existiert ein studentisches Nachfolgeprojekt „RoboCup II". Hier werden ca. 30 Studierende einen zusätzlichen Beitrag zu dem Vorhaben leisten können.

5.2 Zusammenarbeit mit anderen Wissenschaftlern

5.2.1 Innerhalb des Schwerpunktprogramms

Seit März 2001 gibt es eine enge Zusammenarbeit mit der Arbeitsgruppe um Prof. Hans-Dieter Burkhard on der HU Berlin. Die Universitäten Bremen und die HU Berlin sind gemeinsam an dem GermanTeam in der Sony Legged League beteiligt und haben entscheidenden Anteil an den Arbeiten in diesem Team. Die Kooperation soll fortgesetzt werden.

Auch im Bereich der Simulationsliga wird mit der Arbeitsgruppe im Prof. Burkhard kooperiert. So ist vorgesehen, bei der Entwicklung einer qualitativen Verhaltenskomponente zusammen zu arbeiten.

Mit der RWTH Aachen, Arbeitsgruppe Prof. Lakemeyer, wird in dem Vorhaben ebenfalls kooperiert. Hier steht insbesondere die qualitative räumliche Modellierung im Vordergrund, die ebenfalls der Verhaltenskomponente zuzurechnen ist.

Die Arbeitsgruppe um Prof. Lakemeyer wird zudem die von der Universität Bremen in der ersten Phase entwickelten einfachen Planerkennungsmethoden einsetzen.

Im Gegenzug werden wir die Deliberationskomponenten der RWTH Aachen in unsere Modelle integrieren.

5.2.2 Außerhalb des Schwerpunktprogramms

Dr. Röfer ist Teamleader des GermanTeams in der Sony Legged League.

Es besteht eine enge Zusammenarbeit mit den anderen Universitäten, die an dem German Team beteiligt sind: TU Darmstadt (Prof. von Stryk), Universität Dortmund (Prof. Banzhaf).

Dr. Thomas Röfer ist Mitantragsteller des DFG-Sonderforschungsbereichs/Transregio „Spatial Cognition – Reasoning, Action, Interaction", der sich in der Beantragung befindet. Dadurch bestehen hier zahlreiche Kooperationen zu den anderen Antragstellern, z.B. Christian Freksa (Universität Bremen), Bernhard Nebel und Wolfram Burgard (beide Universität Freiburg). Aus dem Schwerpunktprogramm „Raumkognition" bestehen zusätzlich noch gute Kontakte zur Arbeitsgruppe von Prof. Wilfried Brauer (Technische Universität München).

Die Weiterentwicklung des Bremer Autonomen Rollstuhls geschieht in Zusammenarbeit mit der Firma Meyra, der Marktführerin für Rollstühle in Deutschland.

5.3 Arbeiten im Ausland und Kooperation mit ausländischen Partnern

International bestehen im Bereich Robotik Kontakte zu den Arbeitsgruppen von Prof. Anibal T. de Almeida (Universität Coimbra, Portugal), Prof. Spyros Tzafestas (Universität Athen, Griechenland) und Dr. Ulrich Nehmzow (Universität Manchester, England), letztere bis Ende 1999 gefördert durch die DAAD im Rahmen des ARC-Programms.

Dr. Röfer ist Mitglied des Technical Committee der Sony Legged-League in der RoboCup Federation.

Im Bereich der Intelligenten Systeme besteht eine langjährige Kooperation mit dem Machine Learning Research Centre MLRC (das ehemalige Neurocomputing Research Centre) der Queensland University of Technology. Prof. Dr. Joachim Diederich vom MLRC und Dr. Ubbo Visser arbeiten z. Zt. in zwei Forschungsprojekten zusammen. Weiterhin besteht ein enger Kontakt zum International Knowledge Discovery Institute (IKDI), ein weltweit agierendes Forschungs- und Entwicklungsinstitut, bei dem Dr. Visser Senior Partner ist. Der Bereich Intelligente Systeme kooperiert zudem mit Prof. Dr. Ryszard Michalski vom Machine Learning and Inference Laboratory der George Mason University in Fairfax, VA. Eine weitere Kooperation besteht mit Prof. Mathew Palakal von der Indiana University-Purdue University Indianapolis, U.S.A.

Dr. Visser ist Initiator eines international angelegten Workshops, der auf der IJCAI 2003 zum ersten Mal stattfinden soll. Der Titel des Workshops ist "Issues in Designing Physical Agents for Dynamic Real-Time Environments: World modeling, planning, learning, and communicating." Der Fokus liegt auf der Entwicklung von Technologien, die für Roboter in *dynamischen Echtzeitsystemen* geeignet sind. Diese Initiative ist auf der letzten Sitzung des SPP in Aachen diskutiert worden und fand die breite Unterstützung. Das Proposal wird zurzeit begutachtet und mit einer wahrscheinlich positiven Antwort ist Anfang November zu rechnen. Das SPP hat damit eine Chance, auch außerhalb Deutschlands Impulse zu geben. Weiterhin sollte erwähnt werden, dass dieser Workshop kein RoboCup – Workshop sein wird, sondern dass andere namhafte Wissenschaftler, die nicht der internationalen RoboCup-Gemeinde angehören, ebenfalls beteiligt sind (z.B. Patrick Doherty, Schweden).

5.4 Apparative Ausstattung

Der Arbeitsgruppe „Kognitive Robotik" steht ein Rollstuhl mit Sensorik und On-Board-Rechner, sowie vier Sony-Aibo-Roboter zur Verfügung. Der Rollstuhl ist mit 27 Ultraschallsensoren, einem Laser-Entfernungsmesser der Firma Sick und einem Omnivision-System ausgestattet.

Die vorhandene Infrastruktur der Bereiche Intelligente Systeme und Kognitive Robotik des TZI steht zur Nutzung des Projekts zur Verfügung. Hierzu zählen neben zahlreichen Unix-Workstations und PCs auch die Nutzung eines Multi-Prozessor PC-Servers und eines Multi-Prozessor Unix-Servers. Im Weiteren wird soweit möglich vorhandene Software genutzt.

5.5 Laufende Mittel für Sachausgaben

Laufende Mittel für Sachausgaben stehen an der Universität Bremen in ausreichendem Maße zur Verfügung.

5.6 Sonstige Voraussetzungen

Keine.

6 Wirtschaftliche Verwertung

Der Bremer Autonomen Rollstuhl „Rolland“ ist dem Gebiet der *Service Robotik* zuzuordnen. Dieser Bereich wird nach jüngsten Studien (United Nations Economic Commission for Europe, 1999) in Zukunft zum Wachstumsbereich und bietet große wirtschaftliche Chancen. Eine Bewegungserkennung für den Rollstuhl wird das Fahrverhalten des Systems und damit den Komfort für die Benutzer erheblich verbessern und dadurch die Akzeptanz bei der Zielgruppe weiter steigern, so dass Firmen wie z.B. *Meyra* derartige Systeme marktreif zu Ende entwickeln können. Fest geplant ist eine solche Vermarktung jedoch bisher noch nicht.

Dem elektronischen Handel (E-Commerce) wird gerade im Businees-2-Business Bereich ein hohes Potential zugesprochen. Für den Einsatz intelligenter Verfahren müssen geeignete Schutzmechanismen für solche elektronischen Marktplätze entwickelt werden. Bisher ist noch keine konkrete Verwertung des Projektes in diesem Rahmen angedacht, aber es ist möglich und denkbar in Kooperation mit einem Klein- oder Mittelständischen Unternehmen die hier entwickelten Methoden zu effektiven Schutzmechanismen weiterzuentwickeln.

7 Erklärungen

7.1

Es besteht keinerlei thematische Verbindung zwischen dem Vorhaben und Arbeiten in einem am Ort befindlichen Sonderforschungsbereich.

7.2

Ein Antrag auf Finanzierung dieses Vorhabens wurde bei keiner anderen Stelle eingereicht. Wenn einer von uns einen solchen Antrag stellt, wird er die Deutsche Forschungsgemeinschaft unverzüglich benachrichtigen.

7.3

Der Vertrauensdozent der DFG an der Universität Bremen ist über die Antragstellung informiert.

8 Unterschriften

Bremen, den 1. November 2002

(Ubbo Visser) (Thomas Röfer)

9 Verzeichnis der Anlagen

Arbeitsbericht

Lebensläufe und Veröffentlichungslisten von

- Dr. Ubbo Visser
- Dr. Thomas Röfer

Angebote für

- Sony Aibo Zentraleinheit (SuperCore)
- Batterien für Rollstuhl
- Batterien für Sony Aibo

Tabellarischer Lebenslauf Dr. Ubbo Visser

1970 – 1974	Besuch der Grundschule in Norden/Ostfr.
1974 – 1983	Besuch des Ulrichs-Gymnasiums in Norden/Ostfr., abgeschlossen mit dem Abitur
1983 – 1988	Studium der Geographie, Fachrichtung Landschaftsökologie an der Universität Münster, Nebenfächer Geologie und Mineralogie, abgeschlossen mit dem Diplom
1986 – 1988	Studentische Hilfskraft in der Arbeitsgruppe Umweltinformationssysteme von Herrn Prof. Dr. Streit und am Historischen Seminar bei Herrn Prof. Dr. H.-J. Teuteberg
Oktober 1988	Diplomarbeit zum Thema "Die Verdunstungsmessung an Binnenseen als Problem der angewandten Hydrologie", betreut von Prof. Dr. Julius Werner
1988 – 1992	Studium der Informatik an der Universität Münster, abgeschlossen mit der Promotionsprüfung (als Promotionsnebenfach)
1988 – 1989	Wissenschaftlicher Mitarbeiter in der Arbeitsgruppe Windklimatologie von Herrn Prof. Dr. J. Werner
1989 – 1991	Wissenschaftlicher Mitarbeiter in der Arbeitsgruppe Agrarinformatik von Herrn Prof. Dr. U. Streit
1991 – 1995	Wissenschaftlicher Mitarbeiter am Institut für Agrarinformatik der Landwirtschaftskammer Westfalen/Lippe
Dezember 1995	Promotion in Geoinformatik zum Dr. rer. nat. mit dem Dissertationsthema Ëntwicklung und Anwendung neuronaler Netze im Rahmen des regelbasierten Pflanzenschutz - Beratungssystems PRO_PLANT", Erstgutachter Prof. Dr. U. Streit
1996 – 06/1996	BMBF-Stipendium am Neurocomputing Research Centre (heute Machine Learning Research Centre), School of Computer Science, Queensland University of Technology, Brisbane, Australien, bei Prof. Dr. J. Diederich
07/1996 – 07/1997	Post-Doctoral Fellow am Neurocomputing Research Centre, School of Computer Science, Queensland University of Technology, Brisbane, Australien bei Prof. Dr. J. Diederich
07/1997 – 10/1998	Geschäftsführer des Bereiches 'Intelligente Systeme' am Technologie-Zentrum Informatik der Universität Bremen, Prof. Dr. Herzog
seit 11/1998	Wissenschaftlicher Assistent im Bereich 'Intelligente Systeme' am Technologie-Zentrum Informatik der Universität Bremen bei Prof. Dr. Herzog und Prof. Dr. Schlieder

Publikationen Ubbo Visser:

Dissertation:

[1] U. Visser, "Entwicklung und Anwendung neuronaler Netze im Rahmen des regelbasierten Pflanzenschutz - Beratungssystems PRO_PLANT," in *Fachbereich Geowissenschaften*: Westfälische Wilhelms-Universität Münster, 1995, pp. 156.

Buchkapitel und Zeitschriften:

[2] [Visser, U., H. Stuckenschmidt, C. Schlieder, H. Wache, and I. Timm, *Terminology Integration for the Management of distributed Information Resources.* Künstliche Intelligenz (KI), Special Issue Knowledge Management, 2002. **16**(1): p. 31-34.

[3] Visser, U., H. Stuckenschmidt, G. Schuster, and T. Vögele, *Ontologies for Geographic Information Processing.* Computers & Geosciences, 2002. **28**(1): p. 103-118.

[4] Visser, U., H. Stuckenschmidt, H. Wache, and T. Vögele, *Using Environmental Information Efficiently: Sharing Data and Knowledge from Heterogeneous Sources*, in *Environmental Information Systems in Industry and Public Administration*, C. Rautenstrauch and S. Patig, Editors. 2001, IDEA Group: Hershey, USA & London, UK. p. 41-73.

[5] Visser, U., U. Voges, K. Epke, R. Pfeiffer, and U. Streit, *Umweltschonender Pflanzenschutz mit Hilfe des Expertensystems PRO_PLANT.* Künstliche Intelligenz (KI), 1993. **93**(3): p. 34--43.

[6] Visser, U. and C. Schlieder, *Modelling with Ontologies*, in *The Ontology and Modeling of Real Estate Transactions in European Juristictions*, H. Stuckenschmidt, E. Stubkjaer, and C. Schlieder, Editors. 2002, Ashgate, in print.

Weitere begutachtete Artikel:

[7] Wache, H., U. Visser, and T. Scholz. *Ontology Construction - An Iterative and Dynamic Task*. in *Florida Artificial Intelligence Research Society Conference (FLAIRS)*. 2002. Pensacola, FL, USA: AAAI press.

[8] Visser, U. and G. Schuster. *Finding and Integration of Information - A Practical Solution for the Semantic Web -*. in *ECAI 02, Workshop on Ontologies and Semantic Interoperability*. 2002. Lyon, France: ECCAI.

[9] Visser, U., T. Vögele, and C. Schlieder. *Spatio-Terminological Information Retrieval using the BUSTER System*. in *EnviroInfo*. 2002. Vienna: Berger Druck, Horn, Austria.

[10] Visser, U., H. Stuckenschmidt, and C. Schlieder. *Interoperability in GIS - Enabling Technologies*. in *5th AGILE Conference on Geographic Information Science*. 2002. Palma de Mallorca, Spain: Universitat de les Illes Balears.

[11] Visser, U. and H.-G. Weland. *Using online learning to analyze the opponents behavior*. in *Proceedings of the RoboCup-2002: Robot Soccer World Cup VI*. 2002. Fukuoka, Japan.

[12] Miene, A. and U. Visser, *Interpretation of spatio-temporal relations in real-time and dynamic environments*, in *RoboCup 2001: Robot Soccer World Cup V*, A. Birk, S. Coradeschi, and S. Tadokoro, Editors. 2002, Springer: Seattle, WA. p. 441-446.

[13] Meyer, J., R. Adolph, D. Stephan, A. Daniel, M. Seekamp, V. Weinert, and U. Visser. *Decision-making and Tactical Behavior with Potential Fields*. in *Proceedings of the RoboCup-2002: Robot Soccer World Cup VI*. 2002. Fukuoka, Japan.

[14] Drücker, C., S. Hübner, U. Visser, and H.-G. Weland, *"As time goes by" - Using time series based decision tree induction to analyze the behaviour of opponent players*, in *RoboCup 2001: Robot Soccer World Cup V*, A. Birk, S. Coradeschi, and S. Tadokoro, Editors. 2002, Springer Verlag: Seattle, WA. p. 325-330.

[15] Wache, H., T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. *Ontology-based Integration of Information - A Survey of Existing Approaches*. in *IJCAI-01 Workshop: Ontologies and Information Sharing*. 2001. Seattle, WA.

[16] Visser, U., C. Drücker, S. Hübner, E. Schmidt, and H.-G. Weland. *Recognizing Formations in Opponent Teams*. in *RoboCup 2000, Robot Soccer World Cup IV*. 2001. Melbourne, Australia: Springer-Verlag.

[17] Stuckenschmidt, H., T. Vögele, U. Visser, and R. Meyer. *Intelligent Brokering of Environmental Information with the BUSTER system*. in *In: Information Age Economy: Proceedings of the 5th International Conference 'Wirtschaftsinformatik'*. 2001. Ulm, Germany: Physica-Verlag.

[18] Stuckenschmidt, H., U. Visser, C. Schlieder, T. Vögele, and H. Neumann. *Spatial Reasoning for Information Brokering*. in *Florida Artificial Intelligence Research Society Conference (FLAIRS)*. 2001. Key West, FL: AAAI Press.

[19] Schlieder, C., T. Vögele, and U. Visser. *Qualitative Spatial Representation for Information Retrieval by Gazetteers*. in *Conference of Spatial Information Theory COSIT*. 2001. Morrow Bay, CA: Springer.

[20] Neumann, H., G. Schuster, H. Stuckenschmidt, U. Visser, and T. Vögele. *Intelligent Brokering of Environmental Information with the BUSTER System*. in *International Symposium Informatics for Environmental Protection*. 2001. Zürich, Switzerland: Metropolis.

[21] Vögele, T., H. Stuckenschmidt, and U. Visser. *Towards Intelligent Brokering of Geoinformation*. in *Urban and Rural Data Management (UDMS)*. 2000. Delft: Electronic.

[22] Vögele, T., H. Stuckenschmidt, and U. Visser. *BUISY - Using Brokered Data-Objects for Environmental Information Systems*. in *Hypermedia im Umweltschutz*. 2000. Ulm, Germany: Metropolis-Verlag.

[23] Stuckenschmidt, H. and U. Visser. *Semantic Translation Based on Approximate Re-Classification*. in *Workshop on Semantic Approximation, Granularity and Vagueness, Workshop of the Seventh International Conference on Principles of Knowledge Representation and Reasoning*. 2000. Breckenridge.

[24] Drücker, C., S. Hübner, E. Schmidt, U. Visser, and H.-G. Weland. *Virtual Werder: Using the Online-Coach to Change Team Formations*. in *4th International Workshop on RoboCup*. 2000. Melbourne, Australia: Carnegie Mellum University Press.

[25] Vögele, T., K.C. Ranze, and U. Visser. *Innovative Methods for the Collection and Automatic Evaluation of Monitoring Data Acquisition and Analysis*. in *PETRA99, Euro-Med Worskhop for IT Applied to Natural Resources Management*. 1999. Petra, Jordan.

[26] Visser, U. and H. Stuckenschmidt. *Intelligent Location-Dependent Acquisition and Retrieval of Environmental Information*. in *21st Urban Data Management Symposium*. 1999. Vienna, Italy: The Urban Data Management Society.

[27] Timm, I., P. Knirsch, M. Petsch, U. Visser, K. Fischer, O. Herzog, S. Kirn, and S. Zelewski, *Agententechnologie - Multiagentensysteme in der Informationslogistik und wirschaftswissenschaftliche Perspektiven der Agenten-Konzeptionalisierung*, in *KI, Künstliche Intelligenz 99*. 1999, TZI, University of Bremen: Bonn, Germany. p. 114.

[28] Stuckenschmidt, H., U. Visser, G. Schuster, and T. Vögele. *Ontologies for Geographic Information Integration*. in *Workshop "Intelligent Methods in Environmental Protection: Special Aspects of Processing in Space and Time", 13. International Symposium of Computer Science for Environmental Protection (CSEP '99)*. 1999: University of Bremen.

[29] Brinkkötter-Runde, K. and U. Visser. *Wearable computing, wireless communication and knowledge discovery for mobile data and analysis*. in *International Workshop on Mobile Mapping Technology*. 1999. Bangkok, Thailand.

[30] Visser, U., R. Nayak, and M.T. Wong, *Rule extraction from trained neural networks and connectionist knowledge representation for the determination of pesticide mixtures*, in *Connectionist Systems for Knowledge Representation and Deduction*, T. Downes, Editor. 1998, University of Queensland Press: Brisbane. p. 138--142.

[31] Conrad, R. and U. Visser. *Das Konzept und die Technologien des Internet-basierten Bremer Umweltinformationssystems*. in *Umweltinformatik*. 1998. Marburg: Metropolis-Verlag.

[32] Visser, U. and K. Hell, *PRO_PLANT - Information System - Canola Pests*, in *ICCIMA '97*, B. Verma and X. Yao, Editors. 1997, Watson Ferguson \& Company (Griffith University): Gold Coast, Australia. p. 59--64.

[33] Visser, U., A. Tickle, R. Hayward, and R. Andrews, *Rule-Extraction from trained neural networks: Different techniques for the determination of herbicides for the plant protection advisory system PRO_PLANT*, in *Rules and Networks*, J. Diederich and R. Andrews, Editors. 1996, QUT, Society for the Study of Artificial Intelligence and Simulation of Behavior: Brisbane. p. 133--139.

[34] Visser, U., E. Pop, R. Hayward, and J. Diederich. *Regelextraktion aus neuronalen Netzen am Beispiel der Ermittlung von Herbiziden im Pflanzenschutz - Beratungsystem PRO_PLANT*. in *Space and Time in Environmental Information System*. 1995. Berlin: Metropolis Verlag, Marburg, GI--Informatik im Umweltschutz.

[35] Visser, U., U. Voges, and U. Streit, *Integration of AI-, Database- and Telecommunication-Techniques for the Plant Protection Expert System PRO_PLANT*, in *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, D.F. Anger, R.V. Rodriguez, and M. Ali, Editors. 1994, Gordon and Breach Science Publishers: San Antonio. p. 367--374.

[36] Voges, U. and U. Visser, *PRO_PLANT - Expertensystem für den umweltschonenden Einsatz von Pflanzenschutzmitteln in der Landwirtschaft.* Berichte zur Leistungsschau der 2. deutschen Tagung für Expertensysteme, 1993: p. 13--14.

[37] Voges, U., U. Visser, A. Johnen, and K. Hell. *PRO_PLANT - Pflanzenschutz mit Hilfe eines Expertensystems*. in *17. Fachtagung für künstliche Intelligenz*. 1993. Stuttgart.

[38] Visser, U. and U. Streit, *Möglichkeiten zur Einbindung neuronaler Netze am Beispiel des Beratungssystems PRO_PLANT*, in *Referate der 13. GIL-Jahrestagung*, R. Ackmann, B. Petersen, and H. Geidel, Editors. 1991, Ulmer Verlag: Giessen. p. 32--36.

Vollständige Publikationsliste:

[1] Wache, H., U. Visser, and T. Scholz. *Ontology Construction - An Iterative and Dynamic Task*. in *Florida Artificial Intelligence Research Society Conference (FLAIRS)*. 2002. Pensacola, FL, USA: AAAI press.

[2] Visser, U. and G. Schuster. *Finding and Integration of Information - A Practical Solution for the Semantic Web* -. in *ECAI 02, Workshop on Ontologies and Semantic Interoperability*. 2002. Lyon, France: ECCAI.

[3] Visser, U., T. Vögele, and C. Schlieder. *Spatio-Terminological Information Retrieval using the BUSTER System*. in *EnviroInfo*. 2002. Vienna: Berger Druck, Horn, Austria.

[4] Visser, U. and C. Schlieder, *Modelling with Ontologies*, in *The Ontology and Modeling of Real Estate Transactions in European Juristictions*, H. Stuckenschmidt, E. Stubkjaer, and C. Schlieder, Editors. 2002, Ashgate, in print.

[5] Visser, U., H. Stuckenschmidt, and C. Schlieder. *Interoperability in GIS - Enabling Technologies*. in *5th AGILE Conference on Geographic Information Science*. 2002. Palma de Mallorca, Spain: Universitat de les Illes Balears.

[6] Visser, U. and H.-G. Weland. *Using online learning to analyze the opponents behavior*. in *Proceedings of the RoboCup-2002: Robot Soccer World Cup VI*. 2002. Fukuoka, Japan.

[7] Visser, U., H. Stuckenschmidt, C. Schlieder, H. Wache, and I. Timm, *Terminology Integration for the Management of distributed Information Resources*. Künstliche Intelligenz (KI), Special Issue Knowledge Management, 2002. **16**(1): p. 31-34.

[8] Visser, U., H. Stuckenschmidt, G. Schuster, and T. Vögele, *Ontologies for Geographic Information Processing*. Computers & Geosciences, 2002. **28**(1): p. 103-118.

[9] Miene, A. and U. Visser, *Interpretation of spatio-temporal relations in real-time and dynamic environments*, in *RoboCup 2001: Robot Soccer World Cup V*, A. Birk, S. Coradeschi, and S. Tadokoro, Editors. 2002, Springer: Seattle, WA. p. 441-446.

[10] Meyer, J., R. Adolph, D. Stephan, A. Daniel, M. Seekamp, U. Visser, and V. Weinert, *BUGS - Team Description*, in *RoboCup 2002: Robot Soccer World Cup VI*. 2002, Springer Verlag: Fukuoka/Busan.

[11] Meyer, J., R. Adolph, D. Stephan, A. Daniel, M. Seekamp, V. Weinert, and U. Visser. *Decision-making and Tactical Behavior with Potential Fields*. in *Proceedings of the RoboCup-2002: Robot Soccer World Cup VI*. 2002. Fukuoka, Japan.

[12] Drücker, C., S. Hübner, U. Visser, and H.-G. Weland, *"As time goes by" - Using time series based decision tree induction to analyze the behaviour of opponent players*, in *RoboCup 2001: Robot Soccer World Cup V*, A. Birk, S. Coradeschi, and S. Tadokoro, Editors. 2002, Springer Verlag: Seattle, WA. p. 325-330.

[13] Wache, H., T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. *Ontology-based Integration of Information - A Survey of Existing Approaches*. in *IJCAI-01 Workshop: Ontologies and Information Sharing*. 2001. Seattle, WA.

[14] Visser, U., H. Stuckenschmidt, H. Wache, and T. Vögele, *Using Environmental Information Efficiently: Sharing Data and Knowledge from Heterogeneous Sources*, in *Environmental*

Information Systems in Industry and Public Administration, C. Rautenstrauch and S. Patig, Editors. 2001, IDEA Group: Hershey, USA & London, UK. p. 41-73.

[15] Visser, U., C. Drücker, S. Hübner, E. Schmidt, and H.-G. Weland. *Recognizing Formations in Opponent Teams*. in *RoboCup 2000, Robot Soccer World Cup IV*. 2001. Melbourne, Australia: Springer-Verlag.

[16] Stuckenschmidt, H., T. Vögele, U. Visser, and R. Meyer. *Intelligent Brokering of Environmental Information with the BUSTER system*. in *In: Information Age Economy: Proceedings of the 5th International Conference 'Wirtschaftsinformatik'*. 2001. Ulm, Germany: Physica-Verlag.

[17] Stuckenschmidt, H., U. Visser, C. Schlieder, T. Vögele, and H. Neumann. *Spatial Reasoning for Information Brokering*. in *Florida Artificial Intelligence Research Society Conference (FLAIRS)*. 2001. Key West, FL: AAAI Press.

[18] Schlieder, C., T. Vögele, and U. Visser. *Qualitative Spatial Representation for Information Retrieval by Gazetteers*. in *Conference of Spatial Information Theory COSIT*. 2001. Morrow Bay, CA: Springer.

[19] Neumann, H., G. Schuster, H. Stuckenschmidt, U. Visser, and T. Vögele. *Intelligent Brokering of Environmental Information with the BUSTER System*. in *International Symposium Informatics for Environmental Protection*. 2001. Zürich, Switzerland: Metropolis.

[20] Vögele, T., H. Stuckenschmidt, and U. Visser. *Towards Intelligent Brokering of Geoinformation*. in *Urban and Rural Data Management (UDMS)*. 2000. Delft: Electronic.

[21] Vögele, T., H. Stuckenschmidt, and U. Visser. *BUISY - Using Brokered Data-Objects for Environmental Information Systems*. in *Hypermedia im Umweltschutz*. 2000. Ulm, Germany: Metropolis-Verlag.

[22] Visser, U. and H. Pundt, *Information sharing: Methods and Applications*. 2000, TZI, Center for Computing Technologies: Bremen. p. 46.

[23] Stuckenschmidt, H., H. Wache, T. Vögele, and U. Visser. *Enabling Technologies for Interoperability*. in *Workshop: Information Sharing: Methods and Applications at the 14th International Symposium of Computer Science for Environmental Protection*. 2000. Bonn: TZI.

[24] Stuckenschmidt, H. and U. Visser. *Semantic Translation Based on Approximate Re-Classification*. in *Workshop on Semantic Approximation, Granularity and Vagueness, Workshop of the Seventh International Conference on Principles of Knowledge Representation and Reasoning*. 2000. Breckenridge.

[25] Drücker, C., S. Hübner, E. Schmidt, U. Visser, and H.-G. Weland. *Virtual Werder: Using the Online-Coach to Change Team Formations*. in *4th International Workshop on RoboCup*. 2000. Melbourne, Australia: Carnegie Mellum University Press.

[26] Vögele, T., K.C. Ranze, and U. Visser. *Innovative Methods for the Collection and Automatic Evaluation of Monitoring Data Acquisition and Analysis*. in *PETRA99, Euro-Med Worskhop for IT Applied to Natural Resources Management*. 1999. Petra, Jordan.

[27] Visser, U. and H. Pundt, *Workshop: Intelligent Methods in Environmental Protection: Special Aspects of Processing in Space and Time*, in *13. International Symposium of Computer Science for Environmental Protection*. 1999, University of Bremen: Magdeburg, Germany. p. 126.

[28] Visser, U. and H. Stuckenschmidt. *Intelligent Location-Dependent Acquisition and Retrieval of Environmental Information*. in *21st Urban Data Management Symposium*. 1999. Vienna, Italy: The Urban Data Management Society.

[29] Timm, I., P. Knirsch, M. Petsch, U. Visser, K. Fischer, O. Herzog, S. Kirn, and S. Zelewski, *Agententechnologie - Multiagentensysteme in der Informationslogistik und wirschaftswissenschaftliche Perspektiven der Agenten-Konzeptionalisierung*, in *KI, Künstliche Intelligenz 99*. 1999, TZI, University of Bremen: Bonn, Germany. p. 114.

[30] Stuckenschmidt, H., U. Visser, G. Schuster, and T. Vögele. *Ontologies for Geographic Information Integration*. in *Workshop "Intelligent Methods in Environmental Protection: Special Aspects of Processing in Space and Time", 13. International Symposium of Computer Science for Environmental Protection (CSEP '99)*. 1999: University of Bremen.

[31] Brinkkötter-Runde, K. and U. Visser. *Wearable computing, wireless communication and knowledge discovery for mobile data and analysis*. in *International Workshop on Mobile Mapping Technology*. 1999. Bangkok, Thailand.

[32] Visser, U., R. Nayak, and M.T. Wong, *Rule extraction from trained neural networks and connectionist knowledge representation for the determination of pesticide mixtures*, in *Connectionist Systems for Knowledge Representation and Deduction*, T. Downes, Editor. 1998, University of Queensland Press: Brisbane. p. 138--142.

[33] Visser, U. *Information Warehousing and Mining*. in *The First International Conference on Wearable Computing, ICWC-98*. 1998. Fairfax VA.: http://wearcam.org/icwc.htm.

[34] Conrad, R. and U. Visser. *Das Konzept und die Technologien des Internet-basierten Bremer Umweltinformationssystems*. in *Umweltinformatik*. 1998. Marburg: Metropolis-Verlag.

[35] Visser, U. and K. Hell, *PRO_PLANT - Information System - Canola Pests*, in *ICCIMA '97*, B. Verma and X. Yao, Editors. 1997, Watson Ferguson \& Company (Griffith University): Gold Coast, Australia. p. 59--64.

[36] Visser, U., A. Tickle, R. Hayward, and R. Andrews, *Rule-Extraction from trained neural networks: Different techniques for the determination of herbicides for the plant protection advisory system PRO_PLANT*, in *Rules and Networks*, J. Diederich and R. Andrews, Editors. 1996, QUT, Society for the Study of Artificial Intelligence and Simulation of Behavior: Brisbane. p. 133--139.

[37] Voges, U. and U. Visser, *PRO_PLANT II - Umfassende Pflanzenschutzberatung und -information*, in *Expertensysteme 95. Leistungsschau. 3. Deutsche Expertensystemtagung (XPS-95), Universitat Kaiserslautern, 1.-3. März 1995*, B. Bachmann and F. Maurer, Editors. 1995: Universität Kaiserslautern, FB Informatik, AG Expertensysteme. 67653 Kaiserslautern. p. 61-62.

[38] Visser, U., *Neuronale Netze in der Herbizidberatung*. 1995, Universität Münster: Institut für Agrarinformatik.

[39] Visser, U., *Entwicklung und Anwendung neuronaler Netze im Rahmen des regelbasierten Pflanzenschutz - Beratungssystems PRO_PLANT*, in *Fachbereich Geowissenschaften*. 1995, Westfälische Wilhelms-Universität Münster. p. 156.

[40] Visser, U., E. Pop, R. Hayward, and J. Diederich. *Regelextraktion aus neuronalen Netzen am Beispiel der Ermittlung von Herbiziden im Pflanzenschutz - Beratungsystem PRO_PLANT*. in *Space and Time in Environmental Information System*. 1995. Berlin: Metropolis Verlag, Marburg, GI--Informatik im Umweltschutz.

[41] Visser, U., U. Voges, and U. Streit, *Integration of AI-, Database- and Telecommunication-Techniques for the Plant Protection Expert System PRO_PLANT*, in *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, D.F. Anger, R.V. Rodriguez, and M. Ali, Editors. 1994, Gordon and Breach Science Publishers: San Antonio. p. 367--374.

[42] Voges, U. and U. Visser, *PRO_PLANT - Expertensystem für den umweltschonenden Einsatz von Pflanzenschutzmitteln in der Landwirtschaft.* Berichte zur Leistungsschau der 2. deutschen Tagung für Expertensysteme, 1993: p. 13--14.

[43] Voges, U., U. Visser, A. Johnen, and K. Hell. *PRO_PLANT - Pflanzenschutz mit Hilfe eines Expertensystems*. in *17. Fachtagung für künstliche Intelligenz*. 1993. Stuttgart.

[44] Visser, U., U. Voges, K. Epke, R. Pfeiffer, and U. Streit, *Umweltschonender Pflanzenschutz mit Hilfe des Expertensystems PRO_PLANT.* Künstliche Intelligenz (KI), 1993. **93**(3): p. 34--43.

[45] Visser, U. and U. Streit, *Möglichkeiten zur Einbindung neuronaler Netze am Beispiel des Beratungssystems PRO_PLANT*, in *Referate der 13. GIL-Jahrestagung*, R. Ackmann, B. Petersen, and H. Geidel, Editors. 1991, Ulmer Verlag: Giessen. p. 32--36.

[46] Streit, U., A. Remke, and U. Visser. *Expertensysteme und neuronale Netze für die Umweltforschung*. in *Berliner Geographische Arbeiten*. 1991: Wiss. Zeitschriften der Humbold--Universität zu Berlin.

[47] Visser, U., *Die Verdunstungsmessung an Binnenseen als Problem der angewandten Hydrologie*, in *Fachbereich Geowissenschaften*. 1988, Westfälische Wilhelms-Universität Münster: Münster/Westf.

Lebenslauf von Thomas Röfer

Geburtstag:	1. März 1967
Adresse:	FB3 Mathematik und Informatik, Universität Bremen
Telefon:	+49 (421) 218-4659
Fax:	+49 (421) 218-3054
E-Mail:	roefer@informatik.uni-bremen.de
WWW:	http://www.informatik.uni-bremen.de/~roefer

Ausbildung und Abschlüsse

	Universität Bremen
1998	Doktor der Ingenieurwissenschaften (Dr. Ing.)
1993 – 1996	Stipendiat des DFG-Graduiertenkollegs „Raumorientierung und Handlungsorganisation autonome Systeme“
1993	Diplom in Informatik

Wissenschaftlicher Werdegang

	Universität Bremen
seit 2000	Wissenschaftlicher Assistent für „Kognitive Robotik“
1997 – 2000	Wissenschaftlicher Mitarbeiter im DFG-Schwerpunktprogramm „Raumkognition“

Forschungsprojekte und Forschungskoordinierung

seit 2001	DFG-Projekt „Automatische Diagnose von Strategien anderer mobiler Roboter in einer kooperativen, dynamischen Umgebung“ im Schwerpunktprogramm “Kooperierende Teams Mobiler Roboter in dynamischen Umgebungen“ (gemeinsames Projekt mit B. Krieg-Brückner, C. Schlieder und U. Visser)
2000 – 2002	DFG-Project „Navigation in dynamischen Umgebungen“ im Schwerpunktprogramm „Raumkognition“ (gemeinsames Projekt mit B. Krieg-Brückner)

Diverses

Preis

1999	Preis für die beste Dissertation vom Arbeitskreis Deutscher KI-Institute (AKI)

Organisation von Workshops

2001	Mitglied des wissenschaftlichen Komitees bei der „Conference on Spatial Information Theory 2001“ (Cosit‘01)
2000	Mitveranstalter des Workshops „Service Robotics“ bei der „European Conference on Artificial Intelligence 2000“ (ECAI-2000)
1999	Veranstalter des Tutorials „Cognitive Robotics“ bei der „Conference on Spatial Information Theory 1999“ (Cosit‘99)
1998	Mitveranstalter des Workshops „Cognitive Robotics: Architectures“ bei der Jahrestagung “Künstliche Intelligenz 1998” (KI-98)
1998	Veranstalter des Tutorials „Kognitive Robotik“ bei der Herbstschule “Kognitionswissenschaften 1998” in Hamburg

Gutachter für

Biological Cybernetics, IEEE Robotics and Automation Magazine, Information Sciences, COSIT 2001, Spatial Cognition 2002, TIMR 1997 und diverse Workshops zum Thema „Kognitive Robotik"

Forschungsaufenthalte

Universität Manchester

1998 Computer Science Department, finanziert vom Deutschen Akademischen Austauschdienst im Rahmen des ARC-Programms

2000 Computer Science Department, finanziert vom Deutschen Akademischen Austauschdienst im Rahmen des ARC-Programms

Relevante Publikationen der letzten fünf Jahre

Düffert, U.; Jüngel, M.; Laue, T.; Lötzsch, M.; Risler, M.; Röfer, T. (2003). GermanTeam 2002. In: *RoboCup 2002*, Lecture Notes in Artificial Intelligence, Springer, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings).

Röfer T. (2003). An Architecture for a National RoboCup Team. In: *RoboCup 2002*. Lecture Notes in Artificial Intelligence. Springer, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings, 388-395).

Brunn, R., Düffert, U., Jüngel, M., Laue, T., Lötzsch, M., Petters, S., Risler, M., Röfer, T., Spiess, K., Sztybryc, A. (2002). GermanTeam 2001. In: *RoboCup 2001*. Lecture Notes in Artificial Intelligence 2377. Springer. 705-708.

Lankenau, A., Röfer, T. (2002). Mobile Robot Self-Localization in Large-Scale Environments. In: *Proceedings of the IEEE International Conference on Robotics and Automation 2002 (ICRA-2002)*. IEEE. 1359-1364.

Lankenau, A., Röfer, T., Krieg-Brückner, B. (2002). Self-Localization in Large-Scale Environments for the Bremen Autonomous Wheelchair. In: *Spatial Cognition III*. Lecture Notes in Artificial Intelligence. Springer, im Erscheinen.

Röfer, T. (2002). Using Histogram Correlation to Create Consistent Laser Scan Maps. In: *Proceedings of the IEEE International Conference on Robotics Systems (IROS-2002)*.

Röfer, T., Lankenau, A. (2002). Route-Based Robot Navigation. In: Freksa, C. (Hrsg.): Künstliche Intelligenz - Themenheft Spatial Cognition. Fachbereich 1 der Gesellschaft für Informatik e.V., arenDTaP, im Erscheinen.

Lankenau, A., Röfer, T. (2001). A Safe and Versatile Mobility Assistant. In: *IEEE Robotics and Automation Magazine* 7, No. 1, March 2001. 29-37.

Lankenau, A., Röfer, T. (2000). Smart Wheelchairs - State of the Art in an Emerging Market. In: *Zeitschrift Künstliche Intelligenz 4/00*. Schwerpunkt Autonome Mobile Systeme. Fachbereich 1 der Gesellschaft für Informatik e.V., arenDTaP. 37-39.

Müller, R. Röfer, T., Lankenau, A., Musto, A., Stein, K., Eisenkolb, A. (2000). Coarse Qualitative Descriptions in Robot Navigation. In: Freksa, Ch., Brauer, W., Habel, Ch., Wender, K. F. (Eds.): *Spatial Cognition II*. Lecture Notes in Artificial Intelligence 1849. Springer. 265-276.

Musto, A., Stein, K., Eisenkolb, A., Röfer, T., Brauer, W., Schill, K. (2000). From Motion Observation to Qualitative Motion Representation. In: Freksa, Ch., Brauer, W., Habel, Ch., Wender, K. F. (Eds.): *Spatial Cognition II*. Lecture Notes in Artificial Intelligence 1849. Springer. 115-126.

Röfer, T., Lankenau, A. (2000). Architecture and Applications of the Bremen Autonomous Wheelchair. In: Wang, P. (Ed.): *Information Sciences 126*:1-4. Elsevier Science BV. 1-20.

Musto, A., Stein, K., Eisenkolb, A., Röfer, T. (1999). Qualitative and Quantitative Representations of Locomotion and their Application in Robot Navigation. In: *Proc. of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*. Morgan Kaufman Publishers, Inc. San Francisco, CA. 1067-1073 (also published as: Forschungsbericht Künstliche Intelligenz, FKI-228-99. Technische Universität München).

Röfer, T. (1999). Route Navigation Using Motion Analysis. In: Freksa, C., Mark, D. M. (Eds.): *Spatial Information Theory*, Proc. COSIT '99. Lecture Notes in Computer Science 1661. Springer. 21-36.

Röfer, T., Lankenau, A. (1999). Ensuring Safe Obstacle Avoidance in a Shared-Control System. In: J. M. Fuertes (Ed.): *Proc. IEEE 7th International Conference on Emergent Technologies and Factory Automation*. IEEE. 1405-1414.

Krieg-Brückner, B., Röfer, T., Carmesin, H.-O., Müller, R. (1998). A Taxonomy of Spatial Knowledge for Navigation and its Application to the Bremen Autonomous Wheelchair. Freksa, Ch., Habel, Ch., Wender, K. F. (Eds.): *Spatial Cognition*. Lecture Notes in Artificial Intelligence 1404. Springer. 373-397.

Röfer, T. (1998). Panoramic Image Processing and Route Navigation. Dissertation. *BISS Monographs 7*. Shaker-Verlag.

Röfer, T. (1998). Strategies for Using a Simulation in the Development of the Bremen Autonomous Wheelchair. In: Zobel, R., Moeller, D. (Eds.): *Simulation-Past, Present and Future*. Society for Computer Simulation International. 460-464.

Röfer, T., Lankenau, A. (1998). Architecture and Applications of the Bremen Autonomous Wheelchair. In: Wang, P. P. (Ed.): *Proc. 4th Joint Conference on Information Systems 1*. Association for Intelligent Machinery. 365-368.

Röfer, T., Müller, R. (1998). Navigation and Routemark Detection of the Bremen Autonomous Wheelchair. In: Lüth, T., Dillmann, R., Dario, P., Wörn, H. (Eds.): *Distributed Autonomous Robotics Systems*. Springer. 183-192.

Zwischenbericht zum Forschungsvorhaben

Automatische Plan- und Intentionserkennung fremder mobiler Roboter in kooperierenden und konkurrierenden, dynamischen Umgebungen

- erster Antragszeitraum -

im Rahmen des Schwerpunktprogramms
„Kooperierende Teams mobiler Roboter
in dynamischen Umgebungen“

an der
Universität Bremen

Bremen, November 2002

1 Allgemeine Angaben

Der vorgelegte Zwischenbericht gibt einen Überblick über den Arbeitsfortschritt des Forschungsvorhabens „Automatische Plan- und Intentionserkennung mobiler Roboter in kooperierenden und konkurrierenden, dynamischen Umgebungen“, das vom Technologie-Zentrum Informatik der Universität Bremen im Rahmen des DFG-Schwerpunktprogramms 1125 „Kooperierende Teams mobiler Roboter in dynamischen Umgebungen“ bearbeitet wird.

1.1 Antragsteller

Das Forschungsvorhaben wurde beantragt von:

Dr. Ubbo Visser, Wissenschaftlicher Assistent (C1)
Geburtsdatum: 23. 6. 1964
Nationalität: deutsch
FB3 Mathematik und Informatik
Universität Bremen
Postfach 33 04 40
D-28334 Bremen
Tel. (0421) 218 – 78 40
Fax (0421) 218 – 71 96

Private Anschrift:
Gustav-Heinemann-Str. 22
D-28215 Bremen
Tel. (0421) 43 40 860
Email: visser@informatik.uni-bremen.de

Dr. Thomas Röfer, Wissenschaftlicher Assistent (C1)
Geburtsdatum: 1. 3. 1967
Nationalität: deutsch
FB3 Mathematik und Informatik
Universität Bremen
Postfach 33 04 40
D-28334 Bremen
Tel. (0421) 218 – 46 59
Fax (0421) 218 – 30 54

Private Anschrift:
Rudolf-Alexander-Schröder-Str. 96
D-28215 Bremen
Tel. (0421) 35 06 361
Email: roefer@informatik.uni-bremen.de

1.2 Thema

Automatische Plan- und Intentionserkennung fremder mobiler Roboter in kooperierenden und konkurrierenden dynamischen Umgebungen

1.3 Berichtszeitraum und Förderzeitraum

Das Forschungsvorhaben wird von der DFG seit dem 01.06.2001 gefördert.

Das Vorhaben wird im Rahmen des ersten Antragszeitraums für zwei Jahre gefördert. Ein Fortsetzungsantrag über einen zweiten Zeitraum von zwei Jahren ist gestellt.

Der vorliegende Zwischenbericht stellt die Arbeiten dar, die seit Beginn des Vorhabens November 2001 bis Ende Oktober 2002 geleistet wurden.

1.4 Liste der Publikationen

Aus der bisherigen Bearbeitung des Forschungsvorhabens sind die nachfolgend aufgeführten Publikationen hervorgegangen:

Vollständig begutachtete Veröffentlichungen

Drücker, C.; Hübner, S.; Visser, U.; Weland, H.-G. (2002). ”As time goes by“ – Using time series based decision tree induction to analyze the behaviour of opponent players. In: Birk, A.; Coradeschi, S.; Tadokoro, S. (Eds.), RoboCup 2001: Robot Soccer World Cup V. Lecture Notes in Artificial Intelligence 2377. Springer. 325-330.

Meyer, J.; Adolph, R.; Stephan, D.; Seekamp, M.; Weinert, V.; Visser, U. (2003). Decision-making and Tactical Behavior with Potential Fields. In: RoboCup 2002: Robot Soccer World Cup VI. Lecture Notes in Artificial Intelligence, Springer, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings, 300-307).

Miene, A.; Visser, U. (2002). Interpretation of spatio-temporal relations in real-time and dynamic environments. In: Birk, A.; Coradeschi, S.; Tadokoro, S. (Eds.), RoboCup 2001: Robot Soccer World Cup V. Lecture Notes in Artificial Intelligence 2377. Springer. 441-446.

[illegible]r, T. (2003). An Architecture for a National RoboCup Team. In: RoboCup 2002: Robot Soccer World Cup VI. Lecture Notes in Artificial Intelligence, Springer, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings, 388-395).

Visser, U.; Drücker, C.; Hübner, S.; Schmidt, E.; Weland, H.-G. (2001). Recognizing Formations in Opponent Teams. In: Balch, T., Stone, P., Kraetzschmar, G. (Eds.): 4th International Workshop on RoboCup. Carnegie Mellon University Press, Melbourne, Australia, 391-396.

Visser, U.; Weland, H.-G. (2003). Using online learning to analyze the opponents behavior, In: RoboCup 2002: Robot Soccer World Cup VI. Lecture Notes in Artificial Intelligence, Springer, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings, 72-81).

Regulär veröffentlichte Team-Berichte

Bernau, D., Brachmann, L., Enns, W., Hübner, S., Kolweyh, M., Silber-Mankowsky, A., Weidner, J., & Wollny, B. (2002). Team Description Virtual Werder 2002. In: RoboCup 2002: Robot Soccer World Cup VI. Fukuoka/Busan: Springer Verlag, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI, Pre-Proceedings, S. 555).

Brunn, R.; Düffert, U.; Jüngel, M.; Laue, T.; Lötzsch, M.; Pette[illegible]; Risler, M.; Röfer, T.; Spiess, K.; Sztybryc, A. (2001). German Team 2001. In: Birk, A.; Coradeschi, S.; Tadokoro, S. (Eds.), RoboCup 2001: Robot Soccer World Cup V, Lecture Notes in Artificial Intelligence 2377. Springer. 705-708.

Düffert, U.; Jüngel, M.; Laue, T.; Lötzsch, M.; Risler, M.; Röfer, T. (2003). GermanTeam 2002. In: RoboCup 2002: Robot Soccer World Cup VI, Lecture Notes in Artificial Intelligence, Springer, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings, S. 451).

Meyer, J., Adolph, R., Stephan, D., Daniel, A., Seekamp, M., Visser, U., & Weinert, V. (2003). BUGS - Team Description. In: RoboCup 2002: Robot Soccer World Cup VI. Fukuoka/Busan: Springer Verlag, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI, Pre-Proceedings, S. 521).

Online Team-Berichte

Burkhard, H.-D.; Düffert, U.; Jüngel, M.; Lötzsch, M.; Koschmieder, N.; Laue, T.; Röfer, T.; Spiess, K.; Sztybryc, A.; Brunn, R.; Risler, M.; v. Stryk, O. (2001). GermanTeam 2001. Technischer Bericht, 42 Seiten. http://www.robocup.de/germanteam.

Burkhard, H.-D.; Düffert, U.; Hoffmann, J.; Jüngel, M.; Lötzsch, M.; Brunn, R.; Kallnik, M.; Kuntze, N.; Kunz, M.; Petters, S.; Risler, M.; v. Stryk, O.; Koschmieder, N.; Laue, T.; Röfer, T.; Spies[illegible]; Cesarz, A.; Dahm, I.; Hebbel, M.; Nowak, W.; Ziegler, J. (2002). GermanTeam 2002. Technischer Bericht, 178 Seiten. http://www.robocup.de/germanteam.

Eingereichte Beiträge

Röfer, T.; Jüngel, M. (2003). Vision-Based Fast and Reactive Monte-Carlo Localization. Eingereicht bei: International Conference on Robotics and Automation 2003 (ICRA-2003).

2 Arbeits- und Ergebnisbericht

Dieses Kapitel ist wie folgt gegliedert: im Unterkapitel *Architektur* werden zunächst grundsätzliche Beriffe wie Strategie und Taktik definiert. Der Rest dieses Unterkapitels beschreibt den Aufbau des GermanTeams in der Sony Legged League inklusive der Software-Architektur und einer Simulationskomponente. Das nächste Unterkapitel beschreibt Arbeiten, die im Zuge der Entwicklung des Roboterteams bezüglich der *Weltmodellierung* angefallen sind. Der letzte Teil beschäftigt sich mit der *Plan- und Intentionserkennung* und stellt die wesentlichen Entwicklungen (z.B. das Rahmenwerk für die Planerkennung) dar. Hieraus leiten sich u.a. die im Folgeantrag beschriebenen Arbeitspakete ab.

2.1 Architektur

2.1.1 Definition von Strategie und Taktik

Um Strategieplanung und Strategieerkennung betreiben zu können, musste zuerst eine Definition der Begriffe Strategie und Taktik gefunden werden. Hierzu wurden Beispiele aus verschiedenen Domänen (Kriegsführung, Sport, Wirtschaft, Lernen) betrachtet, in denen diese Begriffe ebenfalls eine große Rolle spielen. Eine ältere aber auch aussagekräftige Beschreibung stammt dabei von dem deutschen Kriegsphilosophen Carl von Clausewitz (1780-1831):

"In der Taktik sind die Mittel die ausgebildeten Streitkräfte, welche den Kampf führen sollen. Der Zweck ist der Sieg. [...] Die Strategie hat ursprünglich den Sieg, d.h. den taktischen Erfolg, nur als Mittel, und in letzter Instanz die Gegenstände, welche unmittelbar zum Frieden führen sollen, als Zweck."

In den übrigen Domänen finden sich ähnliche Beschreibungen, die Taktik und Strategie – entgegen der heutigen umgangssprachlichen Verwendung – klar voneinander trennen. Die Betonung der Strategie liegt darauf, die *richtigen* Dinge zu tun („*was* ist zu tun?“), die der Taktik, die *Dinge* richtig zu tun („*wie* ist es zu tun?“). Strategien sind daher abstrakte Vorgehensweisen, die ein globales Ziel verfolgen und auch von äußeren Umständen abhängig sind. Taktiken sind abhängig von der gewählten Strategie und der spezifischen Situation, besitzen eher absehbare, lokale Ziele und beschreiben die konkretisierten Mittel. Taktiken sind daher einer oder mehreren Strategien untergeordnet.

In der Fußballdomäne bzw. im RoboCup lässt sich durch diese Trennung eine Planungshierarchie erstellen. Die Strategie ist eine auf die ganze Mannschaft und ein übergeordnetes Ziel bezogene Einstellung, beispielsweise das Spiel zu gewinnen, wenig Gegentore zu erhalten oder möglichst hohe Anteile am Ballbesitz zu erzielen, defensiv oder offensiv zu spielen. Die Wahl kann von Tabellensituation, Kenntnis über den Gegner oder von äußeren Umständen (Wetter, Licht) abhängig sein, das *Spiel* an sich fordert dabei heraus (s. Abb. 1). Eine Taktik beschreibt die Möglichkeit, ein lokales Problem zu lösen, z.B. einen Gegner zu umspielen, d.h. der *Gegner* fordert heraus. Abhängig von der Situation und der Strategie kann eine Taktik ausgewählt werden, z.B. Umspielen des Gegners durch Dribbling oder Doppelpass (offensiv) oder durch Ausweichen nach hinten (defensiv).

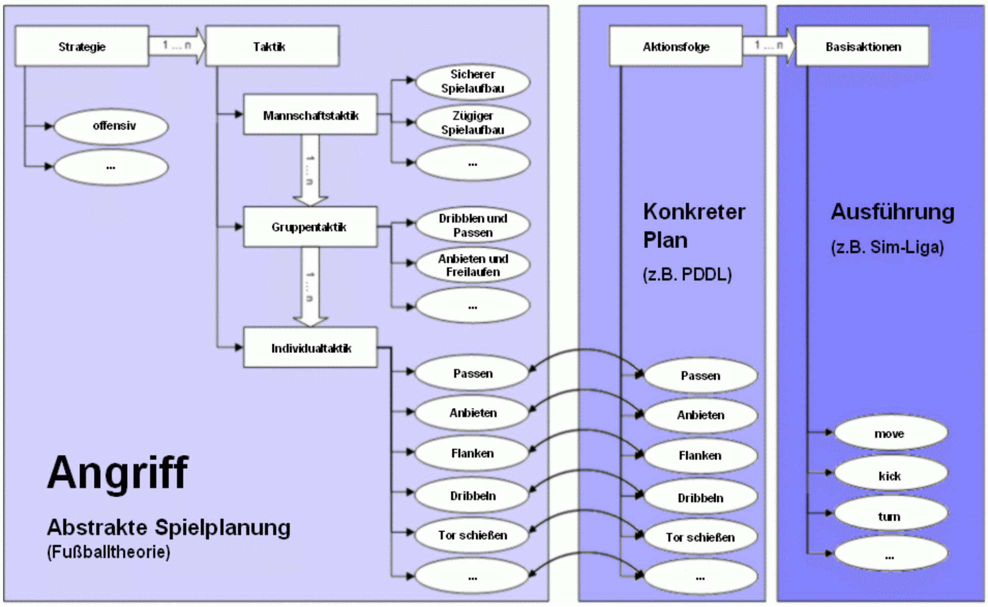


Abbildung 1. Beispiel zu Repräsentation und Umsetzung abstrakten Fußballwissens

Definition und Zusammenhang der Begriffe „Strategie“ und „Taktik“ wurden für die Strategieerkennung im eigenen Teilprojekt unbedingt benötigt, außerdem aber auch als eine allgemeine Begriffsdefinition in der AG3 Architektur vorgestellt und angenommen.

2.1.2 GermanTeam

Mit Beginn des Schwerpunktprogramms startete auch die Zusammenarbeit von fünf Universitäten in der Sony Four-legged Robot League im GermanTeam (HU Berlin, FU Berlin, TU Darmstadt, Uni Bremen, Uni Dortmund). Das Team wurde auf Initiative von Hans-Dieter Burkhard von der Humboldt-Universität zu Berlin gegründet, der sein eigenes Team, die Humboldt Heroes, für Mitglieder anderer Universitäten öffnete, um größere Fortschritte in der Liga erzielen zu können. Zur Weltmeisterschaft 2001 in Seattle trugen nur drei der fünf Partneruniversitäten zum Ergebnis bei (HU Berlin, TU Darmstadt, Uni Bremen) und die Zusammenarbeit startete zu spät, um bereits Früchte zu tragen. Nach der Weltmeisterschaft stießen noch Mitglieder der Universität Dortmund zum GermanTeam hinzu. Dieses Team schaffte es, beim RoboCup 2002 in Fukuoka, Japan leistungsmäßig in das obere Viertel der Liga vorzustoßen. Bremen stellt gegenwärtig den Team-Leader des GermanTeam (Thomas Röfer), der auch die Erstellung der teilweise sehr umfangreichen Team-Berichte koordiniert hat (Brunn et al., 2002; Düffert et al., 2003; Burkhard *et al.*, 2001; 2002).

2.1.3 Software-Architektur für ein Nationalteam

Aus der Zusammenarbeit im GermanTeam ergaben sich ganz spezielle Aufgabenstellungen, da die einzelnen Universitäten bei den German Open gegeneinander antreten, zwei Monate später bei der Weltmeisterschaft aber gemeinsam auftreten. Daher wurde eine gemeinsame Architektur entwickelt, die es erlaubt, verschiedene Implementierungen in einem einzigen System zu unterstützen (Röfer, 2003; Burkhard *et al.*, 2002). Dazu wurden die verschiedenen Aufgaben identifiziert, die für das Fußballspielen in der Sony Legged League gelöst werden müssen. Diese können grob in fünf Schichten eingeteilt werden: Sensorvorverarbeitung, Objekterkennung, Weltmodellierung, Verhaltenssteuerung und Bewegungssteuerung (vgl. Abb. 2). Für diese Aufgaben wurden Schnittstellen definiert, die durch verschiedene Lösungen implementiert werden können. Es ist auch möglich, mehrere dieser Aufgaben zusammenzufassen, so dass auch Lösungen implemen-

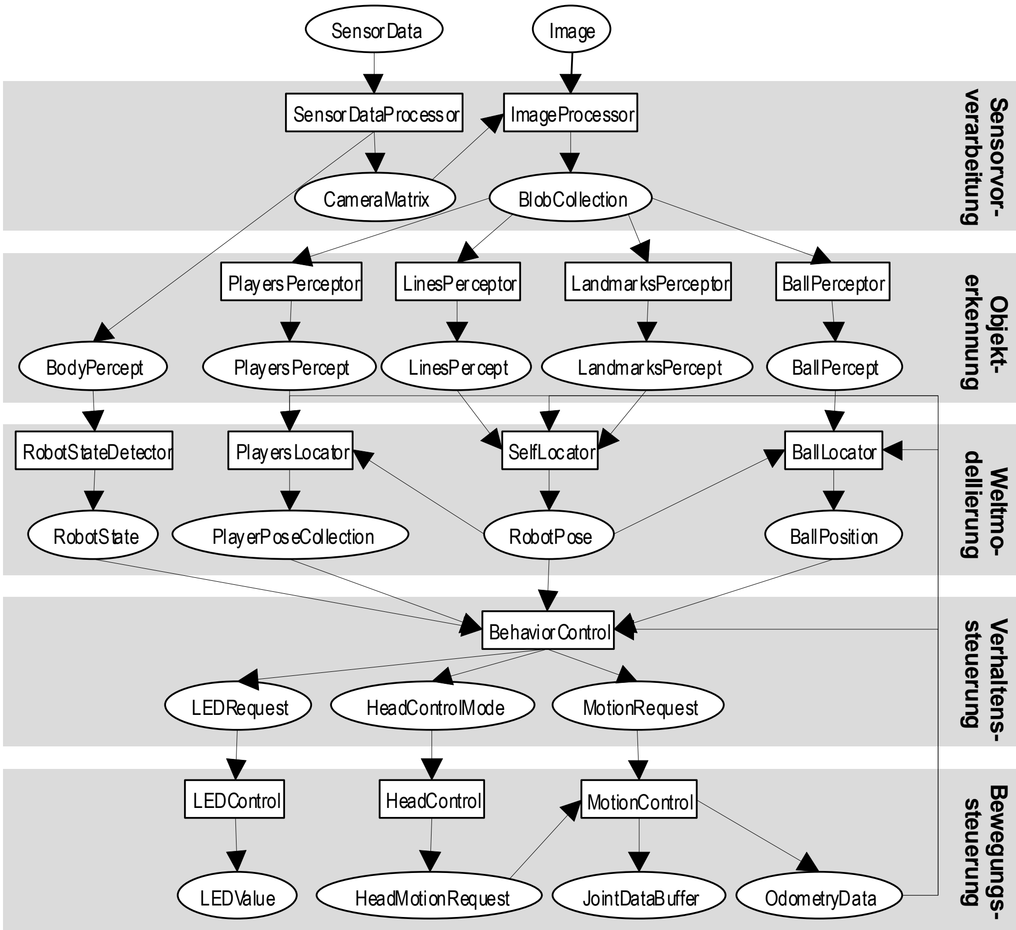


Abbildung 2. Software-Architektur des GermanTeam

tiert werden können, die z.B. eine direkte Sensor-Motor-Kopplung realisieren. Hiervon wurde im Bereich Sensorvorverarbeitung und Objekterkennung auch Gebrauch gemacht (Röfer und Jüngel, 2003). Alle Lösungen können zur Laufzeit ausgetauscht werden, was einen guten Vergleich in verschiedenen Situationen erlaubt.

In einer Echtzeitumgebung spielt zudem die parallele Ausführung der implementierten Module eine Rolle. Um auch hier verschiedene Varianten testen zu können, wurde die Möglichkeit geschaffen, Lösungen zu unterschiedlich vielen, parallel ablaufenden Prozessen zusammenzufassen. In Bremen wurde dabei versucht, möglichst viele Aufgaben parallel abarbeiten zu lassen und die kritischen Aufgaben, wie z.B. die Ballverfolgung, mit hoher Priorität ablaufen zu lassen (Röfer, 2003). Leider bot dieser Ansatz keine wesentlichen Vorteile gegenüber einer weit geringeren Parallelisierung, weil die größte Rechenlast in der Sensorvorverarbeitung (d.h. hauptsächlich Bildverarbeitung) anfällt, die die Eingaben für die meisten anderen Module liefert.

2.1.4 Simulation

Der in Bremen entwickelte 3D-Simulator SimRobot (Röfer, 1998) wurde so erweitert, dass er bis zu zwei vollständige Teams von Sony-Aibo-Robotern simulieren kann (Burkhard et al., 2002; Röfer, 2003). Damit die Roboter-Steuerprogramme sowohl im Simulator als auch auf den Robotern laufen können, wurde eine Abstraktionsschicht eingeführt, die mittlerweile für drei Plattformen zur Verfügung steht (Sony-Aibo, MS Windows, Linux). Das Ergebnis ist der erste 3D-Simulator in der Sony-Legged-League, der die ganze Spannweite der Entwicklung von der Bildverarbeitung bis zur Verhaltenssteuerung unterstützen kann. In der Zusammenarbeit mit der Berliner Arbeitsgruppe wurde er auch in das zentrale Werkzeug des GermanTeams, *RobotControl*, integriert. Gegenwärtig werden Möglichkeiten diskutiert, wie diese Entwicklung in die Bemühungen der AG1-Systemintegration um einen einheitlichen Simulator für die Middle-Sized-League einfließen kann.

2.2 Weltmodellierung – Aufbau des Weltmodells

Wie sich auch in einigen Workshops der AG3-Architektur gezeigt hat, besteht das Weltmodell eines Agenten in allen RoboCup-Ligen im Wesentlichen aus der eigenen Position auf dem Spielfeld, der Position des Balles, der Mitspieler und der Gegner. All diese Informationen sind mit Unsicherheiten behaftet. Das Weltmodell ist sowohl die notwendige Basis für die Verhaltenssteuerung, wie auch für die Analyse des gegnerischen Verhaltens, d.h. des zentralen Anliegens des Bremer Engagements im Schwerpunktprogramm. Im ersten Halbjahr der Projektlaufzeit wurde das GermanTeam mit einer Weltmodellierung ausgestattet, die auch wesentlich zu der erfolgreichen Teilnahme in Fukuoka beigetragen hat:

Selbstlokalisierung

Aufbauend auf dem Ansatz der sog. Monte-Carlo-Lokalisierung (Fox et al., 1999) wurde eine Verfahren zur Selbstlokalisierung implementiert, das aus der Literatur bekannte Ansätze wie die *Sensor Resetting Localisation* (Lenser und Veloso, 2000) integriert, aber auch eigene Erweiterungen enthält (Burkhard et al., 2002; Röfer und Jüngel, 2003), die auf die speziellen Anforderungen in der Sony-Liga zugeschnitten sind (z.B. ungenaue und spärliche Sensorinformationen, geringe Rechenleistung und schnelle Relokalisierung nach Schiedsrichtereingriff). Dadurch war es dem GermanTeam z.B. möglich, dass die Roboter automatisch ihre Startpositionen auf dem Spielfeld eingenommen haben.

Bei dem im Wettbewerb verwendeten Verfahren orientierten sich die Roboter noch an den am Spielfeldrand platzierten Farbmarken. In einer Zusammenarbeit zwischen Berlin und Bremen wurde aber auch mit der Entwicklung einer Methode begonnen, die ohne diese künstlichen Markierungen auskommt und bei der sich der Roboter an Feldlinien orientiert (Burkhard et al., 2002; Röfer und Jüngel, 2003). Ein erster Prototyp wurde in Fukuoka bereits vorgestellt, was u.a. dazu geführt hat, dass es beim RoboCup 2003 wahrscheinlich einen sog. *Challenge* geben wird, bei dem sich Roboter ohne Marken auf dem Spielfeld lokalisieren müssen.

Ballmodellierung

Die Ballmodellierung ist bisher recht rudimentär. Es wird unterschieden, ob ein Ball an seiner geschätzten Position gerade gesehen werden könnte oder nicht, wodurch eine Ballposition nicht vergessen wird, sobald der Roboter in eine andere Richtung blickt. Es wurden erste Verfahren entwickelt, die eine aktive Steuerung der Wahrnehmung des Roboters realisieren, um den Konflikt zwischen den verschiedenen Zielen Selbstlokalisierung (Blick zu den Marken), Ballverfolgung (Blick zum Ball) und Mitspieler/Gegnerverfolgung zu lösen. Dabei wird die Unsicherheit über die eigene Position berücksichtigt. Außerdem wurde ein erster Ansatz zur Integration von zwischen den Robotern kommunizierten Ballpositionen implementiert, der sich an den Arbeiten von Dietl et al. (2001) orientiert.

Mitspieler/Gegnermodellierung

Für die Mitspieler- und Gegnererkennung wurde in Zusammenarbeit mit der Berliner Gruppe eine Lösung erstellt, bei der die Positionen der Roboter in einem Raster als eine Aufenthaltswahrscheinlichkeitsverteilung modelliert werden (Burkhard et al., 2002). In dieser Verteilung wird nach Maxima gesucht, die dann als Positionen der Roboter interpretiert werden. Da diese Lösung noch keine wirklich zufrieden stellenden Ergebnisse liefert, wird gegenwärtig an einer Partikel-basierten Modellierung gearbeitet.

2.3 Plan- und Intentionserkennung

Seit 1999 wurde ein Multiagentensystem für die Simulationsliga des RoboCup unter Leitung von Dr. Visser kontinuierlich weiterentwickelt. Diese Mannschaft ist seit dem Jahr 2000 sowohl auf den nationalen als auch bei den internationalen Veranstaltungen dabei. Schwerpunkt ist die Analyse des gegnerischen Spielsystems und die Ergreifung entsprechender Gegenmaßnahmen. Im Rahmen des Projekts wurde das Spielverhalten gegnerischer Teams auf Basis von Log-Dateien vergangener Spiele aus den Weltmeisterschaften 1998-2001 analysiert.

Ähnliche Ansätze wurden von verschiedenen Forschergruppen in den letzten Jahren verfolgt. Sie beziehen sich auf das Lernen von Teamverhalten in der Simulations- und Middle-Size-League (vgl. (Stone, Riley & Veloso, 2000; Stone & Veloso, 1998a, 1998b, 1999)) und können in Offline- und Online-Analyseverfahren unterschieden werden.

Offline-Analyseverfahren

Bei der Offline-Analyse werden Spiele anhand des erzeugten Logfiles analysiert. Zeit und Rechenaufwand spielen praktisch keine Rolle, da diese Verfahren nach Ende eines Spieles durchgeführt werden. Mit Hilfe des ISAAC Systems (Raines et al., 2000) werden z.B. Regeln generiert, die Einzelspieler, das Zusammenspiel zwischen Spielern sowie das gesamte Spiel beschreiben. Wünstel et al. (2001) stellen einen Ansatz vor, der Self-Organizing Maps zur Analyse des Laufverhaltens einzelner Spieler verwendet.

Online-Analyseverfahren

Frank et al. (2000) präsentieren einen Real-Time Ansatz, der auf statistischen Verfahren beruht. Der „Statistics Proxy Server“ wurde im Rahmen des automatischen Kommentatorensystems MIKE für die Simulations- und Small-Size-League entwickelt. Hier wird auch eher eine Teambewertung vorgenommen als ein Erkennen von Strategien. Der selbst entwickelte Virtual Werder Coach (Visser et al., 2001) erkennt gegnerische Spielsysteme mit Hilfe eines neuronalen Netzes. Die daraus gewonnenen Informationen werden dann dazu verwendet, die Formation des eigenen Teams an die gegebenen Umstände anzupassen.

Die Arbeiten in der ersten Phase des Projektes haben sich sowohl auf die Offline- als auch auf die Online-Analyse bezogen. Die Ergebnisse der Arbeiten zur Erkennung von Aktionen, Folgen von Aktionen und Situationen sind lassen sich in zwei grundsätzliche Bereiche gliedern:

2.3.1 Qualitative raum-zeitliche Relationen (Offline)

Wenn wir die Bewegung der Objekte auf einem Spielfeld betrachten, so kann man die Bewegung als Veränderung über Raum und Zeit sehen. Deswegen ist es hilfreich, das Verhalten der sich bewegenden Objekte mit Hilfe von raum-zeitlichen Relationen zu beschreiben. Es ist möglich, die Beschreibung unabhängig von der Anwendungsdomäne vorzunehmen, so daß der Ansatz sehr generell einsetzbar ist. Sobald die raum-zeitlichen Relationen zwischen den Objekten beschrieben worden ist, ist eine Interpretation des Verhaltens dieser Objekte möglich.

Beschreibung von raum-zeitlichen Relationen

Eine räumliche Relation zwischen zwei Objekten existiert in einem bestimmten Zeitintervall. Die Bewegungsrichtung und die Geschwindigkeit dieser Objekte hat ebenfalls eine bestimmte Dauer. Die Idee ist, sowohl die Richtung und die Geschwindigkeit als auch die räumlichen Beziehungen in Zeitintervallen darzustellen.

Wir haben uns zunächst auf Daten aus der Simulationsliga beschränkt, da hier das Verhalten weitaus komplexer bzw. weiter entwickelt ist als in den anderen Ligen. Die Eingabedaten bestehen aus Zeitreihen von Objektkoordinaten. In einem ersten Schritt werden Zeitintervalle von kontinuierlichen Objektbewegungen (OMI) und die Dauer der räumlichen Relationen zwischen zwei Objekten (SRI) generiert. Dieses wird zur Laufzeit in frei zu wählenden Zeitschritten wiederholt. Zu jedem Zeitpunkt wird für jedes Objekt ein Bewegungsvektor erzeugt, der die Veänderung des Objektes zum vorherigen Zietpunkt über die Länge und den Winkel dokumentiert. Jeder OMI bezieht sich auf exakt ein Objekt O und hat genau einen Startpunkt i_s, einen Endpunkt i_e, die Bewegungsrichtung α und die Bewegungsgeschwindigkeit $v : i_{om} = \left[O\langle \alpha, v \rangle\right]_{i_s}^{i_e}$.

Die räumlichen Relationen zwischen zwei Objekten werden durch die Richtung der Lokalität des zweiten Objektes und der Distanz dahin spezifiziert. Die metrische Distanz sowie der Winkel in die Richtung der Lokalität des zweiten Objekts kann damit berechnet werden. Der Prozess der Ermittlung der Dauer dieser räumlichen Relation ist anlog zu den OMIs. Um nun eine qualitative Beschreibung zu erhalten, wird eine endliche Anzahl von Richtungen und Distanzen festgelegt. Zunächst wurden *meets, near, medium* und *far* unterschieden. Für die Richtung wurde eine Windrose verwendet. Jedes SRI bezieht sich auf genau ein Paar von Objekten (O_1,O_2) und hat genau einen Startpunkt i_s, einen Endpunkt i_e, die Lokationsrichtung l und die Veränderung $d : i_{sr} = \left[O_1\langle l, d \rangle O_2\right]_{i_s}^{i_e}$. Die Beziehung zwischen OMIs und SRIs ist über die sieben zeitlichen Relationen (inkl. der inversen) nach Allen (1983) beschrieben: *before, meets, overlaps, starts, during, finishes* und *equals.*

Interpretation von raum-zeitlichen Relationen

Für die Interpretation sind die Konzepte wichtig, die ein Experte bei der Beobachtung eines Spieles verwendet. Diese Konzepte werden dann in OMIs und SRIs aufgesplittet und danach werden die zeitlichen Intervalle bestimmt. Dieses führt zu einer Definition der Konzepte mit Hilfe von raum-zeitlichen Relationen und es ist möglich, Konzepte in einem Szenario zu erkennen.

Elementare Konzepte beschreiben einfache Bewegungen und sind domänenunabhängig, während komplexe Konzepte aus elementaren Konzepten zusammengesetzt sind und in jeder Domäne unterschiedlich sein können. Die elementaren Konzepte können nach Anzahl der involvierten SRIs und OMIs und deren zeitlichen Relationen in Gruppen unterschieden werden.

Um elementare Konzepte erkennen zu können, müssen Attribute wie Bewegungsrichtung, Geschwindigkeit, Distanz zum zweiten Objekt, etc. betrachtet werden. Zusätzlich müssen sie Bedingungen erfüllen. Zwei Objekte erfüllen beispielsweise *meet* wenn O_1 und O_2 sich annähern, d.h., daß die Geschwindigkeit von $O_1 > 0$ ist, die Bewegungsrichtung identisch mit der Richtung des zweiten Objektes ist und die Geschwindigkeit von $O_2 = 0$ ist.

Ergebnisse

Dieses Verfahren wurde bei Spielen in der Simualtionsliga angewendet. Jedes einfache Ereignis besteht aus mindestens einem Zyklus. Die Zyklen sind Schnappschüsse des Spieles und bilden die Grenzen der Zeitintervalle.Ein elementares Ereignis ist z.B. *departing(dep(b,p)*, d.h. ein Spieler *p* passt den Ball *b*. Dieses Ereignis kann in verschiedene komplexere Ereignisse eingebunden sein, z.B. ein Spieler P_1 passt den Ball zu Spieler P_2. Diese Situation besteht aus vier elementaren Ereignissen und $P_1 \neq P_2$.

$$se_1=meets(p_1,b) \wedge se_2=dep(b,p_1) \wedge se_3=app(b,p_2) \wedge se_4 = meets(b,p_2)$$

Die temporalen Relationen zwischen den elementaren Ereignissen sind $meets(se_1,se_2)$, $equal(se_2,se_3)$ und $meets(se_3,se_4)$ (siehe Abb. 3). Dieses Beispiel zeigt, wie komplexe Situationen durch die Beschreibung von räumlichen und zeitlichen Relationen erkannt werden können.

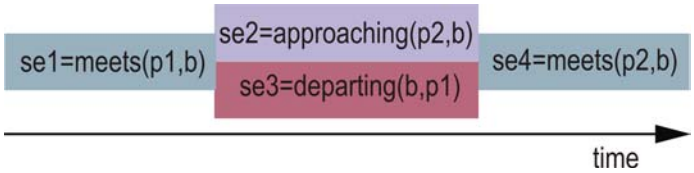


Abbildung 3. Spieler 1 passt den Ball zu Spieler 2

Die Ergebnisse haben gezeigt, dass eine derartige Beschreibung der Objekte auf dem Spielfeld möglich ist, um einfache und komplexe Aktionen oder Teilpläne zu erkennen. Diese Ergebnisse haben dazu beigetragen, das Rahmenwerk für die Erkennung von Plänen auf qualitativer Basis voranzutreiben. Eine detailliertere Darstellung des Ansatzes und der Ergebnisse ist in Miene & Visser (2002) zu finden.

2.3.2 Planerkennung mit zeitreihenbasierten symbolischen Lernverfahren (Online)

Frühere Arbeiten, die im Bereich Online-Analyse einzuordnen sind, betrafen die generische Formationserkennung. Die daraus gewonnenen Informationen wurden dann dazu verwendet, die Formation des eigenen Teams an die gegebenen Umstände anzupassen. Eine Eigenschaft dieses Verfahrens ist es jedoch, daß die gesamte Mannschaft bzw. deren Verhalten Gegenstand der Analyse gewesen ist und damit lediglich grundsätzliche Informationen erzielt werden konnten. Eine Erkennung von Einzelsituationen, Aktionen oder gar das Verhalten einzelner Spieler war mit dem Ansatz nicht möglich. Genau dieses ist das Ziel der neueren Arbeiten aus der ersten Projektphase. Mit Hilfe des Online-Coaches aus der Simulationsliga soll gegnerisches Verhalten während des Spieles erkannt werden. Die zu entwickelnden Verfahren müssen daher in der Lage sein, in kurzer Zeit zu sinnvollen Ergebnissen zu kommen. Die folgenden weiteren Anforderungen haben sich in der Vorbereitungsphase ergeben:

- Echtzeitumgebung (möglichst schnelle Ergebnisse)
- Reelwertige Attribute müssen verarbeitet werden können
- Möglichst geringer Ressourcenverbrauch (alle Teams haben die gleichen Ressourcen, der Coach teilt sich die Ressourcen mit den Spielern)
- Fehlerhafte und widersprüchliche Eingabedaten sind vorhanden
- Keine Einflussnahme von Außen möglich
- Informationsgewinnung durch Lernen

Methode

Nach Überprüfung der möglichen Methoden wurde die Entwicklung eines neuen Verfahrens notwendig, das den o.g. Anforderungen gerecht wird. Es handelt sich dabei um eine zeitreihenbasierte Entscheidungsbauminduktion. Diese Methode basiert auf multiplen Zeitreihen und setzt auf die bekannten Verfahren ID3 und C4.5 (Quinlan, 1993) auf. Die neue Idee basiert auf einer signifikante Reduktion der potentiellen Splitpunkte, was bei den bekannten Verfahren ein gewisses Problem in Echtzeitumgebungen darstellt, weil es zu viele Kandidaten für diese Punkte gibt. Boronowsky (2001) hat allerdings gezeigt, daß die Entropie-Minimalisierungsheuristik bei kontinuierlichen Gleichverteilungen nur an echten Randpunkten minimal werden kann. Echte Randpunkte sind die Punkte, die an den Rändern der Intervalle einer Klassenzugehörigkeit liegen. Eine stückweise Linearisierung der Zeitreihen hilft dabei: solche lineare Klassenintervalle sind kontinuierlich gleichverteilt. Man kann ausserdem zeigen, daß eine stückweise Linearisierung einer Zeitreihe ausreicht, um ein Klassenintervall zwischen zwei Randpunkten zu beschreiben. Es ist also möglich, eine effiziente Entscheidungsbauminduktion für reellwertige Zeitreihen durchzuführen, wenn diese Zeitreihen aus stückweise linearen Funktionen bestehen.

Voraussetzung für den Einsatz dieser Methode sind zwei Dinge: (1) die qualitative Abstraktion der Zeitintervalle für die Erzeugung der Klassen und (2) die Auswahl der Attribute, die die Klassenzugehörigkeit bedingen.

Ergebnisse

Die Testszenarien betreffen das Verhalten des Torwarts sowie das Passverhalten von Spielern. Für das Verhalten des Torwartes sind die folgenden Klassen und Attribute gewählt worden:

Klassen:
- Torwart verlässt das Tor
- Torwart bleibt im Tor
- Torwart geht ins Tor

Attribute:
- Abstand Ball-Torwart
- Geschwindigkeit des Balles
- Abstand Ball-Tor
- Anzahl der Abwehrspieler im Strafraum
- Anzahl der Stürmer im Strafraum
- Anzahl der Abwehrspieler, die einen Torschuss abfangen können

In den Testspielen sind geeignete Mannschaften, die in der Liga bekannt und performant sind, ausgewählt worden (z.B. Karlsruhe Brainstormers, RoboLog Koblenz). Die Methode ist dann in mindestens 20 Spielen gegen diese Mannschaften getesttet worden. Die Abbildung 4 zeigt die Regeln, die in einem Spiel zwischen RoboLog Koblenz und den Karlsruhe Brainstormers erzeugt worden sind (nach 1000 Zyklen, =1/6 der Spielzeit). In diesem Szenarion wird der Torwart der Koblenzer betrachtet. Die Regeln aus der Abbildung können folgendermaßen interpretiert werden:

1. Der Torwart bleibt im Tor, wenn der Ball weniger als 8,21m vom Tor entfernt ist und sich nicht bewegt.

2. Wenn der Ball mehr als 8.21m entfernt ist, geht er ins Tor zurück (mit einer Wahrscheinlichkeit von 51%) oder bleibt an seiner Stelle (29%)

3. Mit einer Wahrscheinlichkeit von 50% geht er zu einem Ball, der weniger als 14,55m entfernt ist und sich bewegt.

4. Torwart reagiert erst, wenn der Ball weniger als 14.55m entfernt ist (~85%)

```
if   AbstBallTw < 14.551129
and  GeschwBall < 0.021304
and  AbstBallT  < 8.216856
then bleibt(1) verlässt(0) zurück(0)


if   AbstBallTw < 14.551129
and  GeschwBall < 0.021304
and  AbstBallT  > 8.216856
then zurück(0.51) bleibt(0.29) verlässt(0.18)


if   AbstBallTw < 14.551129
and  GeschwBall > 0.021304
then verlässt(0.5) zurück(0.11) bleibt(0.04)


if   AbstBallTw 0 > 14.551129
then bleibt(0.84) zurück(0.11) verlässt(0.04)
```

Abbildung 4. Torwartverhalten, Regeln nach 1000 Zyklen aus einem Spiel der RoboLog Koblenz gegen Karlsruhe Brainstormers

Für das Passverhalten sind die folgenden Klassen und Attribute gewählt worden:

<u>Klassen</u>

- Erfolgreicher Pass des Gegners
- Erfolgreiches Dribbeln des Gegners
- Fehlpass des Gegners
- Eigene Mannschaft im Ballbesitz

<u>Attribute</u>

- Abstand zum nächsten Gegner
- Winkel zum nächsten Gegner
- Abstand zum übernächsten Gegner
- Winkel zum übernächsten Gegner
- Abstand zum nächsten Mitspieler
- Winkel zum nächsten Mitspieler
- Abstand zum übernächsten Mitspieler
- Winkel zum übernächsten Mitspieler
- Mannschaft des Passgebers
- x-Position des Balles

Auch hier sind Testspiele gespielt worden und es ergeben sich ähnlich gute Ergebnisse. Auf eine detailliertere Darstellung wird aus Platzgründen verzichtet, ausführliche Informationen finden sich in (Visser & Weland, 2002; Weland, 2002).

Wünschenwert wäre eine automatische Auswertung der gefundenen Regeln. Prinzipiell ist ein „Wiedererkennen“ des Gegners möglich ist, ein Indiz dafür, das Mannschaften ähnliches Spielverhalten zeigen, wenn sie gegen einen bestimmten Gegner immer wieder antreten. Dieses führt u.a. dazu, eine Wissensbasis aufzubauen, in der gegnerspezifische Informationen abgelegt werden können und zu einem späteren Zeitpunkt verwendet werden können. Die Regeln können allerdings auch direkt verwendet werden. Ein wichtiger Punkt ist die Interpretation der Ergebnisse und deren Umsetzung in Anweisungen an die eigenen Spieler.

Die Ergebnisse zeigen, daß diese Art der Planerkennung im laufenden Spiel machbar ist und daß die gefundenen Regeln für die weitere Verarbeitung brauchbar sind. Nach ca. 1/6 des Spieles werden die ersten Informationen über den Gegner gefunden und werden alle 1000 Zyklen weiter verfeinert.

Obwohl diese Methode bisher nur in der Simulationsliga ausprobiert worden ist, wurde von Anfang an auf den potentiellen Einsatz in anderen Ligen Wert gelegt. Die Methode ist als C++ Klassenbibliothek frei verfügbar und wird in der Phase 2 u.a. von der RWTH Aachen eingesetzt.

2.3.3 Konzeptionelle Architektur zur Plan- und Intentionserkennung

In dem im letzten Abschnitt beschriebenen Ansatz zur Plan- und Intentionserkennung wird gegnerisches Verhalten unabhängig von den eigenen Absichten mittelfristig zur Laufzeit erkannt. Damit ist ein wichtiger erster Grundstein gelegt, um Planerkennung in einem praktischen Umfeld validieren zu können. Um Planerkennung jedoch universell in verschiedenen Ligen, mit hochgradig differierenden Rahmenbedingungen flexibel anwenden zu können, ist eine konzeptionelle Architektur erforderlich, die die Rahmenbedingungen beschreibt, die erfüllt sein müssen, um Planerkennung praktisch einsetzen zu können. Dabei ist ein breites Spektrum an Anforderungen zu berücksichtigen:

Granularität: Ob ein erkannter Plan oder eine erkannte Intention für einen Agenten wirklich nützlich ist, hängt entscheidend vom Detaillierungsgrad der Informationen ab. In einer Zweikampfsituation will ein Spieler unmittelbar wissen in welche Richtung sich der Gegner bewegen wird. Ein Abwehrchef hingegen ist auch an dem mittelfristigen taktischen Angriffsverhalten des Gegners interessiert und ein Trainer kann noch weitaus abstraktere strategische Informationen sinnvoll verwenden und Schlussfolgerungen für das Spielverhalten der eigenen Mannschaft daraus ziehen.

Effizienz: Es reicht nicht, dass eine Intention korrekt und eventuell vollständig erkannt wird, sondern sie muss auch so effizient generiert werden, das sie noch in die Planung zukünftiger Handlungen einbezogen werden kann (es hilft nicht, zu erkennen, dass ein Gegenspieler links an mir vorbeispielen will, wenn ich diese Information erst erhalte, wenn er bereits links neben mir steht). Darüber hinaus muss die Effizienz im Rahmen der gesamten Agentenarchitektur betrachtet werden. Planerkennung muss so effizient sein, dass der zusätzliche Ressourcenverbrauch in Relation zum Wert der neuen Information positiv ist.

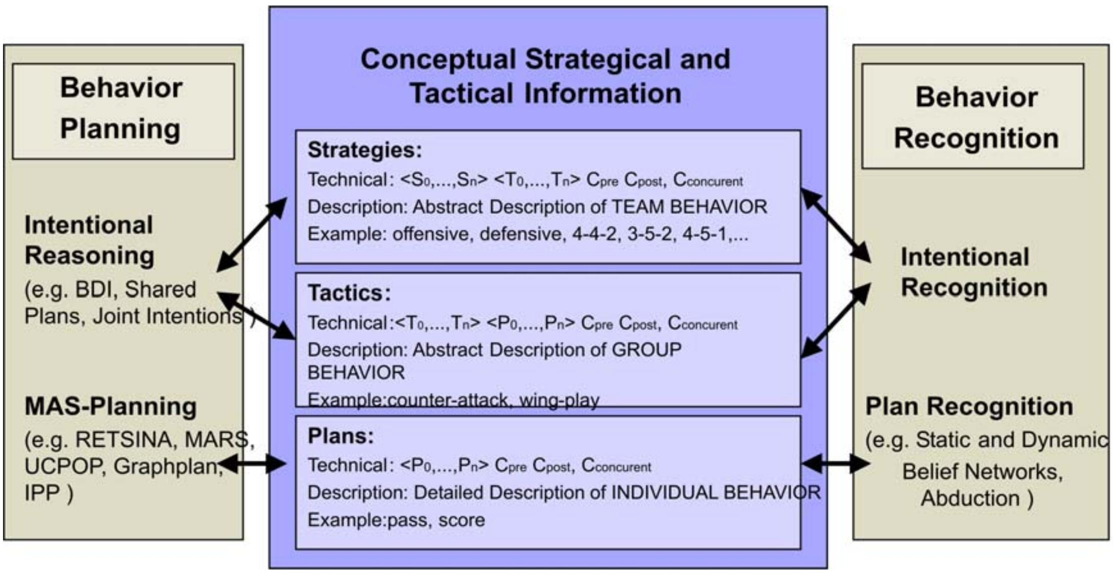


Abbildung 5. Anforderungen an Agentenarchitektur

Um diesen Anforderungen gerecht zu werden muss man Plan- und Intentionserkennung im Kontext der Intentionen und Absichten des Agenten betrachten. In Relation zu den Absichten kann zum einen der Wert einer Information besser bewertet werden, ebenso wie die Granularitätsebene auf der ein Plan oder eine Intention erkannt werden soll. Daraus leiten sich direkt Anforderungen an eine Agentenarchitektur ab (s.Abb. 5). Um keine unzumutbaren Redundanzen zu erzeugen, sollte es ein einziges Verhaltensmodell geben, das sowohl zur Planung als auch zur Planerkennung verwendet werden kann. Ein weiterer Vorteil neben der Vermeidung von Redundanzen ist, dass ein erkannter Plan bzw. eine erkannte Intention immer in der Sprache des Verhaltens eines Agenten beschrieben wird und damit unmittelbarer in den eigenen Handlungsprozess mit einbezogen werden kann. Das heisst auch, dass ein gemeinsames (qualitatives) Verhaltensmodell auf verschieden Abstraktionstufen repräsentiert werden sollte, wenn Informationen gezielt auf verschiedenen Granularitätsstufen erkannt werden soll. In der RoboCup-Fussball-Domäne bietet sich eine Unterteilung in Strategien, Taktiken und einfachen Handlungen wie sie zum natürlichen Sprachgebrauch eines (echten) Trainers gehört (s. Abschnitt 2.1.1).

Abbildung 6. Abhängigkeitszyklus

Eine Konsequenz aus den Effizienzanforderungen ist, dass Plan- und Intentionserkennung zielgerichtet bezogen auf die eigenen Intentionen angewendet werden sollte. In diesem Kontext kann Planerkennung sogar dazu genutzt werden, die Effizienz eines Agenten zu optimieren (neben dem der Verbesserung der Qualität der Handlungsplanung!). Die Absichten eines Agenten können dazu genutzt werden, abzuleiten welche Informationen über den Gegner besonders wichtig sind. Dabei gibt es zwei Klassen von Informationen: (a) (FI in der Abb. 6) Zur Durchführung einer Handlung müssen spezifische Vorbedingungen (in der Welt) erfüllt sein, damit eine Handlung durchgeführt werden kann. Es ist eine sehr hilfreiche Information, wenn von einer Planerkennungskomponente erkannt werden kann, ob die spezifischen Bedingungen noch zum Ausführungszeitpunkt der Handlung erfüllt sein werden. (b) Daneben kann es Bedingungen (z.B. Abseits) geben die in keinem Fall eintreten dürfen, damit eine Handlung erfolgreich durchgeführt werden kann (EOB in der Abb. 6). Wenn eine dieser Bedingungen verletzt wird, kann die Planung und Durchführung einer Handlung abgebrochen werden.

Die Zielgerichtetheit des Planerkennungsprozesses kann dazu verwendet werden die Effizienz z.B. der Bildverarbeitung zu verbessern, indem erkannt wird, dass bestimmte Gegner keinen relevanten Einfluss auf meine Handlungen haben werden, d.h. das auf z.B. auf einige Gegnerlokalisationen verzichtet werden kann.

3 Zusammenfassung

Unser Projekt hat mit einiger Verspätung begonnen. Zu dem Zeitpunkt des Erstellens dieses Zwischenberichtes sind 11 Monate vergangen.

Mit der Gründung des GermanTeams in der Sony Legged Robot League, an der die Universität Bremen initiativ mitgewirkt hat, ist ein schlagkräftiges deutsches Team in dieser Liga aufgebaut worden. Die Softwarearchitektur für dieses Team ist so generisch und modular gehalten, dass verschiedene Implementierungen in einem einzigen System möglich sind. Das erfolgreiche Abschneiden auf der letzten Weltmeisterschaft zeigt, dass die Gründung zum richtigen Zeitpunkt stattgefunden hat und dass die Kooperation mit den Partnern, insbesondere mit der HU Berlin und der TU Darmstadt Früchte trägt. Die Kooperation wird fortgesetzt.

Ein weiteres wichtiges Ergebnis ist die Entwicklung von Sim*GT200x,* eines Simulationswerkzeuges für o.g. Liga. Es ist die erste verfügbare Simulationsumgebung für die Sony Roboter in der RoboCup Community und es ist von verschiedenen Seiten starkes Interesse geäußert worden (z.B. UNSW Australien). Die Arbeiten an diesem Simulator fließen außerdem in die *AG-1 Simulation* des Schwerpunkts ein.

Die Arbeiten an der Erkennung von Aktionen, Aktionsfolgen, Taktiken und Strategien haben gezeigt, dass Plan- und Intentionserkennung in einem umfassenden Kontext betrachtet werden muss. Als Konsequenz wird eine adaptive Agentenarchitektur benötigt, um eine generellere Einsatzfähigkeit z.B. Ligen-übergreifend oder auch für den autonomen Rollstuhl zu gewährleisten. Ausführliche Diskussionen in den Workshops der *AG-3 Architektur* haben verdeutlicht, dass sich strategische und vor allem räumliche Informationen über Weltmodelle Liga-übergreifend gut subsumieren lassen. Gleichzeitig haben wir in Arbeiten zur qualitativen Beschreibung von raumzeitlichen Relationen zwischen Objekten gezeigt, das mit einem qualitativen Ansatz gute Ergebnisse erzielt werden können. Es wurde daher in der AG-3 beschlossen, diesen Weg weiter zu gehen und die Plan- und Intentionserkennung auf qualitativer Ebene voranzutreiben. Die Arbeitspakete in dem Fortsetzungsantrag sind entsprechend formuliert.

Dr. Visser ist Initiator eines international angelegten Workshops, der auf der IJCAI 2003 zum ersten Mal stattfinden soll. Der Titel des Workshops ist „*Issues in Designing Physical Agents for Dynamic Real-Time Environments: World modeling, planning, learning, and communicating.*“ Der Fokus liegt auf der Entwicklung von Technologien, die für Roboter in dynamischen Echtzeitsystemen geeignet sind. Diese Initiative ist auf der letzten Sitzung des SPP in Aachen diskutiert worden und fand die breite Unterstützung. Das Proposal wird zur Zeit begutachtet und mit einer wahrscheinlichen Bestätigung ist Anfang November zu rechnen. Das SPP hat damit eine Chance, auch außerhalb Deutschlands Impulse zu geben. Weiterhin sollte erwähnt werden, dass dieser Workshop kein ausschließlicher RoboCup-Workshop sein wird, sondern dass andere namhafte Wissenschaftler, die nicht der internationalen RoboCup-Gemeinde angehören, ebenfalls beteiligt sind (z.B. Patrick Doherty, Schweden).

Literatur

Allen, J.F., *Maintaining Knowledge about Temporal Intervals.* Communications of the ACM, 1983. **26**(11): p. 832-843.

Boronowsky, M., *Diskretisierung reell-wertiger Attribute mit gemischten kontinuierlichen Gleichverteilungen und ihre Anwendung bei der zeitreihenbasierten Entscheidungsbauminduktion*. Fachbereich Mathematik & Informatik, ed. DISKI. Vol. 246. 2001, Bremen: Aka GmbH, Berlin. 182.

Dietl, M.; Gutmann, J.-S.; Nebel, B. (2002). CS Freiburg: Global View by Cooperative Sensing. In: Birk, A.; Coradeschi, S.; Tadokoro, S. (Eds.), *RoboCup 2001: Robot Soccer World Cup V*. Lecture Notes in Artificial Intelligence 2377. Springer.

Fox D.; Burgard, W.; Dellaert, F.; Thrun, S. (1999). Monte Carlo localization: Efficient position estimation for mobile robots. In: Proc. of the National Conference on Artificial Intelligence.

Frank, I., K. Tanaka-Ishi, K. Arai, and H. Matsubara. *The Statistics Proxy Server*. in *4th International Workshop on RoboCup*. 2000. Melbourne, Australia: Carnegie Mellum University Press.

Lenser, S.; Veloso, M. (2000). Sensor resetting localization for poorly modeled mobile robots. In: Proc. of the IEEE International Conference on Robotics and Automation.

Miene, A. and U. Visser, *Interpretation of spatio-temporal relations in real-time and dynamic environments*, in *RoboCup 2001: Robot Soccer World Cup V*, A. Birk, S. Coradeschi, and S. Tadokoro, Editors. 2002, Springer: Seattle, WA. p. 441-446.

Quinlan, J.R., *C4.5 Programs for Machine Learning*. Morgan Kaufman Series for Machine Learning. 1993: Morgan Kaufmann.

Raines, T., M. Tambe, and S. Marsella, *Automated Assistants to Aid Humans in Understanding Team Behaviors*, in *RoboCup-99: Robot Soccer World Cup III*, M. Veloso, E. Pagallo, and H. Kitano, Editors. 2000, Springer Verlag: Stockholm, Sweden. p. 85-104.

Stone, P. and M. Veloso, *A Layered Approach to Learning Client Behaviors in the RoboCup Soccer Server.* Applied Artificial Intelligence, 1998. **12**(3): p. 165-188.

Stone, P. and M. Veloso, *Towards Collaborative and Adversarial Learning: A Case Study in Robotic Soccer.* International Journal of Human-Computer Studies, 1998. **48**.

Stone, P. and M. Veloso. *Team-Partitioned, Opaque-Transition Reinforcement Learning*. in *Robot Soccer World Cup II*. 1999: Springer Verlag, Berlin.

Stone, P., P. Riley, and M. Veloso, *The CMUnited-99 Champion Simulator Team*, in *RoboCup-99: Robot Soccer World Cup III*, M. Veloso, E. Pagello, and H. Kitano, Editors. 2000, Springer Verlag: Melbourne, AUS. p. 35-48.

Visser, U. and H.-G. Weland. *Using online learning to analyze the opponents behavior*. in *Proceedings of the RoboCup-2002: Robot Soccer World Cup VI*. 2002. Fukuoka, Japan.

Visser, U., C. Drücker, S. Hübner, E. Schmidt, and H.-G. Weland. *Recognizing Formations in Opponent Teams*. in *RoboCup 2000, Robot Soccer World Cup IV*. 2001. Melbourne, Australia: Springer-Verlag.

Weland, H.-G., *Gegneranalyse in dynamischen Echtzeitumgebungen*, in *Fachbereich Mathematik & Informatik*. 2002, Bremen: Bremen. p. 83.

Wünstel, M., D. Polani, T. Uthmann, and J. Perl, *Behavior Classification with Self-Organizing Maps*, in *RoboCup 2000: Robot Soccer World Cup IV*, P. Stone, T. Balch, and G. Kraetschmar, Editors. 2001, Springer: Melbourne, Australia. p. 179-188.

Unterschriften

Bremen, den 1. November 2002

(Ubbo Visser) (Thomas Röfer)

4 Verzeichnis der beigefügten Anlagen

Diesem Zwischenbericht sind folgende Publikationen als Anlagen zum Verbleib bei der DFG beigefügt:

Drücker, C.; Hübner, S.; Schmidt, E.; Visser, U.; Weland, H.-G. (2000). Virtual Werder: Using the Online-Coach to Team Formations. In: Balch, T., Stone, P., Kraetschmar, G. (Eds.): *4th International Workshop on RoboCup.* Carnegie Mellon University Press, Melbourne, Australia, 2000. 217-222.

Drücker, C.; Hübner, S.; Visser, U.; Weland, H.-G. (2002). "As time goes by" – Using time series based decision tree induction to analyze the behaviour of opponent players. In: Birk, A.; Coradeschi, S.; Tadokoro, S. (Eds.), RoboCup 2001: Robot Soccer World Cup V. Lecture Notes in Artificial Intelligence 2377. Springer. 325-330.

Meyer, J.; Adolph, R.; Stephan, D.; Seekamp, M.; Weinert, V.; Visser, U. (2003). Decision-making and Tactical Behavior with Potential Fields. In: RoboCup 2002: Robot Soccer World Cup VI. Lecture Notes in Artificial Intelligence, Springer, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings, 300-307).

Miene, A.; Visser, U. (2002). Interpretation of spatio-temporal relations in real-time and dynamic environments. In: Birk, A.; Coradeschi, S.; Tadokoro, S. (Eds.), RoboCup 2001: Robot Soccer World Cup V. Lecture Notes in Artificial Intelligence 2377. Springer. 441-446.

Röfer, T. (2003). An Architecture for a National RoboCup Team. In: RoboCup 2002: Robot Soccer World Cup VI. Lecture Notes in Artificial Intelligence, Springer, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings, 388-395).

Röfer, T.; Jüngel, M. (2003). Vision-Based Fast and Reactive Monte-Carlo Localization. Eingereicht bei: International Conference on Robotics and Automation 2003 (ICRA-2003).

Visser, U.; Drücker, C.; Hübner, S.; Schmidt, E.; Weland, H.-G. (2001). Recognizing Formations in Opponent Teams. In: Balch, T., Stone, P., Kraetzschmar, G. (Eds.): 4th International Workshop on RoboCup. Carnegie Mellon University Press, Melbourne, Australia, 391-396.

Visser, U.; Weland, H.-G. (2003). Using online learning to analyze the opponents behavior, In: RoboCup 2002: Robot Soccer World Cup VI. Lecture Notes in Artificial Intelligence, Springer, im Erscheinen (bereits veröffentlicht in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings, 72-81).

Drücker, C., Hübner, S., Schmidt, E., Visser, U. and Weland, H.-G. (2000) Virtual Werder: Using the Online-Coach to Change Team Formations, In:Proceedings of 4th International Workshop on RoboCup, (Eds, Tucker Balch, Peter Stone and Kraetschmar, G.), Melbourne, Australia, pp. 217-222.

Virtual Werder: Using the Online-Coach to Change Team Formations

Christian Drücker, Christian Duddeck, Sebastian Hübner, Holger Neumann, Esko Schmidt, Ubbo Visser, Hans-Georg Weland

TZI - Center for Computing Technologies
University of Bremen
D-28334 Bremen, Germany
grp-robocup@tzi.de
WWW home page: http://www.tzi.de/grp/robocup

Abstract. This paper details the Virtual Werder soccer team architecture and describes the characteristic features of the team. Our focus on this matter was determined by the fact that the online coach has become more powerful in the last few years. The opponent's have the option to change tactics at will and therefore, in order to react to these changes and the coach determines the opponent's play system. The coach then provides vital information to our team so it can reorganize itself and choose an effective counter-strategy. All the soccer clients are initialized with a specific behaviour and can change to an appropriate mode depending on the coach's commands. The result is a flexible and effective game played by the eleven soccer clients.

1 Introduction

Over the last few years several attempts have been in learning of team behaviour. These studies were conducted on the teams playing in robot leagues [Stone and Veloso, 1998a], [Stone and Veloso, 1998b], [Stone and Veloso, 1999] and [Stone, 2000]. Raines et al. [Raines et al., 2000] describe a new approach to automate assistants to aid humans in understanding team behaviours for the simulation league. This approach was designed for the analysis of the games, off-line after playing.

While the coach-client has been able to participate through analyzing and control in real matches since 1998 [Corten et al., 1999], the idea of general strategic planning becomes possible. This is the focus point where our idea begin. In real life matches, the coach gives out strategic commands depending on the opponent's strategy, play system, and the current score. Our hypothesis includes the diagnosis of the opponent's strategies and the generation of appropriate moves, which improves the performance of our team.

Our team is based on the sources that were released by the CMUnited99 team [Stone et al., 2000] in place of the reinvention of all the basic skills. Our plan is to use part of the provided functions from the CMU-client to construct a more sophisticated team with individual players. One long-term goal is to use communication among players to execute trained moves. With our other focus being the coach's guidance in a game. While conducting our research for this project we got support from real-life soccer experts, including Thomas Schaaf, the coach of the Werder Bremen 1. Bundesliga team [Timm et al., 1999].

2 Agent Behaviour

The Virtual Werder team consists of individual players which have different behaviours. We created 22 types of players with a variety of characteristics to ensure the flexibility and variability of actions and reactions within a game. There are six different types of forwards, seven different players for both the defense and mid-field, and two types of goalies (figure 1). Therefore, our clients have the option to change their behaviour from one style to another at any stage of the game. This concept is an important key feature to carry out changes in a formation while playing against teams that are switching between different play systems. We plan to use an online-coach so that our clients have the ability to receive the messages, parse them, and change their behaviour accordingly.

On the other hand, the low-level skills of the agents are based on the sources provided by CMUnited99 (see [CMU, 2000]). These skills include functions to locate the ball, the other team members and the opponent players. In addition, methods to communicate via UDP-sockets, a parser for soccerserver messages and other utility functionalities, such as the memory structures, have been utilized.

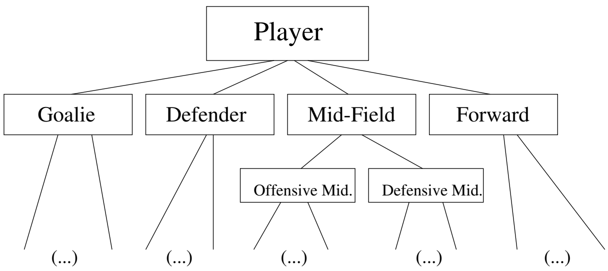


Fig. 1. Players hierarchy

Our clients are initialized with a certain behaviour and the desired formation when connecting to the soccerserver. Furthermore, they have the ability to choose other behaviours and switch to them immediately. These high-level functions include methods to carry out different defense systems such as man-to-man marking and zone defense.

Also, we also included new mid-level functions, e. g. finding a teammate able to catch a pass. Another new skill includes the player moving to a certain point on the field, while keeping track of the ball. This is accomplished through the *turn_neck* command.

In the following section we will describe the characteristics and maneuvers of some of the 22 different team players.

2.1 Player

The basic player searches for the ball and attempts to put it into play. When the ball is within the kickable area the player will attempt to obtain a goal. All players inherit this basic behaviour (figure 1).

Goalie: There are two types of goalies which contain additional characteristics of the basic players. The first goalie remains in the goal area and attempts to catches an incoming goal. The second can also dash forward to obtain the ball when it comes into the penalty area. When the ball has been intercepted both goalies attempt to pass it to an appropriate team member.

Defender: All defenders have a control area, they are responsible in preventing opponents to dribble or pass the ball throughout this area. Normally, the control area consists of a circle surrounding the players starting position however, if the ball is behind them they are able to move their control area backwards. In addition to the different starting positions and control areas, the defenders differ in behaviour, if the ball is played in another area of the field. Some players will mark an opponent forward, while others try to reach a good position to prepare for the next attack. The priority of all defenders is to keep the ball away from their goal. If the ball is kickable, they look for a possibility to pass to a teammate. Otherwise, they can kick the ball further towards the opponent's goal or the nearest sideline.

Mid-Field: The mid-field player are split in the defensive and offensive behaviours. They also differ in their priority of behaviours. The main goal of the defensive mid-field is to intercept the ball from the opponent. The goal of the offensive mid-field is to bring the ball to the forwards and/or score a goal. Although the mid-field player can kick risky passes, they can also carry the ball if there is no other alternative.

Forward: The main objective of the forwards is to score goals. For this reason it is more important for them to get in a position to score than to intercept the ball from the opponent. In order to obtain a good position for receiving a pass they evaluate ten random positions and move to the best possibility.

2.2 Coach

The coach observes the game continually and evaluates the formation of the opponents team at given points in time. The current evaluation takes place twice per second. The position of the players serve as inputs for a neural network, that is trained with the formations most commonly played in our test games and the log-files. Whenever the play mode switches to another state then *PLAY_ON*, the coach generates a message for his team. He tells the players which formation the opponent is currently using giving the information to choose a counter attack.

3 Online-Coach Tasks

We observed that teams in the last RoboCup-tournaments typically relied on their strategy and team formation and often didn't change them within a game. When changes were made they were depended on the score. A common practice was to switch from a offensive to a defensive formation if the team lead with more than three goals. On the other hand some teams remained on the same system regardless of the opponent's strategy.

3.1 Approach

This is the point where the online-coach diagnoses what to do depending on the opponent's formation [Biermann and Fuchs, 1999]. The coach-client had to be fed with formations and information on how to analyse them. Part of the integrated knowledge was obtained from a real expert, the Werder Bremen's head coach Thomas Schaaf [Timm et al., 1999], other parts from literature, e. g. [Biermann and Fuchs, 1999]. Another significant contribution of our knowledge was acquired from games played in the last tournaments. Meaning, we analyzed the appropriate log-files and learned common strategies. These strategies can be used by the online-coach within the decision process. Furthermore, the coach is able to keep track of formations and therefore predicts fitting counter attacks. The coach is only able to broadcasts its message during a break in the game.

3.2 Internals

The observation and analysis of the opponents team is processed in several steps, which are described below. The CMUnited99 sources provide the communication with the soccerserver and the parsing of messages. Furthermore, it supplies the coach with collected information in data structures, which are easily accessible.

It is the task of our high-level functionalities to prepare the data and provide it to an artificial neural network. For this purpose a bounding box is placed around the positions of the opponent players. This box is currently divided into a grid of eight by eight cells, which leads to an arrangement of 64 fields (figure 2). There must by at least one player inside a specific field with the value of this field set to 1.0. Otherwise, the value must by set at 0.0. The sum of the fields defines the input vector for the neural network.

The network in the form of a C-function[1] is now called to calculate the ratings at 16 possible output neurons, which represent the trained opponent's

[1] snns2c by Bernward Kett was used to transform the trained network from an internal SNNS representation to a handy usable C function.

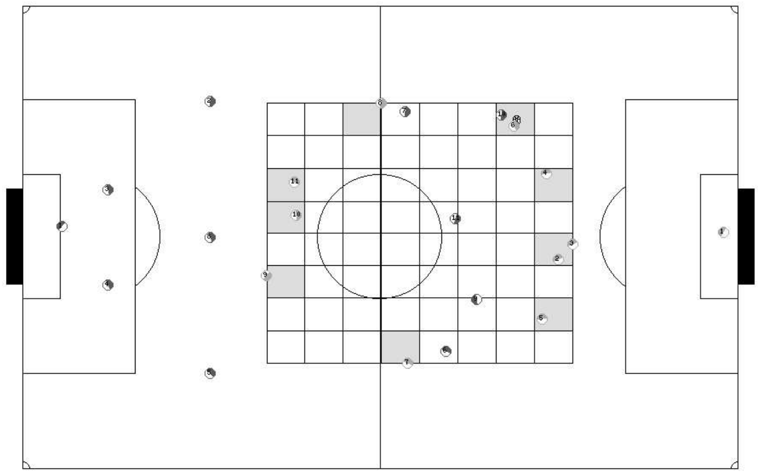


Fig. 2. Positions of opponent players and bounding box; marked cells define the input vector for the neural network.

formations (section 3.3). In the case that the best rated output neuron exceeds a demanded threshold, it is chosen as the result of this function. This together with other information, e. g. the size and position of the bounding box or the current score of the game, results in the determination of the chosen necessary formations. This in turn is memorized by the coach. This information can only be broadcasted during a break within a game. In this case the lastest stored information is formulated as a message and sent immediately to every team member, which eventually results in the players swapping of positions and/or formation.

3.3 Training results

Our coach uses an artificial neural network [Hertz et al., 1991], [Rojas, 1996], to analyze the formation of the opponent's team. The net itself was trained to recognize sixteen different formations. The coach client uses this knowledge about the classified formation to evaluate a proper counter attack.

Soccer formations are typically noted as a combination of defense, mid-field and forward players, e. g. a 5-3-2 represents a team with five defense players (and a goalie), three players in the middle field and two forward players. However, there are some special systems that do not fit into this pattern, these are referred to by name, e. g. the Catenaccio system [Biermann and Fuchs, 1999].

We developed a tool called "ExportPlayer" to obtain formation examples for our network. This program is based on the log-player together with the soccerserver. We extended the X11-interface and added buttons for each of the classified sixteen play systems. The ExportPlayer takes automatic snapshots of existing log-files. It extracts the positions of ten players (not including the goalie)

threshold	classified	classified correctly
0%	100%	48.37%
80%	49.67%	65.53%
85%	41.34%	67.88%
90%	32.52%	69.30%
95%	20.10%	72.27%

Table 1. Correlation between output threshold and correctness

and normalizes the coordinates to a grid of eight by eight cells. Cells labeled '0' do not contain players, cells labeled '1' contain at least one player.

In order to export formations from a log-file, the button corresponding to the formation seen on the soccer-monitor has to be selected. The ExportPlayer jumps to a new position in the log-file until enough samples are collected. The ExportPlayer then generates a pattern file. This pattern file contains the values of all input and output neurons and serves a neural network. We used the Stuttgart Neural Network Simulator (SNNSv4.1) [Zell et al., 1994] to train the examples and to create the code for a MLP-network.

The choice threshold mentioned is section 3.2 is a very important factor for the efficiency of the online-coach. Table 1 shows the correlation between different thresholds, which results in the amount of permitted input patterns and their correctness. The inquiry is based on 680 patterns obtained from log-files and test games. These have been previously classified by us. On ten separate occasions these patterns were randomly divided into 612 training and 68 test units (representing 90% and 10% of the test field), which were processed by the net. The average of these ten different results have been calculated for validity.

Due to the large quantity of test patterns (currently twice per second a snapshot is made), the relatively high amount of rejections is not problematic in this environment. Furthermore, it is similar to real soccer game where distinct formation occurs infrequently.

3.4 Counter Formations

While coaches normally do not change the teams strategy in every play cycle, the fact that the coach broadcasts messages only during a break in play is not considered a handicap. Furthermore, this situation is more comparable to a real-life soccer game, where the coach is also limited to break in play commands. As explained previously it is possible to recognize formations of the opponents. [Biermann and Fuchs, 1999] describes several formations, which can be recognized by the on-line coach.

It is essential to split the opponent's formation in the defense, mid-field and offensive modes so that we can receive a counter formation.

One of the main goals in playing soccer is preventing the opponent from scoring. This would be less complicated if our team is able to keep their opponents in a smaller area of the playing field. While we must score to win the match, our offence must maximize the field by playing the line and passing into the penalty area. The mid-field players should assist the offence in these maneuvers.

Our mid-field players assist both the defense and offense players. When this causes problems the offence should drop back when the opponent attacks allowing the defense to move up during a forward maneuver. In this section we will describe some known formations (see [Biermann and Fuchs, 1999]) and describe counter formations.

2–3–5 Brasil 1958: In this formation the opponent is able to act within two attack lines, while the mid-field is able to assist the outer positions. These three players that comprise the defense are assisted by the central mid-field player.

The defense is a chain of four players. In front of these players three mid-field players operate another chain. The remaining three players build a triangle: one player takes the defensive role, in the middle of the field; another takes the position near one corner of the penalty area and the third operates near the penalty point.

Classical 4–3–3: This classical formation consists of two outer defensive players and two central defensive players. Three of the players operate in a chain while the last acts behind or in front of the chain. In the modern version four defense players operate in this chain. Both the forward and mid-field section consists of three players in a chain, and the mid-field players can assists the offense or defense.

On our team, three players build the defense chain. The chain of four mid-field players are once again able to assist the offense or the defense. The offense consists of three players in the same way as described in section 3.4.

Catenaccio: This defensive formation consists of five players with two players positioned in the middle, providing assistance to the mid-field players. The proper mid-field consists of two players in outer positions. One offensive player is in the middle, who is assisted by two players near the lines.

Our defensive formation consists of three players operating in a chain. In front of two players one operates in the central mid-field (slightly behind the two outer mid-field players). The three remaining players construct the offensive, two of these players operate slightly behind the middle offensive player.

4 Results

Our hypothesis was that we improve the performance of our team by learning knowledge about opponents strategies and obtaining the appropriate counter formation. In this section we show that this hypothesis was true. We describe how important the play system can be in a soccer game.

Our test environment consists of our team and the teams of CMUnited99 and last years Mainz Rolling Brains. We carried out ten games against each team with two different play systems. We decided the formations 5-4-1 (defensive) and 3-4-3 (offensive) to be the most promising for our demonstration.

If playing against the CMUnited99 team Virtual Werder is performing better with a defense formation. The average loss against the CMU-team was 0:14 with the 3-4-3 and 0.1:9 with the defensive strategy (we might add that this is an unacceptable situation in total). Table 2 shows the average results.

When playing against the Main Rolling Brains on the other hand we can see that Virtual Werder performs better with the offensive formation. With an average score of 3.1:0.7 it was better than the 0.5:0.9 score when carrying out the defense formation. We believe that the understaffed mid-field caused this situation (see figure 2).

VW (3-4-3) vs. MRB	**VW (3-4-3) vs. CMU99**
3.1:0.7	0:14
VW (5-4-1) vs. MRB	**VW (5-4-1) vs. CMU99**
0.5:0.9	0.1:9

Table 2. Relation between formation and score

We come to the conclusion that the online-coach can help to 'learn' the opponents strategy. Once a team knows the play system of the opponent, appropriate counter actions can be conducted. We have seen that the Virtual Werder team performs better with this new information. The average score depends heavily upon the chosen play system and whether the team can change their system online.

5 Future Work

Further work needs to be done in the following areas:

– **Keeping track of changes:** This means that the coach-client consists of internal states. With internal states a list of 'scenes' describing the current play system of the opponent can be stored. The next step is to detect when a play system changes. A low pass filter can then be used to determine whether the play system changed temporary or for a longer period.

– **Evaluation of counter attacks:** The evaluation of a fitting counter attack is another issue in our research. Therefore, we will focus on new criteria, such as play cycle and score. The idea is to change formations in situations that do not depend on the opponent's play. An example might be a game with only a few cycles left and 0-1 score. The coach could use the play cycle and score information to change to an offensive formation hoping to even the game.

– **Captain:** Looking at formations is a first step to a more strategic play. The next step to improve the team performance will be the transfer of the coach knowledge to a key-player, which can give commands to the team members during the game, not only within a break. This "captain"-concept could also be extended with the concept of a key defense player, which is responsible for the guidance of the defense.

– **Symbolic learning:** Another idea is to use a symbolic learn algorithm, e. g. C4.5 [Quinlan, 1993], to obtain rules from the collected patterns. These rules can serve a knowledge base, which can be integrated in each player or at least in the key player. The advantage will be the comprehensibility of the rules in comparison to the neural network.

References

[Biermann and Fuchs, 1999] Biermann, C. and Fuchs, U. (1999). *Der Ball ist rund, damit das Spiel die Richtung ändern kann.* Kiepenheuer & Wisch.

[CMU, 2000] CMU (2000). Sources of low-level skills. http://www.cs.cmu.edu/afs/cs/usr/pstone/mosaic/RoboCup/CMUnited99-sim.html.

[Corten et al., 1999] Corten, E., Dorer, K., Heintz, F., Kostiadis, K., Kummeneje, J., Myritz, H., Noda, I., Riekki, J., Riley, R., Stone, P., and Yeap, T. (1999). Soccerserver manual ver5.1 release. Manual.

[Hertz et al., 1991] Hertz, J., Krogh, A., and Palmer, R. (1991). *Introduction to the theory of neural computation*, volume 1. Addison-Wesley Publishing Company, Redwood City, California.

[Kitano et al., 1997] Kitano, H., Kuniyoshi, Y., Noda, I., Asada, M., Matsubara, H., and Osawa, E. (1997). Robocup: A challenge problem for ai. *Artificial Intelligence Magazine*, 18(1):73–85.

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: programs for machine learning.* Morgan Kaufman, San Mateo CA.

[Raines et al., 2000] Raines, T., Tambe, M., and Marsella, S. (2000). Automated assistants to aid humans in understanding team behaviors. In *Fourth International Conference on Autonomous Agents (Agents 2000)*, Barcelona, Spain.

[Rojas, 1996] Rojas, R. (1996). *Neural Networks - A Systematic Introduction.* Springer-Verlag, Berlin.

[Stone et al., 2000] Stone, P., Riley, P., and Veloso, M. (2000). The CMUnited-99 champion simulator team. In Veloso, M., Pagello, E., and Kitano, H., editors, *RoboCup-99: Robot Soccer World Cup III*, Berlin. Springer Verlag.

[Stone and Veloso, 1998a] Stone, P. and Veloso, M. (1998a). A layered approach to learning client behaviors in the robocup soccer server. *Applied Artificial Intelligence*, 12(3):165–188.

[Stone and Veloso, 1998b] Stone, P. and Veloso, M. (1998b). Towards collaborative and adversarial learning: A case study in robotic soccer. *International Journal of Human-Computer Studies*, 48.

[Stone and Veloso, 1999] Stone, P. and Veloso, M. (1999). Team-partitioned, opaque-transition reinforcement learning. In Asada, M. and Kitano, H., editors, *Robot Soccer World Cup II*. Springer Verlag, Berlin.

[Stone, 2000] Stone, P. (2000). *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*. Intelligent Robotics and Autonomous Agents. MIT Press, Cambridge, Massachusetts.

[Timm et al., 1999] Timm, I., Herzog, O., and Visser, U. (1999). Fuballstrategien. Interview with Thomas Schaaf, Coach of Werder Bremen.

[Zell et al., 1994] Zell, A., Mamier, G., Vogt, M., Mache, N., Hubner, R., Herrmann, K., Soyez, T., Schmalzl, M., Sommer, T., Hatzigeorgiou, A., Döring, S., and Posselt, D. (1994). Snns: Stuttgart neural network simulator, user manual version 3.2. Technical Report 3/94, Universitat Stuttgart.

Visser, U., Drücker, C., Hübner, S., Schmidt, E. and Weland, H.-G. (2001) Recognizing Formations in Opponent Teams, In:Proceedings of RoboCup 2000, Robot Soccer World Cup IV, Vol. 2019, (Eds, Stone, P., Balch, T. and Kraetschmar, G.), Melbourne, Australia, pp. 391 - 396.

Recognizing Formations in Opponent Teams

Ubbo Visser, Christian Drücker, Sebastian Hübner,
Esko Schmidt, Hans-Georg Weland

TZI - Center for Computing Technologies
University of Bremen
D-28334 Bremen, Germany
{visser|druecker|huebner|esko|weland}@tzi.de
WWW home page: http://www.virtualwerder.de

Abstract. The online coach within the simulation league has become more powerful over the last few years. Therefore, new options with regard to the recognition of the opponents strategy are possible. For example, the online coach is the only player who gets the information of all the objects on the field. This leads to the idea determine the opponents play system by the online coach and then choose an effective counter-strategy. This has been done with the help of an artificial neural network and will be discussed in this paper. All soccer-clients are initialized with a specific behavior and can change their behavior to an appropriate mode depending on the coach's commands. The result is a flexible and effective game played by the eleven soccer-clients.

1 Introduction

Our team is based on the sources that were released by the CMUnited99 team [10]. We decided to do so because it would have been to time consuming to reinvent all basic skills.[1] Instead, we focus on research w.r.t. high level functions which will hopefully lead to new ideas and results for the RoboCup community. Our long run plan is to use part of the provided functions from the CMU-client to construct a more sophisticated team with individual players.

Over the last few years several attempts have been made in learning of team behaviour. Similar approaches have been developed from numerous research groups. These studies have the focus on learning team behavior within the simulation and middle size league (see [11], [12], [13], [9]). Raines et al. [7], e.g., describe a new approach to automate assistants to aid humans in understanding team behaviours for the simulation league. This approaches are designed for the analysis of games, off-line after playing, to gain new experiences for the next games. Frank et al. ([4]) present a real time approach which is based on statistical methods. A team will be evaluated statistically but there is no recognition of team strategies.

While conducting our research for this project we obtained support from real-life soccer experts. In an interview, Thomas Schaaf, the manager of SV Werder Bremen pointed out the importance of the strategy recognition of the opponent team. While the

[1] We would like to give special thanks to the original authors

coach-client has been able to participate through analysis and control in real matches since 1998 [3], the idea of general strategic planning becomes possible. Like in real life matches, the coach is able to give strategic commands depending on the opponent's system and the current score. We presume that the performance of our team can be improved by analyzing the opponent's strategy.

2 Agents

The Virtual Werder team consists of individual players which have different behaviors.

Players: 22 types of players have been developed with a variety of characteristics to ensure the flexibility and variability of actions and reactions within a game. There are different types of forwards, defenders, mid-fielders and goal keepers. Therefore, our clients have the option to change their behavior from one style to another at any stage of the game. This concept is an important key feature to carry out changes in a formation while playing against teams that are switching between different play systems. We plan to use the online-coach, so that our clients have the ability to receive the messages, parse them, and change their behavior accordingly. On the other hand, the low-level skills of the agents are based on the sources provided by CMUnited99 [2]. These skills include functions to locate the ball, the other team members and the opponent players. In addition, methods to communicate via UDP-sockets, a parser for soccer server messages and other utilities such as the memory structures, have been used. Our clients are initialized with a certain behavior and the desired formation when connecting to the soccer server. Furthermore, they have the ability to choose other behaviors and switch to them immediately. These high-level functions include methods to carry out different defense systems such as man-to-man marking and zone defense. We also included new mid-level functions, e. g. finding a teammate able to catch a pass. Another new skill includes the player moving to a certain point on the field, while keeping track of the ball. This is accomplished through the *turn_neck* command. The characteristics of these 22 types of players are described in the Virtual Werder team description.

Coach: The coach observes the game continually, analyzes the formation of the opponent team at given points in time with an artificial neural network (ANN) ([5], [8]) and broadcasts an adequate counter formation to the players during the next interruption. The current evaluation takes place twice per second. The positions of the players serve as inputs for an ANN, that is trained with the formations most commonly played in our test games and the log-files (see section 3). In order to get a reliable impression only outputs with high ratings are used. Whenever the play mode switches to another state than *PLAY_ON*, the coach generates a message for his team. It instructs the players which formation the opponent is currently using and gives information about the appropriate counter attack.

3 Approach

We observed that teams in the last RoboCup-tournaments typically relied on their strategy and team formation and often didn't change it within a game. When changes were

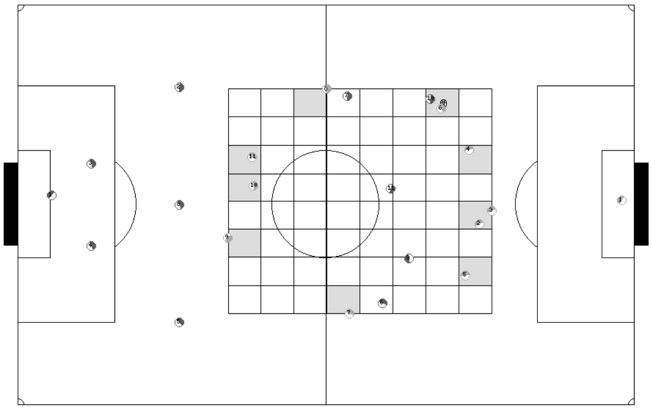


Fig. 1. Positions of opponent players and bounding box; marked cells define the input vector for the neural network.

made they depended on the score. A common practice was to switch from a offensive to a defensive formation if the team lead with more than *n* goals. On the other hand some teams remained on the same system regardless of the opponent's strategy. This is the point where the online-coach diagnoses what to do depending on the opponent's formation. The coach-client had to be fed with formations and information on how to analyze them. Part of the integrated knowledge was obtained from an interview with a real expert, the Werder Bremen's head coach Thomas Schaaf, other parts from literature, e. g. [1] and from games played in the last RoboCup tournaments. This knowledge can be used by the online-coach within the decision process.

The observation and analysis of the opponents team is processed in several steps which are described below. The CMUnited99 sources provide the communication with the soccer server and the parsing of messages. Furthermore, it supplies the coach with collected information in data structures which are easily accessible. This information is then prepared and will act as an input vector for an ANN. In order to prepare the data we first have to decide which variables should be used. Our model consists of 64 binary input variables. For this purpose a bounding box is placed around the positions of the opponent players. The box is currently divided into a grid of eight by eight cells which leads to an arrangement of 64 fields (Fig. 1). There must be at least one player inside a specific field with the value of this field set to 1. Otherwise, the value must be set to 0. The sum of the fields defines the input vector for the ANN. The network classifies the vector in one of 16 output classes, each representing a specific play system.

The network is implemented as a C-function[2] and is called to calculate the ratings at 16 possible output neurons which represent the opponent's formations (section 2). If the output neuron with the highest value exceeds a demanded threshold, this class will be chosen as the result of this function. This result and other information in addition, e. g. the size and position of the bounding box or the current score of the game, deter-

[2] snns2c by Bernward Kett was used to transform the trained network from an internal SNNS representation to a usable C-function [14].

mines the appropriate counter formation. During the next interruption of the game, this information is broadcasted to the team.

Training results: Our coach uses an ANN to analyze the formation of the opponent's team. The network itself was trained to recognize 16 different formations. The coach-client uses this knowledge about the classified formation to evaluate a proper counter-attack. Soccer formations are typically noted as a combination of defense, mid-field and forward players, e. g. a 5-3-2 represents a team with five defense players (and a goalie), three players in the middle field and two forward players. However, there are some special systems that do not fit into this pattern, these are referred to by name, e. g. the Catenaccio system [1].

We developed a tool called “ExportPlayer” to obtain formation examples for our network. This program is based on the log-player and takes automatic snapshots of existing log-files. It extracts the positions of the players (not including the goal keeper) and normalizes the coordinates to a grid of eight by eight cells. Cells labeled '0' do not contain players, cells labeled '1' contain at least one player. The ExportPlayer returns a pattern file which contains the values of all input and output neurons to serve the ANN (see also team description). We used the Stuttgart Neural Network Simulator (SNNSv4.1) [14] to train the examples and to create the code for a feedforward-network. We used standard backpropagation as learning method with the learning parameter $\eta = 1.0$ and the maximum difference $d_{max} = 0.3$ between the teaching value and the output.

The choice of the threshold mentioned in section 3 is a very important factor for the efficiency of the online-coach. Table 1 shows the correlation between different thresholds which results in the amount of permitted input patterns and their correctness. The inquiry is based on 680 patterns obtained from log-files and test games. These have been previously classified by us. On ten separate occasions, these patterns were randomly divided into 612 training and 68 test sets ($\approx$ 10%) and were processed by the ne. The average of these ten different results have been calculated for validity.

Due to the large quantity of test patterns (currently, a snapshot is made twice per second), the relatively high amount of rejections is not problematic in this environment. Furthermore, it is similar to a real soccer game where distinct formation occurs infrequently.

threshold	classified	classified correctly
0%	100%	48.37%
80%	49.67%	65.53%
85%	41.34%	67.88%
90%	32.52%	69.30%
95%	20.10%	72.27%

Table 1. Relation between output threshold and correctness

4 Results

Our hypothesis was that we improve the performance of our team by detecting opponents strategies and obtaining the appropriate counter formation. Our test environment consists of our team and the teams of CMUnited99 and last years' Mainz Rolling Brains. We carried out ten games against each team with two different play systems. We decided that the formations 5-4-1 (defensive) and 3-4-3 (offensive) were the most promising for our demonstration.

Against CMUnited99 Virtual Werder performs better with a defensive formation. The average loss against the CMU-team was 0:14 with the 3-4-3 and 0.1:9 with the defensive strategy (we might add that this is an unacceptable situation in total). Table 2 shows the average results. Against Mainz Rolling Brains on the other hand we can see that Virtual Werder performs better with the offensive formation. The average score of 3.1:0.7 was better than the 0.5:0.9 score with the defensive formation. We believe that the understaffed mid-field caused this situation (see table 2).

We come to the conclusion that the online-coach can help to detect the opponent's strategy. Once a team knows the play system of the opponent, appropriate counter actions can be carried out. However, later experiments have shown that we cannot exclude that the score is caused by other skills such as the individual play style. We think that further investigations with a 'standard team' would be helpful to make a clear point on this issue. In summary, we have seen that the Virtual Werder team performs better with this new information. The average score depends upon the chosen play system and whether the team can change their system online.

	Mainz RB	CMU-99
VW Def. 5-4-1	0.5:0.9	0.1:9
VW Off. 3-4-3	3.1:0.7	0:14

Table 2. Relation between formation and score

The technology of strategy detection could be useful for other application areas. Firstly, the quality of action predictions of physical agents can be improved which plays an important role within the control mechanisms of autonomous agents. Secondly, it is important to improve the robustness and security issues of electronic markets within the area of electronic commerce.

5 Future Work

Further work can be done in the following areas:

Keeping track of changes: This means that the coach-client consists of internal states. With internal states a list of 'scenes' describing the current play system of the opponent's can be stored. The next step is to detect changes in the opponent strategy. A low pass filter can then be used to determine whether the play system changed temporary or for a longer period.

Evaluation of counter attacks: The evaluation of a fitting counter-attack is another issue in our research. Therefore, we will focus on new criteria, such as play cycle and score. The idea is to change formations in situations that do not depend on the opponent's play.

Captain: Looking at formations is a first step to a more strategic play. The next step to improve the team performance will be the transfer of the coach knowledge to a key-player, which can give commands to the team members during the game, not only within a break. This "captain"-concept could also be extended with the concept of a key defense player, which is responsible for the guidance of the defense.

References

1. Christoph Biermann and Ulrich Fuchs. *Der Ball ist rund, damit das Spiel die Richtung ändern kann*. Kiepenheuer & Wisch, 1999.
2. CMU. Sources of low-level skills. http://www.cs.cmu.edu/afs/cs/usr/pstone/mosaic/RoboCup/CMUnited99-sim.html.
3. Emiel Corten, Klaus Dorer, Fredrik Heintz, Kosta Kostiadis, Johan Kummeneje, Helmut Myritz, Itsuki Noda, Jukka Riekki, Ratrick Riley, Peter Stone, and Tralvey Yeap. Soccerserver manual ver5.1 release. Manual, 1999.
4. Ian Frank, Kumiko Tanaka-Ishi, Katsuto Arai, and Hitoshi Matsubara. The statistics proxy server. In Tucker Balch, Peter Stone, and Gerhard Kraetschmar, editors, *4th International Workshop on RoboCup*, pages 199–204, Melbourne, Australia, 2000. Carnegie Mellum University Press.
5. J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the theory of neural computation*, volume 1. Addison-Wesley Publishing Company, Redwood City, California, 1991.
6. H. Kitano, Y. Kuniyoshi, I. Noda, M. Asada, H. Matsubara, and E. Osawa. Robocup: A challenge problem for ai. *Artificial Intelligence Magazine*, 18(1):73–85, 1997.
7. Taylor Raines, Millind Tambe, and Stacy Marsella. Automated assistants to aid humans in understanding team behaviors. In *Fourth International Conference on Autonomous Agents (Agents 2000)*, Barcelona, Spain, 2000.
8. Raul Rojas. *Neural Networks - A Systematic Introduction*. Springer-Verlag, Berlin, 1996.
9. Peter Stone. *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*. Intelligent Robotics and Autonomous Agents. MIT Press, Cambridge, Massachusetts, 2000.
10. Peter Stone, Patrick Riley, and Manuela Veloso. The CMUnited-99 champion simulator team. In Manuela Veloso, Enrico Pagello, and Hiroaki Kitano, editors, *RoboCup-99: Robot Soccer World Cup III*, Berlin, 2000. Springer-Verlag.
11. Peter Stone and Manuela Veloso. A layered approach to learning client behaviors in the robocup soccer server. *Applied Artificial Intelligence*, 12(3):165–188, 1998.
12. Peter Stone and Manuela Veloso. Towards collaborative and adversarial learning: A case study in robotic soccer. *International Journal of Human-Computer Studies*, 48, 1998.
13. Peter Stone and Manuela Veloso. Team-partitioned, opaque-transition reinforcement learning. In M. Asada and H. Kitano, editors, *Robot Soccer World Cup II*, Berlin, 1999. Springer-Verlag.
14. A. Zell, G. Mamier, M. Vogt, N. Mache, R. Hubner, K.U. Herrmann, T. Soyez, M. Schmalzl, T. Sommer, A. Hatzigeorgiou, S. Döring, and D. Posselt. Snns: Stuttgart neural network simulator, user manual version 3.2. Technical Report 3/94, Universität Stuttgart, 1994.

Drücker, C., Hübner, S., Visser, U. and Weland, H.-G. (2002) "As time goes by" - Using time series based decision tree induction to analyze the behaviour of opponent players. In: RoboCup 2001: Robot Soccer World Cup V, Vol. 2377 (Eds, Birk, A., Coradeschi, S. and Tadokoro, S.) Springer Verlag, Seattle, WA, pp. 325-330.

"As time goes by" - Using time series based decision tree induction to analyze the behaviour of opponent players

Christian Drücker, Sebastian Hübner,
Ubbo Visser, Hans-Georg Weland

TZI - Center for Computing Technologies
University of Bremen
D-28334 Bremen, Germany
grp-rcintern@tzi.de
WWW home page: http://www.virtualwerder.de/

Abstract. With the more sophisticated abilities of teams within the simulation league high level online functions become more and more attractive. Last year we proposed an approach to recognize the opponents strategy and developed the online coach accordingly. The coach was able to detect their strategy and then passed this information together with appropriate countermeasures to his team. However, this approach gives only information about the entire team and is not able to detect significant situations (e.g. double pass, standard situations, repeated patterns between two or three players). In this paper we describe a new method for time series which is able to describe the time series by qualitative abstraction and produces samples which then can be used for inductive learning methods such as decision trees.

1 Introduction

As the RoboCup 2000 in Melbourne showed, the differences between the technical abilities of the major teams aren't as big as in previous world-cups. Due to teams which share their progress with the RoboCup community by releasing their source code shortly after the event - a big thanks to CMU at this point -, every team is able to easily arrange eleven pretty good agents. Future teams will have to develop a superior team behaviour to win a tournament. This means a big step forward in terms of multiagent-system research.

Version 7.0 of the soccer server extended the abilities of the online coach further, so that its use becomes even more interesting. It is still the most effective instrument to analyze the opponent, because it possesses all the information about the simulated environment. Therefore, it was important for us to continue the development of the online coach which we used at the RoboCup 2000. In [Visser et al., 2001] we describe how the old coach determines the opponent tactical formation with a neural network and how it is able to change our team formation during a match.

This year our main target is the collection of more information about the opposing team. On one hand we want to recognize more aspects of the opponents global tactics. Beside of their formations, we are interested in their defense behaviour (e.g. the use of offside traps; cover or zone defense) and their preferred offensive plays (e.g. wing play or center breakthroughs; quick passing or long dribbling), but also in their ability to cope with such behaviours when they are used by our team.

On the other hand we want to analyze an individual opponent agent to detect which task in their system it fulfills. To determine this, we have to observe how a certain player reacts to certain situations. Under which circumstances does an agent pass the ball to a fellow player? Why does it pass the ball in this situation to just this player? When does an opponent player shoot at our goal? When does a defense player attack our ball carrier?

If we know how an opponent agent reacts in a certain situation, we can use this information to be always one step ahead of the opponent. For example, if we realize that the opponent forwards only try to score if they have enough space (as it seemed to be performed by the Brainstormers agents), we could prevent them from shooting at the goal by placing our defenders close enough to them.

Similar work has been done over the last couple of years by [Wünstel et al., 2000], [Frank et al., 2000], and [Raines et al., 1999].

Our objective is to find a method that can improve the behaviour of our team during the game and that not only shows us how we could do better next time. Every year new teams debut in RoboCup whose agents differ in their behaviours from all that existed before. Therefore, off-line analysis can prepare our team for existing agents only, whose binaries have been released during the year. Unfortunately only a few teams did that early enough in the past. But even if all would, we couldn't assume that our opponents didn't made major changes in their behaviour.

2 Qualitative abstraction of time series

A new method for the qualitative abstraction of multiple time series has been developed at the Center for Computing Technologies [Boronowsky, 2001]. This method seems especially suitable to us in order to deliver patterns which we then use as a basis for learning algortihms. The method has been developed to analyze continuous valued time series. It uses decision tree induction and therefore generates rules about the analyzed time series, in our case the observed players. The special feature of this method is the way the continuous valued attributes are discretized.

Certain rules, generated by the decision tree, can be evaluated automatically by our coach. This information can be used to improve our agents' behaviour in the running game. Additionally, all rules can can be evaluated by hand after the game has ended. The results of the evaluation by hand can be used to improve our agents' basic behaviour. A big advantage of the used method is, that the discretization of continuous valued attributes is done automatically by our system.

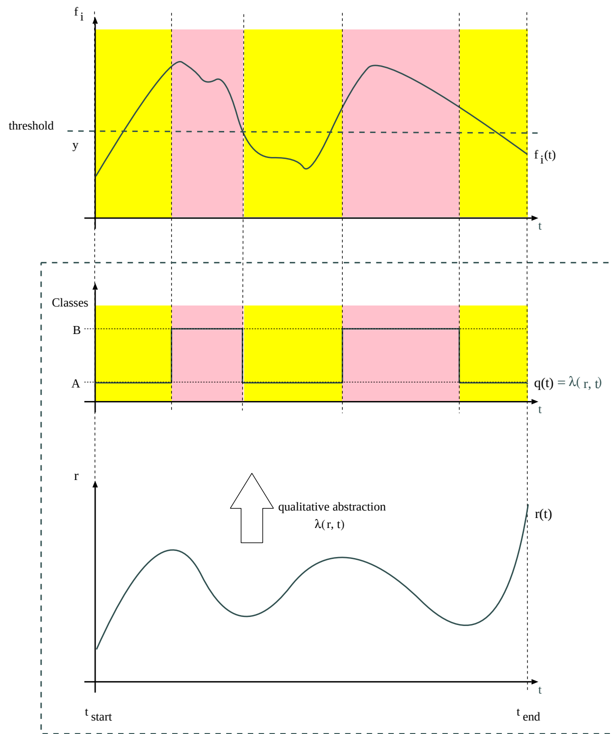


Fig. 1. Qualitative abstraction and horizontal splitting

Thus, there will be no problem with the decision tree induction not generating a proper result, if an adverse discretization has been chosen by the designer. The discretizations resulting from the method particularly hold valuable information about the opposing team. Discretization of continuous valued time series is the main task of the method. The method uses an very efficient algorithm to solve this problem.

To analyze our opponent we record several time series F. Supplementary mathematical transformed time series can be added in order to improve the results. One of the time series $r \in F$ must be qualitatively abstracted. This special series gives the classes to be learned, e.g. a time series containing: *goalie leaves goal*, *goalie stays in goal*, and *goalie returns to goal*. By the qualitatively abstracted time series time slices on all time series are assigned to the classes. One of the time series f_i is split horizontally at a threshold y (fig. 1). To determine this threshold all possible split points must be evaluated with a special heuristic. The best split point is the one which separates the different classes the best.

All the other time series are split vertically (fig. 2) at all points at which the time series f_i crosses the threshold y. With these separated graphs the process is recursively repeated.

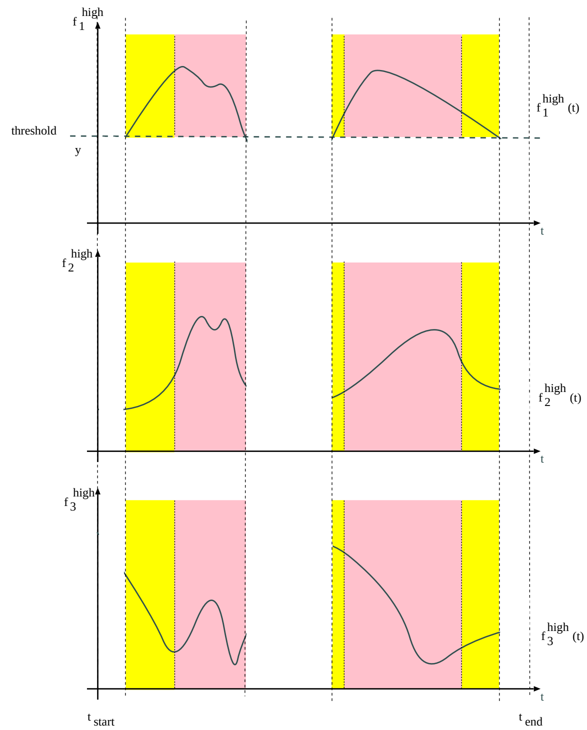


Fig. 2. vertical splitting

To apply the uniform distribution theorem of the entropy minimalization heuristic the time series must be approximated by partially linear functions.

The used method is based on the entropy minimalization heuristic as used in ID3 [Quinlan, 1986] and C4.5 [Quinlan, 1993]. An attribute is split at the point at which the heuristic is minimal.

Fayyad and Irani [Fayyad and Irani, 1992] have proved that this can only happen at class boundaries, we call these points boundary points (BP). By using this knowledge an important increase in efficiency can be reached. But it only works properly for non-overlapping classes. When two classes are overlapping the number of BPs can increase dramatically. In the worst case there can be BP between all examples in the overlapping area.

The method we use in this work allows an efficient splitting of continuous valued attributes in these cases. Boronowsky [Boronowsky, 2001] shows that this problem can be solved by using intervals of uniform distribution. This can be achieved by a linear approximation of the real distribution. He also explains that the entropy minimalization heuristic can only be minimal at the joints of the approximation.

So far the method was used in the system ExtraKT. ExtraKT is a system to aid humans in analyzing time series. It generates hypotheses about coherences in time series which can then be interpreted by a human analyst. The user can also manipulate the decision tree induction to enter his own knowledge into the process.

The system was used to analyze simulated technical systems. In these tests also sudden occurring failures in the technical systems were simulated. ExtraKT didn't know about these errors beforehand, but they were represented in the decision tree and could easily be found by the user. This aspect is very interesting for the use in the RoboCup competition. In a RoboCup match changes in an agent's behaviour can not only happen due to an error in the agent, but also because the team has changed its tactic.

3 New Coach

In order to improve our abilities to analyze the opposing team we integrated the system into the online coach. The coach continuously tracks certain aspects of the game, e.g. the positions of the players, their distance to each other and their distance to the ball. They are stored in memory forming several time series which are analyzed in regular intervals by the system mentioned above. As a result we receive rules about the opponents behaviour. Some of these rules can be used instantly within the game to improve the behaviour of our players and to adapt to the opponent. To achieve this the coach must have some a priori knowledge about the rules to expect and the suitable changes to our behaviour or playing style.

Additionally, there is the possibility to extract the collected data and rules at the end of the game. This information can be revised and corrected by a human expert. They can then be used for further development of the team or to improve the automatic extraction of rules by the coach.

In the first step we use the described method to analyze the opponents goal keeper. The goalie is particulary suitable, because his role in the team is fixed. He is the only player who's function is known from the beginning of the game; all the other players can only be classified as forwards, mid-fielders or defenders by their behaviour in the game, so we focus on the keeper for our initial work. Additionally, the goalie is a very important player. The knowledge of his strengths and weaknesses can be used to optimize the behaviour and configuration of our forwards, leading to better scoring abilities.

We also want to use the method to analyze our own goal keeper. This way we can test the quality of our system by cross checking the results with the details of our implementation. Besides that we may find some aspects in the behaviour of our goalie which could be improved further.

To prepare the analysis our coach stores some time series of game aspects which are related to the goal and the goal keeper. This includes

– the distance between the ball and the goal,

- the distance between the ball and the goal keeper,
- the distance between the goal keeper and the goal,
- the number of opponents and team mates within the penalty area,
- the number of team mates which may intercept the ball when kicked towards the goal.

For us the most important task is to determine, under which circumstances the goal keeper starts to leave the goal to get the ball and when he starts to return to it. According to this problem we have chosen a suitable qualitative abstraction. This abstraction uses the change in the time series corresponding to the distance of the goal keeper to the goal. This is used to determine whether the goal keeper is leaving the goal, is staying in the goal or is returning to the goal

The situations in which the keeper is leaving the goal or is returning to it are of special interest for us, because in these moments he can be taken by surprise more easily.

We plan to modify and extend the methods used to analyze the goal keeper to the other parts of the team in the next step. Then we will be able to improve the ability of all players of our team to adapt to the behaviour of the opponent. The forwards appear to be particular interesting to us. By finding special rules of attacks we can strengthen our defense by using suitable formations and behaviours.

4 Usage of the collected information

After collecting those information, we use them to improve our performance. The off-line usage is rather simple, if we store the data in a human readable form.

During a match the coach has to broadcast its recognitions. As some of the information surely can't be expressed in the standardized language of the coaches, we'll still have to wait for an interruption of the game, to send a message. The communication works just like the change of our formation, which we described in Visser et al. [2001] . The coach generates an announcement, which every agent can parse.

If the coach recognizes that the opponent goalie leaves his goal every time when there is none of his defenders between our ball carrier and him, he could instruct our forwards to pull him to the wings and pass the ball to an unguarded teammate in the center. If the goalie only leaves the goal if the forward achieves a certain distance to his goal, our agents should try to score from about this distance. If the goalie extremely shifts towards the ball during our attacks, quick wing plays should be effective.

Fig. 3 shows an 2-vs-1 situation (the three defenders beneath don't really affect the play). The goalie decides to attack the ball carrier, so as the forward passes the ball to his teammate, he gets a big opportunity to score. If we assume that the decision of the goalie is deterministic, he should react in the same way

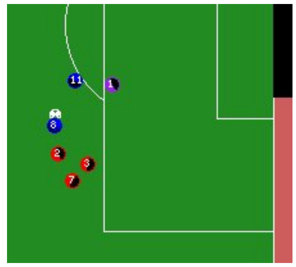


Fig. 3. 2-vs-1 situation

during the next 2-vs-1 situation. The coach should instruct the agents, that the supporting player should depart a little more, to get in an even better scoring position.

If the goalie had suspected and intercepted the pass, the coach would have told the agents that the ball carrier should dribble a little longer next time.

The counter actions of the coach have to be taught by a human instructor. They have to be parameterizable, so not every possible situation has to be taught manually. E.g. a goalie, which leaves his goal at a little different distance to the ball, shouldn't cause a complete different behaviour of our agents.

5 Results

As the described approach is a brand new method and has just been released we have to admit that there are only a few results so far. Currently, we are developing an interface for the RoboCup environment such that the method is provided with the time series information during a game. First test have shown that the approach is very promising, especially because of the ability to discretize automatically. We expect the method to provide the online coach with information such as propositional rules, e.g. *if the forward is closer than 10m and the forward has the ball then the keeper is approaching the ball*. In the final version of this paper we will show the results and discuss them in detail.

Acknowledgement

We would like to thank Michael Boronowsky for the work he has done helping us getting started with his method.

References

[Boronowsky, 2001] Boronowsky, M. (2001). Diskretisierung reellwertiger attribute mit gemischten kontinuierlichen gleichverteilungen und ihre anwendung bei der zeitreihenbasierten entscheidungsbauminduktion. In *DISKI*, volume 246, St. Augustin. infix Verlag.

[Fayyad and Irani, 1992] Fayyad, U. and Irani, K. (1992). On the handling of continuous-valued attributes in decision tree generation. In *Machine Learning*, volume 8, pages 87–102.

[Frank et al., 2000] Frank, I., Tanaka-Ishii, K., Arai, K., and Matsubara, H. (2000). The statistics proxy server. In *The Fourth International Workshop on RoboCup*, pages S.199–204.

[Quinlan, 1986] Quinlan, J. (1986). Induction of decision trees. In *Machine Learning*, volume 1.

[Quinlan, 1993] Quinlan, J. (1993). *C4.5 Programs for Machine Learning*. Morgan Kaufmann.

[Raines et al., 1999] Raines, T., Tambe, M., and Marsella, S. (1999). Automated assistants to aid humans in understanding team behabiors. In *Proceedings of the Third International Workshop on Robocup*, pages S.85–104.

[Visser et al., 2001] Visser, U., Drcker, C., Hbner, S., Schmidt, E., and Weland, H.-G. (2001). Recognizing formations in opponent teams. In *RoboCup-00, Robot Soccer World Cup IV*, Lecture Notes in Computer Science, Melbourne, Australia. Springer-Verlag. to appear.

[Wünstel et al., 2000] Wünstel, M., Polani, D., Uthmann, T., and Perl, J. (2000). Behavior classification with self-organizing maps. In *The Fourth International Workshop on RoboCup*, pages S.179–188.

Miene, A. and Visser, U. (2002) Interpretation of spatio-temporal relations in real-time and dynamic environments. In: RoboCup 2001: Robot Soccer World Cup V, Vol. 2377 (Eds, Birk, A., Coradeschi, S. and Tadokoro, S.) Springer, Seattle, WA, pp. 441-446.

Interpretation of spatio-temporal relations in real-time and dynamic environments

Andrea Miene & Ubbo Visser

TZI - Center for Computing Technologies
University of Bremen
Universitätsallee 21-23
D-28359 Bremen, Germany
{andrea|visser}@tzi.de

Abstract. With the more sophisticated abilities of teams within the simulation league high level online functions become more and more attractive. Last year we proposed an approach to recognize the opponents strategy and developed the online coach accordingly. However, this approach gives only information about the entire team and is not able to detect significant situations (e.g. double pass, standard situations, repeated patterns between two or three players). In this paper we describe a new method which describes spatio-temporal relations between objects. In addition, this approach is able to track the objects and therefore the relations between them online so that we are able to interpret situations over time during the game. This enables us to detect the above mentioned situations. We can implement this in the online coach in order to enrich our team with high level functions. This new method is domain independent.

1 Motivation

Due to the extension of the new soccer server for the online coach, its use becomes even more interesting. It's still the most effective instrument to analyze the opponent, because it can obtain all information about the simulated environment. Therefore, it is important to continue the development of the online coach which we used for the Virtual Werder team in the RoboCup 2000 tournament. In [Visser et al., 2001] we describe how the coach determines the opponent tactical formation with a neural network and how it is able to change a team formation during a match. We showed that it makes sense to recognize strategies and change the own team accordingly.

However, our approach relies on information about the opponent's team in total and is therefore not able to recognize and/or predict 'local' situations. We believe that the detection of the opponents behaviour in smaller areas, e.g. a double pass or a standard situation would help to find the appropriate countermeasures. The online coach is the optimal player for the collection of this kind of information and it is obvious that the coach should be able to process the data and find the appropriate tactic for the own team. Also, the analysis should be available online as the developed methods should be able to function in a real-time environment.

In this paper we describe a new method that is able to track moving objects in real time. The idea is to detect spatio-temporal relations between objects (players and ball) in a first step and then learn from this observations whether there is a repeating pattern, e.g. an attack over the wings with a pass onto the penalty point.

Our approach is related to the work from Raines and his colleagues [Raines et al., 2000] who describe a new approach to automate assistants to aid humans in understanding team behaviours for the simulation league. This approach is designed for the analysis of games, off-line after playing, to gain new experiences for the next games. Frank and colleagues [Frank et al., 2000] presented a real time approach which is based on statistical methods. A team will be evaluated statistically but there is no recognition of team strategies.

The remaining sections provide information about the method in the next section. The application and results of our approach within the soccer domain are discussed in section 3. Conclusions and future work are pointed out in the last section.

2 Approach

In this section we present a new approach on analyzing and describing the behaviour of moving objects. Object motion takes place in space and time. Therefore, it is useful to describe the behaviour of moving objects in terms of spatio-temporal relations. This leads to a domain independent description. Once the spatio-temporal relations between the objects are described one can interpret the behaviour of moving objects in the scenario. On a first level there are general behaviour patterns which are domain independent also. These patterns describe simple events like two objects meeting each other, two objects are approaching or following each other and so on. By using these simple spatio-temporal concepts one can construct more complex situations which are meaningful in a certain application domain such as one vs. two situation in a soccer match or an overtaking procedure in traffic surveillance.

2.1 Description of spatio-temporal relations

To describe spatio-temporal relations between objects two different types of relations have to be brought together, spatial relations between objects on the one hand and temporal relations between time intervals on the other hand. The movement of objects takes place over time and in some cases changes the spatial relations between the objects. Therefore, the main goal is to combine time and space in one description language: spatio-temporal relations.

Within a scenario of moving objects spatial relations between the objects as well as the object motion has to be taken into account. A spatial relation between two objects holds during a time interval and an object motion of continuous direction and velocity also has a certain duration. Therefore, the idea is to describe both, the duration of objects motion as well as spatial relations holding between objects via time intervals.

The input data for the approach consists of a series of object coordinates which are updated at moments which occur within a certain time frequency. This information may be obtained from any source, e.g. an image processing system extracting object motion from image sequences or from any other application such as RoboCup soccer server. The input data represents changes of the object positions over time in a discret and quantitative way. In a first step the time intervals of continuous object motion (OMI) and of duration of spatial relations between pairs of objects (SRI) have to be derived from the input data, i.e. have to be generated step by step at runtime.

At each moment and for each object a new motion vector has to be processed that describes the objects displacement from the last moment to the actual by length and angle. OMIs are established for each object. As long as the length and angle of the motion vector is similar to the average length and direction of the motion vector already belonging to the actual OMI the interval is extended, otherwise a new OMI is started. The metric values for the average length and angle of the vectors are very precise but not intuitive for a human observer. Therefore, length and angle within each OMI are classified to a fixed number of motion directions and speeds to be distinguished. The number of classes to be distinguished depends on the application. For a 2-D application the number of directions is often limited to eight, referring to a wind rose.

The number of speeds to be distinguished also depends on the domain. At least two classes are necessary to distinguish objects in motion from still ones. But in most cases it is useful to distinguish more than one speed and therefore to define classes of different speed such as slow, medium, fast and so on. The qualitative description of an objects movement consists of a continuous sequence of OMIs covering the whole duration of existence of the object. Obviously, each OMI refers to exactly one object O and has exactly one start moment i_s, end moment i_e, motion direction α and motion velocity v:

$$i_{om} = \Big[O\langle\alpha, v\rangle\Big]_{i_s}^{i_e} \tag{1}$$

Several methods on representing spatial relations between objects have been suggested [Güsgen, 1989,Schlieder, 1993]. In this approach a spatial relation between two objects is specified through a direction in which the second object is located and the distance between the objects. On a quantitative level a metric distance and an angle in which direction the related second object is placed is calculated. The procedure of generating time intervals of the duration of spatial relations between two objects (SRI) is the same as for the motion intervals. Due to the fact that there is a maximum distance up to that a spatial relation between two objects is taken into account at all, gaps may occur in the sequence of SRIs referring to a pair of objects. To obtain a qualitative description a fixed number of directions and distances such as *meets*, *near*, *medium* and *far* are distinguished. The wind rose in used again for the direction. Each SRI refers to exactly one pair of objects (O_1, O_2) and has exactly one start moment i_s, end moment i_e, location direction l and displacement d:

$$i_{sr} = \Big[O_1 \langle l, d \rangle O_2 \Big]_{i_s}^{i_e} \tag{2}$$

The relationship between the time intervals – OMI and SRI – are described using the seven temporal relations *before, meets, overlaps, starts, during, finishes* and *equals* and their inverse described in [Allen, 1981].

Any scenarios of moving objects can be described by use of OMIs and SRIs which are temporally related.

2.2 Interpretation of spatio-temporal relations

The first step to interpret scenarios with moving objects is to identify the concepts a human observer uses to describe the movement of objects. Then the concept is split into OMIs and SRIs and the temporal relations between these time intervals are described. This leads to a definition of the concept in terms of spatio-temporal relations and makes it possible to identify the concept within a scenario.

Elementary concepts describing simple motion events are domain independent and build a basis for the construction of more complex events. Complex events are often domain specific and are described as a combination of simple events.

The entire set of simple events can be divided into groups according to the number of involved SRIs and OMIs and their temporal relations. The following group of simple events includes two objects that are spatial related and may move within duration of the spatial relation, i.e. three overlapping time intervals are involved:

$$\begin{aligned} i_{sr} &= \Big[O_1 \langle l_1, d_1 \rangle O_2 \Big], \\ i_{om_1} &= \Big[O_1 \langle \alpha_1, v_1 \rangle \Big] \\ i_{om_2} &= \Big[O_2 \langle \alpha_2, v_2 \rangle \Big] \end{aligned} \tag{3}$$

For an off-line interpretation all three intervals have to overlap to some degree. For an on-line interpretation all three intervals have to exists at the moment the interpretation is generated (and therefore must have started in the past). Note that this is always true for the OMIs because there are no interrupts in the sequence of OMI describing the objects motion, because even if the object does not move at all this time interval is covered by an OMI with zero speed. In opposite to this the SRI may not exist if the distance between the objects exceeds a given threshold.

To distinguish and recognize the different simple events belonging to one group the attributes motion direction, speed, direction of spatial location and distance have to fulfil certain constraints. In the following we present some examples of simple events belonging to the group introduced before:

- Two objects *meet* each other:

$$d_1 = meets \tag{4}$$

- If the motion direction of O_1 is the same than the spatial direction between O_1 and O_2 and O_2 stands still, object O_1 is *approaching* object O_2:

$$v_1 > 0 \wedge \alpha_1 = l_1 \wedge v_2 = 0 \tag{5}$$

- If in contrast to this O_1 moves in opposite direction to the spatial direction between O_1 and O_2 and O_2 stands still, object O_1 is *departing* from object O_2:

$$v_1 > 0 \wedge opposite(\alpha_1, l_1) \wedge v_2 = 0 \tag{6}$$

- If both objects move and their motion directions are the same than the spatial direction the opposite object is located the two objects *approach each other*:

$$v_1 > 0 \wedge \alpha_1 = l_1 \wedge v_2 > 0 \wedge opposite(\alpha_2, l_1) \tag{7}$$

- Object O_1 and O_2 *move in parallel*, i.e. both objects move with similar velocity, their motion directions are the same and the spatial direction between the two objects is perpendicular to the motion direction:

$$v_1 > 0 \wedge v_1 = v_2 \wedge perpendicular(\alpha_1, l_1) \wedge \alpha_1 = \alpha_2 \tag{8}$$

- If both objects move in the same direction and the direction of their movement is the same than the spatial direction between the first and the second object is the same as the motion direction, the first object is *following* the second one:

$$v_1 > 0 \wedge v_2 > 0 \wedge \alpha_1 = \alpha_2 = l_1 \tag{9}$$

  If we have an additional look at the velocity of both object we can also decide if the following object *catches up*, *falls behind* or *follows in a constant distance*.

Another group of simple events describes the change of movement of a single object like start or stop movement, speed up and slow down. In this case two subsequent OMIs of the same object are analyzed.

In addition to this there are several further groups of simple events that are not mentioned in this paper. In the following section we explain the usage of simple events to interpret the behaviour of the players in a soccer game.

3 Application and Results

Most of the simple events introduced in the previous section have a more specific meaning in a concrete domain. As an example we choose the soccer domain: The simple event *meets* between a player and the ball means that the player has the ball. If there is another meets-relation between a player from the opposing team and the ball at the same time (i.e. overlapping time intervals), this is called a 'fight for the ball'. A *meets*-relation between the ball and a goal is called goal.

Another meaningful simple event is $departing(b, p)$, i.e. a player p is passing the ball b. A further evaluation of the balls motion direction leads to a more detailed information whether the ball is played to the front (i.e. in direction to the opposite goal) or backward.

This simple event is part of numerous complex situation such as player p_1 passes the ball (b) to player p_2. This situation consists of four simple events:

1. $se_1 = meets(p_1, b)$
2. $se_2 = departing(b, p_1)$
3. $se_3 = approaching(b, p_2)$
4. $se_4 = meets(b, p_2)$

The temporal relations between the simple events are $meets(se_1, se_2)$, $equal(se_2, se_3)$ and $meets(se_3, se_4)$. This example shows how complex situations are constructed by using temporal related simple events. To make this interpretation more specific it is also possible to distinguish different types of players such as defenders, forwards or keeper semantically, i.e. draw conclusions from their behaviour rather than taking the information provided by the soccer server.

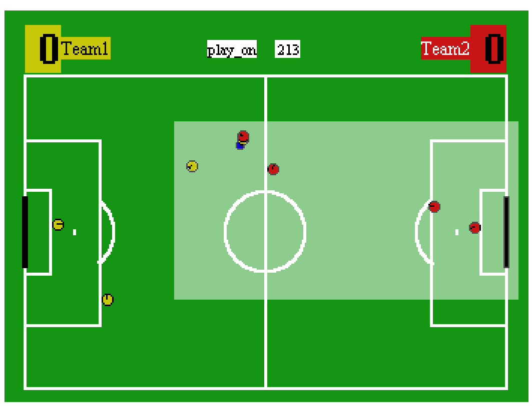
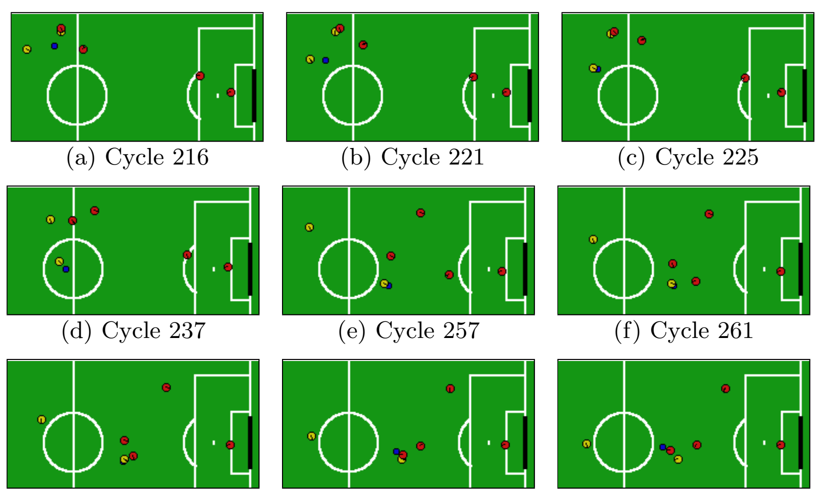


Fig. 1. Two players fighting for the ball

In this section we show first results on how the new method for describing and analyzing spatio-temporal relations can be applied to support the coach in a soccer game. For a coach it is useful to analyze the behaviour of the opposite team and to instruct the players of his own team. With the help of the approach discussed in the previous sections it is possible to analyze the behaviour of both, the players of his own team and the opposite team. Knowing about changes within the spatio-temporal relations between the players over time the coach is able to detect gaps that open up in the defence line of the opposite team. Then he can identify a player of his own team within a good position to take advantage of this. In addition to this, repeating patterns in the strategy of the opposite team can be identified and then detected early when they occur again.

To explain this in more detail we give an example taken from a match with 4 vs. 4 players. The example situation lasts over 60 cycles. The beginning of the situation is illustrated in fig. 1. A forward of team 1 is at the ball while a player the opposing team attacks him to get the ball. For the next cycles we focus on the area high lighted in fig. 1. The ongoing situation is illustrated in fig. 2. The nine detailed images give a brief overview over the sequence of 60 cycles. The description leads to the following interpretation in terms of time intervals



(a) Cycle 216 (b) Cycle 221 (c) Cycle 225

(d) Cycle 237 (e) Cycle 257 (f) Cycle 261

(g) Cycle 265 (h) Cycle 270 (i) Cycle 273

Fig. 2. Ongoing game situation: defense

of simple events:

- Player of team 1 and team 2 are both in meets-relation to the ball. This is interpreted as fighting for the ball.
- In the following time interval the ball is *departing* from both players, i.e. it is passed.
- While the ball is still departing from the two players a second player of team 1 approaches the ball.
- Then the second player of team 1 reaches the ball (*meets*).

– Within the following time intervals the player and the ball move in direction to the opposite goal, while the spatial relation between the player and the ball is *meets* or close (see also fig. 3 for a detailed view).
– While the player moves with the ball some players of team 2 are moving backwards. This is interpreted as building up a defensive position.
– Then two players (defenders) of team 2 are approaching the player with the ball.
– Then one of them is meeting the player with the ball (fight for the ball again).
– The player of team 2 is still in meets relation to the ball whereas the player of team 2 is close to the ball but does not meet it anymore, i.e. he has lost the ball.
– At last the ball is departing from the player of team 2, i.e. he is passing the ball in the opposite direction to avoid the player of team 1 to score.

This description is close to the interpretation an human observer would probably give: The forward succeeds to pass the ball. Another forward of team 1 runs for the ball and tries to approach the opposite goal with the ball. This is noticed by the defenders of team 2 which move backwards and then try to stop it. One of them is approaching the forward and unfortunately there is no other forward of team 1 he could pass the ball to. The defender of team 2 takes the ball from the forward and passes it in the opposite direction.

This defence strategy is typical for team 2 and occurs for several times in the game. To support the coach such situations can be detected by the sequence of time intervals described above. If the situation occurs again, it can be recognized early, so that team 1 could try another way e.g. positioning a second player near to the forward to pass the ball to when the defenders approach.

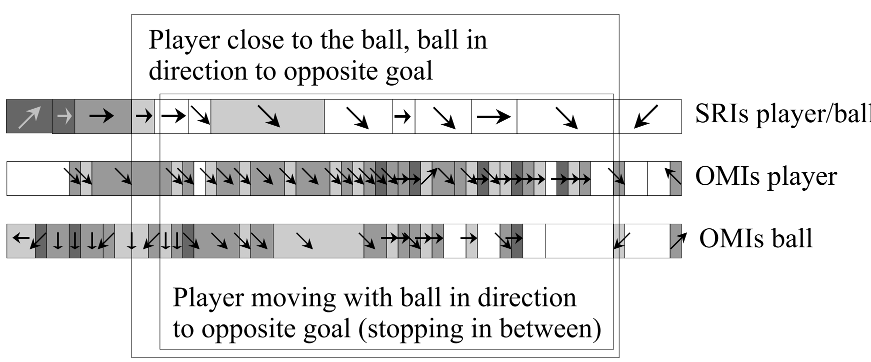


Fig. 3. SRIs and OMIs of the attacking player and the ball.

For a more detailed example on how the temporal relations between the SRIs and OMIs are interpreted refer to fig. 3. The diagram focuses on the attacking player. It shows the SRIs between the player and the ball and the OMIs of the player and the ball. Light colors refer to small distances, dark color to large ones, i.e. white means still stands resp. meets (spatial), light grey slow motion resp. close distance and so on. The arrows represent the motion direction resp. the direction in which the spatial related object (in this example the ball from the viewpoint of the player) is placed. Within the entire sequence of SRIs there is an interval in which the distance is either meets or close and the ball is placed in direction to the opposite goal. This is marked by the large rectangle. Within this time interval both objects – player and ball – are moving in direction to the opposite goal except for short interrupts where they stand still. This time interval is *during* the previous described interval and is marked by the smaller rectangle.

4 Conclusion and future directions

Spatio-temporal relations between objects within real-time environment are challenging by nature. In this paper we presented an approach of how objects are spatially and temporally related to others and track this over time. We showed that simple events such as *meets*, *departing*, *approaching*, *equals* can be detected and combined to more sophisticated events. We also showed that interpretations of situations can be more specific. We applied this idea to the soccer domain and argue that an implementation of this method within the online coach could enhance teams abilities.

Another interesting feature is the ability to analyze games not only online but also off-line. One of the biggest advantages of this approach is the independence from the domain. In the near future, we will also test other domains such as cell tracking in biological systems. Here, the objects are monitored with a camera and the method is able to track the objects over time and describe and store the spatial relations between them as well.

However, there are also difficulties with the approach at the moment. For a human being it is relatively easy to follow a RoboCup soccer game on the monitor and see situations fluently and continuously. Because the objects are described on a discrete level it sometimes happens that the approach is not able to detect a continuous flow. For example, while attacking a goal with the ball the player is moving, kicking the ball in the goal direction and so on. However, there will be moments when either the ball or the player have no movement so that the approach terminates the continuous flow and starts a new situation. Also, we are able to detect relations between two objects but sometimes more than two objects are involved in a situation. This will be on the list for future work.

The next step with the current approach is to detect complex situations over time and learn patterns such as attack over the wing with a pass in the penalty area, double pass etc. In addition, work can be done for the enhancement of this approach from 2D to 3D domains.

References

[Allen, 1981] Allen, J. F. (1981). An interval-based representation of temporal knowledge. In Hayes, P. J., editor, *IJCAI*, pages 221–226, Los Altos, CA.

[Frank et al., 2000] Frank, I., Tanaka-Ishi, K., Arai, K., and Matsubara, H. (2000). The statistics proxy server. In Balch, T., Stone, P., and Kraetschmar, G., editors, *4th International Workshop on RoboCup*, pages 199–204, Melbourne, Australia. Carnegie Mellum University Press.

[Güsgen, 1989] Güsgen, H.-W. (1989). Spatial reasoning based on allen's temporal logic. Technical report TR-89-049, International Computer Science Institute (ICSI).

[Raines et al., 2000] Raines, T., Tambe, M., and Marsella, S. (2000). Automated assistants to aid humans in understanding team behaviors. In *Fourth International Conference on Autonomous Agents (Agents 2000)*, Barcelona, Spain.

[Schlieder, 1993] Schlieder, C. (1993). Representing visible locations for qualitative navigation. In Piera-Carrete, N. and Singh, M., editors, *Qualitative Reasoning and Decision Technologies*, pages 523–532. CIMNE, Barcelona.

[Visser et al., 2001] Visser, U., Drücker, C., Hübner, S., Schmidt, E., and Weland, H.-G. (2001). Recognizing formations in opponent teams. In *RoboCup-00, Robot Soccer World Cup IV*, Lecture Notes in Computer Science, Melbourne, Australia. Springer-Verlag. to appear.

Visser, U. and Weland, H.-G. (2002) Using online learning to analyze the opponents behavior, In:Proceedings of Proceedings of the RoboCup-2002: Robot Soccer World Cup VI, (Eds, Kaminka, G., Lima, P. and Rojas, R.), Fukuoka, Japan, pp. 72-81.

Using Online Learning to Analyze the Opponents Behavior

Ubbo Visser and Hans-Georg Weland

TZI - Center for Computing Technologies, University of Bremen
Universittsallee 21-23, D-28334 Bremen, Germany
{visser|weland}@tzi.de
http://www.virtualwerder.de/

Abstract. Analyzing opponent teams has been established within the simulation league for a number of years. However, most of the analyzing methods are only available off-line. Last year we introduced a new idea which uses a time series-based decision tree induction to generate rules on-line. This paper follows that idea and introduces the approach in detail. We implemented this approach as a library function and are therefore able to use on-line coaches of various teams in order to test the method. The tests are based on two 'models': (a) the behavior of a goalkeeper, and (b) the pass behavior of the opponent players. The approach generates propositional rules (first rules after 1000 cycles) which have to be pruned and interpreted in order to use this new knowledge for one's own team. We discuss the outcome of the tests in detail and conclude that on-line learning despite of the lack of time is not only possible but can become an effective method for one's own team.

1 Introduction

The standard coach language provides the on-line coaches in the RoboCup soccer simulation league with a possibility to rapidly change the behavior of his team. In addition, its existence allows a competition between coaches. In order to achieve a successful coaching, a lot of information about the opponent has to be collected, to which the coach can react and change his own team according to the opponent. For this reason several methods have been introduced in the past to analyze on-line and to adapt to the opponents. These papers demonstrated how to recognize team formation [Visser et al., 2001] or how to adapt to play situations [Riley and Veloso, 2002]. [Drücker et al., 2002] showed the idea and a first prototype of the method presented in this paper.

Very important aspects of the behavior of a soccer team are the goalkeeper and the pass behavior.

A pass is a frequent event within a soccer game. It allows the team passing to move the ball across a great distance in a short time and to defeat the opposing defenders. Thus, successful passing can be a great advantage. On the other hand, a pass is always a risk. When the ball is passing it moves without being guarded by a team member who could change the direction of the ball, if necessary. This gives the opposing players the possibility to intercept the pass. These intercepted passes are a certain disadvantage. When a team is passing less successful in certain situations than in others, it makes sense for the opposing team to try to create these situations as often as possible. However, in order to do this, it must be known which factors lead to these mistakes.

Therefore, analyzing the goalkeeper the moment when he leaves the goal is of special interest. A possibility to play around him or to reach a point near the goal while he does not attack the forwards, produces great chances to score. It is important to know which factors have led to the players' decision. Also, it is important to know the threshold where they react in a certain way. Thus, it is crucial to use a method that is not hampered by pre-discreeted values but does the discreetization itself. As the outcome of such a method should directly be used by the on-line coach, the results require a certain form. They should be employed to generate instructions for the players. For these constraints the time series-based decision tree induction described in [Boronowsky, 2001] seems suitable. As this method operates with continuous-valued time series, a pre-discreetization of the data is not required. The method finds the split points and therefore the thresholds where the analyzed player act.

2 Time series-based decision tree induction

This method consists of a entropy minimization heuristic that only has to be calculated for certain points as opposed to C4.5[Quinlan, 1993] where all possible split points are calculated. This is an important advantage to other decision tree algorithms. Due to the great amount of possible split points, the calculation effort can be very high with continuous data.

The basic ideas for the optimization of the method we use are similar to those of ID3 and C4.5 described in [Fayyad and Irani, 1992]. It shows that the entropy minimization heuristic can only be minimal at the so-called boundary points. A boundary point is a point within the range of a certain attribute whose neighbors belong to two different classes. Thus, the range of an attribute holds for a boundary point T, and

$$A(e_1) < T < A(e_2). \tag{1}$$

$E_1, e_2 \in S$ belong to the set of all samples S, $A(e)$ is the attribute value of e and

$$\neg\exists e_3 \in S : A(e_1) < A(e_3) < A(e_2) \tag{2}$$

where e_1 and e_2 belong to disjunctive classes.

This approach already leads to a noticeable increase in efficiency given appropriate samples. If the samples include overlapping classes the increase of efficiency ends. There are boundary points in the overlapping area in such cases. The calculation then has to be done for all these points because the entropy minimization heuristic can be minimal for all of them. In RoboCup environments such overlapping areas cannot be ruled out.

It can be shown though that for continuous equal distributions the entropy minimization heuristic can only be minimal at so-called true boundary points. True boundary points are those boundary points that are located at the boundaries of the interval of one class assignment. When a continuous equal distribution can be reached, the number of interesting points can be reduced and therefore the calculations. This leads to an significant increase in efficiency.

Since this decision tree induction does not work on a set of examples but on time series, the calculation of the optimal split point does not depend on the amount of examples belonging to a certain class. Instead, it depends on the time interval of such an assignment. Instead of $freq(C_j, S)$ and $|S|$ the interesting issues are

- the duration of the assignment of the time series smaller than the split point y to a class C_j $time_{C_j}(y)$,
- the duration of the assignment of the time series above the split point $\overline{time_{C_j}}(y)$, and
- the total duration of the assignment to one class $tmax_{C_j}$.

Thus, there isn't an assignment to a class for every single sample, but there is a qualitative abstraction to one or a combination of multiple time series. This defines which time intervals are assigned to which class. Figure 1 shows an example of a qualitative abstraction of a time series. It shows the behavior of a goalkeeper. At time t_{begin} the distance between the keeper and the goal remains stable. At time t_1 the distance increases which means that he leaves the goal. This state holds until time t_2. The function will be abstracted to class B_{stays} and C_{leaves} accordingly.

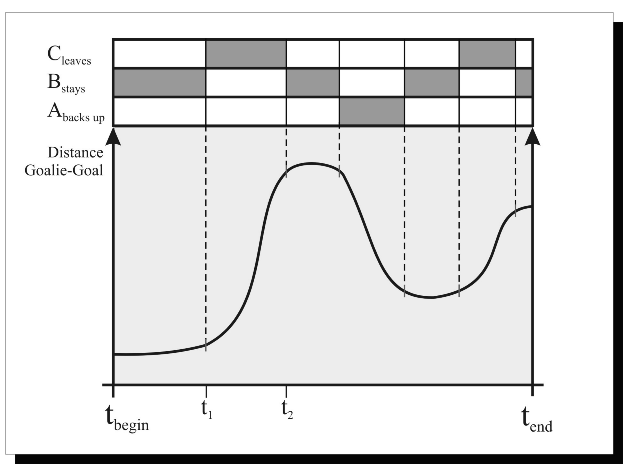


Fig. 1. Qualitative abstraction of the behavior of a goalkeeper

For partial linear measured value gradients it can be shown that the entropy minimization heuristic can be calculated efficiently [Boronowsky, 2001]. This is based on the characteristic that the duration of a class assignment $time_c(y)$ for partial linear measured value gradients can be described by linear functions. Such linear durations of class assignments are continuously equally distributed. Thus, the correlation between a continuous equal distribution and an entropy minimization heuristic, as described above, can be used.

It is therefore possible to perform an efficient decision tree induction for continuous valued time series, if these time series consist of partial linear functions. As it cannot be assumed that such partial linear measured value gradients are found, they must be adapted in an appropriate way. This is done by an approximation of the continuous measured values by partial linear functions. As an approximation does not exactly equal the original function an approximation error occurs. This error should be as small as possible, especially at the split points. In general, one can achieve a better approximation by increasing the number of linear functions. On the other hand, this leads to a higher number of potential split points. Thus, the reduction of the approximation error leads to a loss of efficiency. Although, certain points seems to be very important and should therefore be used as boundary points for the linear functions. These are

- the start and end of the used time series
- the times of a change in the class assignment
- the extreme values of the measurement course

It makes sense to use the start and end of the time series because this is the only way to represent the whole series. By using the points where the class assignment changes as boundary point for the linear approximations, a linear function is always assigned to exactly one class. This makes the calculations of the duration of class assignments easier. The extreme values are interesting because they give the option to find a split point, which separates the time series in such a way that the complete series assigned to one class is under or above this split point.

With these partial linear functions the entropy minimization heuristic only need to be calculated for the boundary points of the linear functions because it can only be minimal at these points. Fig.2 shows an approximation with four boundary points (1-4) and the according four points for potential horizontal splitting (y1-y4). To calculate the entropy, the information contents of the duration of class assignments and $(info(y))$ an above $(\overline{info}(y))$ the potential split point has to be calculated.

$$info(y) = -\sum_{i \in C} \frac{time_i(y)}{\sum_{k \in C} time_k(y)} ln \left(\frac{time_i(y)}{\sum_{k \in C} time_k(y)} \right) \tag{3}$$

$\overline{info}(y)$ has to be calculated in the analogue, by changing $time(y)$ to $\overline{time}(y)$. The entropy can be calculated by

$$entropy = \frac{\sum_{i \in C} time_i(y) info(y) + \sum_{i \in C} \overline{time_i}(y) \overline{info}(y)}{\sum_{i \in C} tmax_i} \tag{4}$$

This calculation must be done for all boundary points in all time series. At the point where the results are minimal the according time series is split horizontally. All the others are split vertically at certain points. These points are those where the horizontally splitted time series crosses the value by which it is split. The result is that two new time series are created for every existing time series. One consists of those time intervals in which the horizontal splitted series is greater than the split threshold and the other of those where it is not. This process is then repeated with the two new sets until a certain end criterion is reached.

The ending criterion cannot be the number of correct classified examples as used in regular supervised symbolic machine learning techniques. As it is a time series-based method the ending criterion should be the correct classified time intervals.

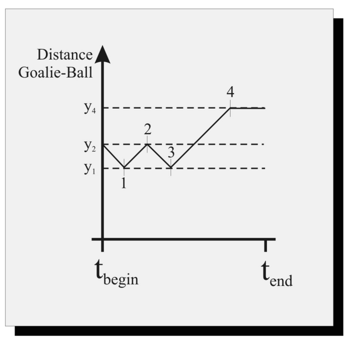


Fig. 2. Partial linear approximation with potential points for horizontal splitting

At the splitting of the time series the approximation error has to be considered. If the splitting is only calculated by the approximations and not on the basis of real functions there can be errors in the horizontal as well as in the vertical splitting. At the horizontal splitting the approximation can be different from the real value on the time axis and at the vertical there can occur an error on the ordinate. This can be prevented by adapting the approximation in a suitable way. Therefore, all points are added to the approximation that should be split horizontally where the real functions crosses the threshold y. To the other approximations new points are added at all vertical split points.

3 Preprocessing of the used data

In order to use the method in an online-coach it has to be defined what should be analyzed and which data should be used for it. In this paper we focus on two scenarios:

1. We analyze the moment when the goalkeeper leaves the goal.
2. We analyze the pass behavior of opponent players

In order to do this a suitable qualitative abstraction must be found. It defines which time intervals are assigned to which class, e.g. if the goalkeeper leaves the goal or if he stays on the line. This leads to the problem that the things that should be learned are not always directly included in the data provided by the soccer server. Therefore, the given data have to be combined or even an element of guessing has to be included. It is important to note that we can only analyze what is happening and that we cannot recognize the player's intention.

The result of this method always depends on the abstraction. If the abstraction is not correct something different is learned. A slight error in the abstraction of the goalkeeper could lead to a tree that has learned the movement of the goalkeeper rather than when he is leaving the goal. These results cannot be used correctly to improve the behavior of a team.

In order to use a decision tree algorithm it is necessary to choose suitable attributes from which the tree is built. It is essential that these attributes represent those values that are important for the decisions of the opponent player. If they are not represented in the attributes the behavior cannot be learned.

3.1 Analysing Goalkeeper Behavior

In the analysis of the goalkeeper's behavior the moment when he leaves the goal is of special interest, because this is when the goal becomes vacant. Thus, a qualitative abstraction is chosen which represents this behavior. The movements of the goalkeeper are represented in three classes, *(a) goalkeeper stays in goal, (b) goalkeeper leaves goal and (c) goalkeeper returns to goal*. This calculation is based upon the movement of the goalkeeper towards the ball. If he moves towards the ball he leaves the goal, and if he goes away from the ball he returns to the goal. In all other cases he stays in the goal. This is not always correct in relation to the goal, but it represents what should be learned. The movement vector of the goalkeeper and the position of the ball are used to compute the abstraction. The length of the movement vector gives the speed, and the angle between the movement vector and a vector from the goalie to the ball are used for the direction.

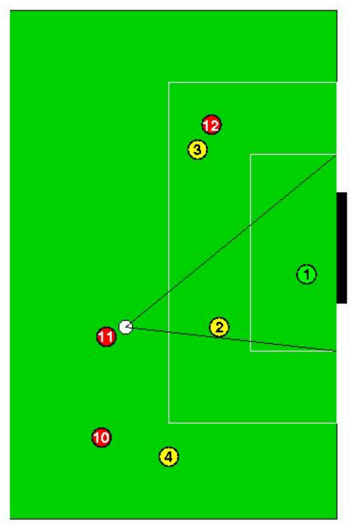


Fig. 3. Cone from ball to both sides of the goal and some positions of opponent players (1-3) as attributes for the learning process

As described above, suitable attributes must be chosen. These attributes should include those that are used by the analyzed goalkeeper to make his decisions. The position of the ball seems to be very important. It will be the major aspect in the goalkeeper's decisions.

Murray [Murray, 1999] even describes a goalkeeper whose decisions are totally based upon the ball position. Because of the relative view of the goalie and the abilities of our own players, the relative distance of the ball to the goal and the goalkeeper seems more interesting than its absolute position. Another interesting fact are the positions of the other players (see figure 3). Especially those who can directly interfere in a possible shot at the goal. Particularly the players within the penalty area are relevant to the goalkeeper because of his ability to catch the ball. This is why he may react differently to players in this area. Thus, the number of forwards and defenders in the penalty area is used to analyze the goalkeeper.

Very important for a goalkeeper is the question whether the opposing ball carrier can run and shoot at the goal without being attacked by a defender. In these cases a goalkeeper should leave the goal and run towards the ball carrier to prevent him from shooting into a free corner of the goal. Hence, the defenders within a cone from the ball to the sides of the goal (figure 3), which are likely to intercept the ball, are used as an attribute for the decision tree.

3.2 Analysing Passes

In order to analyze the pass behavior, a qualitative abstraction and some input values have to be found. This leads to some problems owing to the kind of the present data. The coach only knows the positions and movements of all players and of the ball, but not whether a player kicks the ball or not and in which direction the ball was kicked. Therefore, it is impossible to see in the data given by the soccer server who kicked the ball or even if the ball was passed at all.

However, because there are certain rules describing the changes in speed and direction of the ball, it can be calculated from the movement of the ball whether it was kicked or not. Without being kicked by a player the ball cannot gather speed. The same applies to a sudden stop. In both cases a player must have sent a kick command.

This does not apply to a change of direction. A direction change could also happen as a result of a collision with a player. Thus, it cannot be verified that the ball was kicked when it changes its direction. A bigger problem is the question who kicked the ball. Suppose the ball is within the kickable area of one player and one of the events described above happens at the same time. Then, only that player can be the one who kicked the ball. But if the ball is in the kickable area of several players it cannot be determined who kicked it. It also is not possible to tell which intention the player had when he kicked the not be told. On the other hand, it makes no difference to a team whether it gets the ball by intercepting a pass, by dribbling, or by a goal shot. If a team can provoke situations that lead to an interception always has an advantage.

These considerations lead to a qualitative abstraction based upon two kick events. Every kick event is compared with the prior one. It is important which players did the two kicks. The qualitative abstraction assigns these events into four classes:

– pass between two opponents,
– opponent dribbles,
– pass of an opponent was intercepted, and
– team-mate kicked the ball.

A kick command is supposed to have happened if the ball gathers speed or is suddenly stopped. The player who is next to the ball is assumed to have kicked.

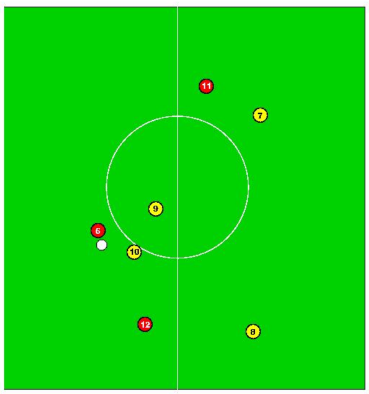


Fig. 4. A passing situation, attributes are positions of the opponent players, surrounding of the ball carrier (e.g. distance and directions of nearest player)

When defining which data should be used as attributes for the learning method one has to take into account that the method has to find out what the player does. Therefore, it is important to use the values used by the player while making decisions. The surroundings of the ball carrier are of special significance for the analysis of pass behavior, especially the distances and directions of the nearest players (see figure 4). It makes no sense to use absolute positions because the player only gets local information and is likely to make his decisions based on relative values, as given to him by the server. Also, players far away from the ball carrier are of no importance. The ball carrier does not see them well, maybe not at all, and a pass across a very long distance is a high risk. Thus it is unlikely that such players play an important part in the decisions of the passer.

While calculating the angles to the other players, it has to be taken into account that in soccer it is more important whether a player is located towards the middle of the field or towards the sideline than to the right or to the left. If e.g. the ball carrier is attacked from his left side it has a different meaning to him whether he is on the right or on the left side of the field. If he is on the right the defender blocks his way to the goal and pushes him towards the sideline. On the other hand, if he is on the left side the defender attacks him from the sideline and the way towards the goal is free. This applies even more in the simulation league because there are only physical differences between the players if a team uses the optional heterogeneous players. And all players can kick the ball to the left as well as to the right. This is why all angles on one half of the field are flipped horizontally. As a result, all passes to the middle have a similar angle and can be represented by the decision tree in the same branch.

Another aspect is the horizontal position of the passing player. The pass behavior may change depending on how far the ball carrier is away from his goal line. E.g. a pass played directly in front of the own goal carries a high risk of being intercepted, thus leading to a good scoring position for the opponent, while a pass in front of the opposing goal may give team-mates the chance to score. Additionally, some parts of a team may pass with a higher risk than others. VirtualWerder 00/01 does take this differences into account. The defenders play as securely as possible while the forwards sometimes even pass into the penalty area without knowing whether anybody is there.

In both cases, goalkeeper and pass behavior, the decision tree algorithm is used every 1000 cycles to generate rules about the opponent's behavior.

4 Results

To test the quality of the method several test games were played with different teams of last year's competition. VirtualWerder, Robolog Koblenz and Karlsruhe Brainstormers were used to evaluate the implemented algorithm. Robolog was used because, in contrary to most other teams, it is logic- based. The were chosen because they represent a reactive team and because they finished second in the last competition.For technical reasons, Tsinghuaeolus, the world champion, could not be used. It is a Windows-based team while the computers available for the tests where Linux computers. The binaries of FC Portugal were not available at the time of the tests.

4.1 Goalkeeper

To analyze the goalkeeper a qualitative abstraction was used based upon the movement of the goalkeeper, as described above, and six time series. Here are the time series:

- Series 0 - distance ball goalkeeper
- Series 1 - speed of the ball
- Series 2 - distance ball goal
- Series 3 - number of defenders within the penalty area
- Series 4 - number of forwards within the penalty area
- Series 5 - number of defenders that may intercept a direct shot at the goal

After 1000 cycles the decision tree is computed for the first time, based on data shown in table 1.

Table 1. Typical input data for the learning algorithm w.r.t. a goalkeeper

	Series 0	Series 1	Series 2	Series 3	Series 4	Series 5	Class
1	50	0	52.5	1	0	3	0
2	50	0	52.5	1	0	3	0
3	50	0	52.5	1	0	3	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1000	88.1415	0.487918	90.3164	1	0	0	1

For the tests with respect to the analysis of the goalkeeper ten test games where played with

– Robolog vs Brainstormers
– Robolog vs VirtualWerder

```
if   series 4 < 2.000000
and  series 0 < 20.105579
then 0(1) 1(0) 2(0)

if   series 4 < 2.000000
and  series 0 > 20.105579
then 0(0.71219) 1(0.245547) 2(0.0422634)

if   series 4 > 2.000000
and  series 0 < 0.385009
then 2(0.950617) 0(0.0493827) 1(0)

if   series 4 > 2.000000
and  series 0 > 0.385009
then 2(0.48) 0(0.34) 1(0.18)
```

Fig. 5. The Brainstormers goalkeeper in a game against Robolog

In the first constellation, both goalkeepers were analyzed, and in the second, for technical reasons, only the one from Virtual Werder.

```
(1) if   series 3 < 1.000000
    then 0(1) 1(0) 2(0)

(2) if   series 3 > 1.000000
    and  series 3 < 4.000000
    and  series 1 < 0.031605
    and  series 0 < 6.774502
    and  series 0 < 5.984714
    then 0(1) 1(0) 2(0)

(3) if   series 3 > 1.000000
    and  series 3 < 4.000000
    and  series 1 < 0.031605
    and  series 0 < 6.774502
    and  series 0 > 5.984714
    then 1(0.8) 0(0.2) 2(0)

(...)

(4) if   series 3 > 1.000000
    and  series 3 > 4.000000
    and  series 2 < 7.680346
    then 1(0.75) 0(0.25) 2(0)

(5) if   series 3 > 1.000000
    and  series 3 > 4.000000
    and  series 2 > 7.680346
    and  series 0 < 6.799984
    then 1(1) 0(0) 2(0)

(...)
```

Fig. 6. The VirtualWerder goalkeeper in a game against Robolog

The rules concerning the Brainstormers' goalkeeper are shown in figure 5. We see that the goalkeeper reacts at different distances to the ball (series 0) depending on the number of attackers in the penalty area (series 4). This change in the distance was often noticed but sometimes it depended on the number of defenders in the penalty area. However, this might be the same because both sounds like a breakaway.

The tests with the VirtualWerder goalkeeper revealed noticeable longer rules than with the two other goalkeepers. One possible explanation is that the VirtualWerder field players cannot keep the ball out of their penalty area and therefore the goalkeeper needs often to react. As this leads to more changes between the classes, he can be better analyzed or he may have a more complex behavior than the others. The rules in figure 6 reflect a part of the behavior of the VirtualWerder goalkeeper in a game against Robolog. The rules show that the goalkeeper changes his behavior depending on the number of defenders in the penalty area (series 3). The first rule reveals that he stays in the goal if there are no defenders present. While the second and third shows that if there are one to four defenders (rules 2 and 3,) he makes his decision based on the speed of the ball (series 1) and the distance from the ball to him. Having more than four defenders within the penalty area (rules 4 and 5,) the decision is based on the distance of the ball to the goal (series 2).

The decision tree method does not produce interesting results about the goalkeeper if there are not enough scenes where he reacts. This can happen if the opponent is too strong for the team of the coach. It may happen that one team is not able to bring the ball into the opposing penalty area. Sometimes teams even have problems to cross the middle line. In these cases the goalkeeper does not have to act often enough to be analyzed, or maybe he doesn't act at all. In such cases the method only delivers one rule saying that the goalie doesn't move at all. But this is no problem, because if the goalkeeper needs not to act it is not an advantage to know what he would do if he had to. At first the other parts of the play have to be improved. If this is possible fast enough there might be enough information to analyze the goalkeeper later on when there is a possibility to draw an advantage out of this knowledge. This problem was revealed by the tests with VirtualWerder. This team was not able to produce enough pressure on the other two teams to produce enough goalkeeper-scenes. Thus, there were no sensible results about the other teams from these games.

4.2 Pass

According to the reflections in 3.2 ten time series had been chosen as attributes to analyze the pass behavior.

- Series 0 - distance to the next opponent
- Series 1 - angle to the next opponent
- Series 2 - distance to the second next opponent
- Series 3 - angle to the second next opponent
- Series 4 - distance to the next team-mate
- Series 5 - angle to the next team-mate
- Series 6 - distance to the second next team-mate
- Series 7 - angle to the second next team-mate
- Series 8 - side of the passer
- Series 9 - x-position of the ball

The distances and angles are always relative to the ball carrier because he makes his decisions based on his own perceptions. Because of the reasons described above, the angles are horizontally flipped in one half of the field, hence, the angles towards the middle of the field are always negative while the angles towards the sidelines are positive. After 1000 cycles the rules are generated from data shown in table 2.

Table 2. Typical input data for the learning algorithm w.r.t a pass

	Series 0	Series 1	Series 2	Series 3	Series 4	...	Series 9	Class
1	9.01388	-2.33172	10.5	-1.5708	10.1623	...	0	0
2	9.01388	-2.33172	10.5	-1.5708	10.1623	...	0	3
3	9.01388	-2.33172	10.5	-1.5708	10.1623	...	0	3
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮
1000	7.93404	0.904754	8.04407	1.64693	8.92419	...	-35.3793	1

The first tests showed that a pass is a frequent event, but owing to the short period of time of the actual passing, the total duration of the 'passing classes' is too short. In less than 10% of the time an actual pass is happening. But this is not sufficient to produce good results.

To get rules about the behavior of the opponent from such a rare event with the described decision tree method no error-based pruning can be done. Splitting the samples into the two not-passing classes dribbling and ball with the other team, leads to rules which are in more than 90% correct. With these values the error used to end the decision tree algorithm must be noticeably smaller than 10%.

```
(...)

(1)  if   series 8 > 1.000000
     and  series 0 < 1.530345
     and  series 0 < 0.862784
     and  series 2 < 3.559328
     then 0(1) 1(0) 2(0)

(2)  if   series 8 > 1.000000
     and  series 0 < 1.530345
     and  series 0 < 0.862784
     and  series 2 > 3.559328
     then 2(0.589744) 1(0.282051) 0(0.128205)

(3)  if   series 8 > 1.000000
     and  series 0 < 1.530345
     and  series 0 > 0.862784
     then 1(0.489796) 0(0.346939) 2(0.142857)

(4)  if   series 8 > 1.000000
     and  series 0 > 1.530345
     then 1(0.797297) 0(0.13964) 2(0.05630063)
```

Fig. 7. The passing behavior of the Brainstormers in a game against Robolog

However, if the according threshold is set to such a low value problems of overfitting occur. This means that the necessary generalization is lost and the tree exactly learns the samples given to the algorithm. But this is not what we want because the results should be used to adapt the own team to the opponent. Overfitted rules describe how the opponent has acted in special situations but not how his general behavior operates. This cannot be used to predict the future behavior of the opponent. Thus, another way to improve the results must be found.

The results of a learning algorithm can also be changed by modifying the input values. In this case the problem is obviously the qualitative abstraction. It does not assign enough pass classes. This is a result of the shortness of the pass event. So if there would be a possibility to increase the duration of such an event this should improve the results noticeably. A close look at the game reveals that the positions of the players does not change much between two cycles, thus the environment short before and after the pass is very similar to the one at the pass itself. Hence, they can also be assigned to the same pass class. As a result the pass event is not longer analyzed, but the situation leading to a pass. If the two cycles before and after the actual event are also assigned to the class, the decision tree can be built without the overfitting problem.

To test the method on passes again Robolog Koblenz, Karlsruhe Brainstormers and VirtualWerder were used. Again, ten games were played between Robolog and Brainstormers, VirtualWerder and Robolog and VirtualWerder and Brainstormers. The game Robolog against Brainstormers revealed for the pass behavior of the Brainstormers rules as in figure 7. The first rule showed that the Brainstormers had problems if they were attacked by two players (series $0 < 0.9$, series $2 < 3.6$). In this case they always lost the ball. But if the second opponent was more than 3.6m away they only lost the ball 12% of the time. The small value used to split series 0 shows that the Brainstormers react very late. The ball is already in the kickable area of the attacker.

In the ten games between Robolog and Brainstormers there were always similar values in the rules, but not all at all times and not in the same order. Although very similar rules to those above could be found in the half of all games, the values just differed slightly .

While analyzing the passing behavior of Robolog, the coach found rules like in figure 8. These rules reveal that Robolog tends to lose the ball if one of their own players is near to the ball carrier (series $4 < 0.5$m) except if the attacker is coming from behind (series $1 <$ -1.36), in this case they are passing very successfully to a team mate. The problem of two Robolog player close to each other was revealed in nearly every game. It was found in the games against VirtualWerder as well.

```
if   series 8 < 1.000000
and  series 4 < 0.462778
and  series 1 < -1.356436
then 2(1) 0(0) 1(0)

if   series 8 < 1.000000
and  series 4 < 0.462778
and  series 1 > -1.356436
then 0(1) 1(0) 2(0)

if   series 8 < 1.000000
and  series 4 > 0.462778
then 1(0.613426) 0(0.208333) 2(0.157407)

(...)
```

Fig. 8. The passing behavior of Robolog in a game against Brainstormers

The tests with analyzing passes also showed that the difference in the quality of the teams influences the results. It is, however, not nearly as great as with the goalkeeper where it could happen that a goalkeeper did not have to move during a whole game which made an analysis of his behavior impossible. It is not as obvious with the passes because even with a weak opponent all teams still passed. Though VirtualWerder could not put much pressure on the Brainstormers there were only few interceptions and thus they have only seldom appeared in the results.

4.3 Related Work

Similar work has be done by Raines and colleagues (1999). They describe a program called ISAAC for off-line analysis that uses C5.0 and pattern matching to generate rules about the success of individual players and team cooperation in certain key events. Key events are events which directly effect the result of the game. The only one used in ISAAC is a shot at the goal. These events are analyzed, similar to the approach in this paper, with a decision tree algorithm. However, ISAAC has to be is used off-line, thus the program is not able to support real-time conditions. The team cooperation is analyzed by a pattern matching algorithm. This patterns are kicks of the ball by certain players which lead to a goal. The rules produced by ISAAC are intended support the development of the analyzed team. Therefore, they show how successful the team is in certain situations but not in which situations the players show which reaction.

An other off-line approach is described in [Wünstel et al., 2000], it uses self organizing maps to analyze the players movement and the players movement in relation to the ball. The trained map can be used to determine which kind of movement is done how often in a game. The results of this method show which kind of movements a player performs, but not in which situations he is doing so.

In the RoboCup 2000 the VirtualWerder coach [Visser et al., 2001] analyzed the opponent with an artificial neuronal network. The network was trained from log-files, from past competitions, to recognize 16 different team formations. To react on the opposing formation the coach was able to change the behavior of his own players.

Riley and Veloso (2002) use a set of pre-defined movement models and compare these with the actual movement of the players in set play situation. In new set play situations the coach uses the gathered information to predict the opponent agent's behavior and to generate a plan for his own players. The main difference to the approach described in this paper is that they analyze the movement of all players in set play situations, while the decision tree approach analyzes certain behaviors of single players and how successful they are in these situations.

5 Conclusion and future work

We showed that on-line learning is possible in time-critical environments as demonstrated in the simulation league. The idea is to see an object in the game as a time series. We applied a qualitative abstraction of those time series and used a new approach which is able to discreetize the time series in a way that the results are useable for symbolic learning algorithms. We implemented the approach and ran various tests in real games. We were using two scenarios to analyze the behavior of the goalkeeper and the pass behavior of opponent players. The discussion indicated that the knowledge derived from our approach is valuable and can be used for further instructions to players of the analyzed team. At present, results are generated every 1000 cycles, however, this depends on the situation to be analyzed.

At the moment we are developing our on-line coach in a way that he can use the results of the described approach and give instructions to his players. In order to use the collected rules and get advantage from them they still need to be processed in order to generate advice and to transmit it through the standard coach language to the players. For this purpose the rules should be refined. This process should e.g. reduce the number of attributes in a rule. One attribute should occur in one rule more than twice to mark a range of a value. Additionally, rules which cannot be transformed into advice for the own team can be deleted. The approach is very promising and we hope that first results can be seen on-line at the competitions in June.

If we try to generalize our approach the following statements can be made. Firstly, the proposed minimization heuristic has been successfully applied in a scenario for a qualitative substitution of sensors [Boronowsky, 2001]. It has been shown that qualitative cohesions between measurement-value time series are generally valid and that these cohesions can be discovered by automatic knowledge extraction procedures. The heuristic was also employed with respect to a qualitative analysis of a technical system. The scenario is purification of sewage water under experimental conditions dealing with two regulation circuits. They are regulating the pH value and the oxygen content. The results show that the regulation of both the pH value and the oxygen content can be modelled qualitatively.

Secondly, the method can be used to generate more rules skipping pruning techniques. Usually the amount of generated rules have to be decreased in order to derive comprehensible rules. This can done with the help of pruning methods while creating the decision tree. There are also ideas of how to automatically reduce the number of rules after being generated. This is investigated currently in a project and is subject of a masters thesis. However, sometimes pruning is not an option at all, e.g. if the number of generated rules are to small. In this case, we would be able to skip the pruning methods and therefore generate more rules. These rules should then enable a domain expert to either verify given hypothesis or producing hypothesis about a certain domain.

References

[Boronowsky, 2001] Boronowsky, M. (2001). *Diskretisierung reellwertiger Attribute mit gemischten kontinuierlichen Gleichverteilungen und ihre Anwendung bei der zeitreihenbasierten Entscheidungsbauminduktion*. PhD thesis, Department of Mathematics and Computer Science, University of Bremen, St. Augustin.

[Drücker et al., 2002] Drücker, C., Hübner, S., Visser, U., and Weland, H.-G. (2002). "as time goes by" - using time series based decision tree induction to analyze the behaviour of opponent players. In *RoboCup-01, Robot Soccer World Cup V*, Lecture Notes in Computer Science, Seattle, Washington. Springer-Verlag. in print.

[Fayyad and Irani, 1992] Fayyad, U. and Irani, K. (1992). On the handling of continuous-valued attributes in decision tree generation. In *Machine Learning*, volume 8, pages 87–102.

[Murray, 1999] Murray, J. (1999). My goal is my castle – die höheren fähigkeiten eines robocup-agenten am beispiel des torwarts.
http://www.uni-koblenz.de/ag-ki/ROBOCUP/PAPER/papers.html.

[Quinlan, 1993] Quinlan, J. (1993). *C4.5 Programs for Machine Learning*. Morgan Kaufmann.

[Raines et al., 1999] Raines, T., Tambe, M., and Marsella, S. (1999). Automated assistants to aid humans in understanding team behabiors. In Veloso, M., Pagello, E., and Kitano, H., editors, *Proceedings of the Third International Workshop on Robocup 2000, Robot Soccer World Cup IV*, volume 1856 of *Lecture Notes in Computer Science*, pages 85–102, Stockholm, Sweden. Springer-Verlag.

[Riley and Veloso, 2002] Riley, P. and Veloso, M. (2002). Recognizing probabilistic opponent movement models. In *RoboCup-01, Robot Soccer World Cup V*, Lecture Notes in Computer Science, Seattle, Washington. Springer-Verlag. in print.

[Visser et al., 2001] Visser, U., Drcker, C., Hbner, S., Schmidt, E., and Weland, H.-G. (2001). Recognizing formations in opponent teams. In Stone, P., Balch, T., and Kraetschmar, G., editors, *RoboCup 2000, Robot Soccer World Cup IV*, volume 2019 of *Lecture Notes in Computer Science*, pages 391 – 396, Melbourne, Australia. Springer-Verlag.

[Wünstel et al., 2000] Wünstel, M., Polani, D., Uthmann, T., and Perl, J. (2000). Behavior classification with self-organizing maps. In Stone, P., Balch, T., and Kraetschmar, G., editors, *RoboCup 2000, Robot Soccer World Cup IV*, volume 2019 of *Lecture Notes in Computer Science*, pages 108–118, Melbourne, Australia. Springer-Verlag.

Meyer, J., Adolph, R., Stephan, D., Daniel, A., Seekamp, M., Weinert, V. and Visser, U. (2002) Decision-making and Tactical Behavior with Potential Fields, In:Proceedings of Proceedings of the RoboCup-2002: Robot Soccer World Cup VI, (Eds, Kaminka, G., Lima, P. and Rojas, R.), Fukuoka, Japan, pp. 300-307.

Decision-Making and Tactical Behavior With Potential Fields

Jens Meyer, Robert Adolph, Daniel Stephan,
Andreas Daniel, Matthias Seekamp, Volker Weinert, and Ubbo Visser

Department of Mathematics and Computer Science
University of Bremen
Postfach 33 04 40
D-28334 Bremen, Germany
jens@informatik.uni-bremen.de

Abstract. Using potential fields is a technique seldom used in RoboCup scenarios. The existing approaches mainly concentrate on world state representations of single actions such as a kick. In this paper we will show how to apply potential fields to assist fast and precise decisions in an easy and intuitive way. We go beyond the existing approaches by using potential fields to determine all possible player actions, basic and advanced tactics, and also general player behaviors. To ensure fast computing, we mainly use basic mathematical computations for potential field-related calculations. This gives us the advantage of both determining and understanding player actions. Therefore, the integration of future features such as a complex online coach and progressive localization methods will be easier. We implemented the approach in our team **B**remen **U**niversity **G**oal **S**eekers (BUGS) and tested it in numerous games against other simulation league teams. The results show that the CPU-time for decision-making has been decreased significantly. This is a crucial improvement for calculations in time-critical environments.

1 Introduction

The idea to use potential fields is based on retrieving knowledge for the best possible place for an agent to act on. These actions are kick, dribble, and dash, consequently it can easily be adapted to all RoboCup leagues. We are able to represent all possible game situations by taking all necessary information from the already existing world model of CMU-99 and interpreting them as objects in the potential fields. The decision for an action is made by a heuristic based on the determination of the distance to this point. A large distance implies kicking the ball to the point while dribbling would be the action when having a short distance. If we don't have the ball we dash to the target.

There have previously been approaches with regard to potential fields. Similar to electric fields by [Johannson and Saffiotti, 2001] and similar to approaches as described in [Latombe, 1991] we use potential fields to represent world model states. In comparison to the mentioned approaches we focus on the fastest decision-making and general usability possible . This means that we use potential fields to derive any decision that has to be made by an agent. [Nagasaka et al., 2000] use potential fields for actions like a single kick. Our general usability approach goes further. [Johansson, 2001] combines decision-making and navigation in using potential fields. Our approach is similar, however, the difference is the environment: it is real-time, dynamic, and more flexible. Therefore, the processes are more difficult.

2 Using Potential Fields In BUGS

For better understanding of the complex associations discussed later in this paper, we have a closer look towards potential fields and show their flexibility and hidden complexity.

2.1 Basic Use Of Potential Fields

For building a potential field it is necessary to lay a grid upon the soccer field. The grid resolution, although it is customizable, used in the BUGS-client is 60*40, which means $\approx 2m^2$ per grid field. Based on information about all visible moving objects, the game situation and extra knowledge about our own tactic and formation, numeric entries (only integer) in all grid fields are made. The relations between the different aspects are controlled by 15 changeable parameters (which are meant to be online manipulated by the coach, depending on various game statistics).

The point about the speed of our algorithm results from various simplifications in calculations and design of potential fields. One reason is that we don't have functions that will interpolate the resulting potential fields. These interpolations are unnecessary because of the predefined areas of effect of each world object (this operates like *stamps* with integer values). Another reason is using a grid instead of the soccer server coordinates.

Every agent, including the coach, calculates every few cycles (2-8) a potential field based on his own world model. Timing depends on game situation and distribution of CPU-power. Although we have enough CPU power, despite of running all clients on one computer, we tend to keep it well balanced to absolutely guarantee complete decisions for all agents. One starting point only allows the next potential field calculation every other turn, starting with half the agents on an even and the other half on an odd cycle. Situation-based timing is obvious: a ball-leading agent should do calculations every other cycle; a position-holding or adjusting agent, with the ball 60m away, will do so again in about 20 cycles or earlier if the ball comes closer to him.

To decide which action is next, the complete field and some more information (e.g. ball possession and position, own position) are necessary. The best value within the grid always means the best position for the next action. Again, these actions are dashing, kicking, and dribbling. Using only these simple player-actions, the whole space of soccer behavior can be emulated. How far this goes and how it exceeds the obvious will be discussed next.

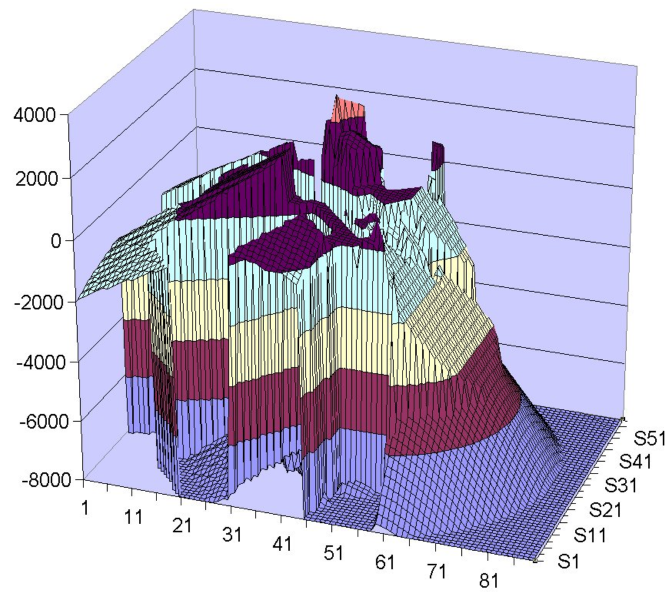


Fig. 1. A typical potential field

Figure 1 shows a typical potential field generated by the ball-leading agent, located on position 32,54. The numerous influences from the other objects on the potential field can be seen (e.g. the cones of the opponents representing the not passable area and also basic potentials indicating the main behaviour). The target point is located on 45,34. Due to the distance between the target and the agents position, the target action is *pass ball*.

2.2 Advanced Use Of Potential Fields

In order to understand the complexity level and the possibilities of potential fields, it is necessary to know their gradual structure. This is the point where a concrete view can be won on later possibilities and implicit conversions of advanced tactics. In fact, the BUGS potential field method includes some tendencies towards planning algorithms. Like a superior plan all clients have a similar basic potential which leads towards the opponent's goal. Each individual action which is decided contains the adherence to these basic guidelines, thus the rough superordinate plan. Whereas following a global intention is not similar to a planning algorithm, viewing all generated potential fields in parallel as one unit means a large step towards a global plan. We need to show the interaction between single potential fields. There are two reasons for the fields to interact with each other. The first reason is rather trivial. Each single field contains its player position such as offense, left mid-field, etc. We get tactical formations owing to tuning these positions and possibly adjusting them to recognized opponent positions (see section 2.4). The second reason seems to be trivial too, but has non-obvious consequences: every potential field is quite similar to the fields generated by neighbor agents, thus based on (nearly) the same inputs which generates similar results. These results are only altered by their own positions and the individual noise transmitted by the soccer server. All agents building potential fields at the same time, each with its own view of the same situation, permanently influence each other with their decisions. While one player holds the ball, the others take position to be passable. This behavior results in building a complete way for the ball into the opponent's goal for most of the time while in ball possession. Due to interceptions, however, most of the time this scenario will not work ; thus, alternatives are created at any time. Similar to planning algorithms we determine sets of action based on the current situation. This might be dangerous because our algorithm has not really a similarity with any planning algorithms from the implementation point of view but in some way the rudimentary behavior is the same, especially for the RoboCup simulation league where world model states and conditions for decision-making change quickly.

2.3 Example for advance use

As we described above, we can assign special values to areas in the grid to gain a special behaviour. The following example shows how it works and gives some views on other tactics, which we can evoke by assigning values to the grid.

Offside A very important tactic in soccer is the use of the offside rule against the other team. Many teams use this tactic to gain free kicks and to interrupt opponents offense easily. Many teams have problems either by setting an offside trap or by recognizing the opponents offside trap. With an potential field we can assign negative potentials to either the own offside area or the opponents offside area. If we assign these potentials to the own offside area we achieve an offside trap. Due to the negative potential in this area, no field player will move into this area on his own. The major exception to this rule is the ball interception after the ball enters this area. Similar happens on the opponents offside area. We assign negative values to this area and achieve that no agent stays in or moves into this area if he don't have the ball or if the ball is already in this area.

Further Examples The method described in the last section can be used on all possible tactical areas. To build up an offensive strategy on the field edges we can simply assign positive potentials in these particular regions. If we want an agent to stay in a specific area (e.g. its position in the team), we can assign negative values to areas outside its tactical area or assigning positive values to his tactical area.

We added some additional points of possibilities for assigning values to this section. This is just a small list, which should show the power of assigning values to the grid within the potential fields:

– The own penalty area is an area where the ball shouldn't stay too long. By assigning a negative value to this area we can achieve that the ball is kicked outside this area quickly if an agent has the ball. Because of the negative value in this area, his target point automatically is set outside this area.
– Assigning positive values to the opponents penalty area and goal. The attraction to this points is high enough to let the attacking agents move to and kick to this specific area.

A very important aspect to the value assigning is the online coach which we plan to use (discussed in the next section). With his clear view onto the game he can gain statistics about the game. So he can easily assign basic values to specific areas for all, some and even single agents. We developed a coach language where we can encode data for assigning values to the agents. The coach is able to get information from his statistics which tells him, what areas of the field is mostly used by the opponents. By assigning positive values to this areas, the agents will be able to intercept the ball or the opponents agents earlier.

2.4 Influence of the tactical online coach

We develop a tactical online coach whose purpose is the statistical evaluation of both our own and the opponent's team. In addition, it will log frequency points of position of all moving objects. Both will be used for game evaluation, which is necessary to re-distribute player-resources, change tactics, and re-arrange player formations. Statistic variables are

– ball losses,
– percentile ball possession,
– percentile ball position per team section (defense, mid-field, offense),
– number of wrongly passed balls,
– gaining of ground and some other variables.

These variables show the quality of each team section and in addition its relative efficiency. Based on these values we will modify various player settings, including player type, position, relations between objects in the potential field or tactics for a single agent, and additionally player formation for a team section or the whole team. All of these changes have an influence on the potential fields, changing tactics for example may tilt the whole field (as described above), formations will simply set new orientation points for the agents, which center the agent's preferred area of action. Special attention should be given to the changes of object relations in the potential field because this is the most subtle way to change behavior, although it could have the greatest effects. Here is an example: raising ball priority will probably do nothing because it is already very high, raising team mate priority slightly may result in passing the ball for a little more percentage rather than dribbling with it. A medium change in opponent priority can change the whole game. Raising it will give an evasive play, lowering may result in nearly ignoring (as long it is possible) while in ball position. Sometimes a change in relations has unpredictable consequences, which makes this way of influence as dangerous as it makes it powerful.

3 Evaluation and Results

The adjustment of the priorities for the evaluation algorithm as described above was probably the hardest work. For this we developed a tool which shows the calculated potential fields of all agents. Also, this enables us to identify errors in priority and to change the potential fields in a way that they fulfill the requirements of our original intentions. We are also able to locate errors in priorities and to bring the real potential fields towards our original intentions. Our agents were running on a Pentium II 400 Mhz Processor with 128 MB of RAM located at the computer pool of the Department of Mathematics and Computer Science at our University. The operating system on these machines is RedHat Linux 7.2. The following table 1 shows our performance test based on a tool called *gprof*. This GNU-tool produces an execution profile of C or C++ programs. All values in the table refer to a complete game. The first column describes percentage of the total running time of the program used by this function. The second column describes the number of seconds counted for this function alone. The third column describes the total number of calls. The last column contains the function's names. Both rows are the most evoked functions of our agent. The function named *estimate_future_pos(...)* is a CMU-function mostly used by the world model itself. The function in the second row is the function which is used to generate potential fields. The result shows that our complete potential field generation uses less than 9% of the time. Until now these figures are difficult to compare in RoboCup scenarios. A comparison of our evaluation algorithm with similar decision algorithms of other teams is difficult because we can't isolate their decision module. The only thing we can compare is the used CPU-time and the amount of memory. The used memory is of lesser interest because there is enough of it available in a tournament. In order to extract these results we simply used **top** (Unix-command) while playing a normal game. Both teams and the soccer server were each running on different computers (the type mentioned above). We repeated each game 15 times and took average values. Karlsruhe-Brainstormers and Mainz-Rolling-Brains ran with the old soccer server v. 7.x, our team and FC-Portugal on soccer server v. 8.x. The use of different soccer servers should not make any difference to the results. The BUGS-team appears twice in the table because of two different grid resolutions to show the relation between resolution and performance. We chose FC-Portugal because it is also based on the CMU-99 sources. Karlsruhe Brainstormers01 was chosen because of its good performance in Seattle, and Mainz-Rolling-Brains completes the list of reference teams. Results are given in the following table 2: We can see that

Table 1. potential field generation based on time and evocation

% time	**self seconds**	**# calls**	**name**
20.62	0.60	288024	estimate_future_pos(...)
8.59	0.25	1526	getEvaluatedAction(...)

Table 2. Best performance test based on a time evaluation relation for the algorithm

Team	Max CPU	Min CPU	Min Memory	Max Memory
FC Portugal 00	12.0%	0.3%	0.5%	1.0
BUGS(90*60)	7.6%	$< 0.1\%$	0.7%	0.8%
BUGS(60*40)	4.6%	$< 0.1\%$	0.6%	0.7%
K. Brainstormers 00	9.8%	0.1%	2.0%	2.1%
Mainz Rolling Brains 00	5.1%	1.5%	1.1%	1.1%

our team BUGS has the best performance with regard to the maximum CPU time used with a grid-resolution of 60*40. It uses only between 40 - 64% of the time that FC Portugal needs and is twice as fast as the Brainstormers, again, with a grid-resolution of 60*40. Similar relations can be seen in the column 'minimum used CPU' where the BUGS team uses less than 0.1%. Here, the team from Mainz has the highest values with 1.5%. As far as memory is concerned, we can note that the Brainstormers always use the same amount of memory. This is probably due to the fact that they are completely based on artificial neural networks. The same relation between maximum and minimum memory used also holds for the BUGS team.It remains constant at a low rate. Only the team from FC Portugal shows a difference in the memory. This indicates that they use various techniques for decision-making. Although we used more than twice the original field size, we were still performing well .

4 Conclusion

We proposed a new idea using potential fields to represent all game situations. In addition to similar approaches such as [Nagasaka et al., 2000] we employ potential fields for *all* possible actions, not only for a kick. They are also used to decide which action to take and to judge the current situation. This method is both intuitive and fast. The main advantage is that we are able to use a single algorithm to determine the agent's action ("One algorithm to fit them all"). Another advantage is the waiving of complex rules and algorithms.

Potential field can be employed to find a teammate to pass the ball as well as to find a position a teammate will pass to. Using an online coach in the near future would make the decision even better. With a coach we are able to give simple advises to the playing agents. Additionally, we can pass messages to single agents indicating specific positions, which makes the potential field even exacter. We use the potential field approach in our own team in the simulation league scenario. At present, we can't make a significant statement about the quality of this decision. However, we have shown that the decision we determined is done due to an easy and especially fast algorithm. Both the CPU-time and the memory used by an agent is very low.

Acknowledgement

We thank the Carnegie Mellon team for letting us use their basic client sources.

References

[Johannson and Saffiotti, 2001] Johannson, S. J. and Saffiotti, A. (2001). Using the electric field approach in the robocup domain. In Birk, A., Coradeschi, S., and Tadokoro, S., editors, *The RoboCup 2001 International Symposium*, Lecture Notes in Artificial Intelligence, Seattle, WA. Springer. in print.

[Johansson, 2001] Johansson, S. J. (2001). An electric field approach - a strategy for sony four-legged robot. M.sc.thesis mse-2001-03, Blekinge Institute of Technology, Sweden.

[Latombe, 1991] Latombe, J. C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Boston, USA.

[Nagasaka et al., 2000] Nagasaka, Y., Murakami, K., Naruse, T., Takahashi, T., and Mori, Y. (2000). Potential field approach to short term action planning in robocup f180 league. In Stone, P., Balch, T., and Kraetschmar, G., editors, *RoboCup Symposium 2000, Robot Soccer World Cup IV*, volume 2019 of *Lecture Notes in Artificial Intelligence*, pages 345–350, Melbourne, Australia. Springer.

An Architecture for a National RoboCup Team

Thomas Röfer

Center for Computing Technology (TZI), FB3, Universität Bremen
Postfach 330440, 28334 Bremen, Germany
roefer@tzi.de

Abstract. This paper describes the architecture used by the GermanTeam 2002 in the Sony Legged Robot League. It focuses on the special needs of a national team, i.e. a "team of teams" from different universities in one country that compete against each other in national contests, but that will jointly line up at the international RoboCup championship. In addition, the tools developed by the GermanTeam will be presented, e.g. the first 3-D simulation used in the Sony Legged Robot League.

1 Introduction

All RoboCup leagues share the problem that there are more teams that want to participate in the world championship than a normal contest schedule can integrate. Therefore, each league has its own approach to limit the number of participants to a certain amount. For instance, teams in the simulation league have to qualify by submitting a team description and two log files, one of a game against a strong opponent, the other of a game against an average one. Based on this material, a committee selects the teams for the championship. In contrast, participants in the Sony Legged Robot League only qualify by submitting a description of their scientific goals—before they have even worked with the robots. Each year, the Sony Legged League grows a little bit, e.g. from 16 to 19 teams from 2001 to 2002. So far, teams that were once accepted in the league were allowed to stay during the following years without a further qualification, amongst other reasons because of their investment in the robots. Thus, only a few new teams have the chance to join the league.

Currently, it is discussed how to provide to chance to participate in the league to more research groups. One solution would be to set up *national teams*, as the GermanTeam that was founded in 2001 [1]. The GermanTeam currently consists of students and researchers at five universities: the Humboldt-Universität zu Berlin, the Universität Bremen, the Technische Universität Darmstadt, the Universität Dortmund, and the Freie Universität Berlin. The members of the GermanTeam are allowed to participate as separate teams in national contests, but will jointly line up at the international RoboCup championship as a single team.

The other solution would be to install national leagues, of which the winning team will get the ticket to participate in the international contest. On the one hand, this approach would enforce the competition, but on the other hand, the goal of the RoboCup initiative is to promote research, and competing teams do not work together very well.

Therefore, it may be a good compromise to support collaboration on the lower, national level, while stressing the element of competition on the international level.

The GermanTeam is an example of a national team. The members will participate as separate teams in the German Open 2002, but will form a single team at Fukuoka. Obviously, the results of the team would not be very good if the members will develop separately until the middle of April, and then try to integrate their code to a single team in only two months. Therefore, an architecture for robot control programs was developed that allows to implement different solutions for the tasks involved in playing robot soccer. The solutions are exchangeable, compatible to each other, and they can even be distributed over a varying number of concurrent processes. The approach will be described in section 2. Finally, in section 3, the tools that were implemented to support the development of the robot control programs are presented.

2 Multi-Team Support

The major goal of the architecture presented in this paper is ability is to support the collaboration between the university-teams in the German national team. Some tasks may be solved only once for the whole team, so any team can use them. Others will be implemented differently by each team, e.g. the behavior control. A specific solution for a certain task is called a *module*. To be able to share modules, interfaces were defined for all tasks that could be identified for playing robot soccer in the Sony Legged League. These tasks will be summarized in the next section. To be able to easily compare the performance of different solutions for same task, it is possible to switch between them at runtime. The mechanisms that support this kind of development are described in section 2.2. However, a common software interface cannot hide the fact that some implementations will need more processing time than others. To compensate for these differences, each team can use its own *process layout*, i.e. they can group together modules to processes that are running concurrently (cf. section 2.2).

2.1 Tasks

Figure 1 depicts the tasks that were identified by the GermanTeam for playing soccer in the Sony Legged Robot League. They can be structured into five levels:

Sensor Data Processing. On this level, the data received from the sensors is preprocessed. For instance, the image delivered by the camera is segmented, and then it is converted into a set of blobs, i.e. image regions of the same color class. The current states of the joints are analyzed to determine the direction the camera is looking at. In addition, further sensors can be employed to determine whether the robot has been picked up, or whether it fell down.

Object Recognition. On this level, the information provided by the previous level is searched to find objects that are known to exist on the field, i.e. landmarks (goals and

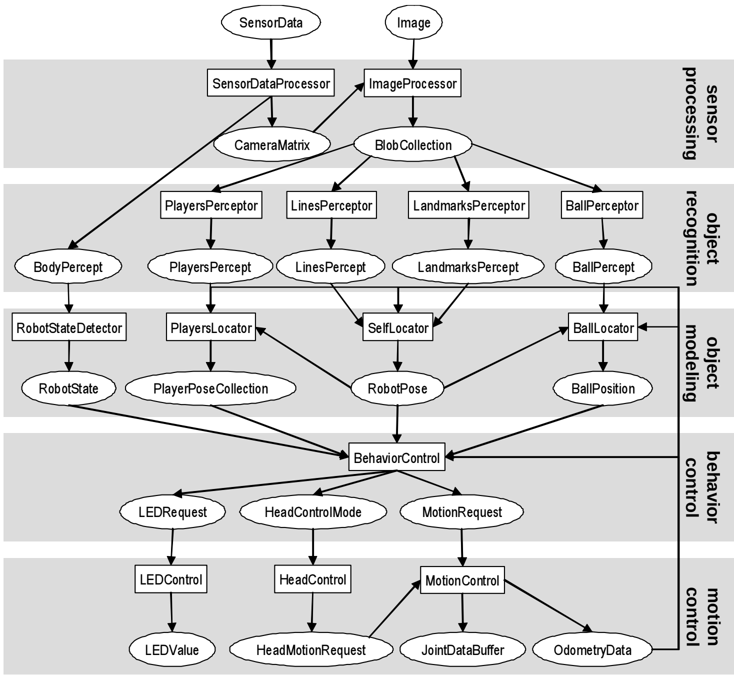


Fig. 1. The modules implemented by the GermanTeam 2002

flags), field lines, other players, and the ball. The sensor readings that were associated to objects are called *percepts*.

Object Modeling. Percepts immediately result from the current sensor readings. However, most objects are not continuously visible, and noise in the sensor readings may even result in a misrecognition of an object. Therefore, the positions of the dynamic objects on the field have to modeled, i.e. the location of the robot itself, the poses (i.e. the (x, y, θ) positions on the field) of the other robots, and the position of the ball. The result of this level is the estimated *world state*.

Behavior Control. Based on the world state and the role of the robot, the fourth level generates the behavior of the robot. This can either be performed very reactively, or deliberative components may be involved. The behavior level sends requests to the fifth level to perform the selected motions.

Motion Control. The final level performs the motions requested by the behavior level. It distinguishes between motions of the head and of the body (i.e. walking). When walking or standing, the head is controlled autonomously, e.g., to find the ball or to look for landmarks, but when a kick is performed, the movement of the head is part of the whole motion. The motion module also performs dead reckoning and provides this information to many other modules.

2.2 Debugging Support

One of the basic ideas of the architecture is that multiple solutions exist for a single task, and that the developer can switch between them at runtime. In addition, it is also possible to include additional switches into the code that can also be triggered at runtime. The realization is an extension of the debugging techniques already implemented in the code of the GermanTeam 2001 [2]: *debug requests* and *solution requests*. The system manages two sets of information, the current state of all *debug keys*, and the currently active solutions. Debug keys work similar to C++ preprocessor symbols, but they can be toggled at runtime. A special infrastructure called *debug queues* is employed to transmit requests to all processes on a robot to change this information at runtime, i.e. to activate and to deactivate debug keys and to switch between different solutions. The debug queues are also used to transmit other kinds of data between the robot(s) and the debugging tool on the PC (cf. section 3). For example, motion requests can directly be sent to the robot, images, text messages, and even drawings can be sent to the PC. This allows to effectively visualize the state of a certain module, textually and even graphically. These techniques work both on the real robots and on the simulated ones (cf. section 3.1).

2.3 Process-Layouts

As already mentioned, each team can group its modules together to processes of their own choice. Such an arrangement is called a *process layout.* The GermanTeam 2002 has developed its own model for processes and the communication between them:

Communication between Processes. In the robot control program developed by the GermanTeam 2001 for the championship in Seattle, the different processes exchanged their data through a shared memory [2], i.e., a blackboard architecture [3] was employed. This approach lacked of a simple concept how to exchange data in a safe and coordinated way. The locking mechanism employed wasted a lot of computing power and it only guaranteed consistency during a single access, but the entries in the shared memory could still change from one access to the other. Therefore, an additional scheme had to be implemented, as, e.g., making copies of all entries in the shared memory at the beginning of a certain calculation step to keep them consistent. In addition, the use of a shared memory is not compatible to the new ability of the Sony Aibo robots to exchange data between processes via a wireless network.

The communication scheme introduced in GT2002 addresses these issues. It uses standard operating system mechanisms to communicate between processes, and therefore it also works via the wireless network. In the approach, no difference exists between inter-process communication and exchanging data with the operating system. A single line of code is sufficient to establish a communication link. A predefined scheme separates the processing time into a communication phase and a calculation phase.

The inter-object communication is performed by *senders* and *receivers* exchanging *packages*. A sender contains a single instance of a package. After it was directed to send the package, it will automatically transfer it to all receivers as soon as they

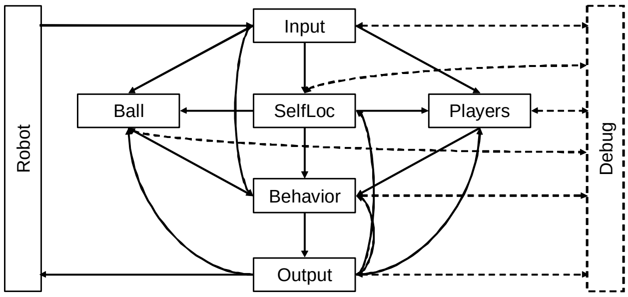


Fig. 2. Process layout of the Bremen Byters. The broken lines indicate the debugging part.

have requested the package. Each receiver also contains an instance of a package. The communication scheme is performed by continuously repeating three phases for each process:

1. All receivers of a process receive all packages that are currently available.
2. The process performs its normal calculations, e.g. image processing, planning, etc. During this, packages can already be sent.
3. All senders that were directed to transmit their package and have not done it yet will send it to the corresponding receivers, if they are ready to accept it.

Note that the communication does not involve any queuing. A process can miss to receive a certain package if it is too slow, i.e., its computation in phase 2 takes too much time. In this aspect, the communication scheme resembles the shared memory approach. Whenever a process enters phase 2, it is equipped with the most current data available.

The whole communication is performed automatically; only the connections between senders and receivers have to be specified. In fact, the command to send a package is the only one that has to be called explicitly. This significantly eases the implementation of new processes.

Different Layouts. The figures 2 and 3 show two different process layouts. Both contain a debug process that is connected to all other processes via debug queues. Note that debug queues are transmitted as normal packages, i.e. a package contains a whole queue. Comparing the two process layouts, it can be recognized that on the one hand, the Bremen Byters try to parallelize as much as possible; on the other hand, Humboldt 2002 focuses on using only a few processes, i.e. the first four levels (cf. Fig. 1) are all integrated into the process *Cognition*. In the layout of the Bremen Byters, one process is used for each of the levels one, four, and five, and three processes implement parts of the levels two and three, i.e. the recognition and the modeling of individual aspects of the world state are grouped together. Odometry is used to decompose information that is dependent: although both the *players* process and the *ball* process require the current pose of the robot, they can run in parallel to the self-localization process, because the odometry can be used to estimate the spatial offset since the last absolute localization. This allows running the ball modeling with a high priority, resulting in a fast update rate, while the self-localization can run as a background process to perform a computationally expensive probabilistic method as, e.g., the one described in [4] or the method used by the GermanTeam 2001 [2].

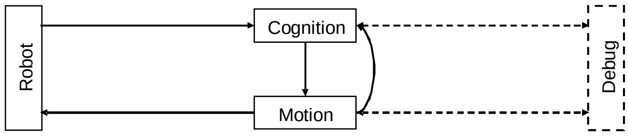


Fig. 3. Process layout of Humboldt 2002. The broken lines indicate the debugging part.

Currently, it is not known which process layout will be the more successful one. The Darmstadt Dribbling Dackels are using a third approach that is a compromise between the two discussed here, and all three will compete against each other at the German Open. So the best can be used for the world championship.

3 Development Tools on the PC

Two tools were implemented on the PC to ease the development of the robot control programs. The first is a 3-D simulator, and the second is a general tool that provides nearly any support imaginable, even the simulator is integrated.

3.1 SimRobot

SimRobot is a kinematic robotics simulator that was developed in the author's group [1]. It is written in C++ and is distributed in public domain [6]. It consists of a portable simulation kernel and platform specific graphical user interfaces. Implementations exist for the *X Window System*, *Microsoft Windows 3.1/95/98/ME/NT/2000/XP*, and *IBM OS/2*. Currently, only the development for the 32 bit versions of Microsoft Windows is continued.

A simulation in SimRobot consists of three parts: the simulation kernel, the graphical user interface, and a controller that is provided by the user. The GermanTeam 2002 has implemented the whole simulation of up to eight robots including the inter-process communication described in section 2.3 as such a controller, providing the same environment to robot control programs as they will find on the real robots. In addition, an object called *the oracle* provides information to the robot control programs that is not available on the real robots, i.e. the robots own location on the field, the poses of the teammates and the opponents, and the position of the ball. On the one hand, this allows implementing functionality that relies on such information before the corresponding modules that determine it are completely implemented. On the other hand, it can be used by the implementators of such modules to compare their results with the correct ones.

3.2 RobotControl

RobotControl is the successor of *DogControl*, the debugging tool used by the GermanTeam in 2001 [2]. Its purpose is to integrate all functionality that is required during the development of the control programs for the Sony Aibo robots.

a) b)

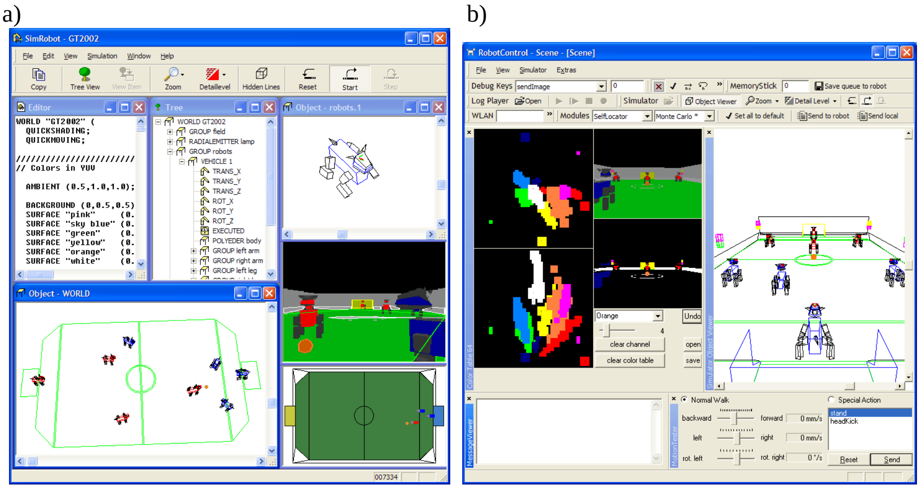


Fig. 4. a) SimRobot simulating the GermanTeam 2002. b) RobotControl: color tool, simulation, message viewer, and motion tester.

Running Robot Controllers. First of all, RobotControl has the ability to run the process-layouts that make up the robot control programs. The simulation kernel of SimRobot is integrated into RobotControl, but in contrast to SimRobot, the robot controller cannot only be provided with simulated inputs. It is also possible to connect it to real robots via the wireless network, and, as a third possibility, the inputs can be generated by replaying log files.

Log Files can either be recorded with a robot, storing them on a memory stick, or the data can be transferred from the robot via the wireless network to RobotControl, and then it will be recorded to a file on the harddisk of the PC. As the same robot controller code runs in all environments, even a simulated robot can produce log files. Log files can contain sensor data and intermediate data as, e.g., blob collections. RobotControl is able to replay log files in real-time or step by step. As RobotControl can run under a debugger, all normal debugging features are available (setting breakpoints, inspecting variables, etc.).

Extensibility. The main purpose of RobotControl is to function as the user interface of the Aibo robots. Therefore, it provides the infrastructure to easily add new toolbars and dialogs. The window layout of RobotControl is always stored and restored on restart. Figure 4b shows a screenshot of RobotControl, in which several toolbars and dialogs can be seen. Toolbars control the replay of log files, they control running the simulation, they allow sending debug keys to real or simulated robots, provide the ability of switching solutions and configuring the wireless network, etc. Dialogs allow generating color tables (for image segmentation, shown in Fig. 4b left), display the simulator scene (Fig. 4b right), control the motions of the robot (Fig. 4b lower right pane), and display debug messages (Fig. 4b lower left pane) and debug drawings (Fig. 4b center). As a result, RobotControl is a very powerful and flexible tool.

4 Conclusion and Future Work

The paper has presented the architecture used by the GermanTeam 2002 in the Sony Legged Robot League. The architecture has been designed for a national team, i.e. a team from different universities that compete against each other in national contests, but that will form a single team at the international RoboCup world championship. The architecture is currently implemented on two different systems, i.e. the Sony Aibo robots and on Microsoft Windows—integrated into the simulator SimRobot and the control software RobotControl. SimRobot is the first 3-D simulator used in the Sony Legged League, and has also been integrated into RobotControl, a universal tool to support the development of the robot soccer team.

Acknowledgements

Although the author contributed to the architecture presented in this paper, namely by realizing the communication scheme and the simulation, the architecture itself is the result of the work of many people in the GermanTeam. From the author's point of view, the main "architects" of the team are Mattias Jüngel, Martin Loetzsch (both Humboldt Universität zu Berlin), and Max Riesler (Technische Universität Darmstadt). The author wants to thank them and also all the other members of the team, who are filling the architecture with life.

The author also thanks the Sony Corporation for their professional support, the Deutsche Forschungsgemeinschaft (DFG) for funding parts of the project through the priority program "Cooperating teams of mobile robots in dynamic environments".

References

1. Burkhard, H.-D., Düffert, U., Jüngel, M., Lötzsch, M., Koschmieder, N., Laue, T., Röfer, T., Spiess, K., Sztybryc, A., Brunn, R., Risler, M., v. Stryk, O.: GermanTeam 2001. Technical report. Only available online: http://www.tzi.de/kogrob/papers/GermanTeam2001report.pdf (2001).
2. Brunn, R., Düffert, U., Jüngel, M., Laue, T., Lötzsch, M., Petters, S., Risler, M., Röfer, T., Spiess, K., Sztybryc, A.: GermanTeam 2001. In *RoboCup 2001*. Lecture Notes in Artificial Intelligence. Springer (2001), to appear.
3. Jagannathan, V., Dodhiawala, R., Baum, L.: *Blackboard Architectures and Applications*. Academic Press, Inc. (1989).
4. Lenser, S., Veloso, M.: Sensor resetting localization for poorly modeled mobile robots. In *Proc. of the IEEE International Conference on Robotics and Automation* (2000).
5. Röfer, T.: Strategies for Using a Simulation in the Development of the Bremen Autonomous Wheelchair. In Zobel, R., Moeller, D. (Eds.): *Simulation-Past, Present and Future*. Society for Computer Simulation International (1998) 460-464.
6. SimRobot homepage. http://www.tzi.de/simrobot.

Vision-Based Fast and Reactive Monte-Carlo Localization

Thomas Röfer†, Matthias Jüngel‡

† Bremer Institut für Sichere Systeme, Technologie-Zentrum Informatik, FB 3, Universität Bremen. E-Mail: roefer@tzi.de
‡ Institut für Informatik, LFG Künstliche Intelligenz, Humboldt-Universität zu Berlin. E-Mail: juengel@informatik.hu-berlin.de.

Abstract— **This paper presents a fast approach for vision-based self-localization in RoboCup. The vision system extracts the features required for localization without processing the whole image and is a first step towards independence of lighting conditions. In the field of self-localization, some new ideas are added to the well-known Monte-Carlo localization approach that increase both stability and reactivity, while keeping the processing time low.**

I. Introduction

THE Sony Four-Legged Robot League is one of the official leagues in RoboCup. In a match, two teams of four robots each compete against each other on a field of approximately $5 \times 3\ \mathrm{m}^2$ in size. All teams consist of robots of the same type, i. e. the Sony Aibo ERS 210 (cf. Fig. 1b). The Aibo is a four-legged robot with 20 degrees of freedom, a color camera, and more than 30 further sensors. The specialty of the league is that, on the one hand, the movements of the robots are the most complex in RoboCup so far, and on the other hand, an on-board camera is the most central sensor. As the robots are only equipped with a 200 MHz processor, all algorithms used, e. g. for image-processing or self-localization, have to be highly efficient to run in real-time.

II. Grid-Based Vision

As the main sensor of the robot is a camera, all objects on the RoboCup field are color coded. There are two-colored flags for localization (pink and either yellow, green, or skyblue), the two goals are of different color (yellow and skyblue), the ball is orange (as in all RoboCup leagues), and the robots of the two teams wear tricots in different colors (red and blue).

A very common preprocessing step for vision-based object recognition in such scenarios is color segmentation using color tables, e. g. [1], [2]. Such methods directly map colors to color classes on a pixel by pixel basis, which has some crucial drawbacks. On the one hand, the color mapping has to be adapted when the lightning conditions change, on the other hand, the mapping results in a loss of information, because the membership of a pixel in a certain class is a yes/no decision, ignoring the influences of the surrounding pixels. Some researchers try to overcome

The Deutsche Forschungsgemeinschaft supports this work through the priority program "Cooperating teams of mobile robots in dynamic environments".

a) b)

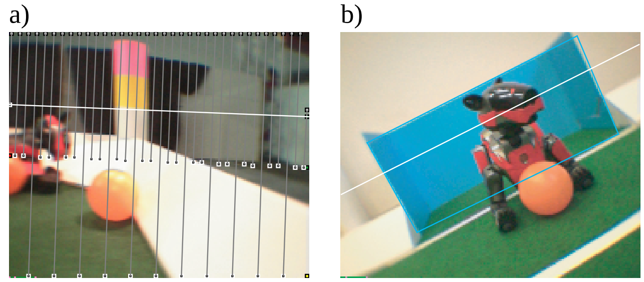


Fig. 1. Images taken by the robot's camera. a) Pink-yellow flag and the horizon-aligned grid. The U-channel of the part of the image near the flag is shown as height map in figure 2d. b) Skyblue goal and Sony Aibo robot.

these limitations [3], but the solutions are too slow to work under real-time conditions on a robot such as the Aibo.

In this paper a method is presented how to detect features in images very quickly without color calibration.

A. Approach

The key ideas of the image-processing method presented in this paper are that speed can be achieved by avoiding to process all pixels of an image, and a certain independence of the lighting conditions can be reached by focusing on contrast patterns in the three different color channels. In case of the Aibo, the camera takes YUV-images. From these images, objects such as flags, goals, and field lines have to be extracted. For such a feature extraction, a high resolution is only needed for far away and thus small objects, but such far away objects cannot appear anywhere in an image, instead they will be close to the *horizon*. The horizon is the intersection of the plane parallel to the ground on the height of the robot's camera and the projection plane. Following this approach, only regions near the horizon need to be scanned at high resolution, while the rest of the image can be searched using a relatively wide spaced grid (cf. Fig. 1a).

The position of the horizon in the image can be calculated from the rotation of the head and the rotation of the body. The roll and tilt of the body are estimated from the readings of the robot's acceleration sensors indicating the direction of the gravity, while the rotation of the head is determined from the angles of the three head joints (tilt, pan, and roll).

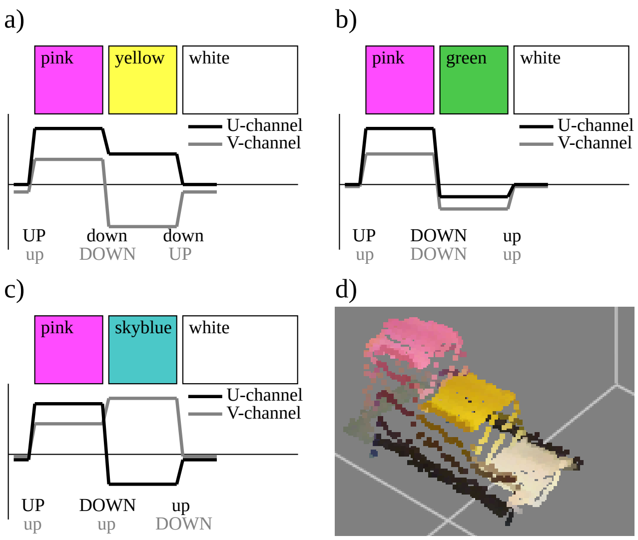


Fig. 2. Detection of flags. a) Pink-yellow flag. b) Pink-green flag. c) Pink-skyblue flag. d) U-channel of a pink-yellow flag as height map

B. Flags and Goals

To detect the flags in the image the vertical grid lines are scanned from top to bottom. During this scan the increase and decrease of the values of the two color channels containing information about the hue (U- and V-channel) is observed. Each position on the grid line with a large or a very large increase or decrease in one of the channels is marked with a *change marker* (up, UP, down, DOWN). This leads to a characteristic sequence of change markers for each grid line. Each flag is characterized by a 3×2 *pattern* containing the expected change markers at the upper edge of the flag at the color change inside the flag and at the lower edge of the flag for both color channels (cf. Fig. 2). Thus a simple pattern matching algorithm is sufficient to detect all flags.

Flags and goals found in an image are represented by the four angles describing their bounding rectangle (top, bottom, left, and right edge) in a system of coordinates that is parallel to the field. The angles to the top and the bottom of the flag are determined from the positions of the first and the last change markers of that flag. To determine the angles to the left and the right edge of the flag, a new scan parallel to the horizon is started from the center of the section with higher contrast to the background, i. e. the pink one, in both directions. The first large decrease in the U-channel on this scan line marks the "end" of the flag and provides a point on a vertical edge.

As goals are coded with only one color they are characterized by a 2×2 *pattern* containing the expected change markers at the upper and lower edges for both color channels. As such small patterns sometimes might match at noise in the image, all matches at the vertical scan lines are compared horizontally to filter the errors. This also provides the leftmost and the rightmost scan line intersecting

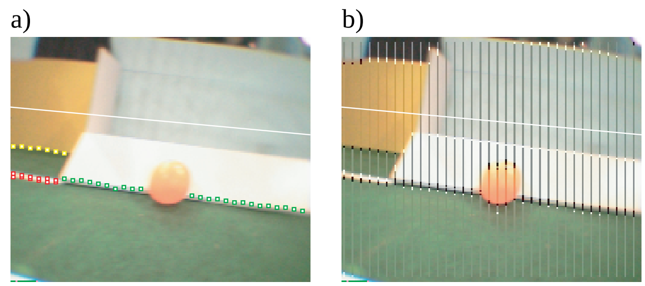


Fig. 3. Detection of lines. a) Three types of lines: field/goal, field/border, and field/line. b) The vertical scan lines are scanned from top to bottom. White pixels: increase in Y-channel, black pixels: decrease in Y-channel.

the goal. The exact horizontal extension of the goal is determined with two more horizontal scans near these scan lines. The angles to the top and the bottom of the goal are determined from the positions of the first and the last change markers of that goal; the angles to the left and the right edges result from the pixels found on the vertical edges.

C. Lines

Four different types of lines can be detected on the RoboCup field: edges between the skyblue goal and the field, edges between the yellow goal and the field, edges between the border and the field, and edges between the field lines and the field (cf. Fig. 3a). The key idea of the method presented here is not to actually extract *lines* from the image, but *pixels on lines* instead. This approach is faster and more robust against misinterpretations, because lines are often partially hidden either by other robots or due to the limited opening angle of the camera.

To find pixels on edges, in the image vertical lines having a distance of five pixels to one another are scanned from top to bottom following the method described in [4] (cf. Fig. 3b). In contrast to this method no color classification is applied. As the green of the field is very dark, all edges are characterized by a big difference of the Y-channel of adjacent pixels. An increase in the Y-channel followed by a decrease is an indication for an edge.

If the color above the decrease in the Y-channel is skyblue or yellow, the pixel lies on an edge between a goal and the field. The differentiation between a field line and the border is a bit more complicated. In most of the cases the border has a bigger size in the image than a field line. But a far distant border might be smaller than a very close field line. For that reason the pixel where the decrease in the Y-channel was found is assumed to lie on the ground. With the known height and rotation of the camera the distance to that point is calculated by projecting it to the ground plane. The distance leads to expected sizes of the border and the field line in the image. For the classification these sizes are compared to the distance between the increase and the decrease of the Y-channel in the image. The projection of the pixels on the field plane is also used to determine their relative position to the robot.

III. Self-Localization in RoboCup

An approach to self-localization is the so-called Monte-Carlo localization (MCL) by Fox *et al.* [5]. It is a probabilistic method, in which the current location of the robot is modeled as the density of a set of particles (cf. Fig. 5a). Each particle can be seen as the hypothesis of the robot being located at that position. Therefore, such particles mainly consist of a robot pose (x, y, θ), i. e. a vector representing the robot's x/y-coordinates and its rotation θ.

In many implementations, MCL was used on robots equipped with distance sensors such as laser scanners or sonar sensors, e. g. in the original one [5]. Only in a few approaches, vision is used for self-localization [6], [7]. Self-localization in RoboCup is different, because the area the robots can be located at is relatively small, i. e. the field, but in that area the position of the robots has to be determined quite precisely to allow different robots of the same team to communicate about objects on the field, and to follow some location-based rules of the game. Odometry is very unreliable, because the robots walk, and they tend to push each other around. As the Aibo is equipped with a sensor with a narrow opening angle of 58°, only a few objects usable for self-localization can be seen at once, and sometimes misreadings are in the majority. The method presented here takes these circumstances into account.

A. Monte-Carlo Localization

A Markov-localization method requires both a motion model and an observation model. The motion model expresses the probability for certain actions to move the robot to certain relative positions. The observation model describes the probability for taking certain measurements at certain locations.

The localization approach works as follows: first, all particles are moved according to the motion model of the previous action of the robot. Then, the probabilities p_i are determined for all particles on the basis of the observation model for the current sensor readings. Based on these probabilities, the so-called *resampling* is performed, i. e. moving more particles to the locations of samples with a high probability. Afterwards, the average of the probability distribution is determined, representing the best estimation of the current robot pose. Finally, the process repeats from the beginning.

The rest of this section will describe all these steps except from the observation model. This is described in two different versions in the sections IV and V.

B. Motion Model

The motion model represents the effects of actions on the robot's pose. First of all, an odometry position is maintained that is derived from the motions performed (gaits, kicks, etc.). As this value is only a rough estimate, in addition a random error Δ_{error} is assumed that depends on the distance travelled and the rotation performed since the last self-localization. For each sample, the new pose is determined as $pose_{new} = pose_{old} + \Delta_{odometry} + \Delta_{error}$. Note that the operation $+$ involves coordinate transformations based on the rotational components of the poses.

C. Resampling

In the resampling step, the samples are moved according to their probabilities. There is a trade-off between quickly reacting to unmodeled movements, e. g., when the referee displaces the robot, and stability against misreadings, resulting either from image processing problems or from the bad synchronization between receiving an image and the corresponding joint angles of the head. Therefore, resampling must be performed carefully. One possibility would be to move only a few samples, but this would require a large number of particles to always have a sufficiently large population of samples at the current position of the robot. The better solution is to limit the change of the probability of each sample to a certain maximum. Thus misreadings will not immediately affect the probability distribution. Instead, several readings are required to lower the probability, resulting in a higher stability of the distribution. However, if the position of the robot was changed externally, the measurements will constantly be inconsistent with the current distribution of the samples, and therefore the probabilities will fall rapidly, and resampling will take place.

The filtered probability p' is calculated as

$$p'_{new} = \begin{cases} p'_{old} + 0.1 & \text{if } p > p'_{old} + 0.1 \\ p'_{old} - 0.05 & \text{if } p < p'_{old} - 0.05 \\ p & \text{otherwise.} \end{cases} \quad (1)$$

Resampling is done in two steps: First, the samples are copied from the old distribution to a new distribution. Their frequency in the new distribution depends on the probability p'_i of each sample, so more probable samples are copied more often than less probable ones, and improbable samples are removed. In a second step that is in fact part of the next motion update, the particles are moved locally according to their probability. The more probable a sample is, the less it is moved. This can be seen as a probabilistic random search for the best position, because the samples that are randomly moved closer to the real position of the robot will be rewarded by better probabilities during the next observation update steps, and they will therefore be more frequent in future distributions. The samples are moved according to the following equation:

$$pose_{new} = pose_{old} + \begin{pmatrix} \Delta_{trans}(1 - p') \times rnd \\ \Delta_{trans}(1 - p') \times rnd \\ \Delta_{rot}(1 - p') \times rnd \end{pmatrix} \quad (2)$$

rnd returns random numbers in the range $[-1 \ldots 1]$. Typical values used for Δ_{trans} and Δ_{rot} are 10 cm and 30°.

D. Estimating the Pose of the Robot

The pose of the robot is calculated from the sample distribution in two steps: first, the largest cluster is determined, and then the current pose is calculated as the average of all samples belonging to that cluster. To calculate the largest cluster, all samples are assigned to a grid that discretizes the x-, y-, and θ-space into $10 \times 10 \times 10$ cells. Then, it is searched for the $2 \times 2 \times 2$ sub-cube that contains the maximum number of samples. All samples belonging to that sub-cube are used to estimate the current pose of the robot. Whereas the mean x- and y-components can directly be determined, averaging the angles is not straightforward, because of their circularity. Instead, the mean angle θ_{robot} is calculated as:

$$\theta_{robot} = \text{atan2}\left(\sum_i \sin\theta_i, \sum_i \cos\theta_i\right) \qquad (3)$$

IV. LANDMARK-BASED SELF-LOCALIZATION

Instead of using the distances and directions to the landmarks in the environment, i. e. the flags and the goals, this localization approach only uses the directions to the vertical edges of the landmarks. The advantage of using the edges for orientation is that one can still use the visible edge of a landmark that is partially hidden by the image border. Therefore, more points of reference can be used per image, which can potentially improve self-localization.

A. Flags and Goals

The image-processing described in section II determines bearings on the edges of flags and goals. These have to be related to the assumed bearings from hypothetical positions. To determine the expected bearings, the camera position has to be determined for each particle first, because the real measurements are not taken from the robot's body position, but from the location of the camera. From these hypothetical camera locations, the bearings on the edges are calculated. As the flags are cylinders, the edges of the flags seen are the tangents to these cylinders starting in the camera center. In case of the goals, their front posts are used as points of reference. As the goals are colored on the inside, but white on the outside, the left and right edges of the colored area even correlate to the posts if the goal is seen from the outside.

B. Probabilities

The observation model only takes the bearings on the edges into account that are actually seen, i. e., it is ignored whether the robot has *not* seen a certain edge that it should have seen according to its hypothetical position and the camera pose. Therefore, the probabilities of the particles are only calculated from the similarities of the measured angles to the expected angles. Each similarity s is determined from the measured angle ω_{seen} and the expected angle ω_{exp} for a certain pose by applying a sigmoid function to the difference of both angles:

$$s(\omega_{seen}, \omega_{exp}) = \begin{cases} e^{-\sigma d^2} & \text{if } d < 1 \\ e^{-\sigma(2-d)^2} & \text{otherwise} \end{cases} \qquad (4)$$
$$\text{where } d = \frac{|\omega_{seen} - \omega_{exp}|}{\pi}$$

A typical value for σ is 50. The probability p of a certain particle is the product of these similarities:

$$p = \prod_{\omega_{seen}} s(\omega_{seen}, \omega_{exp}) \qquad (5)$$

C. Inserting Calculated Samples

Landmark-based self-localization allows some samples to be moved to calculated positions. This approach follows the *sensor resetting* idea of Lenser and Veloso [8], and it can be seen as the small-scale version of the Mixture MCL by Thrun *et al.* [9]: on the RoboCup field, it is often possible to directly determine the position of the robot from the bearings on landmarks. The only problem is that these positions are not always correct, because of misreadings and noise. However, if a calculated position is inserted into the distribution and it is correct, it will get high probabilities during the next observation steps and the distribution will cluster around that position. In contrast, if it is wrong, it will get low probabilities and will be removed very soon. Therefore, calculated positions are only position hypotheses, but they have the potential to speed up the localization of the robot.

Two methods were implemented to calculate possible robot positions. They are used to fill a buffer of *position templates*. The first one uses a short term memory for the bearings on the last three flags seen. From the buffer and the actual bearings on goal posts, all combinations of three bearings are used to determine robot positions by triangulation. The second method only employs flags and goals currently seen. It uses all combinations of a landmark with reliable distance information, i. e. a flag, and a bearing on a goal post or a flag to determine the current position. For each combination, one or two possible positions can be calculated.

The samples in the distribution are replaced by positions from the template buffer with a probability of $1 - p'_i$. Each template is only inserted once into the distribution. If more templates are required than have been calculated, random samples are employed.

V. LINES-BASED SELF-LOCALIZATION

The previous observation model uses the colored flags and goals for self-localization. However, there are no flags on a real soccer field, and as it is the goal of the RoboCup initiative to compete with the human world champion in 2050, it seems to be a natural thing to develop techniques

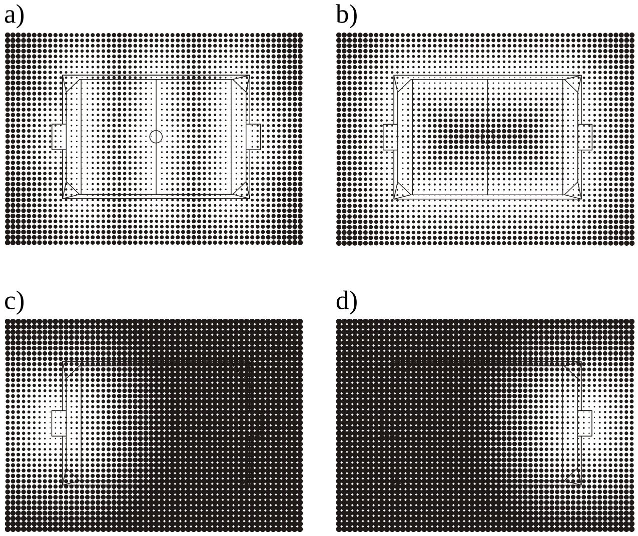


Fig. 4. Distances from lines. Distance is visualized as thickness of dots. a) Field lines. b) Border. c) One goal. d) The other goal.

for self-localization that do not depend on artificial clues. Therefore, a method to use the field lines to determine the robot's location on the field is currently under development, i. e. the approach only differs in the observation model from the one described in section IV.

A. Approach

The localization is based on the pixels on lines determined by the image-processing system (cf. Sect. II-C). Each pixel has a line type (field, border, yellow goal, or blue goal), and by projecting it on the field, a relative offset from the body center of the robot is determined. Note that the calculation of the offsets is prone to errors because the pose of the camera cannot be determined precisely. In fact, the farther away a point is, the bigger the errors will be.

The projections of the pixels are used to determine the probability of each sample in the Monte-Carlo distribution. As the positions of the samples on the field are known, it can be determined for each sample, where the measured points would be located on the field if the position of the sample would be correct. For each of these points in field coordinates, it can be calculated, how far the closest point on a real field line of the corresponding type is away. The smaller the deviation between a real line and the projection of a measured point from a hypothetic position is, the more probable the robot is really located at that position. However, the probability also depends on the distance of the measured point from the robot, because farther points will contain larger measurement errors, i. e. deviations of farther away points should have a smaller impact on the probability than deviations of closer ones.

B. Optimizations

Calculating the probability for all points on lines found and for all samples in the Monte-Carlo distribution would be a costly operation. Therefore the number of points is fixed (e. g. to ten), and these points are selected by random, but according to the following criteria: on the one hand, it is tried to select the same number of points from each line type, because points belonging to border lines and field lines are more frequent, but the points belonging to the goals determine the orientation on the field, because the field is mirror symmetric without the goals. On the other hand, closer points are chosen with a higher probability than farther away points because their measurements are more reliable.

Calculating the smallest distances of a small number of points to the field lines is still an expensive operation if it has to be performed for, e. g., 200 samples. Therefore, the distances are pre-calculated for each line type and stored in two-dimensional lookup tables with a resolution of 2.5 cm (cf. Fig. 4). That way, the distance of a point to the closest line of the corresponding type can be determined by a simple table lookup. Therefore, the method is computationally not slower than landmark-based Monte-Carlo localization.

However, there is a drawback in lines-based self-localization. It is not possible to directly calculate positions from the points on lines, i. e. a sensor resetting localization cannot be performed (cf. Sect. IV-C). However, this shortcoming is partially compensated by the fact that field lines are seen more often than flags.

VI. Results

The image-processing system presented in this paper was used in many RoboCup games of the GermanTeam in the Sony Four-Legged Robot League. Figure 1b shows a typical example of a detected goal, while figure 3a illustrates the results of line recognition.

Figure 5 depicts some examples for the performance of the landmarks-based self-localization using 100 samples. The experiments shown were conducted with SimRobot [10], [11]. The results demonstrate how fast the approach is able to localize and re-localize the robot. At the RoboCup 2002, the method also proved to work on real robots. For instance, before the beginning of a game the robots of the GermanTeam were just started somewhere on the field, and then—while still many people were working on the field—they autonomously walked to their initial positions. In addition, the self-localization worked very well on fields without an outer barrier, i. e. without a white background behind the flags and the goals.

The self-localization using lines is still in an experimental state. At the moment, experiments have only been performed in the simulator, but their results are quite promising. Figure 6 shows an experiment conducted with 200 samples. The method is always able to find the position of

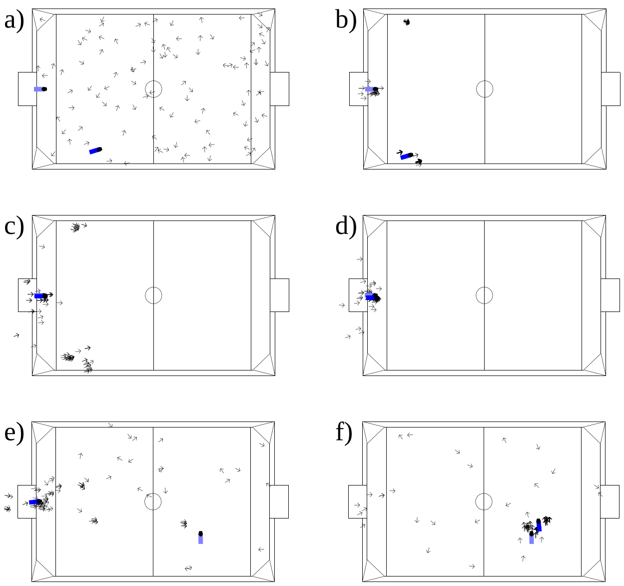


Fig. 5. Distribution of the samples during landmark-based localization while turning the head. The bright robot body marks the real position of the robot, the darker body marks the estimated location. a) After the first image processed (40 ms). b) After eight images processed (320 ms). c) After 14 images (560 ms). d) After 40 images (1600 ms). e) Robot manually moved to another position. f) 13 images (520 ms) later.

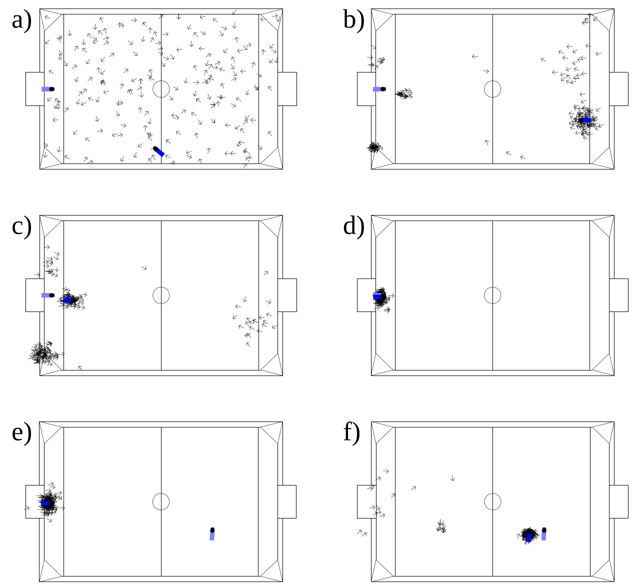


Fig. 6. Distribution of the samples during lines-based localization while turning the head. The bright robot body marks the real position of the robot, the darker body marks the estimated location. a) After the first image processed (40 ms). b) After 16 images processed (640 ms). c) After 30 images (1200 ms). d) After 100 images (4000 ms). e) Robot manually moved to another position. f) 40 images (1600 ms) later.

the robot, but it takes longer than the landmark-based localization. In addition, the mirror symmetry of the field is a problem. In figure 6b, the method first selects the wrong side of the field. However, when a goal comes into view, the position flips over to the correct side (cf. Fig. 6c), and it remains stable. Figures 6e and 6f demonstrate that the approach is also capable of re-localization after the robot was moved manually.

VII. Conclusions

This paper presents two approaches for vision-based self-localization in RoboCup. They are based on a vision system that extracts features without processing whole images, and that has reached a certain independence of lighting conditions. One method uses landmarks for localization, the other is based on field lines. Both approaches are variants of the well known Monte-Carlo localization. While using only a small number of samples, they increase the stability of the localization by a slow adaptation of the probabilities of the samples, and they speed up and increase the precision of the localization by a so-called probabilistic search that moves samples locally dependent on their probabilities. This results in a fast, reactive, and precise self-localization of the robot.

References

[1] J. Bruce, T. Balch, and M. Veloso, "Fast and inexpensive color image segmentation for interactive robots," in *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, 2000, vol. 3, pp. 2061–2066.

[2] F. K. H. Quek, "An algorithm for the rapid computation of boundaries of run length encoded regions," *Pattern Recognition Journal*, vol. 33, pp. 1637–1649, 2000.

[3] Robert Hanek, Thorsten Schmitt, Sebastian Buck, and Michael Beetz, "Fast image-based object localization in natural scenes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2002, Lausanne*, 2002.

[4] M. Jamzad, B. Sadjad, V. Mirrokni, M. Kazemi, H. Chitsaz, A. Heydarnoori, M. Hajiaghai, and E. Chiniforooshan, "A fast vision system for middle size robots in robocup," in *5th International Workshop on RoboCup 2001 (Robot World Cup Soccer Games and Conferences)*. 2002, number 2377 in Lecture Notes in Computer Science, pp. 71–80, Springer.

[5] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo localization: Efficient position estimation for mobile robots," in *Proc. of the National Conference on Artificial Intelligence*, 1999.

[6] F. Dellaert, W. Burgard, D. Fox, , and S. Thrun, "Using the condensation algorithm for robust, vision-based mobile robot localization," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[7] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization for mobile robots using an image retrieval system based on invariant features," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[8] S. Lenser and M. Veloso, "Sensor resetting localization for poorly modeled mobile robots," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[9] S. Thrun, D. Fox, and W. Burgard, "Monte carlo localization with mixture proposal distribution," in *Proc. of the National Conference on Artificial Intelligence*. 2000, pp. 859–865, AAAI.

[10] T. Röfer, "Strategies for using a simulation in the development of the Bremen Autonomous Wheelchair," in *Simulation-Past, Present and Future*, R. Zobel and D. Moeller, Eds. 1998, pp. 460–464, Society for Computer Simulation International.

[11] T. Röfer, "An architecture for a national robocup team," in *RoboCup 2002*. 2003, Lecture Notes in Artificial Intelligence, Springer, to appear (already published in RoboCup 2002: Robot Soccer World Cup VI Pre-Proceedings, 388-395).