Rüdiger Dillmann
Jürgen Beyerer
Uwe D. Hanebeck
Tanja Schultz (Eds.)

LNAI 6359

KI 2010: Advances in Artificial Intelligence

33rd Annual German Conference on AI
Karlsruhe, Germany, September 2010
Proceedings

Springer

Lecture Notes in Artificial Intelligence 6359

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Rüdiger Dillmann Jürgen Beyerer
Uwe D. Hanebeck Tanja Schultz (Eds.)

KI 2010: Advances in Artificial Intelligence

33rd Annual German Conference on AI
Karlsruhe, Germany, September 21-24, 2010
Proceedings

Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Rüdiger Dillmann
KIT
Institute for Anthropomatics
Karlsruhe, Germany
E-mail: ruediger.dillmann@kit.edu

Jürgen Beyerer
KIT
Lehrstuhl für Interaktive Echtzeitsysteme (IES)
Karlsruhe, Germany
E-mail: juergen.beyerer@iosb.fraunhofer.de

Uwe D. Hanebeck
KIT
Lehrstuhl für Intelligente Sensor-Aktor-Systeme (ISAS)
Karlsruhe, Germany
E-mail: uwe.hanebeck@kit.edu

Tanja Schultz
KIT
Cognitive Systems Lab (CSL)
Karlsruhe, Germany
E-mail: tanja@ira.uka.de

Library of Congress Control Number: 2010935082

CR Subject Classification (1998): I.2, H.4, F.1, H.2.8, I.2.6, H.5.2

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-16110-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-16110-0 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

The 33rd Annual German Conference on Artificial Intelligence (KI 2010) took place at the Karlsruhe Institute of Technology KIT, September 21–24, 2010, under the motto "Anthropomatic Systems." In this volume you will find the keynote paper and 49 papers of oral and poster presentations. The papers were selected from 73 submissions, resulting in an acceptance rate of 67%.

As usual at the KI conferences, two entire days were allocated for targeted workshops—seven this year—and one tutorial. The workshop and tutorial materials are not contained in this volume, but the conference website, `www.ki2010.kit.edu`, will provide information and references to their contents. Recent trends in AI research have been focusing on anthropomatic systems, which address synergies between humans and intelligent machines. This trend is emphasized through the topics of the overall conference program. They include learning systems, cognition, robotics, perception and action, knowledge representation and reasoning, and planning and decision making. Many topics deal with uncertainty in various scenarios and incompleteness of knowledge.

Summarizing, KI 2010 provides a cross section of recent research in modern AI methods and anthropomatic system applications.

We are very grateful that José del Millán, Hans-Hellmut Nagel, Carl Edward Rasmussen, and David Vernon accepted our invitation to give a talk.

We cordially thank all colleagues who accepted our invitations and submitted workshop and tutorial proposals and papers. We gratefully acknowledge the work and support of all chairs, especially the area chairs. We would also like to thank the technical committee for their dedication in reviewing the papers. We greatly appreciate the efforts and hard work of the organizing team. Steering meetings to compile the technical program and to organize the conference as a whole were time consuming. The conference management system PaperPlaza was employed, which significantly facilitated the organization of the event from an administrative point of view.

We thank all companies and institutions listed on the following pages, who sponsored this conference. Without their support, the KI 2010 conference would not have been possible. We thank KIT and Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB for hosting the conference. Finally, we would like to thank the publishers for their great support in preparing and printing this volume.

July 2010

Rüdiger Dillmann
Jürgen Beyerer
Uwe D. Hanebeck
Tanja Schultz

Organization

Organization Committee

General Chairs	Rüdiger Dillmann Jürgen Beyerer
Program Chairs	Uwe D. Hanebeck Tanja Schultz
Workshop and Tutorial Chairs	Rainer Stiefelhagen Marius Zöllner
Publicity Chairs	Rudi Studer Alex Waibel
Exhibition Chair	Heinz Wörn
Local Chair	Fernando Puente León
Administrative Consulting	Christine Harms

Local Arrangements

Kristine Back
Patrick Dunau
Ioana Gheta
Dominic Heger
Rainer Jäkel
Martin Lösch
Felix Sawo
Elena Simperl
Marcus Strand

Area Chairs

Michael Beetz	Technical University of Munich
Kasten Berns	Technical University Kaiserslautern
Susanne Biundo	University of Ulm
Cristóbal Curio	MPI for Biological Cybernetics Tübingen
Christian Freksa	University of Bremen
Joachim Hertzberg	University of Osnabrück
Rainer Malaka	University of Bremen
Bernhard Nebel	University of Freiburg
Gerhard Rigoll	Technical University Munich
Helge Ritter	Bielefeld University
Lutz Schröder	DFKI Bremen

Program Committee

Saleh Al-Takrouri
Sven Albrecht
Klaus-Dieter Althoff
Dejan Arsic
Arthur Asuncion
Amit Banerjee
Daniel Beauchêne
Sven Behnke
Vaishak Belle
Brandon Bennett
Martin Berchtold
Ralph Bergmann
Bettina Bläsing
Daniel Borrajo
Gerhard Brewka
Werner Brockmann
Holger Burbach
Wolfram Burgard
Martin Butz
Marc Cavazza
Lawrence Cayton
Daniel Cernea
Girija Chetty
Eliseo Clementini
Daniel Cremers
Kerstin Dautenhahn
Klaus Dräger
Patrick Dunau
Frank Dylla
Stefan Edelkamp
Norbert Elkmann
Dominik Endres
David Engel
Nicholas Evans
Florian Eyben
Amir Massoud Farahmand
Yvonne Fischer
Stefan Funke
Johannes Fürnkranz
Thomas Gärtner
Dirk Gehrig
Christopher Geib
Ioana Gheta
Claudio Gori Giorgi
El Hadji Amadou Gning
Horst-Michael Gross
Martin Günther
Martin Hägele
Fred Hamker
Ronny Hartanto
Malte Helmert
Dominik Henrich
Joachim Hertzberg
Otthein Herzog
Martin Hofmann
Eyke Huellermeier
Christian Igel
Joris Ijsselmuiden
Winfried Ilg
Dominik Jain
Yaochu Jin
Eugen Käfer
Moritz Kaiser
Gabriele Kern-Isberner
Alexander Kleiner
Jens Kober
Roman Kontchakov
Oliver Kramer
Peter Krauthausen
Ralf Krestel
Torsten Kroeger
Rudolf Kruse
Kai-Uwe Kühnberger
Sebastian Kupferschmidt
Sabine Kuske
Bogdan Kwolek
Gerhard Lakemeyer
Tobias Lang
Nicolas Lehment
Benedikt Loewe
Volker Lohweg
Martin Lösch
Bernd Ludwig
Robert Mattmüller
Bärbel Mertsching
Bernd Michaelis

Ralf Möller
Meinard Müller
Heiko Neumann
Hannes Nickisch
Oliver Niggemann
Sandra Paterlini
Angelika Peer
Justus Piater
Robert Porzel
Felix Putze
David Pynadath
Marco Ragni
Anita Raja
Olaf Ronneberger
Thomas Ruehr
Jose M. Sabater
Alessandro Saffiotti
Jan Salmen
Malte Schilling
Ute Schmid
Tim Schmidt
Ulrich Schmucker
Carsten Schürmann
Emrah Akin Sisbot
Jan-Georg Smaus
Luciano Spinello
Jochen Sprickerhof
Steffen Staab
Andreas Starzacher
Rainer Stiefelhagen
Jörg Stückler
Ingo Timm
Volker Tresp
Rudolph Triebel
Thomas Villmann
Frank Wallhoff
Toby Walsh
Matthias Westphal
Thomas Wiemann
Heinz Wörn
Christian Wöhler
Stefan Wölfl
Dirk Wollherr
Diedrich Wolter
Florentin Wörgötter
Andreas Zell
Christoph Zetzsche
Jianwei Zhang

Invited Talks

José del Millán	École Polytechnique Fédérale de Lausanne
Hans-Hellmut Nagel	Karlsruhe Institute of Technology
Carl Edward Rasmussen	University of Cambridge
David Vernon	

Sponsors

ontoprise GmbH, Karlsruhe
PTV Planung Transport Verkehr AG, Karlsruhe
Springer Verlag, Heidelberg

Partners

Fakultät für Informatik, Karlsruher Institut für Technologie (KIT)
Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung (IOSB)
FZI Forschungszentrum Informatik
Gesellschaft für Informatik e.V. (GI)

Table of Contents

Cognition

Human-Machine Interaction

Knowledge

Machine Learning and Data Mining

Planing and Reasoning

Robotics

Special Session: Situation, Intention and Action Recognition

Vision, Logic, and Language – Toward Analyzable Encompassing Systems

Hans-Hellmut Nagel

Fakultät für Informatik, KIT, 76128 Karlsruhe, Germany
(formerly Institut für Algorithmen und Kognitive Systeme IAKS)
nagel@iaks.uni-karlsruhe.de

Abstract. Some time ago, Computer Vision has passed the stage where it detected changes in image sequences, estimated Optical Flow, or began to track people and vehicles in videos. Currently, research in Computer Vision has expanded to extract descriptions of single *actions* or concatenations of actions from videos, sometimes even the description of agent *behavior* in the recorded scene.

This transition from treating mostly *quantitative, geometric* descriptions to becoming concerned with more *qualitative, conceptual* descriptions creates contacts between Computer Vision, Computational Linguistics, and Computational Logic. The latter two disciplines have studied the analysis and combination of conceptual constructs already for decades.

Based on selected examples, attention will be drawn to the potential which can be tapped if the emerging thematic overlap of research in these three disciplines is investigated collaboratively. This applies in particular to the development of encompassing systems which rely on methods from all three disciplines, for example by providing Natural Language interfaces to more generally applicable combinations of Knowledge Bases with Computer Vision systems.

> Logic surely is, like democracy, an unsatisfactory answer to problems of life. Unfortunately, alternatives are worse.
> (based on a cue by W.S. Churchill)

1 Introduction

In order not to mislead potential readers, the title should be interpreted to refer to Computer Vision (CV), Computational Logic, and Computational Linguistics (CL) as an engineering challenge. It is not intended to moonlight in the areas of Biological Vision, of Brain Sciences, Cognitive Sciences, Philosophy, or Linguistics.

Two points of departure have been chosen for the subsequent discussion, one of which considers the evolution of Computer Vision as a discipline in its own right, and a second one associated with a – possibly idiosyncratic – view onto a Human-Machine-Interface (HMI) in cars. Starting with the first aspect provides the background for turning to the second one.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 1–22, 2010.

Initially, most efforts in CV concentrated on the *segmentation* of a static image into components corresponding to *concepts* and on the *recognition* of components which appeared relevant according to a gamut of practical or theoretical reasons. The study of *relations* between concept representations extracted from an image relies on a *stable extraction* process for the concepts themselves and thus took more time to gain momentum. Attempts to detect and study particular *combinations of mutually related concepts* – i.e. the recognition of 'compound concepts' – took even more time to attract broader attention. A similar progression regarding the complexity of research problems can be observed if the time dimension is taken into account. Starting with change detection, research interests widened via 'coherent changes' ascribed to the movement of gray value structures in the image plane ('Optical Flow'), to atomic actions, concatenations of atomic actions, and from there into the broad field of behavior recognition. CV research thus can be seen to gradually progress from the recognition of *atomic concepts* to the study of concept *combinations*, in particular regarding temporal aspects.

The *first tenet* of this presentation holds that (i) the study of concept combinations has been a longstanding topic of both Formal Logic (inferences from axioms) and sentence formation as part of Linguistics and, therefore, (ii) it is worthwhile to look for analogies between research in CV, in Formal Logic, and in Computational Linguistic (CL). This emphasis provides the excuse for skipping the recognition of objects, their attributes, (basic) spatial relations between objects, and *atomic* actions in videos. One might argue that even the recognition of instances of an atomic concept may require the structured composition of elementary pieces of evidence ('features') extracted from an image or an image (sub)sequence. As illustrated by the comprehensive and well-argued recent survey of Poppe [23], the choice of features depends on diverse boundary conditions and idiosyncrasies of individual approaches which makes it difficult to analyse published approaches regarding generally valid *structural* commonalities in the extraction and combination of features.

A *second tenet* of this presentation holds that a *combination* of concepts has to be studied with respect to a well circumscribed 'discourse domain'. This aspect, covered to some extent in the survey of Turaga et al. [30], distinguishes it from the survey of Poppe who restricts himself to treating actions. This second tenet suggests to design, implement, and evaluate an entire algorithmic system capable to handle a substantial fraction of the tasks which are characteristic for the chosen discourse domain. Only then will it be possible to uncover and assess the relative (dis-)advantages of an approach with some confidence.

2 Treating Agent-Related Temporal Variations in 3D-Scenes at a Conceptual Level

Vision is a major source for the avalanche of phenomena which continuously threatens the human capability for a proper reaction. *Concepts* constitute a means to subdivide this avalanche into manageable items. *Language* allows to package these items into a form communicable between humans. *Logic* emerged as an alternative to power[1] in order to prevent an ever growing body of concepts from collapsing under its own weight – in analogy to bones embedded into the tissue of vertebrates.

The following characterisation of candidate discourse (sub-)domains for the algorithmic description of video content is suggested as a starting point for a discussion of *non-stationary scenes*.

1. *Rigid* bodies
   (a) without relevant spatial extent (i.e. trajectories of *point-like* objects), or
   (b) with relevant spatial extent (i.e. which may allow to define characteristic orientations for an object such as its front).
2. *Non*-rigid bodies:
   (a) Deformable bodies;
   (b) Articulated bodies
      - Technical systems;
      - Living systems
         - Animals,
         - Humans.

Following a terse characterisation of temporal variations associated with video recordings of *humans* (last item in the preceding specialisation hierarchy), an example for a temporal combination of declaratively defined concepts will be discussed by comparing two different representational approaches. From there, the discussion will jump to an analogous treatment of concept combinations for *extended rigid bodies* (second item in the preceding specialisation hierarchy), in particular road vehicles.

2.1 Characterisation of Temporal Variations in Videos with Humans

Changes are treated as the *most general concept* for the description of temporal variations.

Movement of a single agent (i) without emphasis on a particular goal pursued by the agent which moves or (ii) without an immediately discernible goal, for example in case of indeterminate, uncoordinated, or nervous movements.

Atomic actions of a single agent Movements which are likely to be performed in order to achieve a goal (this is the specializing requirement which differentiates an action from the more general notion 'movement') and which can not be decomposed into a concatenation of different actions. The latter requirement differentiates an 'atomic' action from a *non*-atomic or *compound* action. Note that the attribute 'atomic' is not bound to a length of the period during which it can be recognized.

Usage of the notion 'action' by Turaga et al. [30] corresponds essentially to what is denoted as atomic action according to the preceding definition.

[1] Is it a mere accident of history that the concepts for both logic and democracy happened to appear at about the same time in the same place?

Poppe [23] uses the terminology introduced and explained by Moeslund et al. [17]: their 'action primitive' corresponds to the notion of atomic action as used here. Their 'action' corresponds to what is introduced in the sequel as compound action, but includes cases where an intention of the agent becomes discernible based on a small sequence of situated actions (see below).

Activity Set of actions by a single agent or more than one agent *without* any reference to *a particular relation* between these actions.

This understanding of the notion activity appears to be compatible with its usage by Robertson & Reid [25] and that by Grimson and coworkers [31]. Turaga et al. use 'activities' to cover all the notions which are differentiated below, starting with compound action.

Note that 'activity' as used by Poppe and Moeslund et al. 'are larger scale events that typically depend on the context of the environment, objects, or interacting humans' [17, p. 110, right column, next-to-last paragraph]). This definition can be understood to cover the situated concatenation of actions (without the finer differentiations which are suggested in the following items).

Single agent *compound* actions constructed based on the concatenation or parallel execution of atomic actions.

Single agent *situated* action execution where conditions can be recognized which influence (enable, modify, or suppress) the execution of an action.

Recognition of *intentions* of a single agent as the situated concatenation of a small number of actions with the potential that the agent achieves an immediately discernable (short-term) goal-state.

Single agent *behavior* Situated concatenation of actions into longer action-sequences in order to achieve a long-term goal. This corresponds, e.g., to the use of the notion 'behavior' by Robertson & Reid [25].

Cooperative behavior of two or more agents

Competition between two or more agents

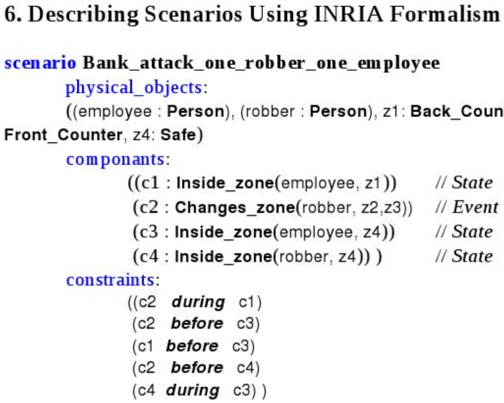
Single-agent *scenario* formulation Integration of potential single-agent behaviors into a coherent framework (i.e. an attempt to collect *all* situated action sequences which are admissible under specified boundary conditions into a unified representation of single-agent behaviors which are considered to be relevant for the chosen discourse domain).

***Multi-agent* scenario formulation**

2.2 On Scenario Representations

The survey on 'Machine Recognition of Human Activities' by Turaga et al. draws attention to the advantage of what the authors refer to as 'activity' definitions which are independent of particular algorithms to instantiate them [30, p. 1482, right column: 2) Ontologies].

Figure 1 shows an excerpt for just a single scenario out of a collection of similar scenarios which have been developed by M. Thonnat and coworkers at INRIA Sophia Antipolis. The predicate-oriented scenario specification should be largely self-explanatory, given the suggestive choices for variable and predicate identifiers.



6. Describing Scenarios Using INRIA Formalism

```
scenario Bank_attack_one_robber_one_employee
        physical_objects:
        ((employee : Person), (robber : Person), z1: Back_Counter, z2: Entrance_Zone, z3:
Front_Counter, z4: Safe)
        componants:
                ((c1 : Inside_zone(employee, z1))       // State
                 (c2 : Changes_zone(robber, z2,z3))     // Event
                 (c3 : Inside_zone(employee, z4))       // State
                 (c4 : Inside_zone(robber, z4)) )       // State
        constraints:
                ((c2  during  c1)
                 (c2  before  c3)
                 (c1  before  c3)
                 (c2  before  c4)
                 (c4  during  c3) )
```

Fig. 1. Copy of the scenario '*Bank_attack_one_robber_one_employee*' from [12, subdomainDrafts/BankMonitoringOntology030910.doc]. This scenario formulation constitutes a part of the documentation about a 'Video Event Challenge Workshop' organized in 2003. The website [12], quoted in [30], comprises additional information about the goal of this endeavor and the conventions which have been agreed on during this workshop.

The appearance of temporal predicates `during` and `before` together with temporal interval identifiers *c1*, *c2*, *c3*, and *c4* indicates a non-standard (time-dependent) logic notation. The declarative structure of this definition stimulated a reformulation of this scenario in the Situation-Graph-Tree (SGT) Formalism – see Figure 2.

The SGT-formalism ([28]) and associated support packages have been developed at the IAKS for the description of, e.g., vehicle behavior extracted from videos of innercity road traffic scenes. As a basis for the subsequent discussion, a terse outline of this formalism has been added in Appendix A, together with references to additional explanations and examples.

The graphical representation of the SGT for this scenario has been generated automatically by the SGTEditor-component (see [1]) based on a textual specification given in Appendix B. The textual specification of an SGT is treated as the canonical one; a graphical representation constructed or modified using the SGTEditor will be converted back to a textual specification in the specification language SIT++ (see [28]). It thus is possible to switch between these representations depending on the preferences for the task to be solved. During the traversal of an SGT based on geometric results obtained by a CV-subsystem, strings will be emitted which subsequently can be converted into a NL-text describing the developments in the video analysed by the CV-subsystem. This illustrates a system structure which facilitates an algorithmical conversion between different representational modalities – (i) a-priori knowledge in the form of a graph like Figure 2, (ii) in a time-dependent logic language like the example in Appendix B, or (iii) a NL-text describing instantiated knowledge obtained by an SGT-traversal.

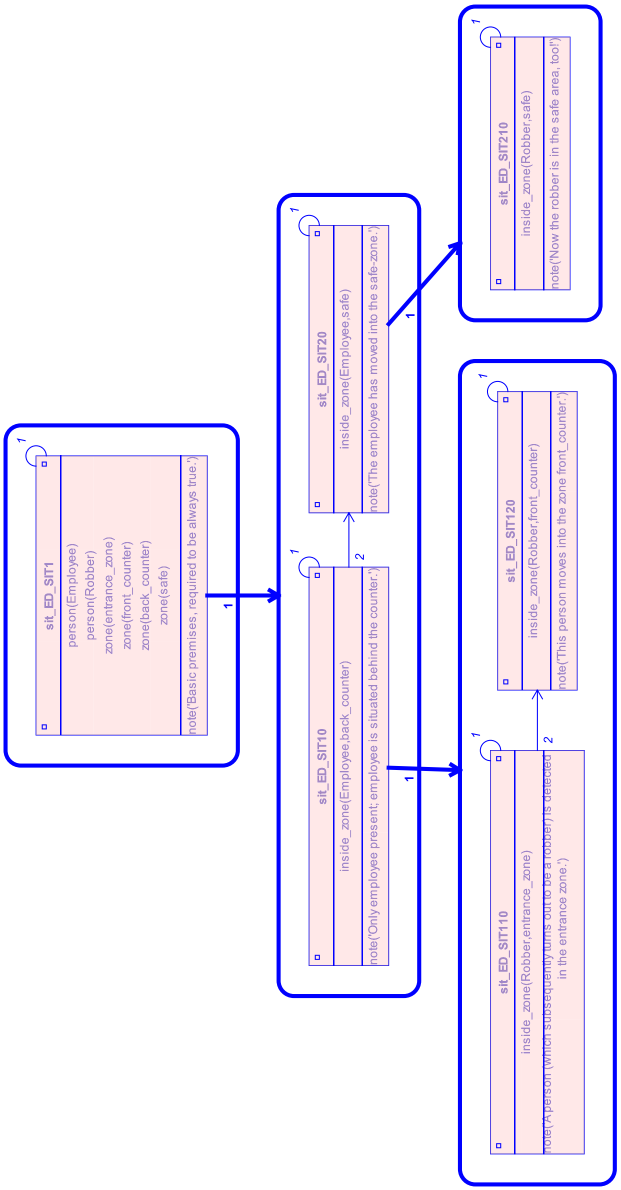

sit_ED_SIT1
person(Employee)
person(Robber)
zone(entrance_zone)
zone(front_counter)
zone(back_counter)
zone(safe)
note('Basic premises, required to be always true.')
sit_ED_SIT10
inside_zone(Employee,back_counter)
note('Only employee present; employee is situated behind the counter.')
sit_ED_SIT20
inside_zone(Employee,safe)
note('The employee has moved into the safe-zone.')
sit_ED_SIT110
inside_zone(Robber,entrance_zone)
note('A person (which subsequently turns out to be a robber) is detected in the entrance zone.')
sit_ED_SIT120
inside_zone(Robber,front_counter)
note('This person moves into the zone front_counter.')
sit_ED_SIT210
inside_zone(Robber,safe)
note('Now the robber is in the safe area, too!')
1
2

Fig. 2. Reformulation of the scenario from Figure 1 as a Situation-Graph-Tree (SGT). The text version (in SIT++) of this graphical presentation can be found in Appendix B. See text for further discussions.

2.3 On Transformations between Different Scenario Representations

The two scenario representations depicted in Figures 1 and 2 will now be compared in more detail. The temporal relations between different 'states' as given by the 'constraints' in Figure 1 disappeared from the SGT-representation in Figure 2. With other words, the *explicated* temporal predicates `during` and `before` in Figure 1 have been incorporated into the tree-*structure* of Figure 2 (or the textual formulation of this scenario in Appendix B).

It occurred to me while preparing this contribution to study the design of an algorithmic (possibly invertible) conversion between a scenario representation with *explicitly* and *implicitly* formulated temporal predicates. Such an *algorithm* would perform for a general case what I performed 'by hand' while translating the scenario-representation from Figure 1 into that of Figure 2 (see, e.g., Appendix C). This idea gave rise to a number of questions which to investigate further I did not yet find time so far:

- Will an analogous transformation be possible for *all* other scenarios documented in [12, subdomainDrafts/]?
- If not, what conditions have to be satisfied in order to allow such a transformation?
- Can one represent a 'storyline' obtained by Machine Learning from annotated videos (see [13]) as a scenario in either of the two modalities mentioned above?
- In case this would appear possible, what needs to be done in order to turn 'learned storylines' *algorithmically* into *logic*-based representations like the ones illustrated above?

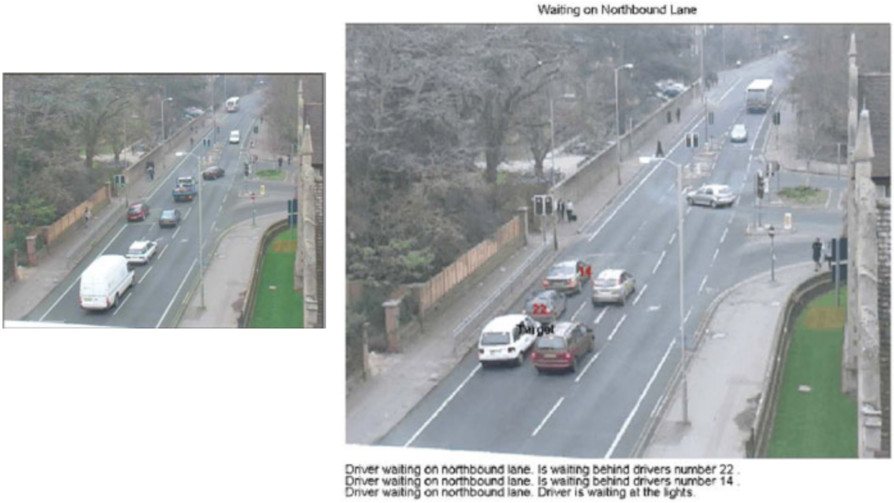
Obviously, the efforts to be invested into an algorithmic conversion between different scenario representations can only be justified if the rules for a representation corresponding to Figure 1 have been laid down precisely, in analogy to the definition of SIT++.

In view of the impression that the participants of the workshop documented in [12] aimed at conventions modeled along the lines of Description Logic (see [3]), it appears tempting to think about a time-dependent Description Logic which comprises provisions for both explicit and implicit temporal predicates for the definition of scenarios. Such a definition would justify to study algorithmic conversion routines between these two scenario representations. The user will then have the option to choose that representation which suits him most for treating a particular problem.

2.4 On Action-Predicates in Scenario Definitions

Inspection of the scenario definition depicted in Figure 1 reveals that it does not contain *movement concepts* for the humans involved. This, of course, is due to the fact that the video-based evidence for a particular temporal development had to be extracted in real-time with insufficient computing power at the time when these scenarios have been developed (about ten years ago, I guess). As a consequence, one attempted to characterize developments by a succession of discrete state-descriptions rather than by descriptions of continuous movements. This observation provides a starting point for another consideration which will reflect on the relation between CV and Formal Logic.

Although investigations have been started already to incorporate action-predicates into an SGT (see [14]), I shall switch the discourse domain from human movements to those of a car treated as a rigid body in order to push the considerations into technical details which can quickly become relevant.



Driver waiting on northbound lane. Is waiting behind drivers number 22 .
Driver waiting on northbound lane. Is waiting behind drivers number 14 .
Driver waiting on northbound lane. Driver is waiting at the lights.

Fig. 3. (Left panel) Representative image frame from the `Banbury Road` traffic video sequences studied by Robertson & Reid at Oxford. (Right panel) Frame from the `queue.avi`-video of the set illustrated by the left panel, showing natural language textual descriptions derived by N. Robertson together with vehicle references overlaid to the image frame [24] (©N.M. Robertson; my thanks to NMR for his permission to reproduce these figures; see, too, [25]).

The right panel in Figure 3 illustrates vehicles waiting in front of a traffic light at a T-intersection. Due to lack of suitable calibration data, we could not track those vehicles immediately using our 3D-tracking approach `Motris` (MOdel-based Tracking in Image Sequences, see [5]). An attempt to provide an SGT-based scenario for geometric results (obtained by a 2D-region-based tracking approach) must cope with the problem that the vehicle velocity has to be estimated by the difference between, e.g., reference coordinates of bounding boxes which enclose the vehicle image in two consecutive image frames. A scenario which comprises the action 'slow_down' for a vehicle will preferably rely on an estimate of the *deceleration*, i.e. based on second differences of a vehicle reference location as a function of time. Such an estimate, however, is likely to vary so much from frame to frame that it may become useless.

At this point one could ponder to fall back on 3D-model-based tracking and to exploit action-definitions based on recognition automata in conjunction with the use of Fuzzy Metric-Temporal Horn Logic (FMTHL) ([28]) as they have been reported in [10]. In such a case, the geometric tracking results obtained by `Motris` are made available to conceptual processing via a predicate `has_status` (see [10, p. 358]):

time	!	actor	x	y	θ	v	ψ
[frame #]	!		[m]	[m]	°	[m/s]	°
614	!	has_status(object_4,	8.941,	1.849,	146.9,	2.93,	-2.06).

where ‘object_4’ represents the actor-identifier, x and y the estimated ground-plane-coordinates of the vehicle, θ its estimated orientation, v its estimated speed, and ψ its estimated steering angle. As far as the subsequent reasoning steps are concerned, it is assumed that these estimates have no associated error.

It can be seen that the predicate `has_status` does not comprise an estimate for the *acceleration*. Whenever this is needed, it will be computed by a five-point difference filter obtained by sampling the derivative of a Gaussian at five equal-distanced argument values centered around zero (see, e.g., [10, ‘derivative’ on p. 387]).

Such an anti-symmetric filter must be treated as ‘a-causal’ because it uses velocity estimates for ‘future’ time-points $t+1$ and $t+2$ in order to estimate the acceleration at time point t. One can, of course, finesse a solution by just ‘delaying’ the evaluation for two time-points such that the required speed-estimates are available. This does not work, however, if an action-predicate ‘slow_down’ appears in an SGT which is used for ‘behavioral feedback’ from the conceptual level of representation to the geometric one (for details, see the literature quoted in Appendix A). In such a case, the predicate will be needed at time t in order to decide which information has to be provided to the geometric tracking component for the next time-point.

Obviously, a solution could consist in an extension of the status vector beyond the five components given above, including a sixth component with an estimate of the acceleration. At this point of the problem exposition, four aspects will have to be considered, two being concerned with more fundamental methodological questions and two requiring (non-trivial) implementation efforts.

- Coursebooks on Kalman-Filtering recommend to keep the status-vector as small as possible in order to reduce irritating effects of noise and limited numerical accuracy ([9, p. 232]).
- Extending the status-vector requires to re-tune the matrices for the start- and system-covariances required by the Kalman-Filter used in `Motris`.
- Suppose these changes will have been performed. The system then will have to be tested extensively in order to make sure that nothing unexpected has happened. The investigations reported in [5] and [21,22] indicate what has to be done, but they give at the same time some hint regarding the efforts required.

- The program ('just') has to be changed to a six-component status vector comprising the acceleration. This requires to scrutinize a hundred or more routines in order to make sure that this sixth component will be handled correctly – an effort which has been postponed continuously so far in favor of 'more urgent' modifications.

The discussion has been pushed deliberately to such technical details in order to illustrate that a conceptual approach in many cases can be influenced by considerations which are dictated by CV-aspects rather than by ideas about which concepts to use and about how to use them. The upshot is, of course, that the integration of a CV-subsystem and a subsystem for the exploitation of conceptual knowledge may justify research efforts which are difficult to justify if only one discipline is taken into consideration.

From here, the discussion will 'zoom back out' again in order to facilitate a more encompassing view.

3 The Emerging Overlap between Computer Vision, Computational Linguistics, and Formal Logic

Sketching a potential application may be helpful for an assessment of considerations to be presented in subsequent Subsections 3.2 and 3.3.

3.1 The 'Electronic Co-Driver' – a (still) Hypothetical System

Increasingly, automobiles are offered which have been equipped with one or more video-camera(s), Radar- and/or Laser-range sensors as well as ultrasound proximity sensors. These sensors enable such a vehicle (henceforce referred to as 'EGO-vehicle') to obtain an estimate of the surrounding traffic situation. This information about the status outside the vehicle is supplemented by sensors which provide information about the vehicle itself, for example its speed, the acceleration or deceleration of individual wheels, or the different orientation angles of the vehicle's body. The primary challenge consists in detecting safety threats before these may exceed a driver's capability to handle them. Such a multi-sensor-based system could be part of an 'electronic co-driver' comprising a multi-modal user interface with sound, light, and vibration facilities for warning the driver of imminent danger. If and when the driver expresses a corresponding interest, moreover, a (post-critical-phase) *explanation dialogue* with the driver could defend and elaborate a terse warning. The entire HMI has to be unobtrusive and consistent enough so that a driver may quickly become accustomed to its use. The designers of an electronic co-driver thus should strive to *convince* its user rather than make her/him submit to an anonymous 'technical authority'.

Such a capability implies that the electronic co-driver can present a chain of arguments which relates a description of the sensorial input analysed by the EGO-vehicle to relevant items of abstract knowledge about the road traffic domain. To *elaborate* a warning implies to either present additional sensor-based evidence or to explicate substeps in an earlier *compound* argument which had been chosen in order to control the complexity and brevity of an explanatory step. Computational Linguistics will be required in order to analyse a NL-inquiry by a driver and to package a (partial) answer in such a manner that a driver can absorb the argument(s) while continuing to drive the EGO-vehicle. In order to optimally adapt the response of an electronic co-driver to an explanation request by an individual human driver, the discourse history – in particular the context of specific driver utterances – have to be taken into account in a mixed-initiative NL-dialogue.

The electronic co-driver has to start with an adequate set of premises shared with its potential users. These premises comprise, e.g., knowledge about traffic legislation and rules, basic knowledge about how to manipulate a road vehicle, including knowledge about safety limits for the operation of a road vehicle. In principle, the necessary knowledge could be encoded in other forms than that of explicitly formulated logical sentences. It is postulated, however, that a logic-based representation of such a-priori knowledge will simplify the task to create and update the required knowledge base, for example by having an inference machine perform consistency checks following incremental extensions or modifications. This argument carries beyond mere good software engineering practice. Obviously, an electronic co-driver may influence the performance of safety-critical operations – a fact which can result in *litigation with the vendor* in case a traffic accident has happened. As a consequence, the more capable a system appears to its user, the more responsibility accrues to the system's designer.

3.2 Equivalence Classes of Scenarios?

The reader is invited to tolerate momentarily a number of seemingly bold generalisations and abstractions. These steps are taken with the aim to characterize the hypothetically emerging overlap between CV, CL, and Computational Logic.

Algorithms exists already which extract a NL-description of temporal developments in a recorded scene from a video. For examples see, e.g., Figure 3 (right panel), the handling of objects in an office [16], recent results published by L.S. Davis and coworkers on the extraction and exploitation of a 'storyline' model from videos of baseball games [13, Figure 7], results reported by [8] and [11]. Examples can be found, too, for a converse algorithmic process, i.e. the generation of a video from NL-text, e.g., [18] or [2]. Disregarding for the moment the linguistic quality of these texts, it justifies the postulate that (some) videos and (some) NL-texts can be converted algorithmically into each other. Videos and NL-descriptions will be denoted as 'surface manifestations' of temporal developments in a corresponding recorded scene.

If one accepts that an algorithmic system can answer NL-questions about such temporal developments in NL, it will have to be endowed with some reasoning capabilities. The associated system-internal representation of information about the recorded scene can be considered to constitute another manifestation of the recorded temporal developments which has to be extracted either from a given video or the source text. This, then, allows to claim that it is possible to compute three manifestations of a temporal development in a recorded scene.

It is well known that, given suitable conditions, a-priori knowledge about the scene and its associated temporal developments will reduce the search effort required to obtain such manifestations and, moreover, is likely to increase the 'quality' of computed manifestations. This consideration suggests to endow the envisaged system with some *schematic* representation about temporal developments in the scenes to be recorded and analysed. It does not make sense to provide a-priori knowledge specifically for each scene whose videos will have to be analysed. The schematic representation of a-priori knowledge thus needs to be formulated in general terms. Algorithms will have to be made available which associate, e.g., evidence extracted by a CV-subsystem from a video with the appropriate parts of this general schematic representation in order to derive a manifestation specific for some temporal development in the recorded scene. The reason why it has been studiously avoided to use the word 'instantiation' for this processing step will become clear later.

Let a 'scenario' denote the totality of schematic a-priori knowledge which has been incorporated into a system with respect to developments in scenes whose videos have to be analysed. Turning the argument around, one may consider the scenario as a precise specification of the discourse domain to which the corresponding scenes might be attributed. The notion 'discourse domain' constitutes a colloquial term. It thus should not be used without qualification in the restricted sense required here when referring to a computational system. In order to avoid a mix-up, it is suggested to denote that subset of a discourse domain, which is specified by a scenario, as an 'algorithmically circumscribed discourse domain (acDD)'.

The formalism has not yet been specified which is required to define a scenario. It thus is possible that different groups come up with apparently different formal specifications for the same scenario (compare the assumptions underlying the discussion of Figures 1 and 2). This consideration leads to the postulate that an algorithm exists which may allow to convert one scenario representation into another one (which need not be the case, a question for specialists). Scenarios which can be converted algorithmically into each other can then be considered to constitute an equivalence class of scenarios. Given such an equivalence class, only one representative scenario needs to be analysed further.

In a next step, scenarios will be considered which do *not* belong to the same equivalence class obtained by converting between different *formal* representations. This assumption raises the question how such essentially different formal representations might be related. It is imaginable that a scenario specifying acDD_1 (i) fully contains another acDD_2, (ii) does not fully contain acDD_2, but exhibits a non-empty intersection with acDD_2, (iii) does not have a non-empty intersection with acDD_2 (i.e. differs completely from acDD_2), and finally that it can not be decided whether any of the enumerated cases (i) through (iii) prevails. In this latter case one would have to ask for restricting conditions which could allow to decide between the cases. The question treated in the preceding paragraph, namely to decide to which equivalence class a scenario belongs, relates to the *form* of a scenario specification whereas the question raised in this paragraph is concerned with the *content*, so to speak, of a scenario. These two questions thus formulate different classification problems.

Assume now that these scenario classification problems can be solved algorithmically. It then should be possible to determine the non-empty intersection part of two essentially different scenarios and to compute the union of these two scenarios without duplication of the common part. Such an algorithm would allow to construct scenarios *incrementally* by successive extension. In effect, scenarios are treated as 'objects' which can be manipulated algorithmically.

3.3 Manifestation vs. Instantiation

Remember that scenarios have been introduced as *schematic* representations of what 'can be said' about a scene. Two important problems have not been addressed so far, namely which (dis-)advantages may be associated with different formalisms for the specification of scenarios and how one can associate a given scenario with evidence extracted from a video. The first question boils down to investigations analogous to those associated with Description Logics (DL), namely which restrictions have to be met in order to make processing feasible. B. Neumann and coworkers pioneered investigations on the use of DL for the interpretation of images (static scene) and two-image comparisons – in particular the Doctoral Thesis of C. Schröder [29] has to be mentioned here. In [20], B. Neumann and R. Möller explored – among other questions – the search efforts required for the interpretation of a system-internal DL-representation of a-priori knowledge in the case of a single image. Although I understand that the algorithmic system Racer used in this context recently has been accelerated significantly (see [4, Footnote 4]) by optimizations designed for handling a specific large database, it is currently not clear to me whether similar accelerations can be realized for image interpretation tasks.

Apart from the search efforts required to associate a video with a suitable part of a scenario, an even more fundamental question remains, namely the theoretical relation between the evidence extracted from a video and the schematic representation. 'Evidence' extracted from a video by CV-methods is principally corrupted by noise-effects. Such principally uncertain items thus can not be admitted to populate the interpretation universum as it is traditionally used in Formal Logic. The schematic representation of a scenario has been tacitly assumed to be rigorously specified according to the requirements of (some still rigorous) variant of Formal Logic. The desire to stick to this framework makes it impossible to treat the association of evidence obtained by a CV-process with a scenario as an 'interpretation' process in the strictly formal logic sense. This is the reason why the notion 'manifestation' has been used because it appears to be not yet constrained by the discipline of Computational Logic. It appears that one has to choose what is wanted:

A description of temporal developments in the scene: in this case, a probabilistic association could be the appropriate alternative.

A decision for an action by the system where the system has to decide which action to take itself (see, e.g., the action-part in the definition of a situation node in Appendix A), based on evidence obtained from some sensor like a video-camera. In this case, the system has to define and evaluate the risks associated with alternative actions. The formalism would have to incorporate *decision-theoretic* approaches (see, e.g., [27, Chapter 16.8, p. 636]). Apart from an isolated reference [27, p. 639] to early work by Feldman & Yakimovsky [7], I did not encounter associated hints in the recent CV-literature.

A proof To the best of my knowledge, K. Popper suggested that one can not *prove* a hypothesis by experimental evidence; the only thing possible is to reject ('falsify') a hypothesis based on overwhelming evidence against it. The consequence is that one may prove sentences which refer exclusively to the schematic representation of a-priori knowledge (the scenario specification), but has to renounce the desire to raise the association of a manifestation of a scenario into the rank of anything that can be *proven* in the strict formal logic sense.

4 Conclusions

After nearly 40 years of research to make computers understand image sequences, KI-2010 offers itself as an opportunity to pause and ponder. How can the field be characterized? The abundance of past and current research activities makes it impossible to do justice to each one. Two broadly accepted trends, however, should be mentioned first before the discussion returns to the topic of this presentation.

The estimation of *scene* geometry from image sequences has clearly established itself during this period as a solid subdiscipline of CV. Today, moreover, cameras are exploited routinely as *sensors in the control loop* of moving robots and road vehicles – a development which was not that obvious in the early days.

Whereas it has been a goal right from the beginning of CV to recognize the images of *objects,* attempts to extract *conceptual* descriptions of *temporal developments* from videos have gained considerable momentum only during the past decade. Related research projects are still mostly concerned with the recognition of isolated (atomic) actions of a single agent in the scene. The associated transition from quantitative geometric descriptions to qualitative conceptual ones, however, provides basic conceptual elements for the systematic construction of descriptions of extended temporal developments. *This is the frontier line where CV begins to clearly overlap Linguistics*, of course with an emphasis on the computational variant of this discipline. Supporting arguments can be found, too, in contributions to a special volume on 'Connecting Language to the World' edited by Roy & Reiter [26].

Investigations into CV- and NL-systems are treated here not as an end in itself, but as a means to eventually *engineer* systems. Formal Logic, the science of truth-preserving (de-)composition of concepts, offers a solid foundation for such an endeavor. Logic-based approaches establish an *analyzable* connection across the abyss between signal- as well as geometry-oriented *quantitative* processing of videos and a NL-based Human-Machine Interface which enables even a non-expert user to understand and operate an encompassing system. Such a property facilitates incremental, reasoned modifications and improvements of a system which almost by definition is too complex to be fully grasped by a single person.

Examples for local realisations of such transformations have been mentioned, albeit with limited scope and without coherent integration into an encompassing system: transformations from videos to logic-based scenario representations, transformations between different logic-based scenario representations, and from these into NL-texts. Research on, e.g., Discourse Representation Theory (DRT) by Kamp and coworkers [15,6] induces the expectation that a transformation from technically oriented legal texts such as road traffic regulations into a schematic logic-based representation of agent behavior in road traffic should be possible (see, too, [2, Section 3]). This corresponds to the assumption that the legal text for road traffic regulations specifies the legally binding version of the road traffic *scenario*. Why should Ph.D. students (or their advisers) have to dream up a road traffic scenario for their videos if parlament and administration have done this already in a far more encompassing manner? Obviously, this argument has a serious core, namely whether or not tools to design, implement, and verify such an encompassing scenario can be constructed. A look at Description Logic recommends to be wary of complexity barriers during such an attempt. A related problem then arises, namely to work out conditions which specify what can be principally achieved along such a path and what is not possible in full generality.

It might be interesting to see whether the following consideration will be accepted more broadly: Rather than declaring a particular approach for representation and handling of conceptual information as 'the best' one, algorithms are developed which transform one representation into another one across the entire combined field, i.e. *an approach which treats videos, various logic-based system-internal representations of temporal developments within a recorded scene, and NL-descriptions of these developments essentially on an equal footing.* The study of algorithmic transformations between hypothetically equivalent logic-based representations thus facilitates an understanding of differences in emphasis – as opposed to an emphasis on differences.

Acknowledgements

Intensive discussions with H. Harland and N. Pirlo about many questions referred to in this presentation are gratefully acknowledged, as are the (somewhat shorter, but basically no less intensive) telephone discussions with B. Neumann, Universität Hamburg.

References

1. Arens, M.: Representation und Nutzung von Verhaltenswissen in der Bildfolgenauswertung. In: Dissertationen zur Künstlichen Intelligenz (DISKI), vol. 287, Akademische Verlagsgesellschaft Aka GmbH, Berlin (2005); Dissertation, Fakultät für Informatik der Universität Karlsruhe, TH (July 2004) (in German)

2. Arens, M., Ottlik, A., Nagel, H.H.: Natural Language Texts for a Cognitive Vision System. In: Proc. 15th European Conference on Artifial Intelligence (ECAI 2002), Lyon, France, July 21-26, pp. 455–459. IOS Press, Amsterdam (2002)
3. Baader, F., Calvanese, D., McGuiness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook, 2nd edn. Cambridge University Press, Cambridge (2007)
4. Baader, F., Lutz, C., Turhan, A.Y.: Small is Again Beautiful in Description Logic. Künstliche Intelligenz (KI) 24(1), 25–33 (2010)
5. Dahlkamp, H., Nagel, H.H., Ottlik, A., Reuter, P.: A Framework for Model-Based Tracking Experiments in Image Sequences. International Journal of Computer Vision 73(2), 139–157 (2007)
6. van Eijck, J., Kamp, H.: Representing Discourse in Context. In: van Benthem, J., ter Meulen, G. (eds.) Handbook of Logic and Language, pp. 179–237. The MIT Press, Cambridge (1997)
7. Feldman, J., Yakimovsky, Y.: Decision Theory and Artificial Intelligence: I. A Semantics-Based Region Analyzer. Artificial Intelligence 5, 349–371 (1974)
8. Fexa, A.: Erzeugung mehrsprachlicher Beschreibungen von Verkehrsbildfolgen. In: Dissertationen zur Künstlichen Intelligenz (DISKI), vol. 317, Akademische Verlagsgesellschaft AKA GmbH, Berlin (2008); Dissertation, Fakultät für Informatik der Universität Karlsruhe, TH (February 2008) (in German)
9. Gelb, A. (ed.): Applied Optimal Estimation. The MIT Press, Cambridge (1974)
10. Gerber, R., Nagel, H.H.: Representation of Occurrences for Road Vehicle Traffic. Artificial Intelligence Journal 172, 351–391 (2008)
11. Gonzàlez, J., Rowe, D., Varona, J., Roca, F.: Understanding Dynamic Scenes Based on Human Sequence Evaluation. Image and Vision Computing 27(10), 1433–1444 (2009)
12. Guler, S., Burns, J., Hakeem, A., Sheikh, Y., Shah, M., Thonnat, M., Bremond, F., Maillot, N., Vu, T., Haritaoglu, I., Chellappa, R., Akdemir, U., Davis, L.: An Ontology of Video Events in the Physical Security and Surveillance Domain (2003), http://www.ai.sri.com/~burns/EventOntology (download: June 26, 2010)
13. Gupta, A., Srinivasan, P., Shi, J., Davis, L.: Understanding Videos, Constructing Plots Learning a Visually Grounded Storyline Model from Annotated Videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami Beach, FL, June 20-25, pp. 2012–2019. IEEE Computer Society, Los Alamitos (2009)
14. Harland, H.: Nutzung logikbasierter Verhaltensrepräsentationen zur natürlichsprachlichen Beschreibung von Videos (2010) (IAKS/KIT, in preparation)
15. Kamp, H., Reyle, U.: From Discourse to Logic – Introduction to Modeltheoretic Semantics of Natural Language. In: Formal Logic and Discourse Representation Theory. Kluwer Academic Publishers, Dordrecht (1993)
16. Kojima, A., Tamura, T., Fukunaga, K.: Natural Language Description of Human Activities from Video Images Based on Concept Hierarchy of Actions. International Journal of Computer Vision 50(2), 171–184 (2002)
17. Moeslund, T., Hilton, A., Krüger, V.: A Survey of Advances in Vision-based Human Motion Capture and Analysis. Computer Vision and Image Understanding 104(2-3), 90–126 (2006)
18. Mukerjee, A., Gupta, K., Nautiyal, S., Singh, M., Mishra, N.: Conceptual Description of Visual Scenes from Linguistic Models. Image and Vision Computing 18(2), 173–187 (2000)
19. Nagel, H.H.: Steps toward a Cognitive Vision System. AI-Magazine 25(2), 31–50 (Summer 2004)

20. Neumann, B., Möller, R.: On Scene Interpretation with Description Logics. Image and Vision Computing 26(1), 82–101 (2008)
21. Pirlo, N.: Zur Robustheit eines modellgestützten Verfolgungsansatzes in Videos von Straßenverkehrsszenen (2010) (IAKS/KIT, in preparation)
22. Pirlo, N., Nagel, H.H.: Logic-based Trajectory Evaluation in Videos. In: Dillmann, R., et al. (eds.) KI 2010. LNCS (LNAI), vol. 6359, pp. 48–57. Springer, Heidelberg (2010)
23. Poppe, R.: A Survey on Vision-based Human Action Recognition. Image and Vision Computing 28, 976–990 (2010)
24. Robertson, N.: Automatic Causal Reasoning for Video Surveillance. Ph.D. Thesis, Department of Engineering Science, University of Oxford, Oxford, UK (2006)
25. Robertson, N., Reid, I.: A General Method for Human Activity Recognition in Video. Computer Vision and Image Understanding 104(2-3), 232–248 (2006)
26. Roy, D., Reiter, E.: Connecting Language to the World. Artificial Intelligence Journal 167(1-2), 1–12 (2005)
27. Russell, S., Norvig, P.: Artificial Intelligence – A Modern Approach, 3rd edn. Pearson Education/Prentice-Hall, Inc., Upper Saddle River (2010)
28. Schäfer, K.: Unscharfe zeitlogische Modellierung von Situationen und Handlungen in Bildfolgenauswertung und Robotik. In: Dissertationen zur Künstlichen Intelligenz (DISKI), vol. 135, Infix Verlag, Sankt Augustin (1996); Dissertation, Fakultät für Informatik der Universität Karlsruhe, TH (July 1996) (in German)
29. Schröder, C.: Bildinterpretation durch Modellkonstruktion: Eine Theorie zur rechnergestützten Analyse von Bildern. In: Dissertationen zur Künstlichen Intelligenz (DISKI), vol. 196. Infix Verlag, Sankt Augustin (1999) (in German)
30. Turaga, P., Chellappa, R., Subrahmanian, V., Udrea, O.: Machine Recognition of Human Activities. IEEE Transactions on Circuits and Systems for Video Technology 18(11), 1473–1488 (2008)
31. Wang, X., Tieu, K., Grimson, W.: Correspondence-Free Activity Analysis and Scene Modeling in Multiple Camera Views. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(1), 56–71 (2010)

A On Situation-Graph-Trees

A Situation-Graph-Tree (SGT) provides a schematic representation for sequences of (supposedly goal-directed) actions of an agent. Whereas an agent's sequence of actions can be observed, the current *goal* of the agent (represented within the system by an *actor*) is in general unknown to the system. It thus has to be inferred based on the sequence of the actor's actions extracted from a video and on a-priori knowledge of the observer.

An SGT is a tree-like hypergraph of *situation-graph*s. Each situation-graph consists of *situation-node*s connected by *successor-link*s. A situation-node comprises three *situation-node-section*s:

1. a situation-node *identifier*,
2. a situation-node *status-part* comprising a conjunction of time-dependent n-ary *predicates* (with $n \geq 1$) which are formulated according to the rules of Fuzzy Metric-Temporal Horn Logic (FMTHL), see [28].
3. an *action-part* comprising one or more actions which are taken to be executed if the status-part of the situation-node has been found to be satisfied.

Outgoing successor-links are prioritized: if the status-part of a situation-node can no longer be satisfied, the highest-prioritized successor-link will be followed in an attempt to satisfy the associated successor-situation-node. If this highest prioritized successor-node can not be verified based on the evidence obtained by evaluation of the current video-frame, the next lower prioritized successor-node will be tested for satisfiability. Usually, a situation-node constitutes its own highest-prioritized successor-node, reflecting the experience that the relevant evidence frequently requires more than a single video-interframe-interval to change significantly.

There are two different '*particularization*' edge-types which connect a situation-node at one level of an SGT to an entire situation-graph at the next lower (i.e. more detailed) level of this SGT:

A specialization-edge adds at least one specializing attribute or predicate to the status-part of each situation-node in the subordinate situation-graph which may consist of only a single situation-node. Given the more specialized characterization of the associated situation-node(s), the action-part of the subordinate situation-node(s) may differ from that of the super-ordinated situation-node, thereby enabling a differentiated reaction to internal or external conditions for a particular actor.

A (temporal) decomposition-edge decomposes a situation-node into an entire graph of situation-nodes which allows to model a *sub*sequence of actions, conditioned on all predicates in the superordinate situation-node remaining satisfiable, but contingent on the time-ordered satisfiability of additional status-parts specified individually for each situation-node of the subordinate situation-graph.

Each situation-graph contains an *entry*-node and an *exit*-node. Entry- and exit-node may coincide. *Traversal* of an SGT starts at the entry-node of the root-situation-graph. The traversal process searches for the first satisfiable situation-node and then descends along specialization- or decomposition-edges towards leaf-situation-nodes. Depending on the execution option for the traversal process, either all actions in the action-parts of all satisfiable situation-nodes along the descending chain from the root node to the current most leaflike satisfiable situation-node are executed or only the actions in the currently satisfiable situation-node most distant from the root node.

If this situation-node can not be satisfied any longer nor one of its successor-situation-nodes in the subordinate situation-graph, the traversal process steps up one level to its parent node which might still be satisfiable despite the fact that no satisfiable successor-node to the last satisfiable situation-node in the more detailed situation-graph could be found. If an exit-situation-node can no longer be satisfied, the subordinate situation-graph will be left for the superordinate situation-node. This particular step upwards indicates that the expectations regarding the specialization/decomposition conform to a sequence of evidences extracted from a video.

Each level of an SGT corresponds to a particular level of abstraction for the temporal development in the recorded scene. A path through an SGT corresponds to a particular sequence of actions potentially executable by an agent – i.e. expected to be a principally observable sequence of actions by an agent. Traversal of an SGT thus corresponds to the instantiation of a sequence of situation-nodes by CV-results associated with an actor which has been instantiated for an agent. A path through an SGT thus can be looked at as a particular goal-directed sequence of actions subject to temporarily prevailing conditions in the recorded scene. With other words, a path through an SGT can be taken to represent a particular agent behavior. An entire SGT thus specifies a complex *set* of potential behaviors of agent(s) – a *scenario*. A particular actor may instantiate one from this set of paths and thereby may instantiate one of the agent behaviors which are admissible in the scenario formalized by this SGT.

A short *illustrated* introduction of the SGT-Formalism can be found in [19] which comprises further references to related publications. Additional applications of the SGT-Formalism are treated in [21]. [22] reports one of these and comprises a figure with illustrative hints regarding the SGT-Formalism.

B SIT++ Formulation of the SGT Shown in Figure 2

```
//
DEFAULT NONINCREMENTAL GREEDY PLURAL DEPTH TRAVERSAL;


GRAPH gr_ED_SITGRAPH1            /* Top graph of the Situation Graph Tree (SGT), i.e.      */
                                 /* there is no parent node.*/
{
  START FINAL SIT sit_ED_SIT1 :  /* Root node with node-identifier ‘sit_ED_SIT1’ of this   */
                                 /* Situation Graph Tree */
      sit_ED_SIT1
  {
    person(Employee);            /* The variable identifier ‘Employee’ satisfies the       */
                                 /* predicate ‘person’. */
    person(Robber);
    zone(entrance_zone);         /* The constant identifier ‘entrance_zone’ satisfies the  */
                                 /* predicate ‘zone’.   */
                                 /* The value for this constant has been fixed in analogy  */
                                 /* to the value of ‘lseg_1’ in Gerber/Nagel,              */
                                 /* AIJ 172:4-5 (March 2008) 351-391, Section 4.2.         */
    zone(front_counter);
    zone(back_counter);
    zone(safe);
  }
  {
    INCREMENTAL note(’Basic premises, required to be always true.’);
  }
}


GRAPH gr_ED_SITGRAPH10 : sit_ED_SIT1  /* Subgraph which particularises node ‘sit_ED_SIT1’ */
                                      /* in superordinated graph */
{
  START FINAL SIT sit_ED_SIT10 :      /* The situation node ‘sit_ED_SIT10’ is its own     */
                                      /* highest prioritized temporal successor.          */
        sit_ED_SIT10,
        sit_ED_SIT20                  /* with the situation node ‘sit_ED_SIT20’ as the    */
```

```
                                        /* lower prioritized choice if the predicates in the*
                                        /* status part of sit_ED_SIT10 can no longer be     */
                                        /* satisfied.                                       */
  {
    inside_zone(Employee, back_counter);
  }
  {
    INCREMENTAL note('Only employee present; employee is situated behind the counter.');
  }

  START FINAL SIT sit_ED_SIT20 :
        sit_ED_SIT20
  {
    inside_zone(Employee, safe);
  }
  {
    INCREMENTAL note('The employee has moved into the area of the safe.');
  }
}



GRAPH gr_ED_SITGRAPH100 : sit_ED_SIT10
{
  START FINAL SIT sit_ED_SIT110 :
        sit_ED_SIT110,
        sit_ED_SIT120                 /* If `inside_zone(Robber, entrance_zone)' can no */
                                      /* longer be satisfied, the temporal successor    */
                                      /* node `sit_ED_SIT120' will be tested whether it */
                                      /* can be satisfied.                              */
  {
    inside_zone(Robber, entrance_zone);
  }
  {
    INCREMENTAL note('A person (which subsequently turns out to be a robber) is detected
                      in the entrance zone.');
  }

  START FINAL SIT sit_ED_SIT120 :
        sit_ED_SIT120
  {
    inside_zone(Robber, front_counter);
  }
  {
    INCREMENTAL note('This person moves into the zone front_counter.');

         /* Note that the predicate `inside_zone(Employee, back_counter)' is still true */
         /* when the traversal algorithm hits this node because this node is part of the*/
         /* particularization (here as temporal decomposition) of node `sit_ED_SIT10' ! */
  }
}



GRAPH gr_ED_SITGRAPH200 : sit_ED_SIT20
{
  START FINAL SIT sit_ED_SIT210 :
        sit_ED_SIT210
  {
    inside_zone(Robber, safe);
  }
  {
    INCREMENTAL note('Now the robber is in the safe area, too!');
  }
}
```

C Steps toward a Scenario-Conversion-Algorithm

This section sketches some initial ideas to convert the SGT-version of the scenario shown in Figure 2 into the formulation of Figure 1. The pure status-predicates such as `inside_zone` are the same in both representations. The representation of Figure 1 differs from the SGT-representation (i) by the temporal predicates `before` and `during` together with the associated time-interval-identifiers and (ii) by the 'event-predicate' `changes_zone`.

The remainder of this section concentrates on the subtask to explicate the two temporal predicates from the temporal structure coded implicitly by the SGT and its time-dependent depth-first traversal algorithm.

C.1 Preparatory Definitions

Explication of the temporal predicates necessitates to explicate the temporal ordering encoded by an SGT. For this we need to explicate node-succession-relations encoded within an SGT. In accordance with the conventions used in Section B, variable-identifiers start with capital letters whereas constant- and predicate-identifiers start with lower case letters.

Let `enumerated_successor_node`$(N_1,\ N_2)$ be `True` for all successor-nodes N_2 (which are enumerated following the colon in the situation node definition identified by the keyword 'SIT'). Let

$$\texttt{new\_direct\_successor\_node}(N_1, N_2) \;=\; \texttt{enumerated\_successor\_node}(N_1,\ N_2) \wedge (N_1 \neq N_2). \qquad (1)$$

The predicate `new_direct_successor_node`(N_1, N_2) excludes the link where a node is its own successor, i.e. it selects genuine successor nodes.
Let a path of successor-nodes *within the same graph* be characterised recursively by

$$\texttt{successor\_node}(N_1, N_2) \;=\; \texttt{new\_direct\_successor\_node}(N_1,\ N_2) \;\vee\; \exists N\big(\ \texttt{new\_direct\_successor\_node}(N_1, N) \wedge \texttt{successor\_node}(N, N_2)\ \big). \qquad (2)$$

Define `node_has_child_graph`(N, G) to be `True` if there is a backlink to node N following the colon in the definition for graph G (identified by the keyword 'GRAPH'). The following recursive definition prepares the study of temporal relations between a node N_1 and any node in a graph which particularizes node N_1.

$$\texttt{node\_particularizes\_into\_subordinated\_graph}(N_1,\ G_{sub_1}) \;=\; \texttt{node\_has\_child\_graph}(N_1,\ G_{sub_1}) \;\vee\; \Bigg(\exists G\ \texttt{node\_has\_child\_graph}(N_1,\ G) \;\wedge\; \bigg(\exists N\Big(\ (N \in G) \wedge \texttt{node\_particularizes\_into\_subordinated\_graph}(N,\ G_{sub_1})\Big)\bigg)\Bigg)\ . \qquad (3)$$

Let `status_part`(N) denote the conjunction of predicates in the status-part of situation-node N and of all those nodes on the path from N back through the SGT to its root-node, i.e. the conjunction of all predicates which characterize the status when node N is successfully visited during the SGT-traversal.

C.2 Extraction of the Temporal Relation `before` from an SGT

Using the preceding definitions, one can infer from an SGT:

$$\begin{aligned}&\texttt{status\_part}(N_1)\ \texttt{before}\ \texttt{status\_part}(N_2) \quad \Leftarrow \\ &\quad \texttt{successor\_node}(N_1,\ N_2)\ \vee\ \Bigg(\exists N \Bigg(\texttt{successor\_node}(N_1,\ N)\ \wedge \\ &\qquad \texttt{node\_particularizes\_into\_subordinated\_graph}(N,\ G_{sub_1})\ \wedge\ \big(N_2 \in G_{sub_1}\big)\Bigg)\Bigg)\ . \end{aligned} \tag{4}$$

C.3 Extraction of the Temporal Relation `during` from an SGT

In analogy to the preceding subsection, one can infer:

$$\begin{aligned}&\texttt{status\_part}(N_2)\ \texttt{during}\ \texttt{status\_part}(N_1) \quad \Leftarrow \\ &\quad \texttt{node\_particularizes\_into\_subordinated\_graph}(N_1,\ G_{sub_1})\ \wedge\ \big(N_2 \in G_{sub_1}\big)\ . \end{aligned} \tag{5}$$

A Computational Model of Human Movement Coordination

Thorsten Stein[1], Christian Simonidis[2],
Wolfgang Seemann[2], and Hermann Schwameder[1]

[1] Institute for Sport and Sport Science, [2] Institute of Engineering Mechanics,
Karlsruhe Institute of Technology (KIT), Germany
thorsten.stein@kit.edu

Abstract. Due to the numerous degrees of freedom in the human motor system, there exists an infinite number of possible movements for any given task. Unfortunately, it is currently unknown how the human central nervous system (CNS) chooses one movement out of the plethora of possible movements to solve the task at hand. The purpose of this study is the construction of a computational model of human movement coordination to unravel the principles the CNS might use to select one movement from plethora of possible movements in a given situation. Thereby, different optimization criteria were examined. The comparison of predicted and measured movement patterns exhibited that a minimum jerk strategy on joint level yielded the closest fit to the human data.

1 Introduction

The human CNS and musculoskeletal system are highly redundant and enable us to achieve movement tasks (e.g. reaching and grasping) in countless ways [1]. Despite this redundancy, a number of studies revealed [2,3] that humans tend to use a narrow set of stereotypical movements in the context of goal directed movements. If this phenomenon is seen in a larger evolutionary context, the CNS may have learned during evolution to optimize behavior with respect to biologically relevant task goals. This implies that some movements will lead to reward (e.g. food) and some to punishment (e.g. hunger). In a world of limited resources competition between individuals lead to fitter individuals. Since movements are the only way one has for interacting with the world, fitness is determined by movements. Accordingly there will be a selection for those CNS that can provide a fitter behavioral control. Therefore, an evolutionary trend towards optimal control can be expected [4,5]. The optimization of the behavior can be formally defined as cost functions which can be incorporated in optimal control models. Current optimal control models can be grouped into open-loop and closed-loop models [6]. In the context of open-loop models a lot of research has been conducted on relatively simple 2D movements. The characteristics of these movements can be explained and are reasonably well understood. However, in the context of multi-joint movements in 3D space the use of open-loop optimization models is still in an investigate phase [7,8,9]. The vast number of

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 23–32, 2010.

different open-loop models can be assigned to four levels: an extrinsic-kinematic space (e.g. Cartesian coordinates of the hand), an intrinsic-kinematic space (e.g. joint angles or muscle length), an intrinsic-dynamic-mechanical space (e.g. joint torque or muscle tension) and an intrinsic-dynamical-neural space (e.g. motor commands controlling muscle tension or the firing rate of motor neurons) [10]. In this study we focus on an extrinsic kinematic space (minimum hand jerk criterion (MHJ) [3]), intrinsic-kinematic space (minimum angle jerk criterion (MAJ)) and intrinsic-dynamic-mechanical space (minimum torque change criterion (MTC) [11]). The purpose of the following study[1] is to quantitatively examine which of the three optimal control models can best reproduce measured multi-joint pointing movements in 3D space.

2 Methods

2.1 Computational Framework

A computational framework [12] was designed to compute the optimal control solutions with respect to defined optimality criteria and to process recorded motion capture data. Therefore, a multibody model of the upper human body [12] was developed consisting of 32 degrees of freedom (DOFs). Biomechanical standards in the kinematic model design [13], e.g. position and orientation of body reference frames and joint axes (Fig. 1), have been considered and the parameters of the model (segment lengths, mass distribution, segment center of mass and inertia) have been computed based on regression equations [14].

From the model definition a minimal set of the spatial dynamic equilibrium equations is established of the form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{T}, \tag{1}$$

where $\mathbf{q}$ is the vector of generalized coordinates (or rather joint angles), $\mathbf{M}(\mathbf{q})$ is the mass matrix of the system, $\mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}})$ is the vector of bias forces and torques containing coriolis and centrifugal terms as well as gravity and external forces. $\mathbf{T}$ is the vector of generalized forces.

2.2 Motion Analysis

Recorded motion capture data was transferred to the upper body model using a nonlinear optimization approach [15] computing the unknown values $\mathbf{q}(t_n),\ n = 1, ..., n_t$, for each recorded time instance n as a result of inverse kinematics by minimizing the difference between absolute coordinates $\hat{\mathbf{R}}_m(t_n),\ m = 1, ..., n_m$, of the measured markers (Fig. 1) and the $\mathbf{q}$-dependent coordinates $\mathbf{R}_m(\mathbf{q}, t_n),\ m = 1, ..., n_m$, of the markers defined on the segments of the multibody model. Biologically inspired limits $\mathbf{q} \in [\mathbf{q}_{\min}, \mathbf{q}_{\max}]$ of the joint variables were included. The

[1] This work is supported by the German Research Foundation within Collaborative Research Center 588 "Humanoid Robots - Learning and Cooperating Multimodal Robots".

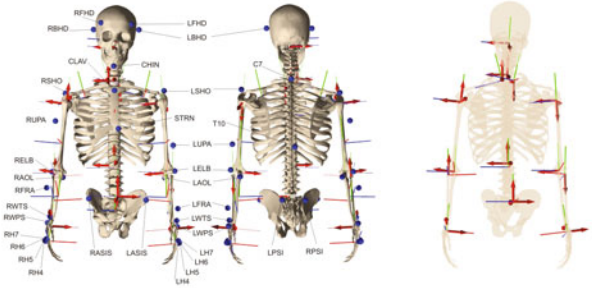


Fig. 1. The skeleton model of the human upper body with attached markers (left and middle) and the visualization of the 32 DOFs with joint axes (right)

resulting kinematic trajectories were Butterworth-filtered and first and second derivatives were computed by cubic spline approximation. Accurate results were obtained using markers on specific anatomical landmarks [16] to reduce influence of soft tissue artifacts and to scale the dimensions of the upper body model to the dimensions of a subject [12]. Inverse dynamics was performed to obtain the net joint torques.

2.3 Motion Synthesis

Synthesis of motion implies the generation of time trajectories of model variables which is realized in this study with open-loop optimal control. The formulation of the optimal control problem is stated as to minimize a scalar cost function:

$$F(\mathbf{z}, \mathbf{u}, t) = \int\limits_{t_0}^{t_f} h(\mathbf{z}(t), \mathbf{u}(t), t)dt, \quad t_0 \leq t \leq t_f \tag{2}$$

subject to an underlying first order system $\dot{\mathbf{z}}(t) = \mathbf{f}(\mathbf{z}(t), \mathbf{u}(t), t)$, with state vector $\mathbf{z}(t) = [\mathbf{q}(t), \dot{\mathbf{q}}(t)]^T$ and control vector $\mathbf{u}(t)$, and subject to start and end conditions $\mathbf{z}(t_0) = \mathbf{z}_0$, $\mathbf{z}(t_f) = \mathbf{z}_f$, and limits on the variables of the system $\mathbf{z} \in [\mathbf{z}_{\min}, \mathbf{z}_{\max}]$, $\mathbf{u} \in [\mathbf{u}_{\min}, \mathbf{u}_{\max}]$.

Trajectories have been computed with respect to the following cost criteria: Minimum hand jerk (MHJ) [3]:

$$F_{HJ} = \frac{1}{2} \int_{t_0}^{t_f} \dddot{\mathbf{R}}^T(t) \dddot{\mathbf{R}}(t) dt, \tag{3}$$

where $\dddot{\mathbf{R}}(\mathbf{t})$ is the third derivative of the position trajectory of the hand segment reference frame in absolute space making up the control $\mathbf{u}(t)$.

Minimum angle jerk (MAJ):

$$F_{AJ} = \frac{1}{2} \int_{t_0}^{t_f} \dddot{\mathbf{q}}^T(t) \dddot{\mathbf{q}} t) dt, \tag{4}$$

where $\dddot{\mathbf{q}}(\mathbf{t})$ is the third derivative of joint angle trajectories making up the control $\mathbf{u}(t)$.

Minimum torque change (MTC) [11]:

$$F_{TC} = \frac{1}{2}\int_{t_0}^{t_f} \dot{\mathbf{T}}^T(t)\dot{\mathbf{T}}(t)dt, \tag{5}$$

where $\dot{\mathbf{T}}$ is the first derivative of actuation torque making up the control $\mathbf{u}(t)$.

Problems are numerically solved using a direct approach [17] discretizing state and control variables with piecewise continuously differentiable quintic splines [12]. The start and end conditions were known from motion analysis and results of optimal trajectories were compared with measured trajectories using statistical methods.

2.4 Validation of Optimality Criteria

To be able to test the different optimization criteria a motion capture study was conducted. Twenty healthy students of the KIT participated voluntarily in the study. Human pointing movements were captured in a kitchen at the Institute for Anthropomatics at the KIT. This kitchen serves as a test center for the development of hardware and software components for humanoid robots (Fig. 2). All subjects stood in a neutral upright posture at the same starting position, looking at a robots image projected onto the opposite wall representing the robot as a communication partner. Four plates with numbers were attached at different heights to the kitchen furniture, the wall, and the floor representing objects which the robot would have to bring to its human user (Fig. 2). Subjects were instructed to perform the gestures like in their daily life. Instructions concerning speed, accuracy or choice of hand were not provided. Besides the starting position, the order of number announcement was standardized. However, subjects were not informed about the order of number announcement before the trial. Each number was called five times resulting in 20 pointing movements. For instance, when the researcher called “Number 1”, the subject pointed into the direction of the corresponding number without touching the target. Prior to data collection, one test trial was performed for each target. All pointing movements were tracked at 120 Hz using a VICON motion capture system. The transformation of coordinates of frame markers into 3D space was done in the VICON Workstation software. Finally, the 3D coordinates of the markers were used as input for the above described computational framework.

In a first step all movements were analyzed [18]. The results showed that the coordination strategies used by the subjects could be assigned to four groups. In this paper we focus on eight subjects of group 1, which did not leave the starting position and pointed with their left hand to targets 1 and 2 and with their right hand to targets 3 and 4. Furthermore, the analysis exhibited that the most relevant DOFs were shoulder abduction/adduction, shoulder rotation, shoulder anteversion/retroversion and elbow flexion/extension. Therefore, these DOFs were optimized and all other DOFs of the multibody model were moved with the experimentally determined movement data.

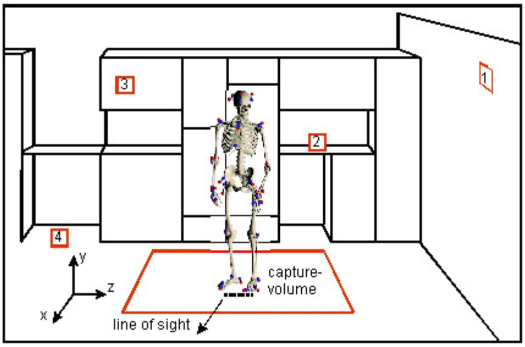


Fig. 2. Starting position of the subjects and positions of the numbers

For validation purposes a similarity coefficient was calculated enabling the comparison of topological course characteristics of movement patterns [19]. All time courses of the measured and predicted hand trajectories, tangential hand velocities, joint angles and joint angular velocities of the optimized joints are correlated with reference functions. Nine Taylor polynomials served as a reference system that satisfied the following condition of orthogonality:

$$\int_0^t f(x)g(x)dx = \begin{cases} 1 \; \forall f(x) = g(x) \\ 0 \; \forall f(x) \neq g(x) \end{cases} \tag{6}$$

By correlating n_q courses (e.g. times series of shoulder and elbow angles) with the nine Taylor polynomials the course characteristics are represented as a topological quantity to a $n_q \times 9$ matrix. The similarity coefficient SIM of n_q courses of measured and predicted movements (e.g. times series of shoulder and elbow angles) is defined with the help of the reference specific correlation coefficients R^{me} and R^{pr}:

$$SIM = \frac{tr(R^{me'}R^{pr})}{\sqrt{tr(R^{me'}R^{me})}\sqrt{tr(R^{pr'}R^{pr})}} \tag{7}$$

Thereby $R^{me'}$ and $R^{pr'}$ are the transpose of the matrices of the reference specific correlation coefficients and tr corresponds to the trace of a matrix. The similarity coefficients SIM can be interpreted as correlation coefficients. To be able to calculate a mean similarity coefficient across a number of trials, the similarity coefficients have to be transformed to Z-values. After the transformation the mean values are calculated and then the means of the Z-values are transformed back to similarity coefficients [20].

3 Results

In this paper results for target 1 are presented. First, variations and similarities between measured and predicted movement data were of special interest. In this context, some representative trajectories are presented and qualitatively analyzed.

In figure 3 representative hand paths for target 1 are illustrated. The MHJ-criterion produced straight hand paths across all trials. In contrast the MAJ-criterion as well as the MTC-criterion generated across all trials more curved hand paths, whereas the MTC-criterion showed sometimes larger deviations. All in all, none of the criteria was able to reproduce the experimentally determined hand paths. The subjects produced single-peaked, almost bell-shaped velocity profiles. The MHJ-criterion produced in all cases single-peaked, bell-shaped velocity profiles with lower peak velocities than the human velocity profiles. The other two optimization criteria produced in most of the cases single-peaked tangential velocity profiles with bell-like shapes. In some cases the profiles of the MTC-criterion revealed distortions. Nevertheless, all three optimization criteria seem to be able to emulate the basic features of the human hand trajectories in most of the test trials.

Below the hand paths typical joint angle trajectories for target 1 are displayed (Fig. 3). Across all trials the two optimization criteria reproduced the angle profiles of the human shoulder abduction best. Furthermore, the MAJ-criterion produced the closest fit to the human trajectories across the four DOFs. In the cases of the shoulder rotation, shoulder anteversion/retroversion as well as elbow flexion/extension the trajectories the MTC-criterion were much more variable with larger movement ranges than the ones produced by the subjects. Below the corresponding joint angular velocities are illustrated. The MAJ-criterion shows the closest fit across all four DOFs. The MTC-criterion reproduced the angular velocity profiles of the human shoulder abduction at least to some extent, but showed large deviations in the other three DOFs with large differences in the peak angular velocities. In the bottom row of figure 3 the corresponding joint torques are displayed. The MTC-criterion showed a tendency to reproduce the human torque profiles only incompletely across all the trials and all four DOFs. However, in few cases the MTC-criterion produced trajectories that emulated the human torques profiles to some extent (Fig. 3, shoulder rot.). The peak torques produced by the MTC-criterion were only seldom larger than the peak torques produced by the subjects. Furthermore, all of the torque profiles were slightly sinuous.

Finally, the similarities between the measured and predicted movements were calculated using the approach described above. The results for the hand paths reveal only small differences in similarity coefficients between the different optimization criteria (Tab. 1). Moreover, the similarity coefficients of the three optimization criteria for the tangential hand velocities are smaller than the similarity coefficients for the hand paths. The MAJ-criterion, however, produced the highest coefficients in extrinsic kinematic coordinates. On joint level the MAJ-criterion showed a closer fit to the human data than the MTC-criterion. In addition, the coefficients for the joint angular velocities are substantially smaller than the coefficients for the joint angles. Finally, the similarity coefficients are smaller in intrinsic kinematic coordinates than in extrinsic kinematic coordinates indicating that the optimization criteria can reproduce human hand movements more precisely than human joint movements.

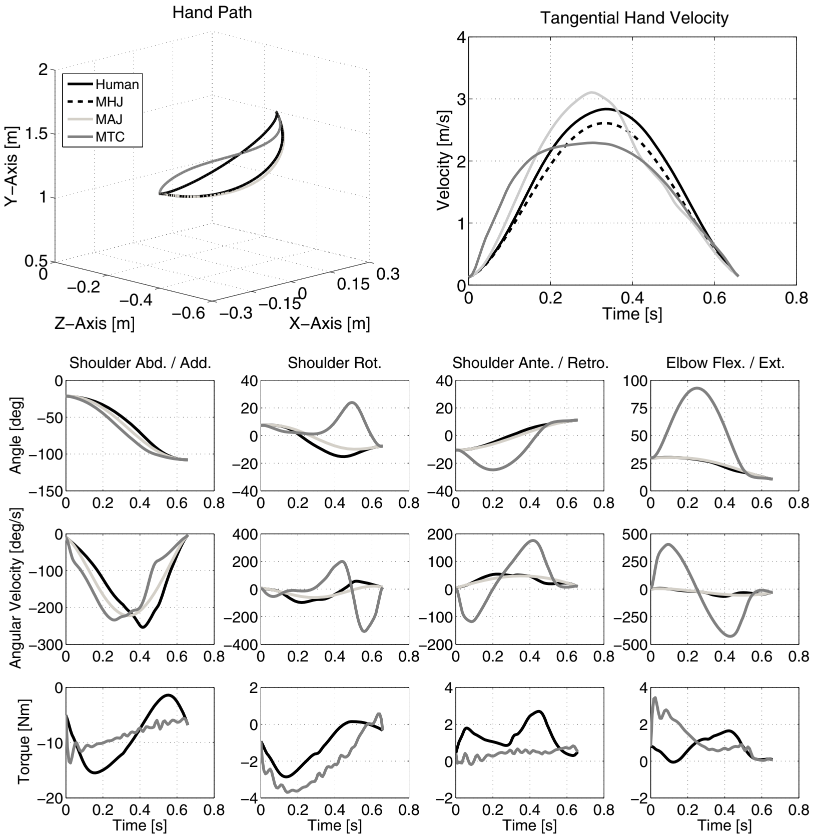


Fig. 3. Representative hand paths and tangential velocity profiles for the optimization of shoulder abduction/adduction, shoulder rotation, shoulder anteversion/retroversion as well as elbow flexion/extension (top row). Below the corresponding angles, angular velocities as well as torque profiles are displayed.

Table 1. Mean of similarity coefficients between the measured and predicted movement patterns (4 DOFs, N = 40) of MHJ-, MAJ- and MTC-criterion

	MHJ-criterion	MAJ-criterion	MTC-criterion
Hand Paths	.996	.998	.994
Tangential Hand Velocities	.963	.969	.917
Joint Angles	–	.955	.671
Joint Angular Velocities	–	.761	.299

4 Discussion

The aim of this study was to quantitatively examine if a MHJ-criterion, a MAJ-criterion or a MTC-criterion can best reproduce measured unconstrained multi-joint pointing movements in 3D space. Till now, only a few studies addressed unconstrained multi-joint pointing movements in 3D space [8]. The advantage of our computational framework consists in the use of a complex biomechanical multibody model instead of an isolated arm model. Therefore, our approach enables the analysis of complex whole body movements.

The results for the optimization of 4 DOFs indicated that the MAJ-criterion yielded the closest fit to the human data. Although the differences in extrinsic kinematic coordinates were rather small, a movement coordination based on angle jerk minimization seems the most plausible one based on our results. The MAJ-criterion did not fully reproduce the measured movement patterns. However, due to noise in the sensorimotor system [21] planned and executed movements will always differ to a certain extent. In addition, it has to be asked if there is a biologically plausible explanation why the human CNS should use a MAJ-strategy for movement coordination. It is debatable whether the CNS is able to estimate such complex quantities. Furthermore, some authors [22] even doubt that the CNS has direct access to these quantities. An alternative to the MAJ-criterion is the minimum variance criterion by Harris and Wolpert [23]. This model assumes that noise exists in the motor commands and that the amount of noise is proportional to the magnitude of the motor commands. There are several important ramifications associated with this model. First, non-smooth movements require larger motor commands than smooth movements and consequently generate an increase in noise. In the case of goal-directed movements, smoothness leads to accuracy but is not a specific goal of its own right. Second, signal-dependent noise inherently imposes a trade off between movement duration and final accuracy in the endpoint of the movement, consistent with Fitts law [24]. Third, the minimum variance model provides a biologically plausible theoretical underpinning for goal-directed movements. This is possible because such costs are directly available to the CNS and the optimal trajectory could be learned from the experience of repeated movements.

MHJ-criterion produced straight hand paths with single-peaked, bell-shaped velocity profiles for all trials. Our analysis revealed that humans produce curved hand paths with single-peaked, bell-shaped velocity profiles. However, it is conceivable that the CNS plans straight trajectories that are distorted during execution due to the complex intersegmental dynamics [25].

The MTC-criterion showed partly low similarity coefficients. However, this does not necessarily mean that the MTC-criterion cannot reproduce human movements and is therefore a strategy that the CNS does not use. The optimization on torque level is complex and the results partly indicated some numerical problems occurred during the optimization process. Accordingly, the optimization algorithm has to be adapted to the individual test trials to further enhance its performance especially in the context of dynamic optimization criteria.

In the future the computational framework is intended to include biologically more plausible optimization criteria, e.g. minimum variance or optimal stochastic feedback control [23,6], as well as combinations of different optimization criteria that can be weighted depending on the movement goal [26].

References

1. Bernstein, N.: The coordination and regulation of movement. Pergamon Press, Oxford (1967)
2. Morasso, P.: Spatial control of arm movements. Experimental Brain Research 42, 223–227 (1981)
3. Flash, T., Hogan, N.: The coordination of arm movements: an experimentally confirmed mathematical model. Journal of Neuroscience 5, 1688–1703 (1985)
4. Harris, C.: On the optimal control of behaviour: A stochastic perspective. Journal of Neuroscience Methods 83, 73–88 (1998)
5. Diedrichsen, J., Shadmehr, R., Ivry, R.B.: Optimal feedback control and beyond. Trends in Cognitive Science 14(1), 31–39 (2010)
6. Todorov, E.: Optimality principles in sensorimotor control. Nature Neuroscience 9, 907–915 (2004)
7. Hermens, F., Gielen, S.: Posture-based or trajectory based movement planning: A comparison of direct and indirect pointing movements. Experimental Brain Research 159, 340–348 (2004)
8. Biess, A., Liebermann, D., Flash, T.: A computational model for redundant human three-dimensional pointing movements: Integration of independent spatial and temporal motor plans simplifies movement dynamics. Journal of Neuroscience 27, 13045–13064 (2007)
9. Gielen, S.: Review of models for the generation of multijoint movements in 3D. In: Sternad, D. (ed.) Progress in Motor Control: A Multidisciplinary Perspective, Adv. in Exp. Med. and Bio., vol. 629, pp. 523–552. Springer, Heidelberg (2009)
10. Nakano, E., Imamizu, H., Osu, R., Uno, Y., Gomi, H., Yoshioka, T., Kawato, M.: Quantitative examinations of internal representations for arm trajectory planning: Minimum commanded torque change model. Journal of Neurophysiology 81, 2140–2155 (1999)
11. Uno, Y., Kawato, M., Suzuki, R.: Formation and control of optimal trajectory in human multijoint arm movement. Biological Cybernetics 61, 89–101 (1989)
12. Simonidis, C.: Methoden zur Analyse und Synthese menschlicher Bewegungen unter Anwendung von Mehrkörpersystemen und Optimierungsverfahren (Analysis and synthesis of human movements by multibody systems and optimization methods) (phD–thesis). Karlsruhe Institute of Technology (2010), http://digbib.ubka.uni-karlsruhe.de/volltexte/1000017136
13. Wu, G., et al.: ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion - part II: Shoulder, elbow, wrist and hand. Journal of Biomechanics 38, 981–992 (2005)
14. De Leva, P.: Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. Journal of Biomechanics 29, 1223–1230 (1996)
15. Lu, T., O'Connor, J.: Bone position estimation from skin marker co-ordinates using global optimization with joint constraints. Journal of Biomechanics 32(2), 129–134 (1999)
16. Cappozzo, A., Croce, U., Leardini, A., Chiari, L.: Human movement analysis using stereophotogrammetry. Part 1: Theoretical background. Gait and Posture 21, 186–196 (2005)
17. von Stryk, O.: Optimal control of multibody systems in minimal coordinates. ZAMM-Zeitschrift für Angewandte Mathematik und Mechanik 78, 1117–1121 (1998)

18. Stein, T., Simonidis, C., Seemann, W., Schwameder, H.: Kinematic analysis of human goal-directed movements. International Journal of Computer Science in Sport (in revision)
19. Schöllhorn, W.: Systemdynamische Betrachtung komplexer Bewegungsmuster im Lernprozess (System dynamic observations of complex movement patterns during the learning process). Peter Lang (1998)
20. Bortz, J.: Statistik für Sozialwissenschaftler. Statistics for Social Scientists. Springer, Heidelberg (1999)
21. Faisal, A.A., Selen, L.P.J., Wolpert, D.M.: Noise in the nervous system. Nature Reviews Neuroscience 9, 292–303 (2008)
22. Feldman, A.G.: The nature of voluntary control of motor actions. In: Latash, M.L., Lestienne, F. (eds.) Motor Control and Learning, pp. 3–8. Springer, Heidelberg (2006)
23. Harris, C.M., Wolpert, D.M.: Signal-dependent noise determines motor planning. Nature 394, 780–784 (1998)
24. Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology 47, 381–391 (1954)
25. Sainburg, R.L., Poizner, H., Ghez, C.: Loss of proprioception produces deficits in interjoint coordination. Journal of Neurophysiology 70, 2136–2147 (1993)
26. Admiraal, M.A., Kusters, M., Gielen, S.: Modeling kinematics and dynamics of human arm movements. Motor Control 8, 312–338 (2004)

BiosignalsStudio: A Flexible Framework for Biosignal Capturing and Processing

Dominic Heger, Felix Putze, Christoph Amma, Michael Wand, Igor Plotkin, Thomas Wielatt, and Tanja Schultz

Cognitive Systems Lab (CSL)
Karlsruhe Institute of Technology (KIT), Germany
{dominic.heger,felix.putze,christoph.amma,
michael.wand,tanja.schultz}@kit.edu,
{igor.plotkin,thomas.wielatt}@student.kit.edu

Abstract. In this paper we introduce BiosignalsStudio (BSS), a framework for multimodal sensor data acquisition. Due to its flexible architecture it can be used for large scale multimodal data collections as well as a multimodal input layer for intelligent systems. The paper describes the software framework and its contributions to our research work and systems.

1 Introduction

In modern statistically based AI research nearly all systems are fundamentally dependent on sensory input data. They infer task-relevant patterns from the outside world and process them by recognition algorithms, which often require large amounts of training data to statistically model relevant information.

From a research perspective, data acquisition is a critical and non-trivial task, particularly when human test persons are involved. We see the need for a tool that supports this task and integrates seamlessly into the development workflow. This recording tool should be flexible, extensible, usable for non-experts and portable to different plattforms.

In this paper we introduce BiosignalsStudio (BSS), a flexible framework for multimodal sensor data acquisition, which is designed as a component of human-machine interaction (HMI) systems for both the development phase with large-scale data collections and the deployment phase for online recording. BSS is available for research purposes by contacting the authors.

2 Motivation

Designing and implementing a multimodal HMI system typically involves the recording of data for two kinds of purposes. First, a large data corpus has to be collected to train statistical models of, for example, user states or actions. Second, once the system is working, data has to be recorded and passed to a classification system with minimal time delay. The setup for the two scenarios

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 33–39, 2010.

may differ in recording parameters, like sensors, the underlying hardware, operating systems, and the requirements to the recording software. In the collection stage, data is archived after recording, while for the deployed system, data has to be passed on to the classifier in real-time. To avoid usage of different tools for the same task, we developed the BSS framework, which can be used in both scenarios. In the development phase BSS can be used to collect multimodal data corpora for analysis and system training and in the deployment phase it provides real-time sensor data as a multimodal input layer for the recognition system.

At the Cognitive Systems Lab, BSS is in use for several large-scale data collections, each with a different focus. All of them have in common that they require the recording software to capture the biosignals and all necessary meta information in a reproducible and coherent way. BSS offers a convenient and streamlined format to store all signals, corresponding timestamps, comments and the configuration of the recordings for each experiment.

For the integration into end-to-end systems, the requirements on the recording software are different. We need stable real-time recording of multiple parallel streams and the possibility to pass the recorded data to a classification system on the same or a different machine. As the recording software may come to use in very different settings, it should be portable, light-weight and integrate seamlessly into new environments.

BSS offers all of those features. By using the same software for data collection and recording for runtime classification, we can use the same tool-chain in both cases. It also provides visualization and data storing facilities at hand.

3 Related Work

There is a huge variety of commercial data-recording tools available. However, to our best knowledge, no tool offers all of the advantages of the BSS.

The first group of software products are commercial systems bundled with recording devices, distributed by hardware manufacturers. They usually have proprietary interfaces and cannot be used in combination with other manufacturers devices or custom developed systems. This causes problems when changing sensor setups (e.g. for testing in different environments) or using sensors of different origin in parallel. Depending on the software's transparency, it is difficult to implement online processing of the recorded data or additional stimulus presentation. Examples for this kind of software are GRECORDER [1] or LABCHART [2].

The second group of software products are more general frameworks which allow recording from arbitrary devices. The most prominent representative of this group is certainly LABVIEW [3]. With a graphical programming interface, LABVIEW allows the design of complex arrangements of building blocks for data acquisition and processing to form a complete and potentially parallel data flow graph. Many sensor manufacturers provide LABVIEW modules which can be integrated as data sources. While LABVIEW definitely is a very powerful tool, there are major differences to the BSS which make the latter more suited for the application in human-computer interaction: While LABVIEW requires a large framework to

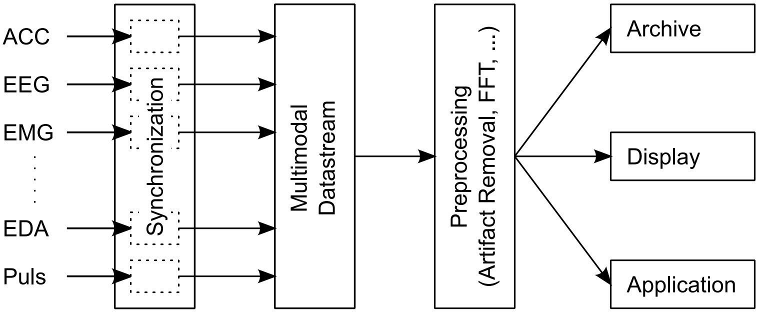


Fig. 1. Overview of the data flow oriented BSS Architecture. BSS can take various modalities in parallel as input, a selection of implemented input modalities is depicted on the left-hand side. Synchronization is performed before merging data streams to ensure minimal synchronization errors. Different output modalities can be selected.

be installed, the BSS only requires a Python interpreter and a few packages and is therefore highly portable. As it is implemented in an open, popular general-purpose programming language, the addition of existing or new modules for signal recording and processing is very simple in the BSS. In [4], an interesting framework called OmniRoute for recording different sensor modalities in the context of affective computing is proposed. However, the development of OmniRoute was discontinued and the framework is not available to the public.

Biofeedback software, such as BioEra [5] or BrainBay [6] has elaborated visualization features, has real-time capabilities and supports different recording hardware. However, these systems are neither designed for large-scale data collection, nor do they have the flexibility to get easily integrated into research applications.

4 BiosignalsStudio

4.1 Architecture

Overview: The BSS framework was designed to be extensible, multimodal, flexible, portable, and to allow short development cycles for application developers. A sophisticated modularization is essential for the system. A simple application based on the framework merely requires the configuration and definition of data connections between existing modules. Figure 1 gives an overview of the architecture.

A module is a component designed to cover one specific task. The fine-grained separation of tasks ensures the greatest possible flexibility of the framework. For example, the tasks of recording audio data and saving it in a specific format are carried out by two different modules. This allows for a flexible migration to another audio data format through replacement of the data saving module. Modules with similar tasks are interchangeable without further modifications due to standardized module interfaces.

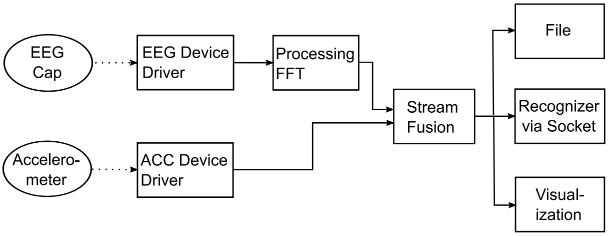


Fig. 2. Example module configuration for a recording setup with an EEG cap and an accelerometer as input modalities. Sensor data is fed into the framework via input modules (device drivers) and processed by a FFT and a fusion module. Data is then written to a file, send to another application and visualized on screen.

Multiple modules have to be connected to build a functional unit. Modules are connected through streams which represent the data flow in a specific application. All modules implement input and/or output streams for exchanging data with other modules. The input/output interface can be connected with any number of data senders or receivers. This allows the distribution of the same data across multiple modules, e.g. for saving and graphically representing the recorded data. Generic standard formats for the data exchange ensure reusability and interoperability of the modules.

The design goal of multimodality is realized by the option to instantiate and to start any number of modules in parallel, which allows recording, synchronization and analysis of multiple data sources at the same time. Despite some Python limitations concerning multi-core development, the system can execute performance-critical modules as sub-processes to use multiple cores. Furthermore, Python is designed to easily cooperate with other programming languages (in particular, C++) so that these languages also can be used to extend BSS.

Streams: The Framework has a data flow oriented architecture. All data is organized in data streams. A stream has a source and a target, the source is typically a device driver for a sensor and a target a file or socket. Data streams consist of modules through which the data is sequentially passed through. The framework offers abstract base classes, that implement the standardized interface to read data from streams and write data to streams. A stream can be split into different individual streams or different streams can be merged together. Figure 2 shows a typical example application scenario. All classes derived from the module base class can be easily plugged together to construct new data streams, thereby ensuring reusability of written modules.

Within a stream, data itself is organized in channels, typically one channel corresponds to a one-dimensional sensory input device. Besides the data, a channel can contain arbitrary additional meta information, for instance, time stamps needed for synchronization can be saved in the meta information.

Synchronization: A key issue in multimodal data acquisition is the synchronization of the different data sources to be able to correlate the modalities with

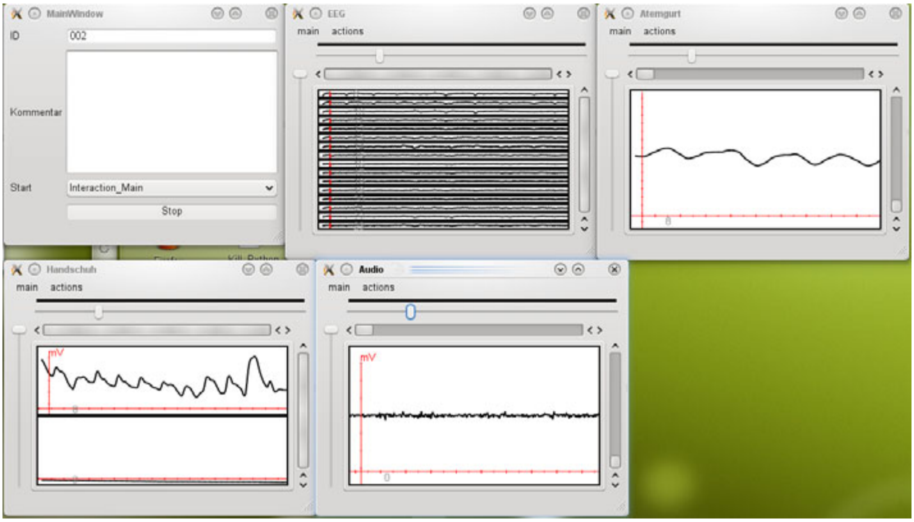


Fig. 3. Screenshot of the visualization module. A live visualization of sensor data from the following devices is shown: a 16 channel EEG (top row, middle), a respiration belt (top row, right), a wireless sensor glove measuring pulse (second row, left), a microphone (second row, right). All windows are resizable and scalable for signal inspection.

each other. Our framework offers a simple but effective approach to deal with this problem. Whenever a device driver module reads new input data, a timestamp depending on the system time is attached to the meta information of the data. The threaded device drivers ensure that new data is read with a minimum delay and thereby the timestamp is as close as possible to the actual time the data was acquired by the sensor.

Visualization: The BSS framework also includes modules to visualize acquired signals in real-time through OpenGL. The visualization of different signals can be freely arranged in individual windows and layouts. Figure 3 shows an example with different signal sources. The visualized signal can be mapped by an arbitray function (e.g. scaling to correct constant offsets, logarithmic axes, etc.).

4.2 Available Modules

The BSS is packaged with a variety of modules which can be used to build new applications. There are different types of modules available with different roles within a project:

Input Modules: Input modules insert new data into the stream. The data can for example be acquired by reading hardware devices or by artifical generation. The current BSS comes with input modules for accessing the biosignal recording interface by Becker Meditec (including EEG, EMG, Respiration, EDA, Plethysmography), by recording interfaces from generic audio devices and video cameras, modules for recording from different input devices (keyboard, steering wheel, etc.), and signal generators for testing. New input modules for different devices can be added easily, in particular when using serial connections (e.g. USB, Bluetooth). It is sufficient to develop a new device driver input module that implements the functionality needed to communicate with the device.

Processing Modules: The BSS is not limited to simple input-output stream topologies. There is a selection of intermediate processing modules that help to manipulate the incoming data before it is sent to other components. Typical processing modules that are available in the BSS are converters between different data formats, filters and mathematical operations, especially for enhanced visualization (Butterworth filters, Fast Fourier Transform, etc.), and internal channel manipulators (e.g. channel merger, temporal storage, etc.).

Output Modules: Data recording with the BSS is typically only the first step in a more complex data processing chain. The BSS therefore offers a variety of output channels. This includes writing to files (with timestamps), online visualization for signal inspection, and generic socket connections to access other components of a human-machine interface, for example statistical classifiers.

Interactive Modules: Interactive components are part of recording setups in which stimuli are presented to the user or in which user input is required. They can be started in parallel to the recording modules, in fact most interactive modules also act as input modules for logging their events. Interactive modules can be of any type, the BSS comes with a set of cognitive tests (e.g. Flanker, Oddball, Stroop, etc.) and modules for multimodal stimulus presentation.

5 Applications

The BiosignalsStudio is already applied in a variety of diverse applications for the investigation of human-machine interaction and human cognition. This includes both, large-scale data collection and online classification tasks. The following section gives an overview over the most interesting ones, highlighting the different requirements for the BSS framework.

HMI in Dynamic Environments: In a driving simulator, we are working on cognitive interaction systems in the car that are able to adapt to inner states of the driver [7], [8]. To detect those inner states, a variety of different biosignals is continuously recorded, including video, voice, several physiological signals, and steering angle. With these many input streams, synchronous recording of parallel streams is critical. As with many input devices comes an increased chance of hardware failure, the BSS has some abilities to recover from situations of lost connections or missing data.

Cognitive Fitness: To investigate the effect of physical activity on cognitive fitness, the BSS contains a whole battery of cognitive tests which can be easily selected and started from a convenient graphical user interface. This is especially useful for larger data collections where student researchers with limited training are employed to start and supervise the recordings. To facilitate later segmentation and labeling of the data, logging of events during a test (e.g. button pressing, stimulus presentation) is completely integrated into BSS.

Brain Computer Interfaces: For research on the recognition of imagined body movements and unspoken speech using EEG streams, we employ BSS to collect data of various test persons to investigate interpersonal differences. BSS has the ability to present a variety of stimuli to the test person (written words, sounds, videos) which are synchronized into the biosignal stream for labeling.

Online Workload Classification: For the development of an empathic humanoid robot that is able to detect and react to its user's mental workload, we developed an online workload recognizer based on EEG data of which the BSS is an integral part [9]. The recognizer works in real-time and receives the data stream from the BSS module via a socket connection.

Airwriting Recognition: For the online recognition of letters and whole words written in the air [10], arm movement is captured by recording acceleration sensor and gyroscope data generated by a wireless sensor glove. As the glove was self-designed, we developed and integrated a new BSS module to capture the data via a bluetooth interface. Python is available for various plattforms, including smartphones, so we currently strive for the integration of the system, including BSS, on a mobile device.

References

[1] Guger Technologies OG, Herbersteinstrasse 60, 8020 Graz, Austria: g.Recorder Biosignal Recording Software, http://www.gtec.at

[2] ADInstruments Pty Ltd, Unit 13, 22 Lexington Drive Bella Vista, NSW, Australia: LabChart (2153), http://www.adinstruments.com

[3] National Instruments Corporation, 11500 N Mopac Expwy, Austin, TX, USA: LabView, http://www.ni.com/labview

[4] Mader, S., Peter, C., Göcke, R., Schultz, R., Voskamp, J., Urban, B.: A Freely Configurable, Multi-modal Sensor Sytem for Affective Computing. In: Proceedings of the Affective Dialogue Systems Workshop, Kloster Irrsee, Germany (2004)

[5] PROATECH LLC: BioEra, http://www.bioera.net

[6] Veigl, C.: An Open-Source System for Biosignal- and Camera-Mouse Applications submission for the Young Researchers Consortium of the ICCHP 2006 (2006), http://www.shifz.org/brainbay

[7] Putze, F., Schultz, T.: Cognitive Dialog Systems for Dynamic Environments: Progress and Challenges. In: Proceedings of the 4th Biennial Workshop on DSP for In-Vehicle Systems and Safety (2009)

[8] Putze, F., Jarvis, J.-P., Schultz, T.: Multimodal Recognition of Cognitive Workload for Multitasking in the Car. In: Accepted for Publication in the Proceedings of the 20th International Conference on Pattern Recognition (2010)

[9] Heger, D., Putze, F., Schultz, T.: Online Workload Recognition during Cogitive Tests and Human-Computer Interaction. In: Dillmann, R., et al. (eds.) KI 2010. LNCS (LNAI), vol. 6359, pp. 402–409. Springer, Heidelberg (2010)

[10] Amma, C., Gehrig, D., Schultz, T.: Airwriting Recognition using Wearable Motion Sensors. In: Proceedings of the First Augmented Human International Conference (2010)

Local Adaptive Extraction of References

Peter Kluegl, Andreas Hotho, and Frank Puppe

University of Würzburg,
Department of Computer Science VI
Am Hubland, 97074 Würzburg, Germany
{pkluegl,hotho,puppe}@informatik.uni-wuerzburg.de

Abstract. The accurate extraction of scholarly reference information from scientific publications is essential for many useful applications like BIBTEX management systems or citation analysis. Automatic extraction methods suffer from the heterogeneity of reference notation, no matter wether the extraction model was handcrafted or learnt from labeled data. However, references of the same paper or journal are usually homogeneous. We exploit this local consistency with a novel approach. Given some initial information from such a reference section, we try to derived generalized patterns. These patterns are used to create a local model of the current document. The local model helps to identify errors and to improve the extracted information incrementally during the extraction process. Our approach is implemented with handcrafted transformation rules working on a meta-level being able to correct the information independent of the applied layout style. The experimental results compete very well with the state of the art methods and show an extremely high performance on consistent reference sections.

1 Introduction

Extracting references from scientific publication is an active area of research and enables many interesting applications. One example is the analysis of cited publications and the resulting citation graph, another is the automatic creation of an internet portal [1]. In general, the extracted fields of a reference are equal to the fields of the well-known BIBTEX format which are used in many applications, e.g. in the social publication sharing systems Bibsonomy[1]. In order to identify duplicate or new references in these applications hash keys could be applied [2] on a restricted number of the BIBTEX fields, i.e. *Author*, *Title*, *Editor* and *Date*, moving these fields into the focus of this paper. Machine learning and sequence labeling approaches [3,4] are often used for such an extraction task. These methods learn a statistical model using training sets with labeled data and apply these models on newly and unseen documents. No information of the unlabeled documents is used to adapt such models in its application phase which is a strong limitation of these approaches. The heterogeneous styles of the references make a suitable generalization difficult and decrease the accuracy of the extraction task.

This paper introduces a local adaptive information extraction approach that improves the extracted information online by using information of unlabeled documents directly

[1] http://www.bibsonomy.org/

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 40–47, 2010.

during the extraction task. The proposed approach utilizes the homogeneity of references in one document which are defined by the author and by the prescribed style guides of the journal or conference. Our proposed solution to handle heterogeneous documents is based on a short term memory and on the analysis of the occurring patterns in the document. Handcrafted rules are then applied on these local patterns and provide an automatic adaption on the internal previously unknown consistency of the documents. This approach considerably improves the accuracy of the extracted references and can be applied in all domains with documents containing several expected information created by the same creation process (c.f. [5]).

The rest of the paper is structured as follows: Section 2 describes the transformation-based extraction technique, the basic idea and the application of the proposed approach. Then, the evaluation setting and experimental results are presented and discussed in section 3. Section 4 gives a short overview of the related work and section 5 concludes with a summary of the presented work.

2 Method

2.1 Transformation-Based Information Extraction

Brill [6] introduced one of the first transformation-based natural language processing approaches for the part-of-speech tagging task. The transformation rules were learnt with a gold standard and corrected the errors of a simple, initial tagger. Nahm [7] utilized this technique to increase the recall of precision-driven matching rules in an information extraction task. The TextMarker system [8] provides a language and development environment for the transformation-based approach. The handcrafted rules match on a combination of annotations and create new annotations or manipulate existing annotations. However, in contrast to other rule engines for information extraction, the rules are executed similar to an imperative programming paradigm in the order they are listed. Each rule element of a TextMarker rule consists of a matching condition and of an optional quantifier, condition part and action part. The following rule with two rule elements tries to match on an *Author* (first rule element) and on a following *EditorIndicator* annotation (second rule element). If the match was successful, then the *Author* annotation is removed and a new *Editor* annotation is created[2].

```
Author{-> UNMARK(Author)} EditorIndicator{-> MARK(Editor,1,2)};
```

This transformation-based approach is an elegant way to create information extraction applications and enables the knowledge engineer to reuse existing information rather efficiently. Additionally to its comprehensible and extensible rule language, the TextMarker system alleviates the knowledge acquisition bottleneck by supporting the knowledge engineer with a test-driven development and rule learning framework.

2.2 Basic Idea of Local Adaptivity

A text document with references like scientific publication is often created in a single creation process, e.g., an author writes an article by using LaTeX. These documents contain similar styles, e.g., an author uses in a single document always the same layout for

[2] A detailed description of the syntax and semantic of the TextMarker language can be found at http://tmwiki.informatik.uni-wuerzburg.de/

headlines or the used BIBTEX style determines the appearance of the references in a paper. Patterns describing these similarities and regularities can be detected and used to improve the extracted information. However, patterns can vary strongly between different documents in a domain. A common problem with extracting references is the different style resulting in conflicting patterns for global models, e.g., different separators between interesting fields or different sequences of the fields. Based on prior work [5], the approach presented in this paper tries to apply a short term memory to exploit these intra-document similarities. A simple information extraction component extracts instances and fields of information similar to transformation-based, error-driven techniques [6,7]. The next phase however does not simply try to correct errors using a gold standard, but rather applies a local adaption process (AIE). An analysis of the occurring patterns returns a set of conflicting information that can be corrected using our transformation-based method. This approach works best if the assumption holds that the document contains several instances of information and was created in the same process. The whole process is depicted in figure 1.

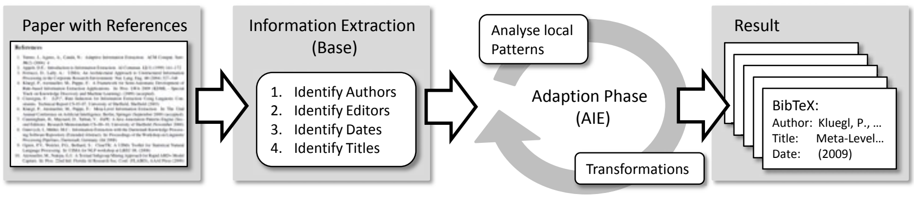


Fig. 1. Overview of the adaptive process with initial information extraction, analysis step and transformation phase

2.3 Implementation

Both components the BASE and the meta approach were handcrafted using the TEXTMARKER system. The BASE component consists of a simple set of rules for the identification of the *Author*, *Title*, *Editor* and *Date*. For a quick start 10 references of the CORA dataset (cf. sec. 3.2) are used to handcraft these initial rules. The given features are token classes and word lists for months, first names, stop words and keywords, e.g., indicators for an editor. Additional features are created during the transformation process which makes it impossible provide a complete list of them in this paper. After some reasonable rules were handcrafted, 11 randomly selected papers of the workplace of the authors are used to develop the adaptive component AIE and to refine the rules of the BASE component (cf. dataset D_{Dev} in sec. 3.2). The separators located at the beginning and the end of the BIBTEX fields and the sequence of the fields are sufficient for a description of the applied creation process. Therefore, handcrafted rules aggregate the separators and sequences of the initialy extracted information to the local model by remembering the most occurring separator and neighbors of each field. This local model is compared with all initially extracted information for an identification of possible conflicts. After this analysis phase, transformation rules are applied that utilize the local model and the information about conflicts in order to restore the consistency of the extracted information and therefore to correct the errors of the BASE component. Processing a reference sections with several conflicting style guides requires the adaptation of our rules to improve the robustness of the approach. Therefore, the rules also rely on normal patterns of the domain, e.g., the *Author* normally contains no numbers. This analysis and transformation step is iterated if necessary in order to incrementally improve the extracted information. The complete set of used rules is available on the web.[3]

2.4 Example

The behavior of our proposed approach is illustrated along the following example which is a typical text fragment representing a single reference of a reference section:

> Kluegl, P., Atzmueller, M., Puppe, F., TextMarker: A Tool for Rule-Based Information Extraction, Biennial GSCL Conference 2009, 2nd UIMA@GSCL Workshop, 2009

A reasonable result of the BASE component contains four errors: "TextMarker:" is part of the *Author* and is missing in the *Title*. The *Title* and *Date* contain additional tokens of the conference. Inline annotations for the *Author* (**<A>**), *Title* (**<T>**) and *Date* (**<D>**) are used as a simple description:

> **<A>**Kluegl, P., Atzmueller, M., Puppe, F., TextMarker:**</A>** **<T>**A Tool for Rule-Based Information Extraction, Biennial GSCL Conference**</T>** **<D>**2009**</D>**, 2nd UIMA@GSCL Workshop, **<D>**2009**</D>**

The analysis of the document results in the following patterns describing the internal consistency of the references: Most of the identified *Author* and *Title* annotations end with a comma. The *Title* follows directly after the *Author* and the *Date* is located near the end of the reference. With this information at hand, conflicts for the first *Date*, the end of the *Author* and *Title* are identified. Several handcrafted transformation rules are applied to solve these conflicts. For our example, the *Author* and *Title* are both reduced to the last comma and only the last *Date* is retained. The following exemplary rule shifts the end of the *Author* by a maximum of four tokens (*ANY*) to the next separator listed in the local model (*EndOfAuthor*) if a conflict was identified at the end of the *Author* annotation (*ConflictAtEnd*):

```
ANY+{PARTOF(Author) -> UNMARK(Author), MARK(Author,1,2)}
  EndOfAuthor ANY[0,4]?{PARTOF(Author)} ConflictAtEnd;
```

However, after these changes the *Title* does not directly follow the currently detected *Author* field. Therefore the *Title* is expanded which results in the following correct annotation of our example:

> **<A>**Kluegl, P., Atzmueller, M., Puppe, F.,**</A>** **<T>**TextMarker: A Tool for Rule-Based Information Extraction,**</T>** Biennial GSCL Conference 2009, 2nd UIMA@GSCL Workshop, **<D>**2009**</D>**

The transformation rules used to correct the errors of the BASE level approach match on a meta-level. They are completely independent of the specific local patterns detected for the current document. This is an immense advantage of our approach, because the same rules will work with different and unknown separators or field sequences.

[3] http://ki.informatik.uni-wuerzburg.de/~pkluegl/KI2010

3 Experimental Study

3.1 Performance Measures

Following the evaluation methodology of other reference extraction publications (cf. [3]) we apply the word level F_1 measure, defined as follows:

$$Precision = \frac{tp}{tp+fp}, \quad Recall = \frac{tp}{tp+fn}, \quad F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision+Recall}$$

The true positives (tp) refer to the amount of all correctly identified alpha-numeric tokens within a labeled field, the false positives (fp) to the number of tokens erroneously assigned to a field and the false negative (fn) to the amount of erroneously missing tokens. As a second measure we will use the $Instance$ accuracy which is the percentage of references in which all fields are correctly extracted.

3.2 Datasets

For a comparable evaluation of the presented approach we refer to the commonly used CORA data set (D_{Cora}: 500 References, D_{Cora}^{All}: 489 References[4]) [1]. However, this data set is not directly applicable for the development and evaluation of the approach as the CORA data set does not contain information about the original document of every reference which we need to derive the local patterns. Therefore, a simple script was developed in order to reconstruct reference sections as they would occur in real publications using only references originated in the available data set. D_{Cora}^{Paper} contains 299 references of D_{Cora}^{All} in 21 documents and represents a selection of papers with a strict style guide applied. Due to the simplicity of the assignment script and the distribution of the reference styles in the dataset a considerable amount of references could not be assigned to a paper. The data set D_{Cora}^{Rest} contains the missing 190 references and the data set D_{Cora}^{All} is the union of D_{Cora}^{Paper} and D_{Cora}^{Rest}. The data set D_{Dev} for the development of the adaptive component was created using 11 randomly selected publications and contains 213 references. All additional data sets are available for download.[5]

3.3 Experimental Results

Table 1 contains the evaluation results of the simple rule-based component (BASE) and the combination with the adaption phase (AIE). The word level F_1 measure was applied on a single field and the instance accuracy was calculated for the complete BIBTEX instance. AIE was only applied on the data set D_{Cora}^{Paper} which are 61% of all references. BASE reached an average F_1 score of 95.3% and an instance accuracy of 85.4% on the development set D_{Dev}. We tuned AIE to achieve an average F_1 score of 100.0% and an instance accuracy of 100% on D_{Dev}.

The BASE component yields results for all data sets which are comparable with knowledge-based approaches. The results of the machine learning methods are considerably better. Merely the score of the *Editor* field outperforms the related results and

[4] D_{Cora} without 10 references (line 100-109) for the development of the BASE component and one reference with damaged markup.

[5] http://ki.informatik.uni-wuerzburg.de/~pkluegl/KI2010

Table 1. Experimental results of the normal rule set (BASE) alone and with the adaption phase (AIE) for the additional data sets. The results for the data set D_{Cora}^{Rest} always refer to the BASE component alone since no adaption was applicable.

	BASE			AIE			Peng[3]	ParsCit[4]
	D_{Cora}^{Paper}	D_{Cora}^{Rest}	D_{Cora}^{All}	D_{Cora}^{Paper}	D_{Cora}^{Rest}	D_{Cora}^{All}	D_{Cora}	D_{Cora}
Author	98.4	98.3	98.4	99.9	98.3	99.3	99.4	99.0
Title	96.4	95.9	96.2	99.2	95.9	97.9	98.3	97.2
Editor	100.0	92.9	94.9	100.0	92.9	94.9	87.7	86.2
Date	98.1	95.8	97.3	99.9	95.8	98.3	98.9	99.2
Average	98.3	95.7	96.7	99.7	95.7	97.6	96.1	95.4
Instance	89.0	81.6	86.1	98.7	81.6	92.0	-	-

implies that transformation rules seem to be very suitable for this task. The overall lower performance for the *Date* field is caused by the fact that the development data set contains no date information with a time span. Hence the BASE component missed several true positives of the test data sets.

The adaptive approach is able to improve the accuracy of the initial rule set for both data sets, D_{Cora}^{Paper} and D_{Cora}^{All}. The results of D_{Cora}^{Rest} refer always to the result of the BASE component since the context of the references is missing and the local adaption cannot be applied. The AIE approach achieves a remarkable result on the (D_{Cora}^{Paper}) data set with an average F_1 score of 99.7%. 98.7% of the references are extracted correctly without a single error. This is a reduction of the harmonic mean error rate by 88.2% for the complete BIBTEX instances. Errors in the extraction process can be observed for complicated references as well as for simpler ones. The presented approach is rather resilient to the difficulty of such references because the approach extracts difficult references correctly by learning from other references of the same document. There was no adaption applied for the *Editor* field. The development data set did not encourage any adaption of the *Editor*, because the BASE component already achieved a F_1 score of 100.0% for this part on the development set D_{Dev}, resulting in a limited amount and quality of the necessary rules.

Although the adaption phase of our approach was only applied to 61% of the references of the data set D_{Cora}^{All}, its evaluation results are able to compete with the results of Peng [3] and ParsCit [4], the state of the art approaches to the best knowledge of the authors. Our results are difficult to compare the results from other researchers. First of all, the achieved outcomes are already very good and admit only marginally improvements for this data set. The results of all three approaches were accomplished with different training or development data sets: A defined amount of references [3], a 10 fold cross evaluation [4] and an external data set for the presented approach. The set of features applied, e.g., the content of the additional word lists, or even the tokenizer used to count the true positives may vary between different approaches. Besides that, it is difficult to compare a knowledge engineering approach with machine learning methods, because the knowledge engineer contributes an intangible amount of background knowledge to the rule set. A different approach to evaluate the extraction result might introduce a better comparability of the applied methods and their benefit in real applications. Matching the extraction results of an available large data set, e.g., the ACL Anthology Reference Corpus [9], against a defined database of references could provide this information.

4 Related Work

The extraction of references is an active area of research mainly dominated by machine learning methods. Techniques based on Hidden Markov Models, Maximum Entropy Models, Support Vector Machines, and several approaches using Conditional Random Fields (CRF) were published. Peng and McCallum [3] underlined the applicability of CRF with their results on the CORA data set and established CRF as the state of the art approach for the reference extraction task. 350 references of the CORA data sets were used for training and 150 references for the evaluation of the approach. Councill et al. [4] also applied CRF on the CORA data set and used two other data sets for their evaluation. The results of both approaches are depicted in table 1.

There are some knowledge engineering approaches. One is Cortez et al. [10] which proposed an unsupervised lexicon-based approach. After a chunking phase each text segment is matched against an automatically generated, domain-specific knowledge base in order to identify the fields. They evaluated their approach on data sets of two different domains from health science and computer science. Day et al. [11] used template matching to extract references from journal articles with a well-known style and achieved an average field-level F_1 score of 90.5% for the extraction of the BIBTEX fields *Author*, *Title* and *Date*.

5 Conclusions

We presented a novel adaptive information extraction approach which we successfully applied on the extraction of references from scientific publications. The information extraction component is extended by a short term memory and utilizes the homogenous information of e.g. one document on a meta-level. The approach is able to increase the accuracy of the initially extracted information and performs considerably well compared to state of the art methods. Applied on consistent references sections, an average F_1 score of 99.7% is achieved. Additionally, the results can easily be improved by extending the approach to other BIBTEX fields, because of its disjoint partition of the information: If, for example, a publisher or journal was identified, then other possible labels like the title can easily be excluded. Furthermore, the nature of a knowledge engineering approach includes also the possibility to improve the extraction component by investing more time in extending and refining the rule set.

There is ongoing work to either combine or replace the presented knowledge engineering approach with machine learning methods. The straightforward combination with statistical methods is given by extending their feature extraction with the results of the adaption phase or by applying the described approach afterwards. The adaption phase is independent of the initial BASE component and can be combined with arbitrary information extraction components. Staying with the rule-based information extraction, well-known coverage algorithms like LP^2 [12] can be adapted to automatically acquire the knowledge necessary for this approach.

References

1. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Automating the Construction of Internet Portals with Machine Learning. Information Retrieval Journal 3, 127–163 (2000)
2. Voss, J., Hotho, A., Jaeschke, R.: Mapping Bibliographic Records with Bibliographic Hash Keys. In: Kuhlen, R. (ed.) Information: Droge, Ware oder Commons? Proceedings of the ISI, Hochschulverband Informationswissenschaft. Verlag Werner Huelsbusch (2009)
3. Peng, F., McCallum, A.: Accurate Information Extraction from Research Papers using Conditional Random Fields. In: HLT-NAACL, pp. 329–336 (2004)
4. Isaac Councill, C.L.G., Kan, M.Y.: ParsCit: an Open-source CRF Reference String Parsing Package. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco, ELRA (2008)
5. Kluegl, P., Atzmueller, M., Puppe, F.: Meta-Level Information Extraction. In: The 32nd Annual German Conference on Artificial Intelligence. Springer, Berlin (September 2009)
6. Brill, E.: Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. Computational Linguistics 21(4), 543–565 (1995)
7. Nahm, U.Y.: Transformation-Based Information Extraction Using Learned Meta-rules. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 535–538. Springer, Heidelberg (2005)
8. Kluegl, P., Atzmueller, M., Puppe, F.: TextMarker: A Tool for Rule-Based Information Extraction. In: Proceedings of the Biennial GSCL Conference 2009, 2nd UIMA@GSCL Workshop, pp. 233–240. Gunter Narr Verlag (2009)
9. Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.Y., Lee, D., Powley, B., Radev, D., Tan, Y.F.: The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco, ELRA (2008)
10. Cortez, E., da Silva, A.S., Gonçalves, M.A., Mesquita, F., de Moura, E.S.: FLUXCiM: Flexible Unsupervised Extraction of Citation Metadata. In: JCDL 2007: Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries, New York, USA, pp. 215–224 (2007)
11. Day, M.Y., Tsai, R.T.H., Sung, C.L., Hsieh, C.C., Lee, C.W., Wu, S.H., Wu, K.P., Ong, C.S., Hsu, W.L.: Reference Metadata Extraction using a Hierarchical Knowledge Representation Framework. Decis. Support Syst. 43(1), 152–167 (2007)
12. Ciravegna, F.: $(LP)^2$, Rule Induction for Information Extraction Using Linguistic Constraints. Technical Report CS-03-07, University of Sheffield, Sheffield (2003)

Logic-Based Trajectory Evaluation in Videos

Nicola Pirlo and Hans-Hellmut Nagel

Fakultät für Informatik
Karlsruher Institut für Technologie (KIT)
76049 Karlsruhe
pirlo@iaks.uni-karlsruhe.de nagel@iaks.uni-karlsruhe.de

Abstract The study of 3D-model-based tracking in videos frequently has to be concerned with details of algorithms or their parameterisation. Time-consuming experiments have to be performed in this context which suggested to (at least partially) automate the evaluation of such experimental runs. A logic-based approach has been developed which generates Natural Language textual descriptions of evaluation runs and facilitates the formulation of specific hints. The implementation of this approach is outlined, results obtained with it on an extended complex real-world road traffic video are presented and discussed.

1 Introduction

Performance assessment can be advanced by a systems approach whose capabilities support its own evaluation. Such an approach is discussed in the sequel for a system which converts a video stream into a **N**atural **L**anguage (NL) text describing the developments within the recorded scene.

The current version of the system approach which has been pursued at the former IAKS during the past decades – see [17] – can be decomposed into four subsystems (see, e.g., [1]): (i) the Interaction Subsystem which comprises camera control and the Human-Machine-Interface, (ii) the Computer Vision Subsystem which detects, initializes, and tracks moving vehicles in a recorded video (see [10]), (iii) the Conceptual Subsystem which relies on the **S**ituation **G**raph **T**ree (SGT) Formalism in order to represent the conceptual a-priori knowledge for the interpretation of the extracted trajectory data (see Section 4), and (iv) the NL Subsystem which transforms an instantiation of the conceptual representation for the frame-period in question into a NL text (see [12] and the literature quoted there).

After having implemented a version of the entire chain from raw video data to the generation of NL text, the emphasis of our research has shifted to finding quantitative and qualitative performance bottlenecks along this chain. It then has to be decided which modifications of algorithms or their parametrisation may reduce or even remove a local bottleneck without decreasing the overall system performance. Such an approach necessitates meticulous evaluation of each tentative modification in order to avoid that an apparent reduction of one deficiency is paid for by the (initially unnoticed) appearance of new deficiencies

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 48–57, 2010.

elsewhere. As a consequence, algorithmic support of evaluation efforts becomes increasingly important.

Two recent developments in our group have helped significantly in coping with this predicament, (i) the provision of accurate 3D-**G**round **T**ruth (GT) for moving road vehicles in the inner-city street scene recorded by a monocular video camera – see Section 3 – and (ii) the adaptation of our conceptual and NL subsystems for the algorithmic generation of a NL *evaluation report*.

We distinguish between 'agent' which denotes a vehicle in the scene and 'actor' which denotes a system internal representation of an agent. Other approaches of performance evaluation are compared with our particular requirements (Section 2). Our acquisition of 3D-GT is outlined in Section 3 and its exploitation for performance evaluation in Section 4, with an illustration and discussion of results in Section 5.

2 Related Work

Computer Vision Systems have reached levels of performance, speed, and cost which have let them become applicable for a variety of purposes – see, e.g., [14, Section 1]. Concomitantly, the evaluation of their performance has led to numerous publications mentioned in Table 1. Space restrictions do not permit to treat each publication in detail. Instead, we assigned these publications to subdivisions for each of three different aspects, namely (i) the representation of GT used as a basis for the evaluation process, (ii) the association between tracking results and GT, and (iii) the characteristics used to rate the performance of the tracking system.

As it can be seen in Table 1, in most cases (hand-labelled) 2D-regions within the image plane are used as GT-representation, usually rectangles or simple polygons, the GT-columns labeled 'boundingbox' and '2D' are well populated. This does not surprise because bounding boxes and 2D-regions facilitate fast tracking with comparatively small computational resources. The GT-column 'none' refers to approaches which exploit internal consistency checks in order to assess the tracking performance without the cost of GT-creation. The column labelled with '3D' will be discussed in Section 3 and the GT-column labelled with 'other' refers to publications which rely on synthetic GT or events.

The comparison between a tracking result and GT can be performed at different levels: (i) at the '*pixel*' level, which is particularly suitable to evaluate segmentation algorithms; (ii) at the '*spatial*' level taking into account frame-wise correspondences between GT-representations and tracked body-images; (iii) at the '*spatiotemporal*' level by constructing a mapping between a GT-representation and tracked agents, taking into account *the entire trajectories* which is a non-trivial task (see [2,14,18,23] and publications referenced there). (iv) Finally, if the evaluation focusses on a higher semantic level rather than details of the tracking process, performance is evaluated via recognition of *events*. In the ETISEO Project, events are used as supplement to the other evaluation levels (see [19, part 4.5]), whereas the approaches in [22,29] are completely event driven.

Table 1. Characterisation of different publications on tracking evaluation. All properties marked with a filled bullet are attributed to the corresponding approach.

	GT					Match			Metrics			
references	*none*	*boundingbox*	*2D*	*3D*	*other*	*none*	*unique*	*multiple*	*pixel*	*spatial*	*spatiotemporal*	*event*
[2]	-	●	-	-	-	-	●	●	-	●	●	-
[3]	-	-	●	-	-	-	-	●	-	●	-	-
[4]	-	-	-	-	●	-	●	-	-	-	●	-
[5]	-	●	-	-	-	-	-	●	-	-	●	-
[6]	●	-	-	-	-	-	-	-	-	-	-	-
[7]	●	-	-	-	-	-	-	-	-	-	-	-
[9]	-	-	●	-	-	●	-	-	-	●	-	-
[10]	-	-	-	●	-	-	●	-	-	-	●	-
[11]	●	-	-	-	-	-	-	-	-	-	-	-
[14]	-	●	-	-	-	-	●	●	●	●	●	-
[15]	-	●	-	-	-	●	-	-	-	●	●	-
[16]	-	●	-	-	-	●	-	-	●	●	-	-
[18]	-	-	●	-	-	●	-	-	-	●	-	-
[19]	-	●	-	-	●	-	-	●	-	●	●	●
[20]	-	-	●	-	-	-	-	●	-	●	●	●
[21]	-	-	-	●	-	-	●	-	-	-	●	●
[22]	-	-	-	-	●	-	-	-	-	-	-	●
[23]	-	●	-	-	-	-	-	●	-	-	●	-
[25]	-	-	●	-	-	-	●	●	-	●	●	-
[26]	-	-	●	-	-	●	-	-	●	-	-	-
[27]	-	●	-	-	-	-	●	-	-	-	●	-
[28]	-	-	●	-	-	●	-	-	●	-	-	-
[29]	-	-	-	-	●	-	-	-	-	-	-	●

3 3D-GT and Its Association with Tracking Results

Whereas most contributions mentioned in Table 1 are concerned with the question whether a particular Computer Vision System approach is suitable for a given application or better than a competing approach, we are interested in assessing the details of algorithms used and their parameterisation. This necessitates the inspection of even infrequently occurring deviations because these might hint at a basic algorithmic insufficiency and thus suggest to study a modification. The Visual Subsystem relies on `Motris` (**Mo**del-Based **Tr**acking in **I**mage **S**equences), a Computer Vision System developed at the former IAKS, which automatically takes into account perspective and pose, given a camera calibration (see [10]). 3D-model information allows to compute the effects of occlusions between different tracked vehicles in the scene. Although we exploited the interactively usable inspection capabilities of `Motris` to generate the required 3D-GT-representation, this is nevertheless a time consuming task. The advantage consists in our ability to *directly compare 3D-tracking results with 3D-GT*. To the best of our knowledge, this appears to be a unique property of our approach (see column '3D' in Table 1).

A 3D-tracking result is associated with the 3D-GT by projecting the current frame estimate for a body and the corresponding frame data for the 3D-GT into the image plane and by comparing these projections according to the **J**accard **C**oefficient (JC)

$$JC = \frac{A \cap B}{A \cup B} \tag{1}$$

where A denotes the image plane area covered by a tracked body and B the area occupied by its tentatively associated GT representation. In contrast to [10, part 4] where two associated bodies are compared by the parameters of their models, the JC is applicable for different kinds of trackers and is not restricted to 3D model based tracking. Merging the assessment of localisation and model fit, it gives a measure compatible with the visual impression of a human evaluator. Note that already Cheetham & Hazel ([8]) have compared the JC with other binary similarity coefficients.

4 Logic-Based Evaluation Process

GT, tracking data, and the corresponding JCs are fed into the logic-based component of the evaluation located in the Conceptual Subsystem (see section 1). The SGT Formalism, originally introduced as a way to model behaviour of an agent and later extended (see [24,1,13]), allows us to implement complex rules within SGTs which in turn can be translated into a **F**uzzy **M**etric-**T**emporal **H**orn **L**ogic (FMTHL) program, to be traversed by an inference machine called `F-Limette` (see [24]).

This approach was chosen for several reasons. (i) The output of logic-based systems is generated by inference, which means that a proof has been created for each result. Therefore, given the evaluation rules, the system allows precise analysis of its behaviour. (ii) The formalism has been shown to be adequate to represent behaviour of moving and interacting agents. Furthermore, it has already been used as basis for NL text generation (see [1,12]). (iii) Finally, related tools integrate well with the IAKS system approach.

The output of the evaluation process is generated as a side effect of the traversal of the specifically designed SGTs. Figure 1 shows and explains a simple SGT. Each node of an SGT is composed of three parts: Identifier, Precondition and Action. The two latter parts are each a conjunction of FMTHL predicates. If during traversal a node is reached and its precondition can be satisfied, the associated action predicates are emitted.

The predicates shown in Figure 2(c) were emitted at (frame-)time 7290. The predictions reached a node whose precondition is satisfied if a tracked body has a localisation rated average or worse at the time of its initialisation and the initialisation occurred at the correct time. All actions on the path from the root of the tree to the node reached during the evaluation are emitted.

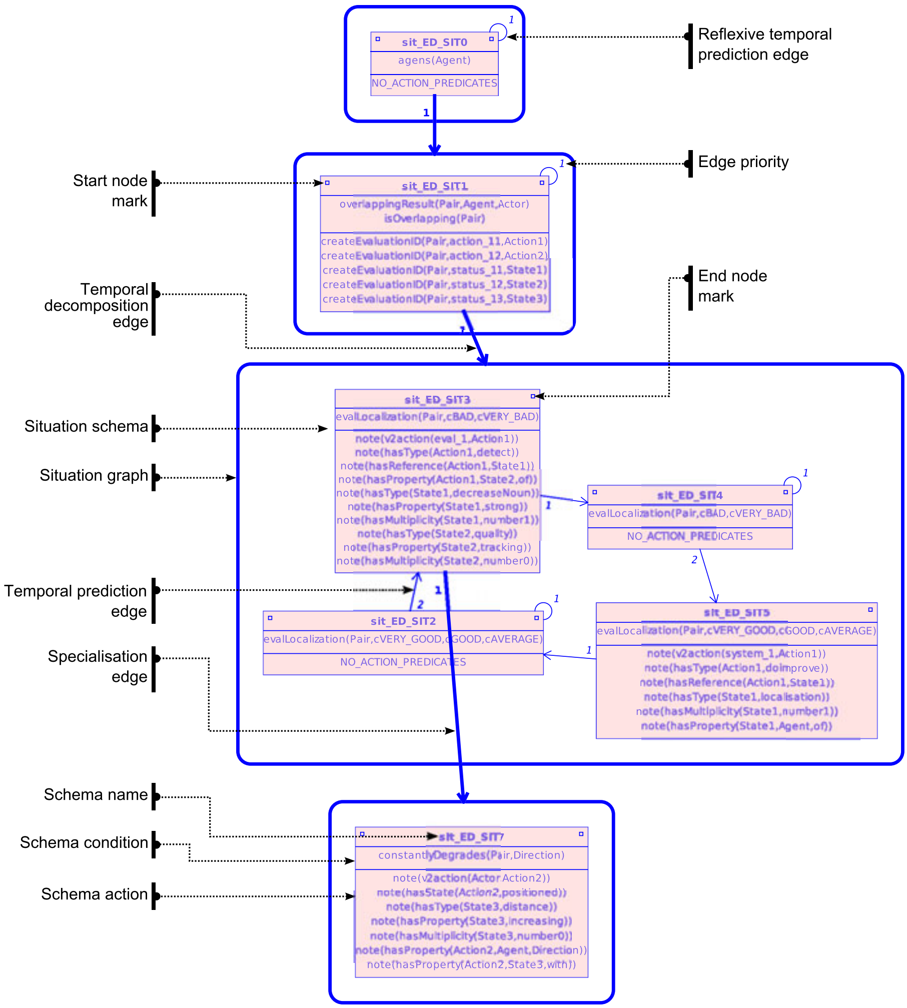

Reflexive temporal prediction edge
1
sit_ED_SIT0
agens(Agent)
NO_ACTION_PREDICATES
1
Edge priority
Start node mark
1
sit_ED_SIT1
overlappingResult(Pair,Agent,Actor)
isOverlapping(Pair)
createEvaluationID(Pair,action_11,Action1)
createEvaluationID(Pair,action_12,Action2)
createEvaluationID(Pair,status_11,State1)
createEvaluationID(Pair,status_12,State2)
createEvaluationID(Pair,status_13,State3)
Temporal decomposition edge
End node mark
1
sit_ED_SIT3
evalLocalization(Pair,cBAD,cVERY_BAD)
note(v2action(eval_1,Action1))
note(hasType(Action1,detect))
note(hasReference(Action1,State1))
note(hasProperty(Action1,State2,of))
note(hasType(State1,decreaseNoun))
note(hasProperty(State1,strong))
note(hasMultiplicity(State1,number1))
note(hasType(State2,quality))
note(hasProperty(State2,tracking))
note(hasMultiplicity(State2,number0))
Situation schema
Situation graph
1
sit_ED_SIT4
evalLocalization(Pair,cBAD,cVERY_BAD)
NO_ACTION_PREDICATES
1
2
Temporal prediction edge
2
1
1
sit_ED_SIT2
evalLocalization(Pair,cVERY_GOOD,cGOOD,cAVERAGE)
NO_ACTION_PREDICATES
Specialisation edge
1
sit_ED_SIT5
evalLocalization(Pair,cVERY_GOOD,cGOOD,cAVERAGE)
note(v2action(system_1,Action1))
note(hasType(Action1,doimprove))
note(hasReference(Action1,State1))
note(hasType(State1,localisation))
note(hasMultiplicity(State1,number1))
note(hasProperty(State1,Agent,of))
Schema name
Schema condition
Schema action
sit_ED_SIT7
constantlyDegrades(Pair,Direction)
note(v2action(Actor,Action2))
note(hasState(Action2,positioned))
note(hasType(State3,distance))
note(hasProperty(State3,increasing))
note(hasMultiplicity(State3,number0))
note(hasProperty(Action2,Agent,Direction))
note(hasProperty(Action2,State3,with))

Fig. 1. Simplified version of an SGT used for evaluation purposes. This SGT monitors the localisation quality of an actor during its time period of tracking. Variables start with an upper case letter and get assigned during the traversal. The SGT is composed of the obligatory root node at the top; a node `sit_ED_SIT1`, which is satisfied as soon as the actor and the corresponding agent overlap; a situation graph with a temporal decomposition containing the main part of the localisation check; and a node specialising schema `sit_ED_SIT3` (bottom row).

The two schemes `sit_ED_SIT2` and `sit_ED_SIT4` marked as start nodes do not emit predicates relevant to NL generation. If a significant change in localisation quality occurs, the traversal will pass by `sit_ED_SIT3` or `sit_ED_SIT5` and the corresponding actions along the path to the specified schema will be emitted. Note that the two last mentioned schemes do not have a reflexive temporal prediction edge.

In contrast to most other evaluation approaches, the SGT Formalism allows to reason about ratings. The path to a specific node may contain knowledge – encoded in the preconditions of its nodes – which can explain a certain rating. A bad initialisation for example may occur because the initialised body was partially occluded at the time of initialisation.

Use of the SGT Formalism allows rapid implementation of new rules. Modularisation is supported by using different SGTs which can be activated selectively for traversal. With this mechanism provided by Harland ([13]), the focus of attention can be changed quickly in order to support analysis of a development. The tool allowed, e.g., to find undesired changes of the steering angle of non-moving vehicles by extending the set of SGTs with an additional one.

5 Results and Concluding Discussion

The `Angus2` version used as output component of the evaluation is an extended version of the system reported in [12].

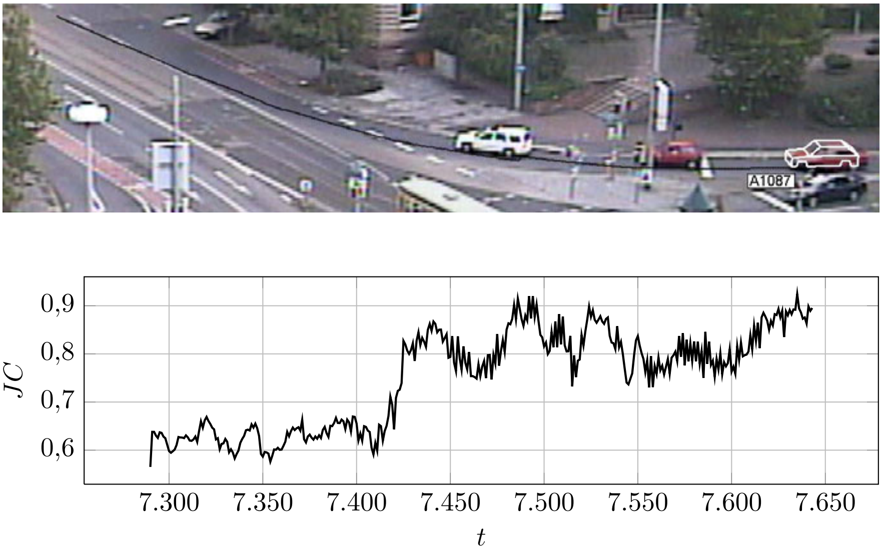
With the introduction of keys to relate several predicates to each other, the extensions allow to process predicates issued by *multiple* SGTs and to combine multiple predicates issued from *different* nodes (see also [13]). Furthermore, with a new design of the algorithm used to unify two **D**iscourse **R**eference **S**tructures (DRSs), the program could be sped up by three orders of magnitude. `Angus2` is now able to generate the text in synchrony with the evaluation process.

Results obtained with `Motris` in several experiments on the `color_rush_hour` image sequence were evaluated with the described system. The `color_rush_hour` image sequence comprises 15.000 frames and presents several difficulties like strong illumination changes, occlusions, and strong compression artifacts (see also [12]). The GT is available for all moving vehicles except trams and bikes.

Due to space restrictions, a single example will be used to demonstrate the evaluation tool. The example will examine an actor entering the scene at the right boundary of the image sequence. Figure 2(a) shows the actor at initialisation time. A human evaluator would remark that the model is not positioned well following its initialisation. The occlusion by traffic lights was recognised automatically and an automatic reinitialisation was successful. The actor was deleted in the upper left corner before it left the field of view of the camera.

Figure 2(b) shows the computed JCs over time. During the evaluation of these coefficients and the corresponding trajectory data, predicates have been emitted (see Figure 2(c)). These predicates are then used as input for `Angus2`, which generates the text shown in Figure 2(d).

The generated text begins with a simple introduction, then the initialisation is rated, and finally the tracking process is described. The text exceeds a simple rating and gives a description similar to observations reported by a human evaluator. For example, it points out that the actor was badly positioned during localisation *despite* full visibility. Furthermore it explains that the average rating of the initialisation was due to a localisation problem. In the example the initialisation rating is determined by the fact that the following



```
7290 ! has_agent_type(agent_43,agent).
7290 ! has_agent_type(actor_1087,actor).
7290 ! has_agent_type(eval_1,evaluation).
7290 ! has_agent_type(system_1,tracker).
7290 ! v2action(actor_1087,action_0107290152).
7290 ! hasState(action_0107290152,initialised).
7290 ! hasProperty(action_0107290152,system_1,by).
7290 ! v2action(eval_1,action_0207290152).
7290 ! hasType(action_0207290152,rating).
7290 ! hasReference(action_0207290152,actor_1087).
7290 ! hasProperty(action_0207290152,status_0107290152,as).
7290 ! hasType(status_0107290152,initialisation).
7290 ! hasMultiplicity(status_0107290152,number0).
7290 ! hasProperty(status_0107290152,agent_43,for).
7290 ! localizationRating(152,cAVERAGE).
7290 ! hasProperty(action_0107290152,cON_TIME).
7290 ! v2action(actor_1087,action_0307290152).
7290 ! hasState(action_0307290152,positioned).
7290 ! hasProperty(action_0307290152,left).
7290 ! hasProperty(action_0307290152,agent_43,of).
7290 ! hasDisplacement(152,cHALF).
7290 ! hasTiming(152,cONTIME).
7290 ! hasProperty(status_0107290152,average).
7290 ! hasProperty(action_0207290152,localization_2,dueto).
7290 ! hasProperty(action_0307290152,occlusion_0,despite).
```

```
@7289: The evaluation starts to check a match labelled as 'match 152'.
Match 152 describes the performance of an agent labelled as 'agent 43'
with an actor labelled as 'actor 1087'. Match 152 has changing quality
from average localisation to very good localisation.

@7290: Actor 1087 is initialised on time by the tracker. Eval 1 rates
Actor 1087 due to average localization as average initialisation
for Agent 43. Actor 1087 is positioned left of Agent 43 despite full
visibility.

@7381: Agent 43 becomes occluded. System 1 correctly recognises partial
occlusion of Agent 43.

@7387: System 1 uses behaviour-based prediction to track Agent 43.

@7403: Agent 43 gets visible again after occlusion. System 1 fails to
reinitialise Actor 1087 at the right time.

@7425: System 1 recovers from false occlusion. Actor 1087 is positioned
left of Agent 43 with very good localisation after reinitialisation.

@7643: System 1 kills Actor 1087 early.
```

Fig. 2. Illustration of the evaluation of a trajectory. The subfigures beginning at the top are: (a) Actor selected as example, (b) computed JCs, (c) excerpt of emitted predicates and (d) generated output text.

precondition holds: `evalLocalisation(Pair,V)`, `neq(V,cGOOD,cVERY_GOOD)`, `eq(Timing,cON_TIME)`. The initialisation rating is degraded, therefore, by the fact that the localisation was assessed only as ‘average’. This is important, because there exist several additional circumstances, which can decrease the initialisation rating, for example a *late* initialisation. Another detail is the sixth paragraph of the algorithmically generated text. It states that the system *recovers* from a false occlusion. This is possible due to the design of the SGT which allows to model temporal relations between situations. The evaluation process is aware of previously encountered situations on the path to the current situation inside the corresponding SGT and can, therefore, tell when the system is in the state of false occlusion. The SGT Formalism allows to define such a kind of dependency interactively by designing an appropriate tree.

This paper gives a proof of concept that the performance evaluation of tracking results can be assessed within an integrated systems approach. The logic-based evaluation shows advantages over purely quantitative approaches: (i) NL output eliminates the need to learn and understand the meaning of numerical output of the metric(s) used during the evaluation process hence allowing evaluation output to be interpreted by non-specialists. The text gives an abstract yet commonly understandable view of the evaluation result. (ii) Explicit rule definition and inference allow to better analyse evaluation results. (iii) Time dependent behaviour of the Computer Vision System can be evaluated within a well defined framework, which allows to implement complex rules with support of CASE tools. (iv) By overlaying the evaluation results as subtitles to the video, consistency between automatic and human evaluation can be checked easily.

While the system already achieves promising results there is still room for improvements. The mapping between bodies of the GT and tracked ones is still done semi-automatically so far. Another interesting extension would be to take into account shadows in order to broaden the possibilities to reason about tracking failures.

Acknowledgments

Discussions with H. Harland are gratefully acknowledged.

This research has been partially supported by the EU 6FP-Project HERMES (#027110).

References

1. Arens, M., Gerber, R., Nagel, H.H.: Conceptual Representations Between Video Signals and Natural Language Descriptions. Image and Vision Computing 26(1), 53–66 (2008)
2. Bashir, F., Porikli, F.: A Complete Performance Evaluation Platform Including Matrix-based Measures for Joint Object Detector and Tracker Systems. In: Ferryman, J.M. (ed.) Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2006), New York, NY, USA, June 18, pp. 7–14. The Print Room, Reading (2006)

3. Bernardin, K., Elbs, A., Stiefelhagen, R.: Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment. In: Sixth International Workshop on Visual Surveillance 2006 (VS 2006), Graz, Austria, May 13 (2006)
4. Black, J., Ellis, T., Rosin, P.: A Novel Method for Video Tracking Performance Evaluation. In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS 2003), Nice, France, October 11-12, pp. 125–132 (2003)
5. Brown, L.M., Senior, A.W., Tian, Y., Connell, J., Hampapur, A., Shu, C., Merkl, H., Lu, M.: Performance Evaluation of Surveillance Systems Under Varying Conditions. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005), Breckenridge, CO, USA, January 7 (2005)
6. Buch, N., Yin, F., Orwell, J., Makris, D., Velastin, S.: Urban Vehicle Tracking Using a Combined 3D Model Detector and Classifier. In: Velásquez, J.D., Ríos, S.A., Howlett, R.J., Jain, L.C. (eds.) KES 2009. LNCS, vol. 5711, pp. 169–176. Springer, Heidelberg (2009)
7. Chau, D.P., Bremond, F., Thonnat, M.: Online Evaluation of Tracking Algorithm Performance. In: The 3rd International Conference on Imaging for Crime Detection and Prevention (ICDP 2009), Kingston University, London, December 3 (2009)
8. Cheetham, A.H., Hazel, J.E.: Binary (Presence-Absence) Similarity Coefficients. Journal of Paleontology 43(5), 1130–1136 (1969)
9. Collins, R., Zhou, X., Teh, S.K.: An Open Source Tracking Testbed and Evaluation Web Site. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005), Breckenridge, CO, USA, January 7 (2005)
10. Dahlkamp, H., Nagel, H.H., Ottlik, A., Reuter, P.: A Framework for Model-Based Tracking Experiments in Image Sequences. International Journal of Computer Vision 73(2), 139–157 (2007)
11. Denman, S., Fookes, C., Sridharan, S., Lakemond, R.: Dynamic Performance Measures for Object Tracking Systems. In: Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2009), Genova, Italy, September 2-4, pp. 541–546 (2009)
12. Fexa, A.: Erzeugung mehrsprachlicher Beschreibungen von Verkehrsbildfolgen. No. 317 in Dissertationen zur Künstlichen Intelligenz (DISKI), Akademische Verlagsgesellschaft AKA GmbH, Berlin (January 2008)
13. Harland, H.: Nutzung logikbasierter Verhaltensrepräsentationen zur natürlichsprachlichen Beschreibung von Videos (in preparation 2010)
14. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., Zhang, J.: Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics and Protocol. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2), 319–336 (2009)
15. Lazarevic-McManus, N., Renno, J., Jones, G.A.: Performance Evaluation in Visual Surveillance Using the F-Measure. In: 4th ACM International Workshop on Video Surveillance and Sensor Networks (VSSN 2006), Santa Barbara, California, USA, October 23-27, pp. 45–52. ACM, New York (2006)
16. Mariano, V.Y., Min, J., Park, J.H., Kasturi, R., Mihalcik, D., Li, H., Doermann, D., Drayer, T.: Performance Evaluation of Object Detection Algorithms. In: 16th International Conference on Pattern Recognition (ICPR 2002), Quebec, Canada, August 11-15, vol. 3, pp. 965–969. IEEE Computer Society, Los Alamitos (2002)
17. Nagel, H.H.: Steps Toward a Cognitive Vision System. AI-Magazine 25(2), 31–50 (2004)

18. Nascimento, J.C., Marques, J.S.: Performance Evaluation of Object Detection Algorithms for Video Surveillance. IEEE Transactions on Multimedia 8(4), 761–774 (2006)
19. Nghiem, A.T., Bremond, F., Thonnat, M., Valentin, V.: ETISEO, Performance Evaluation for Video Surveillance Systems. In: IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2007), London, UK, September 5-7, pp. 476–481. IEEE Computer Society, Los Alamitos (2007)
20. Pingali, S., Segen, J.: Performance Evaluation of People Tracking Systems. In: Storms, P. (ed.) Third IEEE Workshop on Application of Computer Vision (WACV 1996), Sarasota, FL, USA, December 2-4, pp. 33–38. IEEE Computer Society, Los Alamitos (1996)
21. Pirlo, N.: Zur Robustheit eines modellgestützten Verfolgungsansatzes in Videos von Straßenverkehrsszenen (in preparation 2010)
22. Roth, D., Koller-Meier, E., Van Gool, L.: Multi-Object Tracking Evaluated on Sparse Events. Multimedia Tools and Applications 50(1), 29–47 (2010)
23. Saunier, N., Sayed, T., Ismail, K.: An Object Assignment Algorithm for Tracking Performance Evaluation. In: Ferryman, J.M. (ed.) 11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2009), Miami, Fl, USA, June 25, pp. 9–17. The Print Room, Reading (2009)
24. Schäfer, K.: Unscharfe zeitlogische Modellierung von Situationen und Handlungen in Bildfolgenauswertung und Robotik. No. 135 in Dissertationen zur Künstlichen Intelligenz (DISKI), Akademische Verlagsgesellschaft AKA GmbH, Berlin (July 1996)
25. Smith, K., Gatica-Perez, D., Odobez, J.M., Ba, S.: Evaluating Multi-Object Tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005), San Diego, CA, USA, June 20-26. IEEE Computer Society, Los Alamitos (2005), No. of contribution: 36
26. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and Practice of Background Maintenance. In: 7th IEEE International Conference on Computer Vision (ICCV 1999), Kerkyra, Corfu, Greece, September 20-25, vol. 1, pp. 255–261. IEEE Computer Society, Los Alamitos (1999)
27. Yin, F., Makris, D., Velastin, S.: Performance Evaluation of Object Tracking Algorithms. In: Ferryman, J.M. (ed.) 10th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2007), Rio de Janeiro, Brazil, October 14, pp. 17–24. The Print Room, Reading (2007)
28. Young, D.P., Ferryman, J.M.: PETS Metrics: On-line Performance Evaluation Service. In: 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS 2005), Bejing, China, October 15-16, pp. 317–324. IEEE Computer Society, Los Alamitos (2005)
29. Ziliani, F., Velastin, S., Porikli, F., Marcenaro, L., Kelliher, T., Cavallaro, A., Bruneaut, P.: Performance Evaluation of Event Detection Solutions: The CREDS Experience. In: IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2005), Como, Italy, September 15-16, pp. 201–206. IEEE Computer Society, Los Alamitos (2005)

A Testbed for Adaptive Human-Robot Collaboration

Alexandra Kirsch and Yuxiang Chen

Technische Universität München[*]

Abstract. This paper presents a novel method for developing and evaluating INTELLIGENT robot behavior for joint human-robot activities. We extended a physical simulation of an autonomous robot to interact with a second, human-controlled agent as in a computer game. We have conducted a user study to demonstrate the viability of the approach for adaptive human-aware planning for collaborative everyday activities. The paper presents the details of our simulation and its control for human subjects as well as results of the user study.

1 Motivation

Robots that are to interact with humans need to be highly adaptive towards their environment. They have to adapt their behavior to different human individuals, have to respect social rules and respond to human expectations, react to unexpected human actions, and show a high degree of safety, even for unanticipated events. In our research we aim to develop adaptive, model-based planning and plan-execution mechanisms for autonomous robots that are to collaborate with a human partner, for example in assistive scenarios like helping elderly people in their daily chores.

With the current state of the art in robotics research on such topics using real robots is extremely limited. There are only a few projects worldwide where complete robotic systems are available with stable state estimation, reliable low-level actions, knowledge processing and high-level planning. Even the best of these systems only support rudimentary additional capabilities needed in human-robot interaction like tracking human poses, and recognizing human activities. Besides, state-of-the art robot hardware is still relatively slow (or is slowed down for safety reasons) compared to human movements. Because of these limitations, researchers have started to use realistic physics-based simulations of robots to facilitate the development of autonomous robots.

In this paper we propose to extend the simulation-based development to robots interacting with humans by transforming a robot simulator into a simple computer game and letting people control an agent on the screen. We have implemented such an environment based on a physical simulation with two robots. One of these robots is autonomous, the other is controlled by a human operator. The human's

[*] This work was partially funded by the cluster of excellence CoTeSys.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 58–65, 2010.

control of the robot is not prerecorded, but happens at the same time while the autonomous robot is working, allowing all problems of misunderstandings and unexpected activities of the human that could occur in the real world.

For evaluating this simulation, we have performed a user study, where we investigated the usability of our simulation and compared human behavior in real life and in simulation. We had subjects perform a household task (setting and clearing the table) both in reality and simulation.

In the next section we argue for the use of simulation for research on human-robot collaboration. After that we present related work. Then we introduce our simulator and the interface for humans to control an agent. This description is followed by an account of the user study we performed to evaluate the viability of using simulation for HRI research.

2 Simulation-Based Robot Development

In robotics, simulation-based development is often rejected with the comment "in the real world everything is different than in simulation". This statement might be true for low-level control loops, but mostly stems from a time when good physical simulations were unavailable. Meanwhile, some extremely successful researchers use realistic simulation for robot development [9], carrying over the techniques developed in simulation to real robots.

The question is also what you want to develop. If the goal is to program a robot performing a certain well-defined job, you engineer this task to the real hardware, simulation would not help much here. But in our research we are rather interested in *general, adaptive* methods for autonomous robots. Our hypothesis is that if we develop techniques that enable a simulated robot to adapt to several simulated environments, it will also be able to adapt to real-world environments.

Simulation gives us the chance to move the development of adaptive robots from the area of "art" into that of "science":

- Simulation allows us to *focus* research on a particular aspect of robot behavior. Especially, inaccuracy of the state estimation can be ignored or simulated in varying degrees of complexity.
- Experiments can be *repeated.* When using real robots it is extremely hard to get the same conditions for each experiment (e.g. lighting conditions, battery charging level, object positions).
- Simulation makes available *different environments and robots* at low costs.

Beside the "simulation-is-different" objection, HRI researchers often reject simulation claiming that "you can never replace the real user experience". In our case, we are more interested in developing adaptive execution mechanisms and testing them with the dynamics and uncertainty of real user interaction than in the realistic perception of the scene by subjects. We rather rely on the ability of humans to interpret the abstracted view in the simulation than to sacrifice realism for the robot. This is also the reason why we use a *physical simulation* rather than a virtual rendering machine.

Taking these requirements into account there are good reasons to use simulation for interaction experiments as opposed to working with real robots:

- In the simulator, humans and robots operate on *the same time scale*, which means both are quite slow. Even the best available robots are very slow in sophisticated tasks like pick-and-place actions. In a realistic scenario, a human would perform a "collaborative" task mostly on her/his own.
- Interaction in simulation is a lot *safer* than that with real robots. We're not thinking of Wizard of Oz experiments where the robots are controlled by humans, but of experiments with autonomous robots, whose behavior can be hard to predict, even by the developers themselves.

Over time, we expect real robots to become faster and safer. We will then be able to carry over the adaptive methods developed in simulation to real robot applications.

3 Related Work

Ueda et al. [9] report on a realistic simulation of deformable objects for a humanoid robot. The simulation works with the normal robot code and can additionally be used by the robot for predictions and planning of its actions. This work demonstrates how a realistic simulation of the robot and its environment directly enhances the performance of autonomous robots.

For research on social imitation learning, Buchsbaum et al.[1] use two virtual characters, where one character learns motoric movements from the other.

The Restaurant Game [6] is a computer game developed at MIT Media Lab, which people can play over the Internet. The goal of this research was the observation of communication and collaboration patterns to develop a kind of social common sense. Plan networks trained on data acquired from this game could distinguish typical restaurant situations from unusual ones [6]. Moreover, the data was used to generate plans for interaction in the Restaurant Game [7]. The Restaurant Game demonstrates very well, how virtual agents can involve a much larger number of subjects than user studies with robots.

There are other approaches of using virtual reality worlds for research on human-robot interaction [2,10]. But our goal is to support adaptive plan execution for human-robot collaboration. We are mostly interested in the *robot behavior* than in the human reaction, which is in the focus of the well-established Wizard-of-Oz experiments. But still we want our simulated robot to face the dynamic behavior of a human collaborator.

In a similar line Steinfeld et al. describe the "Oz of Wizard" method [8] for HRI research. It shifts the focus from studying humans as practiced in the Wizard of Oz methodology towards the development of robot skills using models of humans.

The OpenRobots Simulator project[1] follows similar goals as our approach to enable human-robot collaboration. This simulator is based on the Blender

[1] https://launchpad.net/openrobots-simulator

software to create virtual worlds including the animation of humans. But it also allows physical interaction between robots, humans and objects to make the robot control realistic. The OpenRobots Simulator will be more versatile than the simulation we have developed for our purposes. In the simulation we introduce, we wanted a low-cost solution using an existing robot simulator.

4 The Simulator

For implementing our HRI simulation platform, we made use of an existing physical simulation of an autonomous household robot [5]. For inserting a human in this simulation, it would be desirable to model a human shape and provide it with human-like movements. But as our simulation is based on realistic physics, we would need a very accurate model and would have to provide accurate motor commands to this human to move realistically. Currently, there is no such simulation available. Therefore, we resorted to using the model of the robot that we already have and let it be controlled by a human operator.

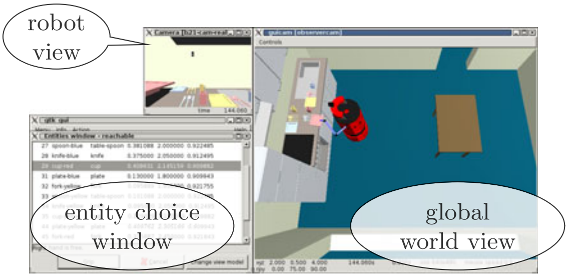
We use the Gazebo simulator, which includes the physical simulation engine ODE. The kitchen is a copy of a real kitchen containing furniture such as cupboards, a table, and a sink. The available objects include cutlery (forks, spoons and knives), dinnerware (plates and cups), and objects necessary for cooking such as pots, colanders and wooden spoons. Beside solid objects, the simulation includes water, which can emerge from the tap and disappear into the sink.

We use the same simulated hardware for the human-controlled robot as for the autonomous one. Our robot is modeled after a B21 robot, which originally comes without arms. We added two arms, which are constructed along the PUMA robot arm, used in industrial environments. To make the arms more agile and expand their operating region, we added four more joints: two joints in the shoulder and two slider joints in the upper and lower arms. The additional joints make the arms more agile and extend the robot's work space.

The user interface for manually controlling one of the robots is a stand-alone program written in C, which uses Player interfaces to control the robot and objects in Gazebo. It uses the GTK library for its graphical user interface and for accessing the keyboard and mouse commands from the user. Figure 1 shows the complete user interface. Currently the graphical interface is only necessary for choosing an object for gripping or a position to put an object down. In the future, we intend to expand it for communication like agreeing on a common goal or giving instructions to the robot.

The robot's position can be controlled by the arrow keys to move the robot forward and backward and turn it. Simultaneous rotation and movement is possible, too.

The user can manipulate objects by using preprogrammed arm routines. Gazebo is not designed for identifying objects by clicking on them on the 2D rendering of the scene. Adding this functionality would have been extremely laborious. Therefore, we present all available objects in a list from which the user can select an object to grasp. When the automatic gripping routine fails to find



27	spoon-blue	table-spoon	0.381088	2.000000	0.922485
28	knife-blue	knife	0.375000	2.050000	0.912495
29	cup-red	cup	0.408431	2.145159	0.909882
31	plate-blue	plate	0.130000	1.800000	0.909843
32	fork-yellow	fork	0.095889	[illegible]	0.921755
33	spoon-yellow	table-spoon	0.131088	2.000000	[illegible]
34	knife-yellow	knife	[illegible]	[illegible]	0.912495
35	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
44	plate-yellow	plate	[illegible]	[illegible]	0.909843
45	fork-red	fork	[illegible]	2.450000	[illegible]

hand is free.

Fig. 1. The simulation and the control GUI. The entity choice window is opened upon command by the user.

a solution, the user is informed about the error. This happens mostly when the robot is positioned too far away from the object.

For the put-down process, the user can choose among some predefined areas where to put an object. In each area, there are predefined positions for a plate, a cup, a knife, a spoon and a fork, where the robot will automatically put them. Also for putting down, the robot has to be navigated to an appropriate position where the simulation control can find waypoints for the arm.

Although these predefined routines have nothing to do with real human movements, they provide a usable interface. The alternative of letting the users control the arm joints would have resulted in a very difficult user interface, which would rather hinder than enhance the interaction with the robot.

5 User Study

When developing adaptive robot behavior in simulation, we must be sure that the (high-level) behavior of humans is comparable in simulation and reality. We also verify that results are not corrupted by different skills in controlling the simulated agent. More specifically, we conducted a pilot study [4] to answer the following questions: (1) Is our simulation control usable for subjects who were previously untrained in handling the simulation? (2) Do subjects show comparable behavior in simulation and the real world when performing everyday activities?

For the second question, we have to keep in mind the kind of research we want to conduct with this simulation. In this user study, we concentrated on the topic, which is most interesting to us: model-based planning and plan execution of joint human-robot tasks in everyday activities. As real-world tasks for this study, we used table setting and table clearing. The actions allowed by the simulation give the users the freedom to move to any spot in the kitchen at any time and to grasp and put down objects. Although the positions for putting down objects are restricted to a few predefined positions, we believe they did not influence the humans' behavior significantly assuming the subjects complied with the task they were given.

In our experiment we only had one agent, the one controlled by the human. The goal is to use this simulation for studies on human-robot collaboration (which in fact we have done for proof-of-concept trials). However, for a human-robot scenario, we have no chance to compare the simulated behavior with that in the real world for the reasons given in Section 2. This is why in this user study (simulating robots with the Gazebo simulator is a generally accepted method).

Nine subjects participated in the study, four female and five male. Eight participants work with the computer regularly, six of which are IT professionals (including students of computer science or related subjects).

The study consisted of two parts for each subject: tasks in the real world and control of the robot in simulation. Five subjects performed the real-world tasks first, the others started with the simulation part. To make the data comparable the subjects were asked in the real world to carry only one item per hand, which corresponds to the capabilities of the simulated agent.

We defined six tasks for the subjects to perform (both in reality and in simulation), three consisted in setting the table, three in clearing it. The order was randomized. The available objects (marked with different colors) were plates, cups, knives and in some tasks also spoons and forks. In total, there were three complete sets available (red, blue and yellow), but not all objects were used in every task.

The trials in the real world were recorded with four cameras from different angles for later evaluation. In the video data we can measure durations and observe which objects were used and placed at which positions. In the simulation, we made use of the data acquisition capabilities of the Robot Learning Language [3] to record the robot's position and orientation 20 times per second, the commands given by the person, failures in the manipulation actions and the original and end positions of objects in the grasping and put down tasks. Beside the quantitative measures, the participants filled out a questionnaire.

We evaluated the ability of the subjects to control the simulation along two criteria: speed and failure rate. The speed of fulfilling the tasks can only be evaluated in comparison to the other subjects. For each scenario we calculated the ratio of the time t_n it took for subject n divided by the average time $\bar{t}$ needed for this scenario by all subjects ($t_n/\bar{t}$). The maximum deviation from the average for a single scenario is 48%. The average score for each participant was up to 16% higher (i.e. slower) than the average and 22% lower than average.

This deviation seems to be high, but we compared the values to those observed in the real world. The values for the real-world tasks are on a different time scale than those from simulation. Therefore, we compared the deviation for each task normalized by the average duration of all tasks in reality and simulation respectively. This normalized deviation ranged from 16% to 25% in reality and was between 14% and 21% in simulation. The average of this value over all six tasks was 19% both for reality and simulation. So the deviation of times from the average time taken in simulation can be explained by the variance with which people execute real-world tasks.

The variance in the failure rates among subjects is a lot higher than for the speed. We plan to give users more advice on the physical properties of the robot to help them evaluate the chances of success. But the different failure rates didn't cause a strong effect in the times people needed for completing the tasks.

As a counter check for the measured data, we asked users how they felt about the simulation using a five-level Likert scale[2]. On average, the participants "partially agreed" that they could handle the simulation well and that they could achieve the tasks as quickly as the simulation allows. However, participants were not too happy about the absolute time scale. The average opinion for "the simulated robot executes the actions fast" was rated with an average value of 3.7. Indeed, the scaling factor between the times in reality and simulation is about 20. The relatively high failure rate was not perceived too bad by the participants. The statement "The simulated robot executes the actions without errors." was answered with an average score of 2.7.

For planning joint human-robot activities, we are mostly interested in the actions and the manipulated objects. The actions for table setting as such are not too interesting and don't differ in reality and simulation (moving, grasping, putting down objects); more interesting are the manipulated objects.

We grouped the execution of a task into *carrying tasks*, which involves the gripping of two objects, carrying them and putting them down. We identified preferences of people to carry the same or similar kinds of objects at a time (similar means cups and plates form one class and cutlery another) or objects of the same color and compared them in reality and simulation (Figure 2).

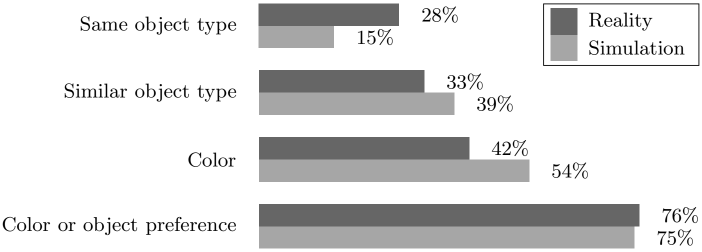


Fig. 2. Comparison of preference types over all subjects. The last line shows the percentage of actions in which any kind of preference (same or similar object or color preference) was applied.

Taken together, these preferences account for 76% of all carry tasks observed in reality and 75% of carry tasks in simulation. Interestingly, when asked about their strategies, the subjects answered that in reality they try to carry objects of the same type or put the plates and cups before they get the cutlery, but that they sacrificed these preferences for the sake of efficiency in simulation. In contrast, our observations suggest that these preferences are also applied in simulation. However, the frequency of each preference is slightly different. In simulation, the subjects carried similar object types slightly more often than in reality and the preference to carry objects of the same type was only visible about half as frequently as in real-world execution. This is compensated by a higher color preference in simulation as compared to reality.

[2] 1: fully agree, 2: partially agree, 3: don't know, 4: partially disagree,5: fully disagree.

Overall, the way people set and clear the table in the simulator is not very different from how they do it in reality. For example, our simulated robot could have learned from observing the subjects in the experiments that people like to carry objects of the same color. Although this preference is not as strong in reality, it is still a valid observation. As the models of humans have to be adapted for each individual anyway, the development of adaptive planning techniques can very well be done in our simulation testbed.

6 Conclusion

We have presented an implemented approach of using a physical simulation to integrate human behavior in the development cycle for research on human-robot collaboration. The user study we have performed has shown that on an abstract level people show similar behavior patterns when they control a simulated robot as when they perform an activity on their own. Taking into account the current state of the art in robotics and the limited capabilities of complete robot systems, there are good reasons to develop high-level methods for human-robot interaction in such a simulation.

References

1. Buchsbaum, D., Blumberg, B., Breazeal, C., Meltzoff, A.N.: A simulation-theory inspired social learning system for interactive characters. In: IEEE International Workshop on Robots and Human Interactive Communication (2005)
2. Howard, A.M., Remy, S.: Utilizing virtual environments to enable learning in human-robot interaction scenarios. The International Journal of Virtual Reality (2008)
3. Kirsch, A.: Robot learning language — integrating programming and learning for cognitive systems. Robotics and Autonomous Systems Journal 57(9) (2009)
4. Kirsch, A.: Be a robot — a study on everyday activities performed in real and virtual worlds. Tech. Rep. TUM-I1006, Technische Universität München (2010)
5. Müller, A., Beetz, M.: Designing and implementing a plan library for a simulated household robot. In: Cognitive Robotics: Papers from the AAAI Workshop (2006)
6. Orkin, J., Roy, D.: The restaurant game: Learning social behavior and language from thousands of players online. Journal of Game Development (JOGD) 3(1), 39–60 (2007)
7. Orkin, J., Roy, D.: Automatic learning and generation of social behavior from collective human gameplay. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, AAMAS (2009)
8. Steinfeld, A., Jenkins, O.C., Scassellati, B.: The oz of wizard: Simulating the human for interaction research. In: Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction (2009)
9. Ueda, R., Ogura, T., Okada, K., Inaba, M.: Design and implementation of humanoid programming system powered by deformable objects simulation. In: 10th International Conference on Intelligent Autonomous Systems (2008)
10. Xin, M., Sharlin, E.: Exploring human-robot interaction through telepresence board games. In: Pan, Z., Cheok, D.A.D., Haller, M., Lau, R., Saito, H., Liang, R. (eds.) ICAT 2006. LNCS, vol. 4282, pp. 249–261. Springer, Heidelberg (2006)

Human Head Pose Estimation Using Multi-appearance Features

Norbert Schmitz, Gregor Zolynski, and Karsten Berns

Robotics Research Lab, Department of Computer Science,
University of Kaiserslautern, Germany
{nschmitz,zolynski,berns}@cs.uni-kl.de
http://agrosy.cs.uni-kl.de/

Abstract. Non-verbal interaction signals are of great interest in the research field of natural human-robot interaction. These signals are not limited to gestures and emotional expressions since other signals - like the interpersonal distance and orientation - do also have large influence on the communication process. Therefore, this paper presents a marker-less mono-ocular object pose estimation using a model-to-image registration technique. The object model uses different feature types and visibilities which allow the modeling of various objects. Final experiments with different feature types and tracked objects show the flexibility of the system. It turned out that the introduction of feature visibility allows pose estimations when only a subset of the modeled features is visible. This visibility is an extension to similar approaches found in literature.

1 Introduction

The development of humanoid robots is directed towards natural interaction following the ideal of inter human interaction. In [4] seven major categories of action which are observable in inter-human interaction are mentioned. Two of them are the body motion (kinesics) and the influence of interpersonal distance (proxemics). The kinesics describe the role of body motions like head shaking and nodding which are stated to have a geat contribution [1]. The second category describes the influence of distance zones on the interaction process [6]. Although various analysis on non-verbal interaction signals do exit they are not often used in human-robot interaction scenarios. This fact is based on the complexity of these signals which are hard to detect in an unmodified environment and hard to interpret since their meaning depends on the situation and various other factors.

These signals and their usage provides a great yet unused potential towards natural human-robot interaction. Having a look at these signals it turns out that the head pose consisting of the position and orientation can help to detect a broad range of signals. With this information distance zones like personal or social can be categorized and gestures like head nodding can be detected. In this paper we therefore present a head pose estimation system which uses a single camera mounted in the head of the humanoid robot ROMAN. The main

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 66–73, 2010.

goal of this pose estimation is a reliable and computationally inexpensive system without any modifications of the environment.

1.1 State of the Art

There are many publications devoted to the topic of head pose estimation [7]. They can be roughly divided into feature-based approaches (finding a few salient points and computing the pose) and global approaches (estimation of the pose using the appearance of the whole face).

Some of the recent approaches (e.g. [13]) use an imaging system with depth information (time-of-flight camera) to obtain color and depth data simultaneously. The combination of the two data sources allows a relative motion estimation of points in 3D space which is then used to determine the overall pose change of a human head. In [3] the face pose and facial expressions are estimated using *Online Appearance Models*. The approach calculates the change of the head pose (and the expression) using the perceived deformation of textured areas. The initialization is done by hand, but, is believed to be possible in an automatic fashion. A work focused on pose retrieval using point-like facial features (eyes, nose, ears, . . .) is presented in [11]. A rough classification into frontal and profile pose is followed by a search for expected facial features and pose calculation using a neural network. An unconventional approach is presented in [5]: The pose of a human head is determined using an image retrieval method (the *vocabulary tree* from [8]) and a large database of images with annotated pose.

The approach proposed in the work at hand is to be seen as an extension to the work of Vatahska [11]. But, instead of the rough initial classification of human head views into “frontal” and “profile”, this work incorporates a more refined, three-dimensional model of facial feature distribution in combination with a new technique considering the appearance of features under different viewing angles. With this modeling approach including feature visibility and feature types a flexible pose estimation can be realized. In comparison to other work in literature higher rotation angles and a higher robustness concerning missing features is achieved.

2 Head Pose Estimation

For head pose estimation, the first step is to find and to track human faces in video streams. The detection of faces is done using a fusion of two simple approaches. The initial search is performed using a boosted cascade of Haar-like features [12]. Assuming that a person, willing to interact with the robot, will at least once look directly at the robots’ head, the search is conducted using detectors for frontal faces only. To reduce the number of false positives, a verification using colors is performed. Only if a region of the currently processed image fits the Harris detector and contains a large enough area of skin-like color, the region is accepted as a candidate face. This candidate is then verified by searching for the most prominent features in a human face: the eyes. Once a face is detected and confirmed, it is tracked in two dimensions using the well-known "Good Features to Track" presented by Shi and Tomasi in [10].

The tracking of faces is not explained in more detail since this work focuses mainly on the problem of pose estimation which is described in the following sections.

This work aims at developing a system capable of pose estimation of any rigid object (as position and orientation relative to the viewer) from a monocular image, as long as the object contains detectable *point-like* features. The properties of those features are – of course – their salience in relation to their surroundings and the possibility of their localization using a single three-dimensional coordinate (in contrast to line features and area features).

Detection of facial features is done using a Haar-like feature detector although the implemented system is not limited to a special feature detector, but, for now, the Haar classifier yields acceptable results and allows for evaluation of our approach. The pose estimation is calculated using a slightly modified implementation of the `Softposit` algorithm of David et al. [2]. The model-to-image registration performed was extended by a feature point type property which is used to reward the convergence of image and model points of the same type and punishes the convergence of points having different types by modifying the entries in the assignment matrix. This will be explained in detail in section 2.2.

2.1 Multi-appearance Feature Model

Many approaches for head pose estimation from a set of facial features implicitly assume that all the features needed for the task are always visible. This is the main reason why those approaches only work with rotations up to 20° or 30°. When a person turns its head further, some features simply disappear. The approach here eliminates this problem. The main assumption that leads to the three-dimensional multi-appearance feature head model is: facial features keep their appearance only within small deviations from a straight view (termed *cone of visibility*, see Fig. 1). Beyond this cone, the features can look quite different. So, in this context, the term *visibility* does not encompass the complete facial

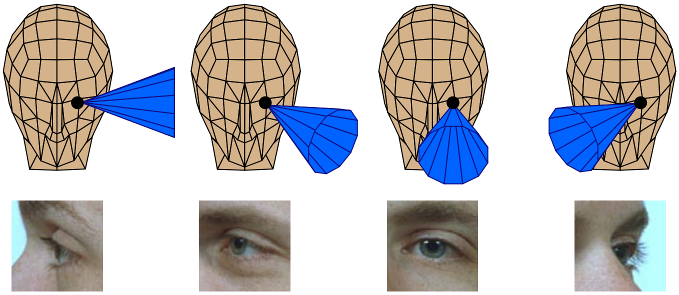


Fig. 1. The model's left eye: four *visibility cones* and the respective appearance of a human eye. Huge variations of the appearance can not be disputed.

View	Feature	Position	Direction	Vis. Cone	Search Area
Frontal	*Left Eye	$(28, -34, 20)$	$(0.25, 0.2, -1)$	0.75	$(0.5, 0.5)$
	*Right Eye	$(-28, -34, 20)$	$(-0.25, 0.2, -1)$	0.75	$(0.5, 0.5)$
	*Nose	$(0, 0, 0)$	$(0, 0, -1)$	0.75	$(0.45, 0.45)$
	*Mouth	$(0, 28, 15)$	$(0, 0, -1)$	0.75	$(0.6, 0.5)$
	Two Eyes	$(0, -34, 15)$	$(0, 0, -1)$	0.9	$(0.7, 0.7)$
Virtual	Virtual Center	$(0, 0, 0)$	$(0, 0, -1)$	0	$(1, 1)$
Left Half Profile	Left Eye	$(30, -34, 22)$	$(1, 0, -1)$	0.7	$(0.6, 0.6)$
	Right Eye	$(-26, -34, 20)$	$(1, 0, -1)$	0.7	$(0.6, 0.6)$
	Nose	$(2, -5, 2)$	$(1, 0, -1)$	0.7	$(0.5, 0.5)$
	Left Mouth Corner	$(20, 28, 20)$	$(1, 0, -1)$,	0.7	$(0.5, 0.5)$
Left Full Profile	Lips	$(0, 28, 15)$	$(1, 0, -0.4)$	1.7	$(0.5, 0.5)$
	Left Eye	$(32, -34, 24)$	$(1, 0, -0.4)$	0.7	$(0.5, 0.5)$
	Nose	$(3, -5, 4)$	$(1, 0, -0.4)$	1.7	$(0.5, 0.5)$

Table 1. Human head model with position in mm, direction as vector, visibility cone size in rad opening angle and feature search area as relative size in reference to the face size. All features marked with a * are required for initialization to start tracking.

feature (*e.g.* the nose), it is a means to describe the appearance of a feature from a particular view. So instead of having one appearance model of the nose and expecting it to work for all views, each feature actually consists of multiple features having the same location and that can be activated and deactivated depending on the current head pose.

The human head model used in the experiments later consists of 13 features listed in 1. Each feature is defined by position, direction and visibility cone. According to the current orientation of the head model the visibility of each feature is estimated by comparing the cone with the current view direction. Each feature is searched within a window centered at the projection of the feature position. The size of the window is defined relative to the overall face size. Additionally to the visible features a virtual head center feature has been introduced. This feature adds "depth" to the head model and reduces the number of orientation ambigiuities. Its position is estimated from the previous head pose and the current face window.

2.2 Extending the Standard SoftPOSIT Algorithm

Pose with Known Correspondences A point in three-dimensional space P with homogeneous coordinates $P = (X, Y, Z, 1)^T$ is projected onto a two-dimensional image point $p = (wx, wy, w)^T$. If the correspondence is known, the transformation, consisting of a rotation matrix R, a translation vector T and a pinhole camera projection, can be computed. With the scaling factor $s = f/T_z$ you get

$$\begin{bmatrix} wx \\ wy \end{bmatrix} = \begin{bmatrix} sR_{11} & sR_{12} & sR_{13} & sT_x \\ sR_{21} & sR_{22} & sR_{23} & sT_y \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, w = \left(\frac{R_{31}X + R_{32}Y + R_{33}Z}{T_z} \right) + 1 \quad (1)$$

With M pairs of model and image points the equation system is

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_M & Y_M & Z_M & 1 \end{bmatrix} \begin{bmatrix} sR_{11} & sR_{21} \\ sR_{12} & sR_{22} \\ sR_{13} & sR_{23} \\ sT_x & sT_y \end{bmatrix} = \begin{bmatrix} wx_1 & wy_1 \\ wx_2 & wy_2 \\ \vdots & \vdots \\ wx_M & wy_M \end{bmatrix} . \tag{2}$$

When $sR_1 = (sR_{11}, sR_{12}, sR_{13})^T$, $sR_2 = (sR_{21}, sR_{22}, sR_{23})^T$, sT_x and sT_y are found, the complete rotation matrix and translation vector can be computed by

$$s = \sqrt{\|sR_1\| \|sR_2\|};\ R_{1|2} = \frac{sR_{1|2}}{s};\ R_3 = R_1 \times R_2;\ T_{x|y} = \frac{sT_{x|y}}{s};\ T_z = \frac{f}{s} \tag{3}$$

Pose with Unknown Correspondences The assignment matrix m encodes the correspondence of each model point and image point. Each column and row of the matrix sums up to exactly 1. For potentially unmatchable points an additional slack row and slack column are added. A value of 1 in one of those slack spaces means the point could not be matched. Throughout the computation the matrix contains real numbers and converges only slowly to a real 0/1-configuration. The assignment "probabilities" are computed from the current distances between image points and projected model points d_{jk}.

$$E = \sum_{j=1}^{N} \sum_{k=1}^{M} m_{jk} \left(d_{jk}^2 - \alpha \right) \tag{4}$$

The additional term α is needed to discourage the algorithm from converging to the trivial solution $m_{jk} = 0$. With β as parameter for iteration annealing the computation of the assignment "probability" of features from Eq. 4 has been modified in this work to incorporate different feature types:

$$m_{jk} = \begin{cases} t_1 \cdot e^{-\beta(d_{jk}^2 - \alpha)} & \text{if types match} \\ t_2 \cdot e^{-\beta(d_{jk}^2 - \alpha)} & \text{else} \end{cases} \tag{5}$$

Obviously, the value for t_1 needs to be greater than 1, and the value for t_2 needs to be smaller than 1. Empirically, $t_1 = 2$ and $t_2 = \frac{1}{2}$ have yielded good results. Increasing the "probability" of a model feature and an image feature belonging to each other helps "pulling" the model registration process into the right direction, by fixing one or more assignments as soon as possible, thus simplifying the subsequent iterations. Too large values may cause false assignments if multiple features of the same type are used.

There is also a very good reason for not setting $m_{jk} = 0$ if the feature types do not match. As explained above, the system is not limited to one type of feature detectors, so type ambiguities should be expected. So, if a detector handles colors, there is a possibility of color mismatch due to unfavorable lighting conditions, e.g. resulting in a green colored feature being classified as blue colored. In this case, blocking the assignment of the two allegedly not matching features would hurt the overall model fitting scheme. The algorithm then iteratively calculates the correspondences m_{jk}.

3 Experiments

This test demonstrates the capabilities of the proposed approach under the assumption that facial features can be detected and tracked reliably. The features used in this test are artificial color landmarks. The test setup consists of a camera mounted in the head of the robot ROMAN and a silicone mask placed on a turntable. The mask has colored strips of tape glued to it, the color of the strips is used for feature visibility simulation. Green features are visible from the front, red and blue from the left and the right side respectively. A software model containing those 13 features, their rough position on the mask, and their orientation is used by the implemented system. The turntable has angle markings on the side which are visible on the captured images and can be used for optical readout. This method is considered sufficiently accurate for rotation angle measurement.

Initially, the turntable is positioned at 0° and the mask is placed on top of it. The camera is set to continuous capturing at 7.5 frames per second (fps). The turntable is rotated left and right, with ±90° being the limits which are reached in frames #502 and #591 respectively. The pose error is always below 10°, and over 50% of the time the error is less than 3°, which is 1.8%, relative to the complete range of 180° as plotted in Fig. 2.

A second test – using database images from the VidTIMIT database [9] – was performed to examine the capabilities of the proposed approach. The VidTIMIT database contains videos and sound samples of 43 persons. The videos are especially interesting for this work, because the persons turn their head into extreme orientations: first to the right, then to the left (both about 90°), then upwards (about 30° to 50°), and finally downwards (30° to 60°). Between each movement phase there is a short pause showing the person in frontal view.

For evaluation the head model described in section 2.1 with the frontal and left half profile/full profile facial features is used. That means, when a person turns her or his head to the right, the system should be able to follow into profile view. Turning the head to the left has only the frontal features at disposal, so the pose estimation should fail much earlier. This way, the test provides direct

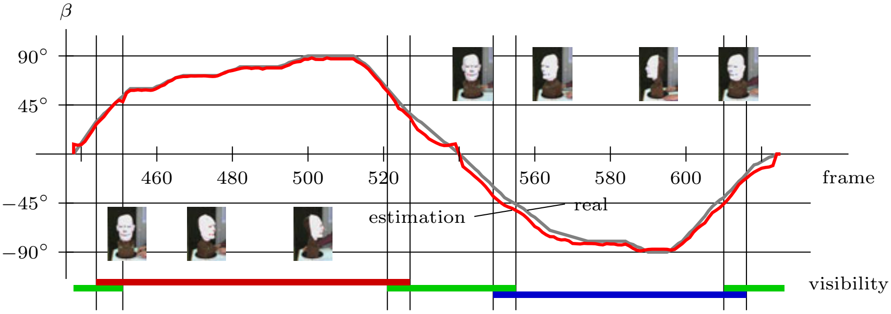


Fig. 2. Performance of the implemented system with artificial landmarks. When the features are tracked properly, the system follows the object very closely.

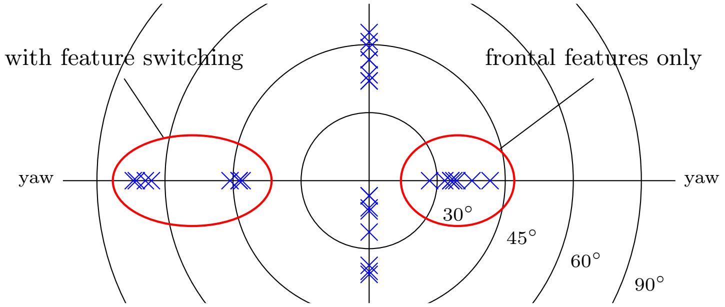


Fig. 3. Plot of the maximum angles tracked in videos from the VidTIMIT database. Each cross marks the maximal correctly detected orientation.

results regarding the question whether the feature switching approach allows pose estimation with much more extreme angles.

After each failure of pose estimation, the system is reinitialized with one of the next frontal images appearing in the image sequence. The evaluation of the systems performance is done solely in visual fashion, because no annotations of the pose are given with the database. The quality of each pose estimation step is evaluated visually and the rotation angles computed by the program are reported for the last frame before pose estimation fails. The videos used in this test are all taken from the `.../video/head/` folder (the first video sequence). The data sets 6, 11, 16, 21, 26, 31, 36, and 41 were chosen for testing.

Fig. 3 summarizes the results from this test. The markers show the maximal correctly tracked head rotation. Rotation to the left used only the frontal features while rotations to the right are assisted by multi-appearance features. Although the tracker is not able to keep the pose estimation up to full profile view, the feature visibility switching method shows a clear advantage over the static, frontal only feature model.

An interesting finding is the ability of the tracker to follow upward rotations without any additional help. Facial features keep their appearance at great angles when the head is turned upwards. Human eyes are pointed slightly downwards, so they stay visible. When a head is lit from above, a dark shadow can be seen under the nose. This shadow is crucial for the appearance-based Haar cascade detector used to detect the facial features. It seems as if the nostrils take over the role of this shadow when the head is turned upwards. On the other hand, turning the head down involves also looking down, which in turn changes the appearance of the eyes completely. Without those two very salient features, the pose tracker implemented in this work can not keep up.

4 Conclusion

In this paper a marker-less head pose estimation system has been presented. Based on the flexible object model the concept of feature types and feature visibility is introduced. This information is used to adapt the model fitting step using a modified Softposit algorithm. Final experiments with an artificial head and a standardized image database show the variability and possibilities of the modeling process. Also real environment tests with ROMAN and a human interaction partner have shown adequate results with high precision as well as large rotations which is not possible with similar approaches in literature. The future development of the model should include a broad set of simple features to reduce the calculation complexity and a motion prediction step to estimate the objects pose change before the fitting process is started.

References

1. Birdwhistell, R.L.: Kinesics and Context: Essays in Body Motion Communication. University of Pennsylvania Press, Philadelphia (1970)
2. David, P., De Menthon, D., Duraiswami, R., Samet, H.: Softposit: Simultaneous pose and correspondence determination. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 698–714. Springer, Heidelberg (2002)
3. Dornaika, F., Orozco, J., Gonzàles, J.: Combined head, lips, eyebrows, and eyelids tracking using adaptive apperarence models. In: Perales, F.J., Fisher, R.B. (eds.) AMDO 2006. LNCS, vol. 4069, pp. 110–119. Springer, Heidelberg (2006)
4. Duncan, S., Fiske, D.W.: Face-To-Face Interaction. Lawrence Erlbaum Associates, Hillsdale (1977)
5. Grujić, N., Ilić, S., Lepetit, V., Fua, P.: 3d facial pose estimation by image retrieval. In: 8th IEEE Int'l Conference on Automatic Face and Gesture Recognition (2008)
6. Hall, E.T.: The hidden dimension. Anchor (1966)
7. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(4), 607–626 (2009)
8. Nistér, D., Stévenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2161–2168 (2006)
9. Sanderson, C., Palival, K.K.: Identity verification using speech and face information. In: Digital Signal Processing, pp. 449–480 (2004)
10. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (June 1994)
11. Vatahska, T., Bennewitz, M., Behnke, S.: Feature-based head pose estimation from images. In: Proceedings of the IEEE-RAS International Conference on Humanoid Robots, Humanoids (2007)
12. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, pp. 511–518 (2001)
13. Zhu, Y., Fujimura, K.: 3d head pose estimation with optical flow and depth constraints. In: Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling, pp. 211–216 (2003)

Online Full Body Human Motion Tracking Based on Dense Volumetric 3D Reconstructions from Multi Camera Setups*

Tobias Feldmann[1], Ioannis Mihailidis[2], Sebastian Schulz[1],
Dietrich Paulus[2], and Annika Wörner[1]

[1] Group on Human Motion Analysis, Institute for Anthropomatics
Department of Informatics, Karlsruhe Institute of Technology
feldmann@kit.edu, s.schulz@kit.edu, woerner@kit.edu
[2] Working Group Active Vision (AGAS), Institute for Computational Visualistics
Computer Science Faculty, University Koblenz-Landau
janni@uni-koblenz.de, paulus@uni-koblenz.de

Abstract. We present an approach for video based human motion capture using a static multi camera setup. The image data of calibrated video cameras is used to generate dense volumetric reconstructions of a person within the capture volume. The 3d reconstructions are then used to fit a 3d cone model into the data utilizing the Iterative Closest Point (ICP) algorithm. We can show that it is beneficial to use multi camera data instead of a single time of flight camera to gain more robust results in the overall tracking approach.

1 Introduction

The estimation of human poses and the pose tracking over time is a fundamental step in the process chain of human motion analysis, understanding and in human computer interaction (HCI). In conjunction with artificial intelligence (AI) pose estimation and tracking are the sine quibus non of motion recognition which in turn is a fundamental step in scene and motion analysis with the goal of automatic motion generation on humanoid robots.

A lot of research has taken place, to automatically exploit information for pose estimation and tracking in video data. The video based techniques can here be separated into two major approaches. First: Approaches based on markers placed on the observed subjects. Second: Marker-free approaches based on image features, on geometrical or on statistical information. The latter approaches usually replace the knowledge about markers with knowledge about a human body model. In HCI tasks it is usually desirable to avoid markers to allow a less intrusive observation of persons.

* This work was supported by a grant from the *Ministry of Science, Research and the Arts of Baden-Württemberg.*

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 74–81, 2010.

Marker-less image based human pose tracking is usually started by the extraction of features like points, edges [1], contours, colors [2], segments [1], boundaries, 2.5d [3] or 3d points [2,4], etc. from one or more given input images. In an additional step a correspondence problems has to be solved to assign the found features to an a priori given kinematic human body model. To obtain the pose of the person within the images, the body pose problem has to be solved, which can be done by using e.g. probabilistic approaches like particle filters [1], geometric approaches like Iterative Closest Point (ICP) in the Voodoo framework [3], iterations of linear approximations [5] or learning approaches, e.g. based on silhouettes [6]. An extensive survey of current approaches can be found in [7].

A key point is the definition of the distance function to assign correspondences. Often, color segments of a priori known color distributions are (skin color, foreground colors, etc.) to determine regions of interest. The color segments can be used to directly calculate 3d information via triangulation, e.g. by the use of stereo information [2] or in form of fore-/background estimation for a subsequent dense 3d reconstruction by utilizing a multi camera setup and the shape from silhouette approach [4].

2 Contribution

We present a purely image based marker-less online capable tracking system based on a human cone model with 25 degrees of freedom (DOF). We focus on the shape from silhouette approach for a volumetric 3d reconstruction and combine it with ICP based tracking. We demonstrate the applicability of the overall approach for online human body tracking in static multi camera scenarios.

The approach works as presented in Fig. 1. Given the input images of n calibrated cameras, the foreground has to be segmented by background subtraction. We use a more complex color model to enhance the segmentation results and, hence, shift the calculations to the graphics processing unit (GPU) to gain online capable performance. The segmentation results are used as input for a dense binary voxel reconstruction. The voxel carving can be parallelized very easily by dividing the volume into subvolumes. Thus, we use multithreading to increase the speed of the reconstruction. In case of m cores we use $m-1$ threads for the dense reconstruction. The results of the reconstruction are then used as input for an ICP based tracking framework, which runs in a separate process on the last unused core of the CPU.

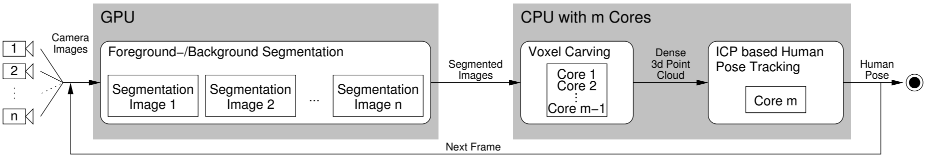


Fig. 1. The integrated pipeline of voxel based human pose estimation and tracking

3 Fore-/Background Segmentation

One key aspect in volumetric 3d reconstruction is the determination of the foreground in the camera images to utilize this information for dense volumetric reconstruction in a later step. A simple approach to distinguish between fore- and background is simple differencing between the camera images and a priori recorded images of the background to extract the silhouettes of the image foreground. The recording scenario is often influenced by changing light conditions e.g. if the recording room is equipped with windows. In realistic scenarios it is hence reasonable to use a color space which splits luminance and chroma. An often used color space which implicitly splits luminance and chroma is HSV. However, color spaces like HSV suffer from singularities near for example $V = 0$, which leads to large errors introduced by only small amounts of image noise in dark areas. Unfortunately, in realistic videos of indoor scenes a lot of dark areas exist below objects which occlude the illumination spread of lamps and windows. Hence, due to the robustness in realistic scenarios the computational color model of [8] is used. The idea is to separate chroma and luminance directly in RGB without previous color space transformations.

Given two colors $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ in RGB color space, two distances have to be calculated. First: The distance of brightness α. Second: The distance of chroma d (c.f. Fig. 2). The distance of brightness is defined as a scalar value α that describes the factor of brightness, the color $\boldsymbol{c_1}$ have to be scaled with to be as bright as the color $\boldsymbol{c_2}$. Geometrically, α defines the root point $\boldsymbol{r}$ of the perpendicular of the vector $\boldsymbol{c_1}$ through the point $\boldsymbol{c_2}$. The value of α can be calculated with the dot product as shown in eq. 1. If $\alpha < 1$, the color $\boldsymbol{c_2}$ is darker than the color $\boldsymbol{c_1}$, if $\alpha > 1$, the color $\boldsymbol{c_2}$ is brighter than the color $\boldsymbol{c_1}$.

$$\alpha = \frac{\|\boldsymbol{c_1}\|}{\|\boldsymbol{r}\|} \quad \text{where} \quad \boldsymbol{r} = \boldsymbol{c_1} + \langle(\boldsymbol{c_2} - \boldsymbol{c_1}), \boldsymbol{n}\rangle\boldsymbol{n} \quad \text{with} \quad \boldsymbol{n} = \frac{\boldsymbol{c_1}}{\|\boldsymbol{c_1}\|} \tag{1}$$

Using the normed direction vector $\boldsymbol{n}$ of $\boldsymbol{c_1}$ from eq. 1 the color distance d can be calculated as described in eq. 2.

$$d = \|(\boldsymbol{c_2} - \boldsymbol{c_1}) \times \boldsymbol{n}\| \tag{2}$$

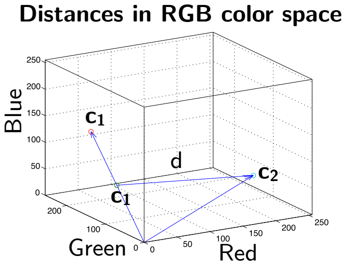


Fig. 2. Distance measure in RGB color space. The distance of two colors $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ is decomposed into the luminance distance α and the chroma distance d.

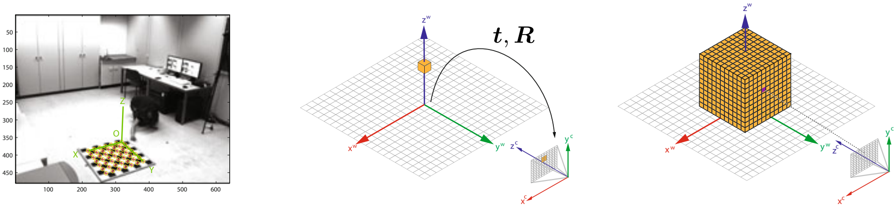
Due to online constraints and in contrast to [8] we are using static thresholds τ_α and τ_d for the discrimination of fore- and background and use the algorithm in parallel on different camera views all at the same time in a multi camera setup. The values are estimated empirically during runtime by using sliders. It is most important to adjust the τ_α correctly to be able to cope with light changes and shadows. Due to the computational complexity of the calculations and the aim of online applications we exploited the computational power of the GPU of the used computer system. By shifting the calculations to the GPU, we were on the one hand able to speed up the segmentation process. On the other hand, in this way we were able to keep the CPU free from segmentation calculations and, thus, use the CPU primarily for additional tasks like the dense volumetric 3d reconstruction and the model based pose tracking of the observed person.

4 Dense Voxel Based 3d Reconstruction

Based on the calibration data of the used calibrated multi camera setup and the silhouettes from the previous section 3, a dense volumetric 3d reconstruction is performed using a voxel carving approach derived from [4].

The camera setup is static and the intrinsic parameters (calibration matrix $\boldsymbol{K}$ and distortion coefficients kc_i with $i \in 1 \ldots 5$) and extrinsic parameters (rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$) are determined using Bouguets Matlab Calibration Toolkit [9], (c.f. Fig. 3(a)) based on a checker board calibration target which defines the world coordinate system (WCS) and stay the same over the whole sequence. For each camera, the position in the WCS is determined in 6d based on rotations $\boldsymbol{R}$ and translations $\boldsymbol{t}$ (c.f. Fig. 3(b)). Therefore, the transformation of a 3d world point $\boldsymbol{p}^{\mathrm{w}}$ into the camera coordinate system (CCS) can be calculated by $\boldsymbol{p}^{\mathrm{c}} = \boldsymbol{R}\boldsymbol{p}^{\mathrm{w}} + \boldsymbol{t}$. The resulting 3d point $\boldsymbol{p}^{\mathrm{c}}$ has to be projected into 2d, distorted according to the distortion coefficients resulting in $\boldsymbol{p}^{\boldsymbol{d}}$ and finally linearly transformed by $\boldsymbol{K}$ with $\boldsymbol{p}^{p} = \boldsymbol{K}\boldsymbol{p}^{d}$ (c.f. [9]).

This projection of a single 3d point can be extended to a projection of a 3d volume grid as depicted in Fig. 3(c). The space of interest is divided into a uniform 3d grid of voxels. Due to the online constraints of our approach, we consider only the center of each voxel and project it into each camera image.



(a) Camera calibration (b) WCS to CCS (c) Voxel projection

Fig. 3. (a) Camera position based on WCS defined by calibration target. (b) Transformation of the orange voxel from WCS to CCS. (c) Center of each voxel of defined voxel space is projected into specific pixel coordinates of each camera.

The projection information is static as the camera setup is static and, thus, can be pre-calculated for each voxel and for each camera before the reconstruction process starts, which helps to speed up the projection process.

The idea of voxel carving is, to define two states of a voxel: The voxel is turned on or off. Initally, all voxels are off. By utilizing the silhouettes from section 3 an iteration over all voxels is started whereas for the voxel's projection to each camera a check is performed, whether the projection falls within the silhouette of the observed person in the camera image. If this is the case for all cameras, the voxel is turned on, otherwise the voxel remains off. After the iteration over all voxels, a discretized 3d reconstruction of the observed foreground has been created. This reconstruction is then repeated over a whole sequence of frames which contain synchronized video images of each camera.

5 Online Full Body Human Motion Tracking

Knoop et al. [3] created the public available[1] cone based *Voodoo* framework for human motion tracking in the *Cogniron* project[2]. The framework is able to track human poses online utilizing ICP and based on a human body model. Knoop et al. used the framework mainly with time of flight cameras and sensor fusion approaches by integrating 2.5d and 2d image data. We present our integration of data derived from a n-view camera setup into the Voodoo tracking framework to enable online tracking on pure video data. In contrast to 2.5d data taken from a single time of flight camera, our approach generates full 3d point clouds from different views and, hence, has the potential to achieve improved tracking results by solving ambiguities which result from occlusions.

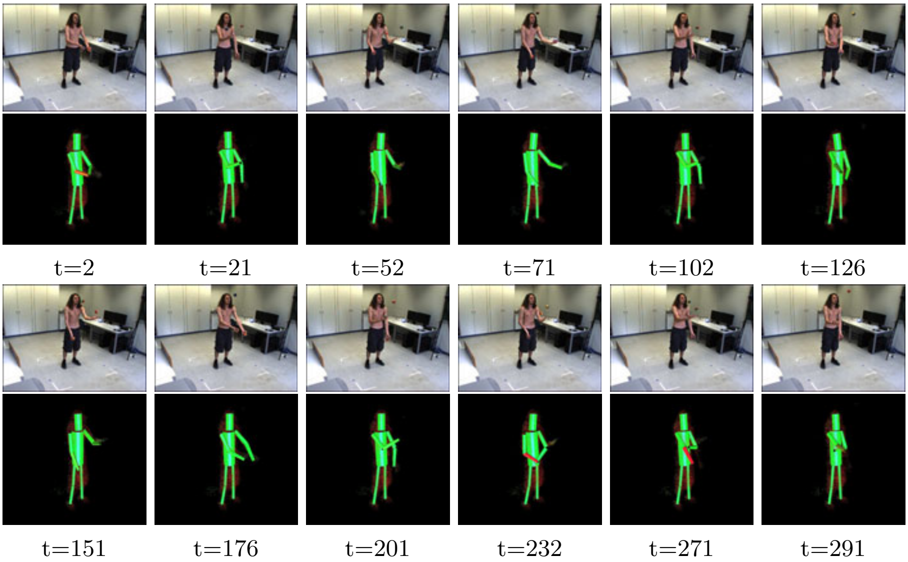
One benefit of the Voodoo framework is it's flexible sensor integration, i.e. the possibility to add an arbitrary number of sensors as data sources. We integrated two additional sensors which use our own point cloud format to enable the data exchange between the 3d reconstruction of section 4 and Voodoo. The first sensor imports recorded sequences and enables Voodoo to replay these sequences for later analysis. The second sensor imports a continuous stream of point clouds from a connected reconstruction instance directly into Voodoo and in this way enables online tracking. The source code of the point cloud interface is available at: http://hu-man.ira.uka.de/public/sources/libPtc.tar.bz2.

6 Evaluation

For the evaluation we used a standard PC (Intel(R) Core(TM)2 Quad CPU Q6600, 2.40GHz and nVidia GeForce 8600 GTS) and a set of 3 to 8 Prosilica GE680C cameras with VGA resolution in a circular setup except as noted otherwise. In case of online human motion tracking, we connected 3 cameras to the PC via Gigabit Ethernet and used the image data directly for segmentation,

[1] http://voodootracking.sourceforge.net/

[2] http://www.cogniron.org



t=2 t=21 t=52 t=71 t=102 t=126

t=151 t=176 t=201 t=232 t=271 t=291

Fig. 4. Images, reconstructions and tracking results of the juggle sequence

3d reconstruction and motion tracking. In case of offline analysis, we recorded sequences with 8 cameras, created 3d point clouds in a second step and used the point cloud reader library to import the data into Voodoo for offline analysis. In the following we will present the results of the segmentation, the benefit of the GPU based calculations and the online and offline motion tracking results.

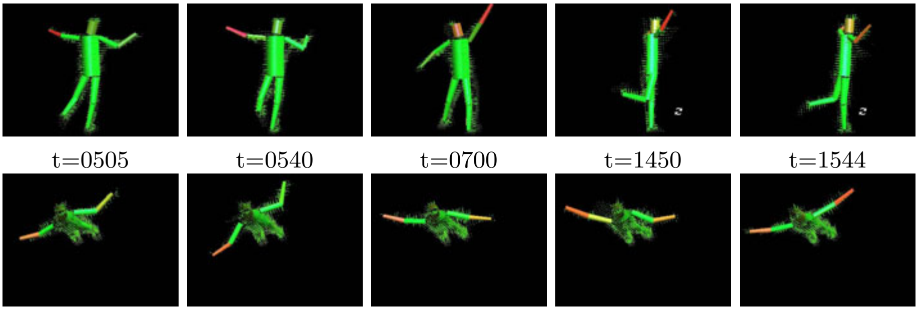
6.1 CPU versus GPU Based Segmentation

The performance increase of the GPU enabled version of the fore-/background segmentation has been measured on an Intel(R) Xeon(R) CPU X5450, 3.0GHz with a nVidia GeForce GTX 295.

The average processing time of an image of 640 × 480 pixels on the CPU in a non-optimized version using the linear algebra of the OpenCV library was 116.9ms, on the GPU it was 0.364ms. The time for images of the dimensions 1280 × 720 was 352.3ms on the CPU and 0.969ms on the GPU. The increase of speed is, hence around a factor of 320 to 360 depending on the image dimensions.

6.2 Full Body Motion Tracking: 8 Cameras (offline)

The first exemplary sequence contains a person juggling with balls in the laboratory (c.f. Fig. 4). The sequence has been recorded with 8 cameras with 50 frames per second. The voxel space dimensions had been 100 × 100 × 100 with a resolution of 20mm per voxel. Despite the fast movements of the arms Voodoo is able to track the motions successful over 200 frames. In frame 232 an interchange of the left arm happens. The positions of the head, torso and legs has been tracked successfully over the whole sequence. The tracking of fast movements is, hence,



t=0505 t=0540 t=0700 t=1450 t=1544

t=1943 t=1976 t=2000 t=2028 t=2164

Fig. 5. Reconstructions and tracking results of the online tracking from different views. The average tracking speed is around 13 frames per second on a Intel Quadcore 2,4GHz and a nVidia 8600GTS.

no problem with our approach but the interconnection of extremities with other body parts can result in the loss of the correct pose of the affected body parts.

6.3 Full Body Motion Tracking: 3 Cameras (Online)

The second exemplary sequence contains the results of an online tracking with the presented approach. The capture volume has been divided into 60^3 voxels with an edge length of 40mm per voxel. Using this configuration and three cameras with 25fps, the online tracking achieves an average speed of around 13 frames per second. The tracking results have been simultaneously captured via screen capture and a selection of the results is presented in Fig. 5. The first three frames of Fig. 5 show the tracking from behind. The point of view is then rotated to observe the person from the right in the next to frames 1450 and 1544. These two frames also demonstrate the advantages of a n-view capture method over a time of flight camera from the front as the bended knees (right knee in frame 1450, left knee in frame 1544) could be recognized and tracked correctly. The following frames 1943-2164 show the scene from above whereas it should be noted, that this is a completely artificial camera position as we did not place a real camera at the ceiling to observe the volume from above. The images show, that the arms could be tracked in all three dimension due to the 3d point cloud. Even though some tracking problems appear during rotation, i.e. the torso does not adapt to the movements of the arms in the shoulder areas. However, this can not be fixed by the data modality but has to be enhanced in the tracking framework. The full video sequence can be found at: http://hu-man.ira.uka.de/public/videos/Tracking.mp4.

7 Conclusion

We presented a marker-less system for online full body human motion tracking based on dense volumetric 3d reconstruction by utilizing the information of an fully calibrated static multi camera system and fore-/background segmentation with a computational color model in RGB and a body model with 25 DOF. We found the tracking to benefit significantly from the integration of multi view information as presented by two exemplary sequences. Hence, if the scenario allows the integration of multiple views, this information source should be exploited. Although, we noticed difficulties in the tracking process during rotations which were based on not recognized and not adapted rotations of the torso element of the tracking framework. In further work, we will focus on solutions to cope with these kind of problems. Additionally, we will perform a quantitative error analysis based on offline recordings, which is not the focus of the current paper.

Overall, the presented results constitute a first fundamental step in a substantial 3d online motion tracking for succeeding motion analysis and understanding, which form the foundation of subsequent AI methods for online scene analysis regarding humans and motion generation for e.g. humanoid robots.

Acknowledgements

We would like to thank the *Humanoids and Intelligence Systems Laboratories* (HIS) headed by Prof. Dr.-Ing. R. Dillmann, and especially Martin Lösch, for their support regarding the Voodoo tracking framework.

References

1. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: CVPR 2000, vol. 2, pp. 2126–2133 (2000)
2. Azad, P., Ude, A., Asfour, T., Dillmann, R.: Stereo-based markerless human motion capture for humanoid robot systems. In: ICRA 2007, pp. 3951–3956. IEEE, Los Alamitos (2007)
3. Knoop, S., Vacek, S., Dillmann, R.: Sensor fusion for 3d human body tracking with an articulated 3d body model. In: ICRA 2006, pp. 1686–1691. IEEE, Los Alamitos (2006)
4. Cheung, G.K., Kanade, T., Bouguet, J.Y., Holler, M.: A real time system for robust 3d voxel reconstruction of human motions. In: CVPR 2000, vol. 2, pp. 714–720 (2000)
5. Rosenhahn, B., Kersting, U.G., Smith, A.W., Gurney, J., Brox, T., Klette, R.: A system for marker-less human motion estimation. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 230–237. Springer, Heidelberg (2005)
6. Hofmann, M., Gavrila, D.M.: Single-frame 3d human pose recovery from multiple views. In: Denzler, J., Notni, G., Süße, H. (eds.) DAGM 2010. LNCS, vol. 5748, pp. 71–80. Springer, Heidelberg (2009)
7. Moeslund, T.B., Granum, E.: A survey of computer vision-based human motion capture. Computer Vision and Image Understanding 81, 231–268 (2001)
8. Horprasert, T., Harwood, D., Davis, L.S.: A statistical approach for real-time robust background subtraction and shadow detection. In: ICCV 1999 Frame-Rate Workshop, Kerkyra, Greece (September 1999)
9. Bouguet, J.Y.: Camera calibration toolbox for matlab (2010), `http://www.vision.caltech.edu/bouguetj/calib_doc/`

On-Line Handwriting Recognition with Parallelized Machine Learning Algorithms

Sebastian Bothe, Thomas Gärtner, and Stefan Wrobel

[1] Department of Computer Science III, University of Bonn, Germany
[2] Fraunhofer Institut für Intelligente Analyse- und Informationssysteme IAIS,
Schloss Birlinghoven, 53754 Sankt Augustin
script@sbothe.de, thomas.gaertner@iais.fraunhofer.de,
stefan.wrobel@iais.fraunhofer.de

Abstract. The availability of mobile devices without a keypad like Apple's iPad and iPhone grows continuously and the demand for sophisticated input methods with them. In this paper we present classifiers for on-line handwriting recognition based on SVM and kNN algorithms and provide a comparison of the different classifiers using the freely available handwriting corpus UjiPenchars2. We further investigate how their performance can be improved by parallelization and how these improvements can be utilized on a mobile device.

1 Introduction

We are shifting to the digital age, nevertheless handwriting remains attractive as a fast and individual way of storing and communicating information. In the wake of the growing popularity of devices like Apple's iPhone or iPad, a shift in the input paradigm for mobiles devices arises. These devices feature only limited or no keyboards due to size limitations, but offer a touch sensitive surface instead. This surface can be used as a new interface for an old skill, a skill that is common and well trained among the general population, the widely known skill of handwriting.

The task of handwriting recognition is addressed in the past using well known machine learning algorithms i.e. neural networks, hidden markov models and others. A detailed presentation of the other approaches is not possible here and we point the reader to the survey [8] by Plamondon and Srihari for more detailed information.

In contrast to the exisiting approaches we focus on the possible benefits of parallel processing for achieving high recognition speeds. We evaluate the recognition accuracy and speed of the alignment kernel proposed in [4], which is not used in the handwriting recognition task before. Addionally, we determine the same metrics for the GDTW kernel proposed by Bahlmann and a k-nearest neighbours approach. All presented results are based on publicly available data. We further propose a client server design that allows the usage of the implemented parallel classifiers on mobile devices.

2 Pre-considerations

Transforming handwriting to a corresponding symbolic representation is commonly known as the task of handwriting recognition. In on-line handwriting recognition the

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 82–90, 2010.

handwriting is represented as a time sequence of vectors. In our case, these feature vectors hold the horizontal and vertical coordinates of the pen tip relative to the writing surface for every sample point in time. Naturally the meaning of a written character is independent from its exact placement and size on the writing surface. We therefore normalize the representation by moving the origin of ordinates to the center of mass (μ_x, μ_y) and by scaling with the vertical standard deviation σ_y, while preserving the aspect ratio.

$$\tilde{x_i} = \frac{x_i - \mu_x}{\sigma_y}, \quad \tilde{y_i} = \frac{y_i - \mu_y}{\sigma_y} \tag{1}$$

We assume that every character can be recognized without analyzing its context. Given training data of this kind, we place the recognition task in a supervised learning framework. The classification techniques used in our approach depend on the ability to determine a distance value between the various instances.

2.1 Distances

Usually the Euclidean vector distance is used to define the proximity of instances. This metric cannot be directly applied to on-line handwriting data due to the variable length of the input sequences. One solution to this problem is to map all sequences to interpolated sequences of fixed length. This can be done by linear interpolation which for a function $f(t)$ with values $f(t_0), f(t_1)$ at times t_0 and t_1 is calculated for a time $\tilde{t}$ with $t_0 < \tilde{t} < t_1$ by:

$$f(\tilde{t}) = f(t_0) + \frac{f(t_1) - f(t_0)}{t_1 - t_0}(\tilde{t} - t_0) \tag{2}$$

This approach generates time equi-distant points and the accumulated point wise Euclidean distance between two compared sequences leads to a (dis)similarity measure. Point wise comparison has the disadvantage that small differences in the shape of the characters lead to a large value for the accumulated distance, despite the fact that the shape remains similar to the eye of a human reader. In the field of speech recognition a technique named dynamic time warping (DTW) was introduced by Sakoe and Chiba in [10], which aligns the sequences before calculating the differences. The alignment ϕ with length N for two sequences R, T is a mapping of alignment positions to pairs of corresponding sequence indices.

$$\phi = (\phi(1), \ldots, \phi(N)) \tag{3}$$

with $|R| = n$, $|T| = m$ and $\phi(i) = (\phi_T(i), \phi_R(i)) \in \{1, \ldots, n\} \times \{1, \ldots, m\}$, $\phi(1) = (1, 1)$, $\phi(N) = (m, n)$. The dynamic time warping distance $DTW(T, R)$ is the distance along the alignment path ϕ that minimizes the accumulated Euclidean distance D.

$$DTW(T, R) = \min_{\phi}\{D_\phi(T, R)\} \tag{4}$$

2.2 k-Nearest Neighbors

As a reference we use the popular k- nearest neighbor (kNN) classification rule. This rule assigns the new instance x_q a label out of a set of possible labels V through a majority vote among the k closest neighbors N. These get determined out of a set of reference patterns called the knowledge base.

$$label = \arg\max_{v \in V} |\{(x_i, y_i) \in N | y_i = v\}| \tag{5}$$

2.3 Support Vector Machine

Support Vector Machines (SVM) have achieved high recognition accuracies in many applications to mention one example the optical character recognition task (c.f. [11]). SVM training for a set of training instances x_i with associated binary labels y_i is done by solving the following optimization problem.

$$\underset{\alpha \in \mathbb{R}^m}{\text{maximize}} \quad L(w, b, \alpha) = \sum_{i=1} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{6}$$

$$\text{subject to} \quad \alpha_i \geq 0, i = 1, \ldots, m \tag{7}$$

$$\text{and} \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{8}$$

Where α_i are Lagrange multipliers, $L(w, b, \alpha)$ is the objective function and $k(\cdot, \cdot)$ is a kernel function. The details can be found in standard literature on this topic, for example in [11]. The SVM classification is achieved by calculating the sign of the following sum.

$$\sum_i \alpha_i y_i k(x, x_i) + b \tag{9}$$

Again there needs to be some preprocessing before we can apply this method to the variable length handwriting sequences. Interpolation as mentioned in Section 2.1 combined with a standard RBF Kernel is one option, but a kernel function that is capable to deal with the handwriting data directly is preferable.

2.4 Kernels

An approach to achieve this is the integration of DTW in a kernel function which has been proposed by Bahlmann et al. in [2].

$$k_{\text{GDTW}}(T, R) = exp(-\gamma DTW(T, R)) \tag{10}$$

Where T, R denote two handwriting sequences and $DTW(\cdot, \cdot)$ the DTW distance (c.f. Eq. 4). A valid kernel function needs to be positive-semi definite and this means the Gram matrix needs to be positive-semi definite (see [11]). The DTW distance function is not a metric as the triangular inequality is violated. An example for this is given in [1] and in this example the Gram matrix is indefinite which proves that their function is strictly seen not a kernel.

Cuturi et al. investigate other kernel functions that are provable positive definite (see [4]). Instead of finding the alignment that minimizes the distance, they consider all possible alignments $\mathcal{A}(x,y)$ and average the distance across them. With this idea they define their alignment kernel as:

$$K_{\text{Align}}(x,y) = \sum_{\pi \in \mathcal{A}(x,y)} \exp\left(\sum_{i=1}^{\pi} \varphi(x_{\pi_1(i)}, y_{\pi_2(i)})\right) \tag{11}$$

$$= \sum_{\pi \in \mathcal{A}(x,y)} \prod_{i=1}^{\pi} k(x_{\pi_1(i)}, y_{\pi_2(i)}) \tag{12}$$

with $k = e^{\varphi}$. In the experiments with speech recognition data they found out that their kernel is diagonal dominant and to resolve this issue they apply the log function on the kernel. This causes the kernel to loose the positive definiteness, but nevertheless they achieved good recognition accuracies with this approach in their speech recognition experiments.

3 Practical Considerations

In on-line handwriting recognition the recognition process is concurrent to the writing process. For this reason we need to achieve high recognition speeds. Modern computers with multi-core CPUs and graphic processing units (GPU) offer a huge amount of parallel computing power. To make use of this power parallel algorithms are required.

3.1 Parallel k-Nearest Neighbors

The kNN algorithm calculates the distances between the test pattern and all reference patterns stored in its knowledge base which happens one after the other within the sequential algorithm. Any of these distance calculations is independent of all other distance calculations, so they can be executed in any order without changing the calculated results. We use this insight for invoking the distance calculations in parallel, which can be achieved with the parallel loop constructs in OpenMP and place the results in an array. Further, the kNN method requires a majority vote among the k-nearest patterns in the knowledge base which we determine by sorting the array of distances and counting the number of occurrences for every label in the k closest patterns. Sorting, counting and determining the label is done by sequential code in our current implementation. We parallelized the DTW calculation, because it has larger asymptotic runtime than the sorting part and sorting is done on the smaller result array in contrast to the operation on all sequence elements in the DTW calculation.

3.2 GPU kNN

The architecture of GPUs requires a large amount of independent and parallel active computations (see [7]). These are needed to hide the memory latency, because the GPU architecture omits the cache hierarchy used on CPUs. This fact makes the above kNN parallelization well suited choice for the implementation on GPUs. In addition the memory layout and memory access patterns must be controlled and optimized to achieve high performance. We place the reference patterns in global memory and the single elements of the time sequences such that threads can read multiple elements at once. This is called coalescing and allows to make better use of the available memory bandwidth. We provide an implementation using a single GPU and one to distribute the work to two GPU wiht help of OpenMP.

3.3 Parallel SVM

Our implementation for the GDTW Kernel and the alignment kernel is based on the open source libsvm by Chang and Lin. Further we follow their hint[1] for the parallelization of the SVM based classifiers. Training a SVM is achieved by finding a solution of the optimization problem in Eq. 6. The solution is found with help of the sequential minimal optimization (SMO) algorithm (see [9]) which in each step updates two α weights and after this recalculates the value of the objective function (Eq 6). For this recalculation the parts of the sum can be computed in parallel by OpenMP loops, because each of the parts is independent of each other. This parallel calculation includes the evaluation of the kernel values $k(x_i, x_j)$ in the second term of Eq. 6. In SVM classification (Eq. 9) the calculation of the sum can be done in parallel code by the same idea. The other parts of the algorithm remain sequential code in the current state of our implementation, as the kernel evaluation time is considered to be the most compute intensive part of the calculation.

3.4 Classifier as a Web service

The invocation of this classifiers by a mobile client is achieved by decoupling the classifier and the user front end. On the mobile device only a XML representation of the electronic ink is generated and send as classification request to a web server. This web server performs the preprocessing and executes one of the classifiers, which one exactly is determined by the request URI. After successful classification the server responds to the request with the recognized symbol.

4 Experiments

The Experiments by Bahlmann were carried out using the UNIPEN dataset [5] which is commercially available. In our experiments we use the Ujipenchars2 database ([6]) which is available for free. To this dataset 60 writers contributed a total of 11,640 handwriting samples. As symbols in this dataset the latin characters as well as some Spanish characters and some non ASCII symbols are used. For our experiments we consider only the symbols included in the ASCII code set and discard the others. The UJIPenchars2 database is divided into a set of 40 training writers and a disjoint set of 20 test writers. This evaluation type is called writer independent as the classifiers are evaluated on a set of writers not available to the classifier while training. This situation is different

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/faq.html#f432

Table 1. Comparison of the recognition accuracies of the different classification approaches. Model selection is done during training with cross validation. The kNN results for lowercase and digits are for choice of $k = 1$, other results for $k = 3$. All results for the different tasks are given for CV measured in training as well as measured with an independent test set.

category	GDTW		RBF		log Align		kNN	
	train	test	train	test	train	test	train	test
lowercase letters	95.2%	87.9%	95.3%	86.1%	93.75%	**88.2%**	90.67%	85.19%
uppercase letters	97.5%	**91.2%**	92.9%	86.3%	93.2%	87.7%	91.25%	87.4%
digits	98.0%	96.75%	96.4%	96.0%	97.6%	**97.0%**	96.25%	94.5%
mixed	81.5%	75.6%	77.7%	72.7%	79.42%	**75.9%**	72.14%	71.21%
ujimap	93.7%	**88.5%**	91.5%	85.3%	90.4%	85.6%	n.a.	n.a.

Table 2. Comparison of training and recognition time of the SVM based approaches on different tasks with the UJIPenchars dataset. All results are for the parallelized algorithm with eight threads running synchronously. The test system is equipped with an Intel Core i7 920 CPU running at 2.67 GHz and having hyperthreading enabled.

category	GDTW		RBF		Align	
	train (s)	test (ms)	train (s)	test (ms)	train (s)	test (ms)
lowercase letters	16.160	5.12	1.734	1.46	58.317	38.636
uppercase letters	12.206	5.04	1.705	0.53	66.555	42.786
digits	1.357	1.31	0.275	0.67	7.612	10.475
mixed	85.868	13.80	8.866	1.59	385.556	125.540
ujimap	60.619	11.47	6.253	0.87	309.852	105.852

from the results published by [2], where a random choice for generating their test set and training sets is used. To enable comparison with results published elsewhere we use the classification task that was used in the respective paper. The first class definition is introduced by Llorens et al. [6] for their UJIPenchars database, where each character and each digit corresponds to one class. The characters are represented case insensitive and zero is furthermore merged with the class “O”. We refer to this class definition as ”UJImap” in the following. The second class definition is the same as used in the evaluation of [2]. Their class definition divides the set into digits, lowercase characters and uppercase characters. They consider every character (case sensitive) to be an own class and further use one class for each digit. As in the work of [2], we train independent classifiers for lowercase, uppercase and digits. Further, we train a classifier for the mixed set composed of all symbols. The achieved recognition accuracies for different classifiers are summarized in Table 1. For the kNN method the best value for k was determined during training through cross validation. The runtime measured for the training and classification process for the SVM based classifiers is given by Table 2.

The effect of parallelization on the SVM training and prediction times is evaluated using the GDTW Kernel. Achieved runtimes are given by Table 3. For the kNN based classifiers the size of the knowledge base is important and for this reason we evaluate the runtime with respect to it (see Table 4).

Table 3. The training and prediction times of the SVM GDTW classifier for the different recognition tasks. The times are measured for one to eight threads concurrently running on the CPU.

	digits		lowercase		mixed	
Threads	train (s)	predict (ms)	train (s)	predict (ms)	train (s)	predict (ms)
1	3.813	6.54	56.396	26.83	302.635	72.42
2	2.316	3.37	34.704	13.36	190.113	38.24
3	1.592	2.38	24.121	8.86	131.428	25.92
4	1.322	1.93	19.517	7.09	104.779	20.09
8	1.057	1.31	16.160	5.12	85.868	13.80

Table 4. Time needed to calculate the DTW distance between a single query instance and all reference instances in the knowledge base. Results are given for varying sizes of the knowledge base and different kNN implementation variants. The two GPUs used in this benchmark are each equipped with a NVIDIA GT260 chip.

Knowledge Base size	4960	10,000	20,000	40,000	85,000
CPU 1 Thread:	714 ms	1410 ms	2820 ms	5647 ms	12888 ms
CPU 2 Threads:	370 ms	741 ms	1481 ms	2926 ms	6705 ms
CPU 3 Threads:	245 ms	496 ms	946 ms	1960 ms	4519 ms
CPU 4 Threads:	248 ms	377 ms	741 ms	1476 ms	3219 ms
CPU 8 Threads:	244 ms	348 ms	720 ms	1269 ms	2788 ms
1 GPU	32 ms	65 ms	133 ms	260 ms	562 ms
2 GPU	108 ms	114 ms	148 ms	232 ms	421 ms

5 Conclusion

We compared the different approaches using the freely available UJIPenchars2 dataset. The kNN method using the dynamic time warping distance achieves good recognition accuracies. This accuracy is outperformed by the SVM classification approaches. Within the different kernel choices the default RBF kernel used together with interpolation shows lowest accuracy. Which of the two other kernels achieves best results depends on the specific task (c.f. Table 1).

Additionally, we have shown that parallel computing allows significant speedups for SVM classification as well as for SVM training. Measured execution times for the different kernels show that there is a trade off between recognition accuracy and time requirements. The most accurate kernel (alignment kernel) turns out to be four to six times slower in training and seven to nine times slower in prediction. The alignment kernel used with the complete character set cannot fulfill the real time requirement, which we defined as being able to evaluate within 40 ms (25 frames per second).

Our parallel kNN implementation scales well with increasing number of processing units and increasing size of the knowledge base. The basic CPU implementation we created as a reference is not suitable for use in a real time environment. The required reduction in computation time can be achieved with our GPU based implementation of the kNN method. If a very large knowledge base with more than 40,000 samples is used, the dual GPU implementation provides better performance compared to the single GPU variant. This is the case, because we transfer all data to both GPUs and the PICE bus bandwidth is shared among the GPUs and the transfer time can not be amortized by speedup if a small size knowledge base is used. Further, we implemented the classifier and handwriting storage in such way that they are decoupled from the front end. This architecture allows the invocation of the complex classification algorithms by computationally limited clients, as for example smart phones. An open issue with the developed architecture is the latency of the networks used in mobile phones these days (GSM or UMTS) which have high latencies and cause additional delay in classification. Due to space restrictions on these devices a larger amount of training data can be stored by the server. This allows training data to be contributed by many different writers and to be collected through the Internet.

Our future research will focus on finding a kernel that is positive definite and can be applied to the data without modification. Another point we want to investigate is how we can move from character recognition to a system that is capable to recognize whole words. Concerning the speed issue we are interested in a parallelization of the SVM algorithm for GPUs. Experiments with the dataset olhwdb1 [12] which contains Chinese characters would be of practical relevance.

Acknowledgements. Part of this work was supported by the German Science Foundation (DFG) under the reference number ‘GA 1615/1-1’.

References

[1] Bahlmann, C.: Advanced Sequence Classification Techniques Applied to Online Handwriting Recognition. PhD thesis, Albert-Ludwigs-Universität Freiburg, Institut für Informatik (2005)

[2] Bahlmann, C., Haasdonk, B., Burkhardt, H.: On-line handwriting recognition with support vector machines - a kernel approach. In: Proc. of the 8th IWFHR, pp. 49–54 (2002)

[3] Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001)

[4] Cuturi, M., Vert, J.-P., Birkenes, O., Matsui, T.: A kernel for time series based on global alignments. CoRR, abs/cs/0610033 (2006)

[5] Guyon, I., Schomaker, L., Plamondon, R., Liberman, M., Janet, S.: Unipen project of on-line data exchange and recognizer benchmarks. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition. Conference B: Computer Vision & Image Processing, vol. 2, pp. 29–33 (1994)

[6] Llorens, D., Prat, F., Marzal, A., Castro, M.J., Vilar, J.M., Amengual, J.C., Barrachina, S., Castellanos, A., España, S., Gómez, J.A., Gorbe, J., Gordo, A., Palazón, V., Peris, G., Ramos-Garijo, R., Zamora, F.: The ujipenchars database: a pen-based database of isolated handwritten characters. In: Calzolari, N. (Conference Chair), Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., Tapias, D. (eds.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), European Language Resources Association (ELRA), Marrakech, Morocco (May 2008), ISBN 2-9517408-4-0

[7] NVIDIA. NVIDIA CUDA Best Practices Guide 2.3. NVIDIA (2009)

[8] Plamondon, R., Srihari, S.N.: Online handwriting recognition: A comprehensive survey. IEEE Transactions on Pattern Analysis and Maschine Intelligence 22(1), 63–84 (2000)

[9] Platt, J.C.: Fast training of support vector machines using sequential minimal optimization, pp. 185–208 (1999)

[10] Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 26, 43–49 (1978)

[11] Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond (Adaptive Computation and Machine Learning). The MIT Press, Cambridge (December 2001), ISBN 0262194759

[12] Wang, D.-H., Liu, C.-L., Yu, J.-L., Zhou, X.-D.: Casia-olhwdb1: A database of online handwritten chinese characters. In: ICDAR 2009: Proceedings of the Tenth International Conference on Document Analysis and Recognition (2009) (to appear)

Planning Cooperative Motions of Cognitive Automobiles Using Tree Search Algorithms

Christian Frese[1] and Jürgen Beyerer[1,2]

[1] Karlsruhe Institute of Technology,
Institute for Anthropomatics, Vision and Fusion Laboratory,
76128 Karlsruhe, Germany
http://www.ies.uni-karlsruhe.de

[2] Fraunhofer Institute of Optronics, System Technologies and Image
Exploitation IOSB, Fraunhoferstraße 1, 76131 Karlsruhe, Germany

Abstract. A tree search algorithm is proposed for planning cooperative motions of multiple vehicles. The method relies on planning techniques from artificial intelligence such as A* search and cost-to-go estimation. It avoids the restrictions of decoupling assumptions and exploits the full potential of cooperative actions. Precomputation of lower bounds is used to restrict the search to a small portion of the tree of possible cooperative actions. The proposed algorithm is applied to the problem of planning cooperative maneuvers for multiple cognitive vehicles with the aim of preventing accidents in dangerous traffic situations. Simulation results show the feasibility of the approach and the performance gain obtained by precomputing lower bounds.

1 Introduction

Wireless communication between cognitive vehicles bears a potential for increasing traffic safety by cooperative actions of multiple vehicles. In many dangerous situations, it is possible to prevent or mitigate an accident by performing an automated cooperative maneuver, whereas the drivers of the individual vehicles cannot achieve this on their own because of reaction times and lack of coordination possibilities [1]. Therefore, a safety system is proposed which intervenes in dangerous situations by autonomously executing cooperative maneuvers within a group of cognitive vehicles equipped with sensors and actuators [6]. This paper is concerned with an algorithm for the planning of cooperative maneuvers, which is an important component of such a system.

Most previous algorithms to plan cooperative motions for multiple vehicles or robots rely on decoupling assumptions to simplify the problem. One possibility is the path-velocity decomposition [10,7]: paths are planned separately for each vehicle, and a subsequent coordination of velocities along fixed paths is used to achieve collision-free motions. This approach has been extended to the coordination of motions within a roadmap [17]. The other common simplifying assumption leads to prioritized planning [3,13]. The motion of the highest-priority vehicle is planned without considering the other vehicles. For the second vehicle,

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 91–98, 2010.

a motion plan is derived which avoids the obstacles and the previously planned trajectory of the first vehicle, but ignores all remaining vehicles, and so on.

The decoupling approaches are appropriate for systems with negligible dynamics. However, in the present context of planning cooperative maneuvers in emergency situations, the cognitive automobiles have significant velocities and dynamics. This implies that many possible solutions are ruled out by the decoupling assumptions, and thus, accidents cannot be prevented in some situations although a feasible cooperative motion exists [5]. This contribution therefore investigates a cooperative motion planning algorithm which does not make these simplifying assumptions. In theory, it is well known that planning with the full freedom of cooperative actions is possible in the composite configuration space [11]. However, these so-called centralized methods have rarely been used in practice due to their high computational complexity [15]. In this paper, artificial intelligence algorithms such as A* search, branch and bound methods, and cost-to-go estimation are applied to cooperative motion planning, and it is shown that they can reduce computing times significantly compared to the naive centralized approach.

Section 2 formulates the cooperative motion planning task as a search problem in a tree of possible actions. In Section 3 it is shown that some of the information computed during the tree search can be reused in subsequent stages of the search. Methods to obtain lower bounds for subtrees are described in Section 4. Section 5 presents and compares two possible search strategies, namely A* search and depth-first branch and bound search. Simulation results and computation times are shown in Section 6, and conclusions are presented in Section 7.

2 Tree Search Formulation of Cooperative Motion Planning

In the proposed approach, both time t and action space $\mathcal{A}$ are discretized. Within the planning horizon, there are T decision points at times $t_0, \dots, t_{T-1}$. At each decision point, an action $\mathbf{a}$ is chosen out of the finite action set $\mathcal{A}$ and is constantly applied until the next decision point is reached. In the domain of cognitive automobiles, time discretization is justified because actuator limits prohibit a quick change of actions. The discrete action set should contain extreme actions like maximum braking and maximum steering admissible by the dynamics of the vehicle in its current state. These extreme actions are known to be optimal for a single vehicle evading an obstacle [16,9].

The state of vehicle i is denoted by $\mathbf{x}_i = (x_i, y_i, \varphi_i, v_i)^{\mathrm{T}}$, where (x_i, y_i) is the planar position of the vehicle centroid, φ_i is the yaw angle, and v_i is the scalar velocity of the vehicle. A kinodynamic vehicle model f_i is used to compute the state $\mathbf{x}'_i$ at time t_{k+1} which is reached by executing action a_i starting from state $\mathbf{x}_i$ at time t_k: $\mathbf{x}'_i = f_i(\mathbf{x}_i, a_i, t_k)$. The composite state vector $\mathbf{x}$ of the cooperative vehicles is obtained by concatenating the individual state vectors $\mathbf{x}_i$. A cooperative action is denoted as $\mathbf{a} = (a_1, \dots, a_m)^{\mathrm{T}}$, where $a_i \in \mathcal{A}_i$ is the action of the ith vehicle. When action $\mathbf{a}$ is applied from t_k until t_{k+1} starting from

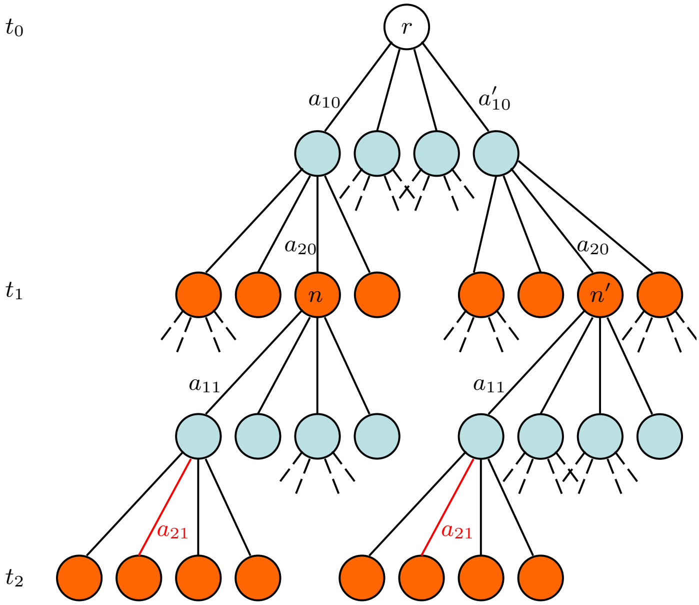


Fig. 1. Tree $\mathcal{T}$ of cooperative actions for $m = 2$ vehicles

state $\mathbf{x}$, a loss $L(\mathbf{x}, \mathbf{a}, t_k) \geq 0$ is incurred. The loss function rates collisions among vehicles or with obstacles, road departure, and control effort. Loss is a notion used in utility theory to determine preferences between alternative actions [2].

The possible action sequences of one vehicle can be arranged in a tree $\mathcal{T}_i$ having A_i^T leaves, with $A_i := |\mathcal{A}_i|$ being the number of actions. When planning cooperative motions of m vehicles, each action of one vehicle can be combined with all possible actions of the remaining vehicles, yielding the composite action set $\mathcal{A} := \mathcal{A}_1 \times \ldots \times \mathcal{A}_m$. If the action sets $\mathcal{A}_i$ are of equal size $A_0 = |\mathcal{A}_i|$ for all vehicles, the composite action set contains $A := |\mathcal{A}| = A_0^m$ actions. Therefore, the tree $\mathcal{T}$ of cooperative decisions has $A^T = A_0^{mT}$ leaves (see Fig. 1).

This results in large trees even for moderate values of A_0 and T. A naive search for the minimum-loss action sequence visiting every tree node cannot be conducted efficiently. There are two possibilities for speeding up the search while retaining the guarantee of obtaining the optimal solution: firstly, the computing time per tree node can be reduced, e. g., by avoiding some redundancies, and secondly, the number of explored nodes can be reduced by pruning subtrees that cannot contain the optimal solution. Both approaches are detailed in the following sections.

The possibility to use explicit vehicle, loss, and action models is an important advantage of the algorithm. This allows to consider geometry, kinematics, and dynamics in the required level of abstraction. More details on the models currently used are described in [4].

3 Reuse of Computed Information

A naive traversal of the tree $\mathcal{T}$ causes a lot of redundant computations to be performed. For example, let the action sequences $((a_{10}, a_{20})^{\mathrm{T}}, (a_{11}, a_{21})^{\mathrm{T}})$ and $((a'_{10}, a_{20})^{\mathrm{T}}, (a_{11}, a_{21})^{\mathrm{T}})$ pass through the nodes n and n' of $\mathcal{T}$, respectively.[1] If $a'_{10} \neq a_{10}$, $n \neq n'$ results (Fig. 1). However, the state of the second vehicle is the same in both nodes:[2] $\mathbf{x}_2(n) = \mathbf{x}_2(n') = f_2(\mathbf{x}_2(r), a_{20}, t_0)$, where r is the root node of $\mathcal{T}$. Therefore, the identical computation $f_2(\mathbf{x}_2(n), a_{21}, t_1)$ is executed both for n and for n' in order to determine the state of the respective successor node.

Hence, it would be beneficial to store the vehicle states in a suitable look-up data structure, avoiding redundant state computations. This can be achieved by annotating the single-vehicle trees $\mathcal{T}_i$, $i = 1, \ldots, m$ with the state information. The single-vehicle trees can be kept in memory as they are much smaller than $\mathcal{T}$ (see Section 2). Mappings $n_i : \mathcal{T} \mapsto \mathcal{T}_i$ can be defined which yield for each node n of the cooperative search tree $\mathcal{T}$ the corresponding nodes $n_i(n)$ in the single-vehicle trees $\mathcal{T}_i$. In the above example, $n_2(n) = n_2(n')$, and therefore the successor state can be stored in $\mathcal{T}_2$ when n is visited and looked up again when n' is visited.

Additionally, some components of the loss function only depend on the action sequence of one vehicle. In particular, these are the terms accounting for collisions with obstacles, road departure, and control effort. In order to avoid redundant computations of loss terms, their sum $L_i(\mathbf{x}_i, a_i, t_k)$ for each applied action $a_i \in \mathcal{A}_i$ can be stored in $\mathcal{T}_i$ in the same way as the vehicle state.

The only remaining component of the loss function is the term for collisions among cooperative vehicles. This term does not depend on the full cooperative action sequence either: it only depends on the actions of a pair of vehicles [12]. Therefore, similar issues with redundant computations arise during a cooperative search with $m > 2$, and the same principle can be applied to store the already computed loss values. Precisely, the two-vehicle trees $\mathcal{T}_{ij}$, $1 \leq i < j \leq m$, can be annotated with the collision loss $L_{ij}(\mathbf{x}_i, \mathbf{x}_j, a_i, a_j, t_k)$, and mappings $n_{ij} : \mathcal{T} \mapsto \mathcal{T}_{ij}$ can be constructed to retrieve the stored values.

4 Precomputing Lower Bounds

During tree search, it is helpful to have a cost-to-go estimate for the optimal loss within a subtree of $\mathcal{T}$. These estimates can be used as a heuristic to guide the search and to avoid subtrees with high loss. In particular, when a lower bound

[1] In the following, $\mathbf{x}_i(n)$ and $t(n)$ denote the state of the ith vehicle at n and the decision time of the node, respectively.

[2] Strictly speaking, this is true only if no collisions have occurred yet. After a collision of two cooperative vehicles, the state of one vehicle is no longer independent of the other vehicle's actions. Currently, this is not considered by the proposed tree search method. Note, however, that only the states in the subtree below the point of collision would have to be recomputed.

for the loss within a subtree is known, it can be proven under certain conditions that the optimal action sequence cannot visit this subtree.

Lower bounds for the cooperative motion planning problem are obtained as follows. The single-vehicle search trees $\mathcal{T}_i$ and the corresponding loss values L_i are precomputed. Then, lower bounds h_i can be established by propagating loss information upwards the trees recursively:

$$h_i(\mathbf{x}_i, t_k) := \begin{cases} 0 & \text{if } k = T \\ \min\limits_{a_i \in \mathcal{A}_i} \{L_i(\mathbf{x}_i, a_i, t_k) + h_i(f(\mathbf{x}_i, a_i), t_{k+1})\} & \text{if } k < T \end{cases} \tag{1}$$

A lower bound for the subtree rooted at node n of the cooperative search tree $\mathcal{T}$ can be computed as follows:

$$h_{\text{SV}}(n) := \sum_{i=1}^{m} h_i(\mathbf{x}_i(n_i(n)), t(n)) \tag{2}$$

In a similar way, lower bounds $h_{\text{coll}}(n)$ are obtained by precomputing the two-vehicle trees $\mathcal{T}_{ij}$ and the collision loss values L_{ij}.

Lower bound precomputation (P) and storage (S) of possibly reusable information can either be performed for the entire problem or only up to certain tree depths $0 \leq P_{\text{SV}} \leq S_{\text{SV}} \leq T$ and $0 \leq P_{\text{coll}} \leq S_{\text{coll}} \leq T$. $h_{\text{SV}}(n)$ and $h_{\text{coll}}(n)$ are defined to be zero if no precomputation has been performed for node n.

5 Search Strategies

The A* algorithm makes use of a so-called heuristic function to search graphs efficiently [8]. If the heuristic function $h(n)$ yields lower bounds of the true loss values from n to a goal node, the search is guaranteed to find the optimal solution [14]. As the precomputed values $h_{\text{SV}}(n)$ and $h_{\text{coll}}(n)$ fulfill this requirement, A* can be used to search the tree of cooperative actions $\mathcal{T}$. Starting with the root r, nodes are inserted into a priority queue sorted by ascending value of $g(n) := L(r, n) + h_{\text{SV}}(n) + h_{\text{coll}}(n)$, where $L(r, n)$ is the loss accumulated on the path from r to n in $\mathcal{T}$. The minimum element of the queue is chosen for processing, and the children of this node are inserted into the queue. The search terminates with the optimal action sequence as soon as a leaf node is reached.

Alternatively, the branch and bound (BB) search strategy can be used. It explores the tree in depth-first order, remembering the best solution found so far and its loss value L^*. Subtrees are excluded from the search if it can be proven that they cannot contain any part of the optimal solution. This is the case if $g(n) > L^*$ for the root node n of the subtree. Then, the total loss of any path passing through the subtree must be greater than L^*.

When compared to BB, the A* strategy has the drawback of an additional logarithmic time complexity for the operations of the priority queue. Moreover, the memory requirements of the priority queue can become problematic, as the queue may contain every leaf node in the worst case. By contrast, the BB strategy has very low memory requirements. Only the current path of the depth-first search and the best action sequence found so far need to be remembered.

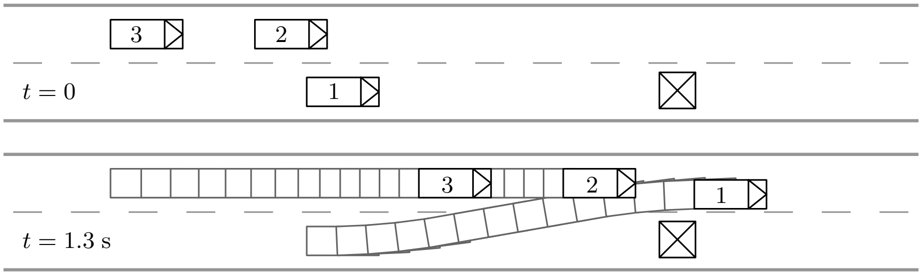


Fig. 2. Example of a cooperative merging maneuver planned with the proposed algorithm. While vehicle 1 changes to the left lane due to the obstacle, vehicles 2 and 3 brake in order to allow the merging. The full planning horizon has been 2.5 seconds.

Table 1. Computing times in seconds averaged over 20 different instances of the merging problem of Fig. 2, with $m = 3$ and $T = 4$. The best results for A* and BB, respectively, are highlighted. Also shown is the number of tree nodes visited. Search is restricted to a small fraction of the tree: the total number of leaf nodes is $1.3 \cdot 10^{10}$ in this example.

method	P_{SV}	S_{SV}	P_{coll}	S_{coll}	precomputation time		total time		nodes visited	
					average	max	average	max	average	max
A*	0	0	0	0	0.000	0.000	0.390	1.679	7 332	33 796
A*	0	4	0	0	0.000	0.000	0.078	0.371	7 332	33 796
A*	0	4	0	3	0.000	0.000	**0.026**	0.098	7 332	33 796
A*	3	4	0	3	0.017	0.020	0.033	**0.090**	4 638	25 438
A*	4	4	4	4	31.955	37.791	31.972	37.807	**477**	**2 303**
BB	0	0	0	0	0.000	0.000	33.151	380.278	2 840 040	34 087 340
BB	0	4	0	0	0.000	0.000	0.189	1.144	16 915	114 761
BB	3	4	0	3	0.017	0.020	**0.094**	0.832	6 467	79 242
BB	4	4	0	3	0.148	0.171	0.156	**0.196**	377	1 725
BB	4	4	4	4	31.581	37.422	31.598	37.439	**97**	**525**

6 Results

The proposed algorithms have been implemented in C++ and integrated with the traffic simulator described in [18]. They have been tested in various obstacle avoidance, merging, overtaking, and intersection scenarios [4]. Fig. 2 depicts a cooperative maneuver planned with the proposed algorithm.

Tables 1 and 2 list the computing times measured for different variants of the algorithm on an off-the-shelf desktop PC. The best performing variants of each A* and BB are shown together with the respective baseline algorithms. On average, A* performed better than BB, but there are some instances of the problem for which BB was significantly faster. For example, computing times of 0.33 s (BB) and 1.2 s (A*) have been observed for a specific overtaking problem involving four vehicles. Concerning the different storage and precomputation variants, reuse of single-vehicle information was always useful. Precomputation

Table 2. Average and maximum computing times in seconds for 150 different intersection situations

m	T	method	P_{SV}	S_{SV}	P_{coll}	S_{coll}	precomp. time		total time		nodes visited	
							average	max	average	max	average	max
2	5	A*	0	0	0	0	0.000	0.000	0.136	0.775	2 927	17 521
2	5	A*	2	5	0	0	0.002	0.003	0.021	0.073	752	4 471
2	5	BB	0	0	0	0	0.000	0.000	0.933	4.028	73 533	306 936
2	5	BB	2	5	0	0	0.002	0.003	0.042	0.091	1 731	7 186
4	3	A*	0	0	0	0	0.000	0.000	0.624	10.547	5 718	163 387
4	3	A*	2	3	0	3	0.005	0.006	0.032	0.171	501	5 973
4	3	BB	0	0	0	0	0.000	0.000	733.349	6499.990	31 882 148	249 583 950
4	3	BB	3	3	0	3	0.048	0.054	0.085	0.781	413	11 643
4	4	A*	4	4	0	3	0.314	0.373	0.329	0.522	3 154	68 936
4	4	BB	4	4	0	4	0.313	0.374	1.497	64.134	13 373	926 700

of single-vehicle lower bounds and reuse of collision loss values were advantageous up to certain tree depths P_{SV}, S_{coll}. For larger S_{coll} the overhead for maintaining the data structures dominates the gain of avoiding redundant collision checking. Precomputation of collision loss was not beneficial in most situations: while it reduced the number of visited nodes even further, the precomputation effort usually more than outweighed the performance gain during the search (Table 1).

In the most difficult setting considered ($m = 4$, $T = 4$) none of the variants with $P_{\mathrm{SV}} < T$ was feasible: BB showed runtimes in the order of hours, and A* aborted with out of memory error. With precomputation, the algorithm has potential for real-time use in all simulated instances of the considered tasks (Table 2). This result highlights the effectiveness of the presented lower bound precomputation scheme, although no guarantees on the number of explored nodes can be given for the theoretical worst case.

7 Conclusions

A new algorithm for cooperative motion planning has been proposed which exploits the full potential of cooperative actions. It has been shown that precomputation of lower bounds is essential for the computational feasibility of the tree search approach. The method has been successfully applied to simulate cooperative maneuvers in dangerous traffic situations. Although A* performed better on average, BB and A* search each have advantages in specific problem instances so that no clear preference between the two search strategies could be derived. Further research is necessary to gain deeper insight into this issue, and possibilities for combining both search strategies will be evaluated.

Acknowledgements

The authors gratefully acknowledge support of this work by Deutsche Forschungsgemeinschaft (German Research Foundation) within the Transregional Collaborative Research Center 28 “Cognitive Automobiles”. Furthermore, the

authors would like to thank all project partners contributing to the development of the simulation system.

References

1. Batz, T., Watson, K., Beyerer, J.: Recognition of dangerous situations within a cooperative group of vehicles. In: Proc. IEEE Intelligent Vehicles Symposium, Xi'an, China, pp. 907–912 (June 2009)
2. Berger, J.O.: Statistical decision theory and Bayesian analysis, 2nd edn. Springer Series in Statistics. Springer, New York (1993)
3. Erdmann, M., Lozano-Pérez, T.: On multiple moving objects. Algorithmica 2, 477–521 (1987)
4. Frese, C.: Cooperative motion planning using branch and bound methods. Tech. Rep. IES-2009-13. In: Beyerer, J., Huber, M. (eds.) Proceedings of the 2009 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory, pp. 187–201. KIT Scientific Publishing (2010)
5. Frese, C., Batz, T., Beyerer, J.: Kooperative Bewegungsplanung zur Unfallvermeidung im Straßenverkehr mit der Methode der elastischen Bänder. In: Dillmann, R., et al. (eds.) Autonome Mobile Systems, pp. 193–200. Springer, Heidelberg (2009)
6. Frese, C., Beyerer, J., Zimmer, P.: Cooperation of cars and formation of cooperative groups. In: Proc. IEEE Intelligent Vehicles Symposium, Istanbul, pp. 227–232 (June 2007)
7. Ghrist, R., LaValle, S.: Nonpositive curvature and pareto optimal coordination of robots. SIAM Journal on Control and Optimization 45(5), 1697–1713 (2006)
8. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on System Science and Cybernetics 4(2), 100–107 (1968)
9. Hillenbrand, J.: Fahrerassistenz zur Kollisionsvermeidung. Dissertation, Universität Karlsruhe, TH (2007)
10. Kant, K., Zucker, S.: Toward efficient trajectory planning: The path-velocity decomposition. Int. J. Robotics Research 5(3), 72–89 (1986)
11. LaValle, S.: Planning Algorithms, 1st edn. Cambridge University Press, Cambridge (2006)
12. LaValle, S., Hutchinson, S.: Optimal motion planning for multiple robots having independent goals. IEEE Transactions on Robotics and Automation 14(6), 912–925 (1998)
13. Regele, R., Levi, P.: Kooperative Multi-Roboter-Wegplanung durch heuristische Prioritätenanpassung. In: Autonome Mobile Systems, pp. 33–39. Springer, Heidelberg (2005)
14. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice-Hall, Englewood Cliffs (2003)
15. Sánchez, G., Latombe, J.C.: On delaying collision checking in PRM planning: Application to multi-robot coordination. Int. J. Robotics Research 21(1), 5–26 (2002)
16. Schmidt, C., Oechsle, F., Branz, W.: Research on trajectory planning in emergency situations with multiple objects. In: Proc. IEEE Intelligent Transportation Systems Conference, pp. 988–992 (September 2006)
17. Švestka, P., Overmars, M.: Coordinated path planning for multiple robots. Robotics and Autonomous Systems 23, 125–152 (1998)
18. Vacek, S., Nagel, R., Batz, T., Moosmann, F., Dillmann, R.: An integrated simulation framework for cognitive automobiles. In: Proc. IEEE Intelligent Vehicles Symposium, Istanbul, pp. 221–226 (June 2007)

Static Preference Models for Options with Dynamic Extent

Thomas Bauereiß, Stefan Mandl, and Bernd Ludwig

Dept. of Computer Science 8 (Artificial Intelligence),
Friedrich-Alexander-Universität Erlangen-Nürnberg,
Haberstraße 2, D-91058 Erlangen, Germany
thomas@bauereiss.name, {Stefan.Mandl,Bernd.Ludwig}@cs.fau.de
http://www8.informatik.uni-erlangen.de

Abstract. Models of user preferences are an important resource to improve the user experience of recommender systems. Using user feedback static preference models can be adapted over time. Still, if the options to choose from themselves have temporal extent, dynamic preferences have to be taken into account even when answering a single query. In this paper we propose that static preference models could be used in such situations by identifying an appropriate set of features.

1 Introduction

Preferences are an important component of personalized information retrieval and recommendation systems (see [4]). Given a recommendation or information retrieval system that interacts with the user such that the user enters queries and the system produces a list of options the user may choose from, the order of the presented items typically is determined by the quality of the item with respect to some *objective* value function. Given that this value function is user-pareto-optimal in the sense, that it cannot be changed to improve from the perspective of one potential user without worsening from the perspective of another potential user, any further improvement of the system can only be achieved by personalized user specific adaptations, hence user models and user preferences.

In this paper, after discussing the use of standard machine learning techniques for classification and regression to represent user preferences in a standard scenario—TV program recommendations—, we focus on a tour recommender. The major difference between tour planning and TV programs is that under the usual granularity of discourse, TV programs typically are considered as singular events (though they have temporal extent) while tours and trips are considered multi-part events.

The goal of this paper is to empirically identify proper features of multi-part options in the tour-planning scenario such that standard models of preferences can be used to enhance user experience.

Section 2 gives a short account on modeling user preferences: Section 2.1 gives a formal definition of preferences, Section 2.2 contains some notes on preference elicitation, Section 2.3 presents various concrete preferences models, and

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 99–106, 2010.

Section 2.4 specifies the performance measures we use to evaluate preference models. Section 3.1 describes a user survey and the use of the preference models discussed before in a standard TV program recommendation scenario. Section 3.2 describes a set of features for multi-part tours such that the previous preference models can be used for this multi-part setup. Results from a preliminary study show that the proposed method of using standard preference models with a dynamic set of features is appropriate for the task. Section 4 concludes the paper and discusses future research.

2 Preference Modeling

2.1 Formalizing Preferences

In [12], several formalizations of preferences are discussed. They have in common that preferences are expressed as binary relations on a set of options. If a user prefers an option a to another option b, then the pair (a, b) is an element of her preference relation. In this paper, we define these relations in the following way:

Definition 1. *A preference relation $P \subset O \times O$ on a set of options O is a strict weak ordering with the properties*

- *Irreflexivity:* $\forall a \in O : \neg P(a, a)$
- *Asymmetry:* $\forall a, b \in O : P(a, b) \longrightarrow \neg P(b, a)$
- *Transitivity:* $\forall a, b, c \in O : (P(a, b) \wedge P(b, c)) \longrightarrow P(a, c)$
- *Transitivity of equivalence:* $\forall a, b, c \in O : P(a, b) \longrightarrow (P(a, c) \vee P(c, b))$.

This definition is consistent with properties commonly associated with preferences, e.g. irreflexivity (an option a is not considered to be preferred to itself). It also implies the existence of a numerical representation of preference relations [12, Theorem 6.2]:

Theorem 1. *For a preference relation $P \subset O \times O$ on a finite set of options O, a mapping $g : O \mapsto \mathbb{R}^+$ exists such that $\forall a, b \in O : (a, b) \in P \text{ iff } g(a) > g(b)$.*

This means that the prediction of preferences may be understood as the prediction of this representation function $g : O \mapsto \mathbb{R}^+$. For every option we aim to calculate a real number such that the ordering induced by the values of g for a set of options matches the user's preferences.

2.2 Preference Elicitation

Filling the preference model is an important task for systems using preferences. Possible ways to acquire the desired information are letting the user specify rules for her preferences [13] or gathering implicit [7] or explicit feedback [10]. In this paper, we use the latter approach and let users rate options on a scale from 1 (disliked) to 9 (liked). The underlying assumption is that we can infer what the user might currently like by using information about what she liked in the past.

The motivation to use explicit feedback instead of implicit feedback is the fact that in this paper we want to focus on the preference model and its *evaluation*. Therefore, features that enhance user experience at the cost of measurability of system correctness – like implicit feedback – are disregarded here.

2.3 Preference Models

Common ways of using feedback information to predict preferences are collaborative filtering [1] and content-based filtering [11], which can also be combined [3]. In this paper, we focus on content-based filtering. For this approach, options have to be turned into feature vectors, which can then be used as input for standard machine learning methods. In Section 3 we use different kinds of regression and classification models provided by the Weka toolkit for data mining [5].

Regression models try to predict a user's rating of an option as a real number, based on the user's previous ratings. These numbers are interpreted as values of a preference representation function g as mentioned in section 2.1.

Classification models assign a discrete class value to an option. The options previously rated by the user have to be separated into two classes, e.g. by labeling options with a rating below the user's average rating as "disliked" and options with a rating above average as "liked". The classifier can then be trained with this labeled set of examples and used to predict class labels for new options. When two options are assigned the same class label, the class probabilities provided by the classifier are used to order the options. The higher the estimated probability of an option the more it is considered to be preferred: The class probabilites are used as preference scores $g(a)$.

The preference models considered here are static, as the dynamics of preferences changing over time and from one situation to another are not explicitly modeled. Still, static preference models can capture long-term changes in preferences by adapting to user feedback. Additionally, we propose that static models can also be used for preferences regarding dynamic options, where the options themselves have temporal extent.

2.4 Performance Measures for Preference Models

In order to evaluate preference models, we gathered rating data from a set of users, performed 10-fold stratified cross-validation [8] for each user and averaged the results. These results were calculated in terms of the normalized distance-based performance measure (NDPM) [14], which is defined as:

$$ndpm(P_u, P_s) = \frac{2C^- + C^u}{2C} = \frac{2|P_u \cap P_s^{-1}| + |P_u \cap I_s|}{2|P_u|} \tag{1}$$

where C^-, C^u, and C are the numbers of contradicting, compatible and total user preference pairs, respectively, P_u and P_s are the user's actual preference relation and the one calculated by the system, respectively, P_s^{-1} is the inverse relation $\{(a, b) \mid (b, a) \in P_s\}$, and the indifference relation corresponding to the preference relation calculated by the system is denoted by $I_s =$

$\{(a,b) \mid (a,b) \notin P_s \wedge (b,a) \notin P_s\}$. Basically, this measure counts the differences between the user and the system relation, where compatible pairs are counted once and contradictory pairs are counted twice. This count is then normalized with the maximum distance $2|P_u|$, such that values of *ndpm* lie in the interval from 0 to 1, where 0 indicates a perfect match between predicted and actual user preference, 1 is the maximum distance, and 0.5 is equivalent to the performance of a random generator (assigning uniformly distributed random preference scores $g(a)$ to all available options).

Additionally, we used two other measures, one based on the symmetric set difference between the relations and one based on the mean squared error of the rank $rank(o,P)$ assigned to an option o according to a preference relation P:

$$rdpm(P_u,P_s) = \frac{|P_u \Delta P_s|}{\mathrm{E}\left[|P_u \Delta P_r|\right]} = \frac{|(P_u \setminus P_s) \cup (P_s \setminus P_u)|}{\mathrm{E}\left[|(P_u \setminus P_r) \cup (P_r \setminus P_u)|\right]} \tag{2}$$

$$rsre(P_u,P_s) = \frac{\frac{1}{|O|}\sum_{o \in O}\left(rank(o,P_u) - rank(o,P_s)\right)^2}{\mathrm{E}\left[\frac{1}{|O|}\sum_{o \in O}\left(rank(o,P_u) - rank(o,P_r)\right)^2\right]} \tag{3}$$

where P_r is a randomly generated preference relation and $\mathrm{E}[\cdot]$ is the expected value (estimated by calculating a sample mean). In case of partial preference orders, we estimated the expected value regarding randomly generated linear extensions to total orders by calculating a sample mean. With these measures, an ideal match between user and system corresponds to a value of 0, the performance of a random generator to 1. We call these measures relative distance-based performance measure (RDPM) and relative squared rank error (RSRE), respectively. Because of the normalization with the empiric performance of a random generator, we expected these two measures to be less optimistic than NDPM.

In the following section, we evaluate several preference models for two application domains, one with static and one with dynamic options. Although the domains are very different in nature, we show that after finding suitable features, the same methods can be applied to predict preferences in both cases.

3 Evaluation of Static Preference Models in Static and Dynamic Domains

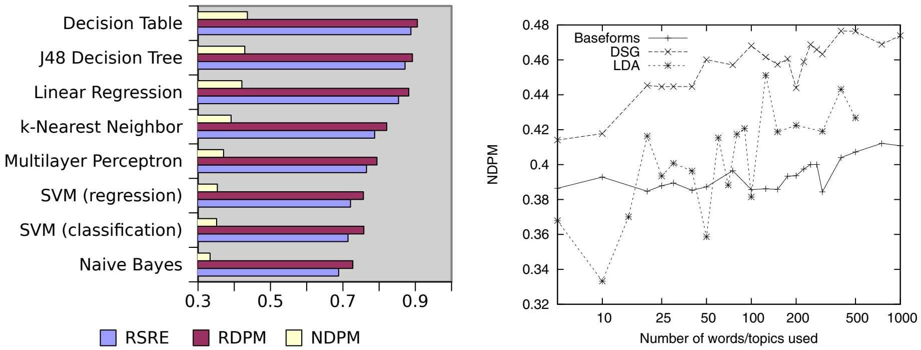
3.1 TV Program Recommender—A Domain with Static Options

We extend the TV Program Recommender that is presented in [9]. Queries have the form of free text terms. Ranking is performed by following the standard *bag of words* approach. For titles and descriptions of TV programs, the following feature sets are used:

- Baseform – every (normalized) word is a dimension in the feature vector
- DSG (Dornseiff group) – baseforms are mapped to topics which are defined in the Dornseiff lexicon for German language.

- LDA (Latent Dirichlet Allocation [2]) – baseforms are mapped to corpus-specific topics using a probabilistic model

Additionally, we use other features available from electronic program guide (EPG) data, namely the channel the program is broadcast on, the weekday and time of broadcast and the duration of the program. We gathered data for evaluating different preference models using an online questionnaire. 41 participants gave ratings (i.e. explicit feedback) for an average number of 45.6 programs.



(a) Comparison of preference models (b) Evaluation of textual features

Fig. 1. Evaluation of models and features for the TV domain (smaller values are better)

The results displayed in Fig. 1(a) show a high consistency between the three performance measures. We evaluated several models provided by Weka: probabilistic methods (Naive Bayes), instance-based methods (k-nearest neighbor), neural networks (multilayer perceptron), decision trees, decision tables and support vector machines. For our data, Naive Bayes achieves the best performance.

We also experimented with different ways of representing and selecting features, in particular for the textual features of program titles and descriptions. Fig. 1(b) shows the results of using three different kinds of textual features with a Naive Bayes classifier. We evaluated the use of baseforms, Dornseiff groups and LDA topics to represent title and description of programs. Additionally, we evaluated different choices for the number of words or topics to be used. We employed a feature selection method that ranks the features according to their mutual information with the class and selects a fixed number of features with the highest scores. The results show that for our application, the best performance is achieved with relatively low numbers of selected words or topics. For LDA, the number of topics must be configured at the time of topic generation. The performance when using LDA features is highly dependent on the choice of this number. In our case, setting the number of topics to 10 produced the best results. LDA features with 10 topics also performed better than the other two kinds of textual features for our data.

3.2 Tour Planner—A Domain with Dynamic Options

The second application domain we investigated is a tour planner that was designed for big events like the "Long Night of Sciences" in Erlangen, Nuremberg and Fürth. Throughout this "long night", hundreds of talks, workshops, film presentations and information booths are offered on a single evening. The system allows the user to select a set of individual events that she is interested in. It then recommends a detailed tour plan consisting of a sequence of events and the details of the journeys from one event location to the next one. In its current version, the system recommends the tour with the maximum number of events.

However, it is desirable to take the preferences of individual users into account. In contrast to the static options in the case of TV programs, the options in this application domain are dynamic in nature. Not only are they generated dynamically for each user query, but the temporal extent and dynamics of the tours themselves like sequence and lengths of events, journeys and pauses are essential for user preferences. Still, we propose static preference models as discussed before to be used for these dynamic options by finding suitable features.

Table 1. Feature evaluation in terms of NDPM for individual users (smaller values are better; bold features are selected with the method described in [6])

Feature	User 1	User 2	User 3
Number of events	**0.374**	0.286	**0.175**
Average event duration	0.264	**0.102**	0.274
Event descriptions	**0.356**	**0.461**	**0.424**
Topic spread of event descriptions	0.520	0.311	0.363
Number of visited places	**0.307**	0.142	0.228
Ratio of events with fixed start and end time	**0.240**	**0.130**	**0.266**
Percentage of time spent at events	0.217	**0.344**	0.596
Ratio of journey duration to event duration	0.228	0.285	0.573
Elapsed time at begin of first event	0.424	0.363	0.395
Number of nearby events during pauses	0.580	0.667	0.476
Longest waiting time	0.710	**0.376**	0.302
Shortest waiting time	0.543	0.500	0.458
Longest foot walk	0.563	0.512	0.432
Longest bus ride	0.373	0.435	0.407
Number of bus changes	0.383	0.254	0.353
Number of different bus lines used	0.358	0.251	0.322
Distance between events	0.287	**0.172**	**0.172**
Performance with feature selection [6]	0.285	0.119	0.217
Performance with all features used	0.180	0.109	0.148

In search of these features, we asked several users for aspects that might be important for them when rating a tour, and formalized these ideas as features. Table 1 lists the features that resulted from this process. They represent different aspects of the composition and temporal dynamics of a tour, like the selection of events (e.g. number of events, descriptions or topic spread of events in the tour), travel modalities (e.g. number of bus changes, duration of longest bus ride or total distance travelled throughout the evening) and pauses (e.g. longest waiting time before an event, average ratio of waiting time to duration of the event or number of nearby events while waiting for another event to begin).

In order to evaluate these features, we created a set of test items by constructing nine different queries and collecting the six top results returned by the tour planner. These 54 tours were labeled according to the preferences of three different users[1]:

- User 1 prefers talks and film presentations to other kinds of events, like workshops and information stands. She also prefers to stay in the same place when possible, and avoids bus travels.
- User 2 wants to visit as many different places as possible, even if this means that she can only visit few and short events at each location.
- User 3 wants to attend as many events as possible throughout the evening. She prefers to have a tight schedule and as little waiting time as possible.

We then compared the same preference models with each other as for the TV recommender domain by performing cross-validation for all models with the labeled sets of tours for each user. The results are shown in Fig. 2. Naive Bayes, the best model for the TV recommender, also does well for the tour planner, but the best performance in this case is achieved with SVM classification. We therefore used SVM classification to calculate performance measures for each of the proposed features. Table 1 lists the results of the evaluation of these features for the three users individually. The results show that all features have predictive value (i.e. an NDPM value less than 0.5, which would be equivalent to a random generator) for at least one of the users. The feature selection method described in [6] does a good job at selecting small subsets of features while retaining performance, but it doesn't improve performance in our case. Overall, the model based on Support Vector Machines in combination with our proposed set of features achieves good performance at predicting preferences for tours, even better results than in the TV recommender scenario.

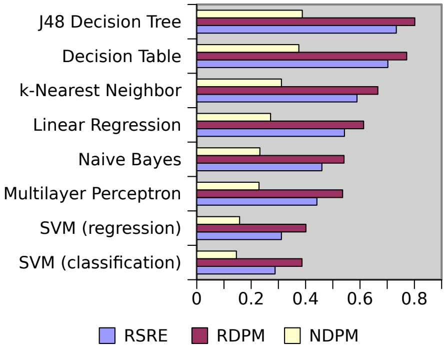


Fig. 2. Comparison of preference models for the tour planner (smaller values are better)

4 Conclusions

In this paper, we described an approach to content-based, static preference modeling. It is based on representing the options available to the user with features

[1] The data set can be downloaded from http://www8.informatik.uni-erlangen.de/en/datasets.html

and then using machine learning methods to adapt a preference model over time by incorporating feedback from the user. We introduced two application domains, and discussed the differences between them. The options to which preferences refer in the TV recommender scenario can be considered singular events, while in the tour planner domain, preference models need to deal with options consisting of multiple parts with temporal extent and dynamics. We presented empirical results from preliminary studies showing that the approach to preference modeling described in the paper can be applied successfully to application domains with static and dynamic options by finding suitable sets of features. In the future, we intend to validate our findings for other application domains and larger data sets, for example by gathering ratings from users at future events like the upcoming "Long Night of Music" in Munich.

References

1. Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Communications of the ACM 40(3), 66–72 (1997)
2. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
3. Burke, R.: Hybrid web recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 377–408. Springer, Heidelberg (2007)
4. Chen, P.M., Kuo, F.C.: An information retrieval system based on a user prole. The Journal of Systems and Software 54 (2000)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA data mining software: An update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)
6. Hall, M.A.: Correlation-based Feature Subset Selection for Machine Learning. Ph.D. thesis, University of Waikato (1998)
7. Kelly, D., Teevan, J.: Implicit feedback for inferring user preference: a bibliography. In: ACM SIGIR Forum, vol. 37, pp. 18–28. ACM, New York (2003)
8. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence, Citeseer, vol. 14, pp. 1137–1145 (1995)
9. Ludwig, B., Mandl, S.: Centering information retrieval to the user. Revue D'intelligence Artificielle 24, 96–120 (2010)
10. McNee, S., Lam, S., Konstan, J., Riedl, J.: Interfaces for eliciting new user preferences in recommender systems. In: User Modeling 2003, pp. 178–187. Springer, Heidelberg (2003)
11. Mooney, R., Roy, L.: Content-based book recommending using learning for text categorization. In: Proceedings of the Fifth ACM Conference on Digital Libraries, pp. 195–204. ACM, New York (2000)
12. Öztürk, M., Tsoukiàs, A., Vincke, P.: Preference modelling. Tech. Rep. 2003-34, DIMACS (2003)
13. Stefanidis, K., Pitoura, E., Vassiliadis, P.: Adding context to preferences. In: Proc. ICDE, Citeseer, pp. 846–855 (2007)
14. Yao, Y.: Measuring retrieval effectiveness based on user preference of documents. Journal of the American Society for Information Science 46(2), 133–145 (1995)

Towards User Assistance for Documents via Interactional Semantic Technology

Andrea Kohlhase

German Research Center for Artificial Intelligence (DFKI), 28359 Bremen, Germany
Andrea.Kohlhase@dfki.de

Abstract. As documents become disseminated widely, establishing the context for interpretation becomes difficult. We argue that high-impact documents require embedded user assistance facilities for their readers (*not* authors) to cope with their semantic complexity. We suggest to illustrate such documents and their players with a semiformal background ontology, an interpretation mapping between domain concepts and semantic objects, and to enable with this setup semantic interaction. We showcase the feasibility and interactional potential of user assistance for documents based on interactional semantic technology with an exemplary implementation for a high-impact Excel spreadsheet.

1 Introduction

Internet technologies enable wide and easy dissemination of documents. Targeted audiences of typical documents are therefore not only much bigger but also much more diverse than before. The first yields more investment into the creation of relevant documents – often resulting in high information complexity, the second calls for adaptive/adaptable documents. Thus, active document formats are used, so that the documents' surface structure can adapt to the environment or user input. They are realized in many forms, for example, HTML documents are elaborated for the web (with JavaScript-based interactions), and Word, Excel and PowerPoint (with VBA macros) or Calc (with OOo macros) ones for office use. In these documents the difference between code and data is vanishing. Even though they still need players to present them and editors to support their authoring, they themselves can be considered a mixture of data and software conglomerated into a unique system. To overcome usability problems, systems are often equipped with user assistance features like help systems. Therefore, the question of *user assistance (UA) for (active) documents* arises. In this paper we are not so much interested in help for the code of such documents, those are typically provided by the player software (e.g. via manuals or support web portals), but rather in the semantic complexity of documents.

Let us look at the problem setting specifically for spreadsheet documents as they represent par-excellence high-impact, active documents. Panko reports in [1] on their usage rate in business firms in 2003/04 which results in the cognition that "*most financial reporting systems today use spreadsheets*". The spreadsheet error rates

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 107–115, 2010.

he found in [2] reach up to astounding 90%; see also [3] to understand the impact of this "*spreadsheet hell*" [4]. Moreover, the typically unassisted dissemination leads to misapprehension of the underlying model (e.g. analyzed as the need for a "*sociology of spreadsheet use*" [5, p. 137]). The losses caused by formula errors and misinterpretation have led to an international task force to battle them [6].

In the next section we report on different approaches to user assistance for spreadsheets. It turns out that there is a UA bias with respect to spreadsheet *authors*. Then we elaborate in Section 3 on our approach for UA targeting spreadsheet readers with interactional semantic technology (by introducing semantic objects, -illustration and -interaction) and showcase in Section 4 SACHS as an exemplary user assistance system for documents that builds on semantic interaction. Here, we revisit SACHS from an Artificial Intelligence perspective onto semantic technology, particularly as it concerns user assistance as part of Human-Computer-Interaction. This is on the one hand in contrast to our previous Knowledge Management point of view taken in [7], where we presented the SACHS system, and in [8], where we introduced new features, and on the other hand to our Semantic technology standpoint used in [9], where we established the notion of "*semantic transparency*". Finally, we explore new research directions in Section 5 and conclude the paper.

2 State of the Art

User assistance facilities reach from informal best practice recommendations (as in [10]) to rigorous equation input forms (e.g. [11]) from which spreadsheets are then generated. To name a few distinct approaches, we start with Abraham and Erwig who introduce structural models for generic spreadsheets that interpret table header cells as types and use this information for type checking (e.g. [12]). Mittermeir and Clermont [13] search for similarities in cell functions and use spatial distances between cells to structure the worksheet. In contrast, Engel and Erwig follow an object-oriented, computational model. They suggest to use "*class sheets*" [14] to derive actual well-formed spreadsheets as their instances. Note that here the real-world usability problems with spreadsheets are tackled by supporting the authoring process of individual spreadsheets.

This line of research is based on Panko's influential report on error states and types in [1] originally published in 1998. If we have a closer look, we realize that he only considers such errors that spreadsheet *authors* introduced like computational errors based on faulty formulae. As a consequence, user assistance for spreadsheets is generally not associated with help for reading spreadsheet documents but for operating or enhancing their players with Software Engineering methods (see e.g. [15,16]).

There are a few exceptions though that aim for supporting spreadsheet readers. For example, Banks and Monday recognize an interpretation issue for spreadsheets in [17]. Support for comprehending specific spreadsheets is addressed in [18] and [19] with data visualization techniques resp. data/formula dependency graphs. A documentation-through-annotation approach is described in [20]. Help systems specifically designed for marked high-impact spreadsheets to overcome the "spreadsheet hell" experience described above have not yet been scientifically addressed – except for our own work with the SACHS system (described in section 4).

Even though we focused on spreadsheets, we can generalize the analysis to documents as the situation for reader support of other high-impact documents is even worse. For instance, *shared* slide collections often loose their 'activeness' completely and text documents offer help for understanding the 'message' only in form of hyperlinks.

Note that for text documents the situation is different if they are embedded in Knowledge Management systems like content management systems. Therefore, applying Knowledge Management techniques to UA for documents seems sensible. On the one hand, help systems need to cope with huge numbers of information fragments to account for the abundance of potential problems of understanding documents, on the other hand a central question for UA consists in the appropriateness of help (e.g. [21]). Thus, we opted for semantic technologies that promise intelligent handling of vast information resources. In the next section we argue for the use of an *interactional* semantic technology approach to meet the appropriateness condition of help.

3 Semantic Interaction

Semantic technologies are technologies that process semantic data, i.e., ordinary data that are extended by explicitly marking up the objects involved and their relations among each other, and which act on these sensibly through inference. They are knowledge management tools for mastering information complexity. In order to use semantic technology for user assistance, we need to explore the interaction with semantic data.

From the perspective of a document user the interface objects that carry meaning are distinguished objects, as the user naturally focuses on them and acts upon them while interacting. These objects are called "**semantic objects**" [9]. For instance, consider the reader of a spreadsheet document. Semantically, spreadsheet documents are functional programs represented as arrays of *spreadsheet variables*, which in turn consist of a *current value* and a *formula* (an arithmetic/logic expression involving spreadsheet variable names) to compute it from the values of other variables. In typical spreadsheet players, spreadsheets can be directly manipulated via the user interface that uses a grid-like arrangement of cells, rows, and columns inspired by the familiar ledger sheets in accounting [22]. Here, obvious semantic objects are tables and cells (points in a two-dimensional table), but co-located cells whose underlying formula is the same are semantic objects too. Note that points-of-pain directly map to semantic objects.

The interaction potential of the document's interface is determined by the user's *understanding* of these semantic objects. Thus, we consider the ability of interface objects to expose their underlying meaning as an important property of user interfaces. A user interface is called "**semantically transparent**" [9], if it enables a user to access its semantic objects and their relations via the corresponding interface objects. For spreadsheets e.g., semantic transparency consists in the tight correspondence of grid rectangles with cells. This correspondence is built into the system by naming the spreadsheet variables by a combination of the column and row identifiers of their corresponding cell. The resulting values are directly presented in the grid, and the formulae are displayed in the formula window — possibly using a color coding scheme to tie cell addresses back to cells as a visual aid. Therefore, the semantics of variables can be accessed by clicking on the corresponding cells, and hence, the semantic objects are transparently accessible to the user. Note that — because semantic objects are acted on by the user — semantic transparency yields ***embedded* user assistance** [23], i.e., user assistance that is provided without having a user to push a help button.

The Semantic Web approach doesn't help for user assistance for documents, as neither the high-impact documents nor their players are explicit on the contained background knowledge. But with a new architecture called "**Semantic Illustration**" [9] semantic technology can be made fruitful nevertheless by complementing existing systems semantically. In particular, instead of enhancing resources into semiformal ontologies by annotating them with formal objects that allow reasoning as in the Semantic Web 'paradigm', here a semantic system *illustrates* a software artifact (an application, program, or document) with a semiformal ontology by complementing it with enough information to render new semantic services. The illustration is based on an interpretation mapping between semantic objects and an at least semiformal domain ontology. The SACHS system, summarized in Section 4, serves as an example for such an architecture. Note that the Semantic Illustration approach opens the 'use of semantic data' for non-semantic software applications: Any system with formal data can be mashed up with semantic applications. As long as an application provides access to its semantic objects, their transparency can be enhanced with this architecture.

Semantic technology is traditionally concerned with the computer side of human-computer interaction. Pointedly spoken, the term 'semantic technology' comprises supportive tools for the generation of semantic data for machine use and respective tools to make machines using these. In particular, semantic technology is only interested in interfaces as it concerns data, e.g. their representation, their presentation and their retrieval. We have already seen that this is not sufficient for user assistance systems, as appropriateness at points-of-pain is hard to assess. For further usefulness of semantic technology, we argue that we need to extend focus from the computer side to the interaction one. This "**interactional semantic technology**" draws on semantic technology, but combines it with reasoning about the formal data within the interface used for interaction.

From a reader's point of view, the document (together with its player) represents the interface for underlying services. For example, let us look at Fig. 3, where we see a person interacting with a spreadsheet. Here, the semantic objects come into play as they are natural access points for interaction, the cell for example. Together with its domain ontology, the spreadsheet document becomes a semantic spreadsheet that allows semantic services (see Fig. 3).

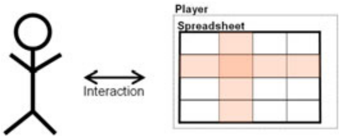


Fig. 1. Interaction with Excel

The Semantic Illustration architecture allows to use the document itself as interface for these services. In particular, the interaction with the semantic spreadsheet is done via its semantic objects, which are connected to the domain ontology via the interpretation mapping. To differentiate the interaction services that follow from this setup from semantic services, we call them "**semantic interaction**". Note that in contrast to semantic services that make the machine smarter, semantic interaction makes the interaction with the smarter machine smarter.

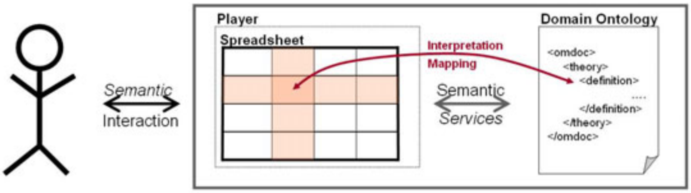


Fig. 2. Semantic Interaction with Excel

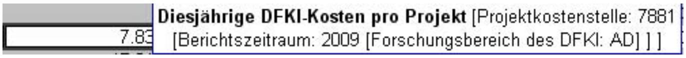
The interactional technology approach was pushed by Dourish in [24]. He suggests to put "*interaction at the center of the picture. [... This approach] considers interaction not only as* what *is being done, but also as* how *it is being done.*" [24, p. 4]. In the field of Artificial Intelligence (AI) we find its traces e.g. among intelligent "embodied interaction" and ubiquitous computing technologies (e.g. [25]). Even though 'interaction' is at the heart of the term "human-computer interaction" used in AI, very often the representational approach for context is at its base. Here, semantic technologies are no exception, in fact, discussions about representation of context are predominantly on 'what can be done'. But there are some exceptional developments. For instance, we find semantic interaction in the *mSpace* interactional interface [26]. There, spatial representation of semantic categories and data is combined with user-defined dimensional sorting of category columns and preview cues with exemplary data for each category. Note that a semantic service provides the spatial context, but the interaction option makes the interaction with this smart machine even smarter.

4 Revisiting SACHS as User Assistance for Documents via Interactional Semantic Technology

In this section we will show our conceptualization of interactional semantic technologies on a concrete example: the SACHS system [7], a user assistance for MS Excel documents. We will pay particular attention to SACHS' reader assistance facilities and how they draw on semantic interaction.

SACHS is a VBA-based extension of Excel that enables the semantic illustration of spreadsheets, we have based our development on "**DCS**" - an Excel-based financial controlling system in daily use at DFKI. SACHS assumes that a spreadsheet is illustrated with a semiformal ontology of the background knowledge in the OMDoc format [27] together with an interpretation mapping that aligns cells with concepts in the respective domain ontology. For DCS we authored the former, the latter was semi-automatically generated. SACHS uses the formal parts of the ontology to control the aggregation of help texts (from the informal part of the ontology) about the objects in the spreadsheet; for details we refer to [7].

UA systems which have the spreadsheet author in mind, mainly consider formulae as the main semantic objects. In contrast, for the development of the SACHS system we started out with an analysis of DCS usage from a document *reader*'s micro-perspective. From this point of view cells are on the one hand understood via their values, on the other hand they are consumed via their location within a grid structure. At first glance former seems to be explicit (e.g. via formulae), the latter implicit knowledge (e.g. cognitive grid recognition). But interestingly, even the former interpretation is not as transparent as it seems, since formulae typically contain other cell addresses. If for example, domain knowledge about such underlying cells is missing, then the transparency dissolves even though the first step was possibly transparent. Therefore, we diagnosed the complexity of understanding Excel documents as an implication of the *incomplete semantic transparency of cells*. In turn, SACHS aims at making implicit parts of this cell-dependant domain knowledge explicit. Importantly, this information is not only user-accessible just somewhere, but through Excel's usual mechanism for manipulation of cells, so that the semantic transparency of cells is considerably strengthened. We realized *embeddedness by using cell clicks* as entry points for the help system, so that every click on a cell generates help. Note that the 'aptness' of help is provided by the mash-up of document interface and standard semantic technology, that is by semantic interaction.



7.83

Diesjährige DFKI-Kosten pro Projekt [Projektkostenstelle: 7881
[Berichtszeitraum: 2009 [Forschungsbereich des DFKI: AD]]]

Fig. 3. A SACHS Label

Concretely, the UA comes in various forms, depending on what the user is interested in at that point in time. We comprise its functionality only very concisely and refer to [8,7] for more information. The semantic service of delivering on-the-fly, context-dependant **help texts** is provided on various granularity levels (ranging from short labels to deep explanations). Labels are recursively extended by *nterspersed domain and value information* – which becomes the more helpful for orientation in the worksheet the larger the grid – see a DCS example in Fig. 4.

SACHS also offers the presentation of **dependency graphs** for a cell (interpreted as a concept in the domain knowledge), see an example in Fig. 4. They serve domain comprehension. In particular, the graph is developed level-by-level: the user of Fig. 4 e.g. already clicked on the node "Salary Costs of SemAnteX" but not on the other ones on the second and third level.

A main feature of SACHS that draws on interactional semantic technology is called **semantic navigation** and is situated with the dependency graph. The selection of marked nodes in the graph allows the navigation in the workbook (and theoretically beyond) whereever concepts are aligned with cells. Especially for complex

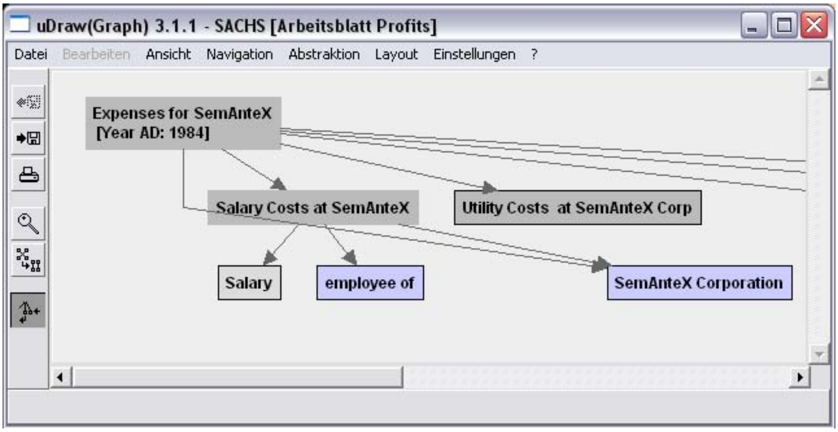


Fig. 4. Dependency Graph of a Cell's Concept

spreadsheets with data distributed over several worksheets, this is very convenient and enhances interaction tremendously. In our examplaric graph the "Salary" node functions as hyperlink to the worksheet where the salary data come from.

Two other more elaborated new interactions are implemented in SACHS. The first draws on exploiting theory morphisms affording **to frame content**, i.e., to customize perspectives on content. In a nutshell, SACHS enables the user to choose from a palette of frames for a cell (and thereby a topic) in question. If a user is interested in the meaning of the value in a cell e.g. as a result of a concrete function, then she likes to know what function was applied, whereas if she already knows the name of the function, but doesn't know what it is about, then she likes to see the definition of this concrete function. She might even be wondering, what the superfluent concept of the function is. Note that the use of content frames enables *content variants*, another form of semantic interaction. As framing offers various access points for understanding the meaning of cells, this semantic interaction individualizes semantic transparency. Additionally, framing enables another innovative interaction, namely the **presentation of formula variants**, i.e., suggesting and applying related formulae for a playful evaluation of cell values for a given cell.

Unfortunately, we cannot yet describe evaluation results for SACHS. As with all semantic technologies SACHS' cost-benefit ratio is hard to assess as the real value only appears rather late in the process. We have prepared a questionnaire to get first usability feedback. To understand the real added-value for document readers we also like to make use of the RGT (Repertory Grid Technique), which explores the user perspective by researching users' *personal* constructs employed when interacting e.g. with a software artifact, here a specific spreadsheet document.

5 Conclusion and Future Work

We motivated and addressed the question of user assistance for documents. Our approach builds on semantic objects as entry door for meaningful interaction, that can be aligned with semantic data via an interpretation mapping based on the Semantic Illustration architecture. Note that semantic interactions are not just interactions based on semantic technology. They are induced by the mash-up of a software artifact and a semantic system. We showcased the SACHS system as an example for user assistance for documents with innovative functionalities like “semantic navigation” based on interactional semantic technology.

Another line of research in the SACHS context is concerned with the kinds of knowledge necessary to be captured into semantic formats to assist the user in understanding a specific spreadsheet. In [28] we evaluated the coverage of the SACHS system with a “Wizard of Oz” experiment. It resulted in the cognition that while SACHS fares much better than DCS alone, it systematically misses important classes of explanations. Even though SACHS certainly offers an improvement, it leaves much more to be desired than we anticipated and room for future work for SACHS specifically as well as for user assistance for documents generally.

Acknowledgements

The work reported here was supported by the FormalSafe project conducted by DFKI Bremen and funded by the German Federal Ministry of Education and Research (FKZ 01IW07002) and an internal grant of the German Center for Artificial Intelligence. I like to thank Michael Kohlhase and Dieter Hutter for valuable discussions.

References

1. Panko, R.R.: What we know about spreadsheet errors. Journal of End User Computing's Special issue on Scaling Up End User Development 10(2), 15–21 (2008); Published 1998 (revised May 2008)
2. Panko, R.R.: Spreadsheet errors: What we know. what we think we can do. In: Symp. of the European Spreadsheet Risks Interest Group, EuSpRIG 2000 (2000)
3. Powell, S.G., Baker, K.R., Lawson, B.: Impact of errors in operational spreadsheets. CoRR abs/0801.0715 (2008)
4. Murphy, S.: Spreadsheet hell. CoRR abs/0801.3118 (2008)
5. Powell, S.G., Baker, K.R., Lawson, B.: A critical review of the literature on spreadsheet errors. Decision Support Systems 46(1), 128–138 (2008)
6. EUSPRIG: European spreadsheet risks interest group (2010), http://www.eusprig.org
7. Kohlhase, A., Kohlhase, M.: Compensating the computational bias of spreadsheets with MKM techniques. In: Carette, J., Dixon, L., Sacerdoti Coen, C., Watt, S.M. (eds.) Calculemus 2009. LNCS, vol. 5625, pp. 357–372. Springer, Heidelberg (2009)
8. Kohlhase, A., Kohlhase, M.: Spreadsheet interaction with frames: Exploring a mathematical practice. In: Carette, J., Dixon, L., Sacerdoti Coen, C., Watt, S.M. (eds.) Calculemus 2009. LNCS, vol. 5625, pp. 341–356. Springer, Heidelberg (2009)
9. Kohlhase, A., Kohlhase, M.: Semantic transparency in user assistance systems. In: Mehlenbacher, B., Protopsaltis, A., Williams, A., Slatterey, S. (eds.) Proceedings of the 27th Annual ACM International Conference on Design of Communication, ACM Special Interest Group for Design of Communication, pp. 89–96. ACM Press, New York (2009)

10. Barnes, J.N., Tufte, D., Christensen, D.: Spreadsheet design: An optimal checklist for accountants. In: Ninth Annual IBER and TLC Conference Proceedings 2009, pp. 1–16 (2009)
11. Paine, J.: Exelsior: Bringing the benefit of modularisation to excel. In: Symp. of the European Spreadsheet Risks Interest Group, EuSpRIG 2005 (2005)
12. Abraham, R., Erwig, M.: Mutation operators for spreadsheets. IEEE Transactions on Software Engineering 35(1), 94–108 (2009)
13. Mittermeir, R., Clermont, M.: Finding high-level structures in spreadsheet programs. In: WCRE 2002: Proceedings of the Ninth Working Conference on Reverse Engineering (WCRE 2002), Washington, DC, USA, p. 221. IEEE Computer Society, Los Alamitos (2002)
14. Engels, G., Erwig, M.: ClassSheets: Automatic generation of spreadsheet applications from object oriented specifications. In: 20th IEEE/ACM International Conference on Automated Software Engineering, pp. 124–155. IEEE Computer Society, Los Alamitos (2005)
15. Grossman, T.A., Ozluk, O.: A paradigm for spreadsheet engineering methodologies. CoRR abs/0802.3919 (2008)
16. Kreie, J., Cronan, T.P., Pendley, J., Renwick, J.S.: Applications development by end-users: can quality be improved? Decision Support Systems 29(2), 143–152 (2000)
17. Banks, D.A., Monday, A.: Interpretation as a factor in understanding flawed spreadsheets. CoRR abs/0801.1856 (2008)
18. Brath, R., Peters, M.: Spreadsheet validation and analysis through content visualization. CoRR abs/0803.0166 (2008)
19. Hodnigg, K., Mittermeir, R.T.: Metrics-based spreadsheet visualization: Support for focused maintenance. CoRR abs/0809.3009 (2008)
20. Dinmore, M.: Documenting problem-solving knowledge: Proposed annotation design guidelines and their application to spreadsheet tools. CoRR abs/0908.1192 (2009)
21. Novick, D.G., Ward, K.: What users say they want in documentation. In: SIGDOC 2006 Conference Proceedings, pp. 84–91. ACM, New York (2006)
22. Winograd, T.: The spreadsheet. In: Winograd, T., Bennett, J., de Young, L., Hartfield, B. (eds.) Bringing Design to Software, pp. 228–231. Addison-Wesley, Reading (1996/2006)
23. Ellison, M.: Embedded user assistance: The future for software help? Interactions 14(1), 30–31 (2007)
24. Dourish, P.: Where the Action Is: The Foundations of Embodied Interaction. The MIT Press, Cambridge (2003)
25. Malaka, R., Porzel, R.: Design principles for embodied interaction. In: Mertsching, B., Hund, M., Aziz, Z. (eds.) KI 2009. LNCS, vol. 5803, pp. 711–718. Springer, Heidelberg (2009)
26. Schraefel, M., Smith, D.A., Owens, A., Russell, A., Harris, C., Wilson, M.: The evolving mspace platform: Leveraging the semantic web o n the trail of the memex. In: HT 2005, Salzburg, Austria. ACM, New York (September 2005)
27. Kohlhase, M.: OMDoc – An open markup format for mathematical documents (Version 1.2). LNCS (LNAI), vol. 4180. Springer, Heidelberg (August 2006)
28. Kohlhase, A., Kohlhase, M.: What we understand is we get: Assessment in spreadsheets. In: Symp. of the European Spreadsheet Risks Interest Group, EuSpRIG 2010 (2010) (in Press)

Flexible Concept-Based Argumentation in Dynamic Scenes

Jörn Sprado, Björn Gottfried, and Otthein Herzog

Centre for Computing and Communication Technologies
Universität Bremen, Am Fallturm 1, 28359 Bremen, Germany

Abstract. Argumentation systems can be employed for detecting spatiotemporal patterns. While the idea of argumentation consists in defending specific positions, complex patterns are influenced by several factors that can be regarded as arguments against or in favor of the realisation of those patterns. The idea is to determine consistent positions of arguments which speak for specific patterns. This becomes possible by means of algorithms which have been defined for argumentation systems. The introduced method of conceptual argumentation is new in comparison to classical, i.e. value-based, argumentation systems. It has the advantage to be more flexible by enabling the definition of conceptual arguments influencing relevant patterns. There are two main results: first, conceptual argumentation frameworks do scale significantly better; secondly, investigating our approach by examining soccer games, we show that specific patterns, such as passes, can be detected with different retrieval performances depending on the chosen spatial granularity level.

1 Introduction

Over the last years there has been an increasing interest in the fields of Decision Support Systems and Artificial Intelligence (AI) to investigate argumentation approaches. As a promising model for reasoning about inconsistent knowledge [1,4] we investigate argumentation frameworks to examine the behaviours of objects, i.e. we are looking for how to provide decision support in the context of spatiotemporal systems. An example in the soccer domain looks like this: from one point of view the behaviours of soccer player might be inconsistent since not all behaviours support a given strategy; however, another point of view might argue for another strategy for which the behaviours are consistent; and yet another view would state that the players were not able to get to a common strategy – it is then the decision of the coach which player to censure in which way and with which kinds of arguments. Argumentation frameworks allow specific conclusions to be derived about what is true or rather forms a consistent argumentation. In our case in order to evaluate spatiotemporal group interactions we wish to identify them by an argumentation system.

The most fundamental group interaction in soccer is to pass the ball between players. This closely links to the individual action of dribbling the ball. We shall therefore focus on these basic actions in order to demonstrate our approach.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 116–125, 2010.

For a complete game of the RoboCup simulation league championships 2007 it is shown how passes can be detected by means of a conceptual argumentation framework representing spatiotemporal knowledge about soccer.

1.1 System Overview

The whole analysis process of spatiotemporal data by means of argumentation frameworks is outlined in this paragraph. Analysing soccer games input of the whole process are raw positional data and the output is an explanation component for the input data. The input data represents at the most basic level spatiotemporal object interactions. Conversely, an explanation of such object interactions is provided by the explanation component. An explanation describes what is going on at an abstract semantic level, while the raw positional input data are only positional measurements. A generated explanation might be useful in particular for making decisions. For this purpose basically two methods are employed:

- *argumentation frameworks* are used in order to describe which concepts form consistent scenarios, and conversely, which concepts cannot be reconciled [7];
- the terminology of both a specific domain and of argumentation frameworks is defined by methods of *description logics* [2].

The process of explanation generation, and hence decision support, is thus a process of looking for consistent sets of arguments. Domain specific arguments are defined by an ontology, and the more abstract level of argumentation frameworks is itself described by another ontology. Before interpreting data, a priori knowledge is modelled by means of these ontologies. Taking the raw positional data, arguments about basic spatiotemporal behaviour patterns are constructed. Then attack-relations among arguments are constructed that define which arguments attack other arguments. Referring to terminological knowledge which is defined a priori, semantic arguments are constructed; they describe at the semantic level how concepts characterise the data.

The body of this paper is structured as follows: Sect. 2 shows the use of description logics in order to formalise domain knowledge. Sect. 3 introduces concept-based argumentation frameworks as an extension to value-based approaches. Experimental results in Sect. 4 indicate how the conceptual argumentation framework enables the search for spatiotemporal patterns. Conclusions are drawn in Sect. 5.

2 Semantic Description of Dialectic Structures

As a groundwork we use ontologies to provide a formal representation for the elements of argumentation systems. A popular definition of ontologies in AI is proposed by Gruber [10].

> An ontology is an explicit formal specification of a shared conceptualisation.

Such a conceptualisation corresponds to a way of thinking about a domain [16]. The idea of a semantic description of the argumentation structures is to represent their meaning explicitly. Arguments and conflicts with clear semantics become machine-interpretable and further inference mechanisms can be applied (cf. large-scale argumentation [11,12]).

The ontologies shown in this paper are expressed by using a description logic (DL). Description logics are part of the family of knowledge representation languages that are subsets of first-order logic (cf. [13]). A DL is a concept-based knowledge representation formalism with well-defined model-theoretic semantics which consists of atomic concepts (unary predicates), atomic roles (binary predicates), and individuals (constants). The expressive power of DL languages is restricted to a small set of constructors for building complex concepts and roles. Implicit knowledge about concepts and individuals can be inferred automatically through inference procedures. For precise formal semantics of DLs see [2]. A description logic knowledge base is naturally separated into two parts: a TBox containing intensional knowledge that describes general properties of concepts and an ABox containing extensional knowledge that is specific to the individuals of the universe of discourse.

Handling complex patterns requires to bridge the gap between pure measurements and meaningful explanations of those measurements. We describe a conceptual argumentation process in the soccer domain which determines consistent positions of pass patterns from the raw positional data, entailing the flexible interpretation of spatiotemporal measurements by referring to domain knowledge. Some concepts of the running example are defined as follows:

$$\begin{aligned}
\mathsf{Argument} &\equiv \top \ \sqcap (= 1\ \mathsf{hasSourceArgument}) \sqcap (= 1\ \mathsf{hasTargetArgument}) \\
\mathsf{Spatial\text{-}Argument} &\equiv \mathsf{Argument} \ \sqcap (\geqslant 1\ \mathsf{tookPlaceAt}) \\
\mathsf{Action\text{-}Argument} &\equiv \mathsf{Spatial\text{-}Argument} \\
&\quad \sqcap \exists \mathsf{hasAnalysisLevel}.(\forall \mathsf{hasAnalysisLevel}.\mathsf{Action\text{-}Level}) \\
\mathsf{Pass} &\equiv \mathsf{Action} \sqcap (= 1\ \mathsf{hasInitialPlayer}) \sqcap (= 1\ \mathsf{hasTargetPlayer}) \\
\mathsf{Pass\text{-}Argument} &\equiv \mathsf{Action\text{-}Argument} \sqcap \exists \mathsf{hasConclusion}.\mathsf{Pass}
\end{aligned}$$

Several ontologies are used to represent conceptual descriptions of this example. A first one describes argumentation structures in general (e.g. concepts like Spatial-Argument) whilst a second one defines concepts and roles of the concrete domain (e.g. a Pass-Argument). A conflict between two arguments can either be defined as a role or concept. The latter enables reasoning mechanisms, in particular classification, on these structures. They have in common that conflicts are automatically inherited by all subconcepts. Fig. 1 gives an overview of relevant concepts.

3 Concept-Based Argumentation

Our running example involves multiple arguments for and against different claims. In this section we generally describe how to reconcile these conflicts with a *concept-based argumentation framework*:

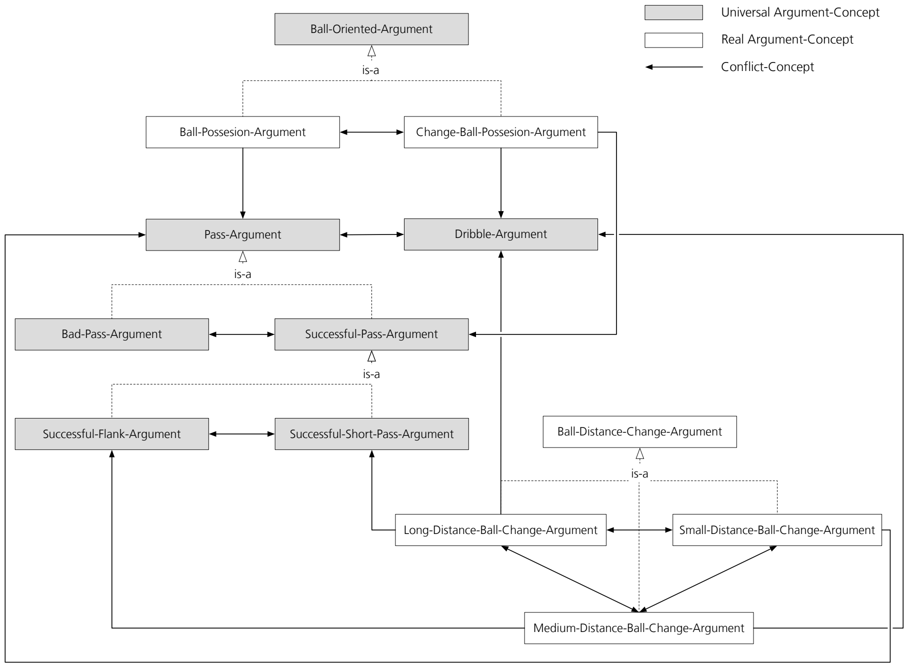

Universal Argument-Concept
Real Argument-Concept
Conflict-Concept
Ball-Oriented-Argument
is-a
Ball-Possesion-Argument
Change-Ball-Possesion-Argument
Pass-Argument
Dribble-Argument
is-a
Bad-Pass-Argument
Successful-Pass-Argument
is-a
Successful-Flank-Argument
Successful-Short-Pass-Argument
Ball-Distance-Change-Argument
is-a
Long-Distance-Ball-Change-Argument
Small-Distance-Ball-Change-Argument
Medium-Distance-Ball-Change-Argument

Fig. 1. Concepts of domain arguments and conflicts of the running example

Definition 1 (Concept-Based AF). *A concept-based argumentation framework (CAF) is a triple*

$$\Phi = \langle \mathscr{KB}(\mathcal{T}, \mathcal{A}), \mathcal{AR}_\Phi, \mathcal{KO}_\Phi \rangle$$

where $\mathscr{KB}$ is a DL knowledge base, consisting of terminological (TBox) and assertional knowledge (ABox), which represents the domain of interest and $\mathcal{AR}_\Phi$ is a set of argument concepts and $\mathcal{KO}_\Phi$ is a set of conflict concepts.

A concept-based argumentation framework is a specialisation of a VAF[1] [3] which associates concepts instead of simple values with arguments. That is, to determine the acceptability of arguments we can refer to VAFs. But a main difference is that semantic arguments are mapped to concepts automatically as a result of inference procedures. For instance, if we change the semantic description of an argument, an appropriate mapping within the argumentation system will be automatically adjusted. Thus, argumentation becomes more scalable.

Based on the proposed framework it should be possible to find an answer whether an argument or a collection of arguments fulfill the requirements of given formal definitions on verification, credulous acceptance or skeptical acceptance [8]. Solving argumentation problems is often computationally complex, e.g. determining the credulous acceptance of preferred extensions falls into

[1] VAF = Value-based argumentation framework.

the class of NP problems [6]. Preferred extensions are of main interest because they form sets of arguments where each incoming conflict is defeated. Employing a given conceptual model in a CAF some of these questions could already be accommodated at an early stage by displacing corresponding processes on the conceptual level. For example, instances of Successful-Pass-Argument, as a specialisation of Pass-Argument which is disjunct to and also in conflict with Dribble-Argument, never form admissible collections of arguments with the concept Dribble-Argument.

For a comprehensive examination in absence of specific instances, all combinations of conflicts have to be involved in the argumentation analysis. This leads generally to an exponential number of cases — however, by means of the conceptual model many of these cases can be rejected. The results of each argumentation process are accumulated in decision structures where extensions are represented as nodes and information on the existence of conflicts as edges. Thus, argumentation is substituted by instance checking.

4 Interpretation of Actions in the Soccer Domain

In our running example we distinguish between *real* and *universal* concepts (cf. Fig. 1). These two kinds of argument concepts form a template-based extension of a CAF. A real argument concept refers to structural path by so-called *semantic mappings* [5]. It describes how instances could be obtained from existing data. In the soccer domain such concepts are commonly used to describe classes of arguments for simple behaviours, e.g. a specific distance change. By contrast, universal ones could be used to represent argument concepts on a more sophisticated level (e.g. an offside-trap position).

When using a qualitative reasoning approach underlying discrete interval definitions would be subjective. In order to ensure the quality of the argumentation results, it is important to choose these intervals in the context of the application carefully. In the following we compare a number of intervals for identifying pass actions using information about distances. For this purpose the queries $\sigma_1, \ldots, \sigma_4$ are applied to obtain instances from raw positional data (cf. Fig. 4). Each structural query corresponds to a conceptual description implied by a parameter. Due to existing management of data, a query template is represented by SQL-statements which contain diverse variables defining placeholders for game identifier (#DBNAME), temporal information (#TIME) and interval definitions (cf. Table 4).

Fig. 2 depicts a specific conceptual argumentation process on instance level for two scenes of the RoboCup 2D-game TsinghuAeolus vs. FC Portugal where a pass action could be derived unambiguously. Fig. 4 shows the overall results of argumentation processes using the discrete interval specifications $qd_1 - qd_4$. A precision of 1 is never achieved. This, as well as all other results, is influenced by several reasons: the chosen arguments and conflict concepts, the employed semantic mappings, the chosen spatial and temporal granularity levels, different modes of acceptances, and eventually, the given data.

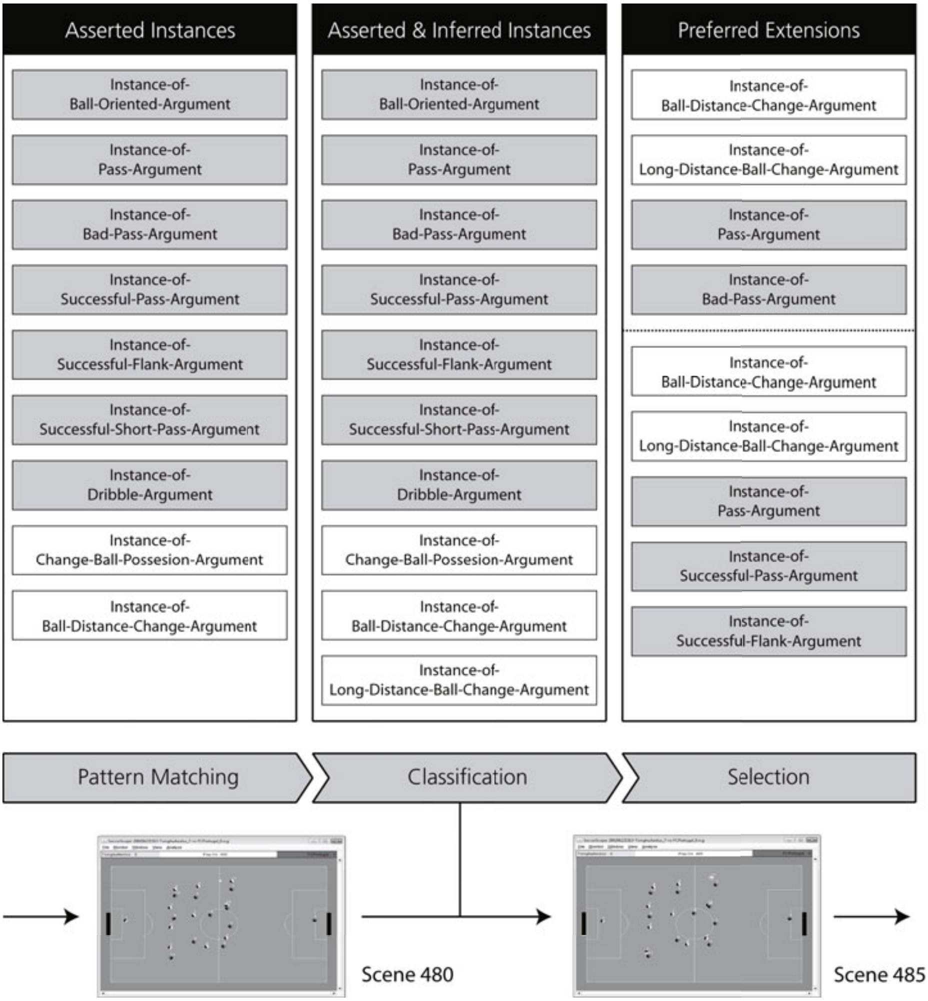
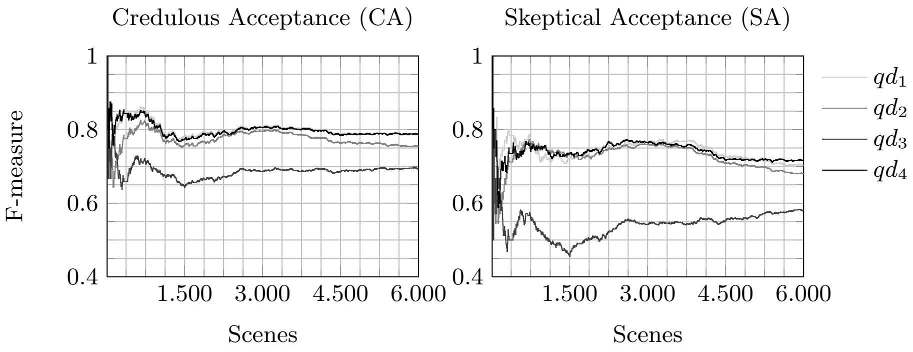

Asserted Instances
Instance-of-Ball-Oriented-Argument
Instance-of-Pass-Argument
Instance-of-Bad-Pass-Argument
Instance-of-Successful-Pass-Argument
Instance-of-Successful-Flank-Argument
Instance-of-Successful-Short-Pass-Argument
Instance-of-Dribble-Argument
Instance-of-Change-Ball-Possesion-Argument
Instance-of-Ball-Distance-Change-Argument
Asserted & Inferred Instances
Instance-of-Ball-Oriented-Argument
Instance-of-Pass-Argument
Instance-of-Bad-Pass-Argument
Instance-of-Successful-Pass-Argument
Instance-of-Successful-Flank-Argument
Instance-of-Successful-Short-Pass-Argument
Instance-of-Dribble-Argument
Instance-of-Change-Ball-Possesion-Argument
Instance-of-Ball-Distance-Change-Argument
Instance-of-Long-Distance-Ball-Change-Argument
Preferred Extensions
Instance-of-Ball-Distance-Change-Argument
Instance-of-Long-Distance-Ball-Change-Argument
Instance-of-Pass-Argument
Instance-of-Bad-Pass-Argument
Instance-of-Ball-Distance-Change-Argument
Instance-of-Long-Distance-Ball-Change-Argument
Instance-of-Pass-Argument
Instance-of-Successful-Pass-Argument
Instance-of-Successful-Flank-Argument
Pattern Matching
Classification
Selection
Scene 480
Scene 485

Fig. 2. Two preferred extensions as a result of a conceptual argumentation process on the instance level for an argumentation system using the concepts from Figure 1 combined with the proposed semantic mappings in Figure 4 and the interval specification qd_4

In [15] we have shown that a specific temporal granularity level of 5 performs best, which we have taken in the current experiment. In particular, we investigated the influence of the selected spatial granularity, for a specific model and specific conflicts; no clear preference can be derived from the results. But, having compared two different modes of acceptances, it shows that different results are obtained, namely for the credulous (CA) and for the skeptical analysis (SA): the precision of SA is significantly higher as for CA; this is because less many irrelevant results will be found by SA than by CA. On the other hand,



	Precision	**Recall**	**Fallout**	**Accuracy**	**F-measure**
$qd_1 - CA$	0.72	0.86	0.50	0.72	0.78
$qd_1 - SA$	0.90	0.58	0.06	0.77	0.70
$qd_2 - CA$	0.67	0.87	0.53	0.69	0.75
$qd_2 - SA$	0.89	0.55	0.08	0.72	0.68
$qd_3 - CA$	0.61	0.81	0.61	0.61	0.69
$qd_3 - SA$	0.74	0.47	0.19	0.63	0.57
$qd_4 - CA$	0.73	0.85	0.50	0.71	0.79
$qd_4 - SA$	0.85	0.61	0.11	0.75	0.72

Fig. 3. Results of argumentation processes to identify pass actions within the RoboCup 2D-game TsinghuAeolus vs. FC Portugal using interval definitions from Table 4

the recall for SA is significantly lower than for CA; this is because more relevant results will be found when just stipulating to have one single extension in the case of CA. However, the f-measure does not show a clear tendency, because it simply combines precision and recall; it balances the reverse results for CA and SA. In other words, using the skeptical argumentation method we aim at excluding other, contrariwise arguments. By contrast, CA is just showing that there exists one set of admissible arguments which does not attack the intended argument.

Identifying preferred extensions (specific patterns) based on the specified argumentation model and the RoboCup 2D-simulation league game TsinghuAeolus vs. FC Portugal, two different algorithms are compared. On the one hand, the logical-program ASPARTIX [9] is used to represent the classical argumentation approach. On the other hand, a prototypical JAVA-implementation of the conceptual approach is applied which deploys the OWL-reasoner PELLET [14] for instance checking. Fig. 5 shows average time periods of 10 cycles for the detection of preferred extensions when increasing the number of argument instances arbitrarily. While the time effort for instance checking on a $\mathcal{SHIQ}$-knowledge base is 2.386 ms, it is 359.526 ms for the logical system.

Identifier	qd_1	qd_2	qd_3	qd_4
#zero	$[0.0, 0.1]$	$[0.0, 0.2]$	$[0.0, 0.1]$	$[0.0, 0.4]$
#very-close	$[0.1, 1.0]$	$[0.2.1.5]$	$[0.1, 3.0]$	$[0.4, 1.0]$
#close	$[1.0, 2.0]$	$[1.5, 2.5]$	$[3.0, 5.0]$	$[1.0, 3.5]$
#medium	$[2.0, 4.0]$	$[2.5, 5.0]$	$[5.0, 10.0]$	$[3.5, 7.0]$
#far	$[4.0, 8.0]$	$[5.0, 10.0]$	$[10.0, 15.0]$	$[7.0, 12.0]$
#very-far	$[8.0, \infty]$	$[10.0, \infty]$	$[15.0, \infty]$	$[12.0, \infty]$

σ_1(Ball-Possesion-Argument ⊓ Distance-Argument):

```
SELECT tab.objectName1 FROM #DBNAME.distances AS tab
WHERE tab.t=#TIME AND NOT tab.objectName1='ball'
AND tab.objectName2='ball' AND distance='#very-close'
```

σ_2(Change-Ball-Possesion-Argument ⊓ Distance-Argument):

```
SELECT tab1.objectName1, tab2.objectName1
FROM ((SELECT * FROM #DBNAME.distances WHERE t=#TIME) AS tab1
INNER JOIN #DBNAME.distances AS tab2 ON (tab1.t=tab2.t)
AND NOT (tab1.objectName1= tab2.objectName1)
AND NOT tab1.objectName1='ball' AND NOT tab2.objectName1='ball'
AND tab1.objectName2='ball' AND tab2.objectName2='ball'
AND tab1.distance='#very_close' AND tab2.distance='#very_close')
```

σ_3(Long-Distance-Ball-Change-Argument ⊓ Distance-Argument):

```
SELECT tab.objectName1 FROM #DBNAME.distances AS tab
WHERE tab.t=#TIME AND tab.objectName1='ball'
AND tab.objectName2='ball' AND distance='#far'
```

σ_4(Medium-Distance-Ball-Change-Argument ⊓ Distance-Argument):

```
SELECT tab.objectName1 FROM #DBNAME.distances AS tab
WHERE tab.t=#TIME AND tab.objectName1='ball'
AND tab.objectName2='ball'
AND (distance='#very-close' OR distance='#close')
```

σ_5(Small-Distance-Ball-Change-Argument ⊓ Distance-Argument):

```
SELECT tab.objectName1 FROM #DBNAME.distances AS tab
WHERE tab.t=#TIME AND tab.objectName1='ball'
AND tab.objectName2='ball' AND distance='#zero'
```

Fig. 4. Different queries and discrete interval specifications [simulation-units]

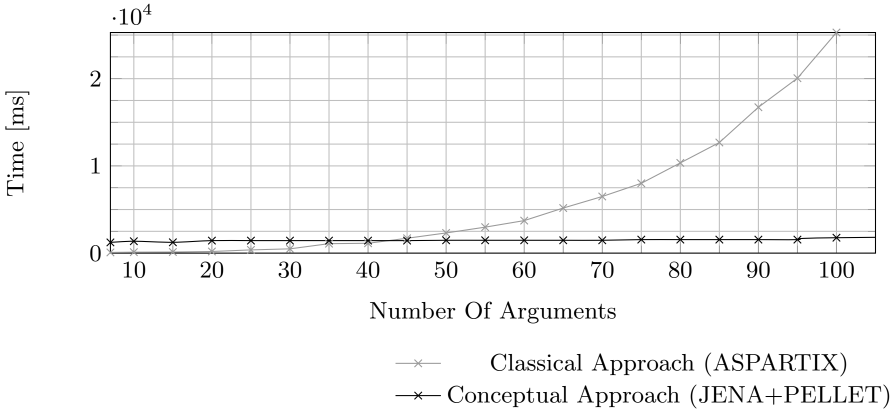


Fig. 5. Computations of preferred extensions scale differently for the classic in contrast to our approach

5 Conclusions

We have presented an approach based on two paradigms: that of argumentation frameworks and that of description logics. The former is employed for analysing consistent sets of arguments, given a set of instantiated arguments at some given time – consistent sets supporting patterns to be detected. The latter is primarily used for defining terminological knowledge in order to characterise arguments at the semantic level; that is to say that instead of value-based systems, concept-based arguments are introduced. Concept-based argumentations are more flexible in that appropriate mappings within an argumentation system can be automatically adjusted if we would change the semantic description of arguments. Moreover, the proposed basics enables to define preferences among arguments at a conceptual level instead of taking simple values, as is the case in preference-based argumentation framework. In a nutshell, there are two advantages we have shown in this work: first, the flexibility of dealing with conceptual knowledge for interpreting measurements; secondly, the performance scaling significantly better for sets of large arguments than the conventional approach.

References

1. Amgoud, L., Cayrol, C., Lagasquie, M.C., Livet, P.: On bipolarity in argumentation frameworks. In International Journal of Intelligent Systems 23(10), 1062–1093 (2008)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge (2003)
3. Bench-Capon, T.J.M.: Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation 13(3), 429–448 (2003)

4. Bench-Capon, T.J.M., Dunne, P.E.: Argumentation in artificial intelligence. Artificial Intelligence 171(10-15), 619–641 (2007)
5. Bowers, S., Lin, K., Ludäscher, B.: On integrating scientific resources through semantic registration. In: 16th International Conference on Scientific and Statistical Database Management (SSDBM), Santorini Island, pp. 21–23 (2004)
6. Dimopoulos, Y., Torres, A.: Graph theoretical structures in logic programs and default theories. Theoretical Computer Science 170(1-2), 209–244 (1996)
7. Dung, P.M.: On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence 77(2), 321–358 (1995)
8. Dunne, P.E., Wooldridge, M.: Complexity of Abstract Argumentation. In: Simari, G., Rahwan, I. (eds.), pp. 85–104 (2009)
9. Egly, U., Gaggl, S.A., Woltran, S.: Aspartix: Implementing argumentation frameworks using answer-set programming. In: Garcia de la Banda, M., Pontelli, E. (eds.) ICLP 2008. LNCS, vol. 5366, pp. 734–738. Springer, Heidelberg (2008)
10. Gruber, T.R.: Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies 43, 907–928 (1995)
11. Rahwan, I.: Mass argumentation and the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web 6(1), 29–37 (2008)
12. Rahwan, I., Banihashemi, B.: Arguments in owl: A progress report. In: COMMA. Frontiers in Artificial Intelligence and Applications, vol. 172, pp. 297–310. IOS Press, Amsterdam (2008)
13. Sattler, U., Calvanese, D., Molitor, R.: Relationship with other formalisms. In: The Description Logic Handbook: Theory, Implementation and Applications, pp. 137–177. CUP, Cambridge (2003)
14. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical owl-dl reasoner. Journal of Web Semantics 5(2), 51–53 (2007)
15. Sprado, J., Gottfried, B.: What motion patterns tell us about soccer teams. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) RoboCup 2008: Robot Soccer World Cup XII. LNCS, vol. 5399, pp. 614–625. Springer, Heidelberg (2009)
16. Uschold, M.: Knowledge level modelling: concepts and terminology. The Knowledge Engineering Review 13(1), 5–29 (1998)

Focused Belief Revision as a Model of Fallible Relevance-Sensitive Perception

Haythem O. Ismail and Nasr Kasrin

German University in Cairo
Department of Computer Science
{haythem.ismail,nasr.kasrin}@guc.edu.eg

Abstract. We present a framework for incorporating perception-induced beliefs into the knowledge base of a rational agent. Normally, the agent accepts the propositional content of perception and other propositions that follow from it. Given the fallibility of perception, this may result in contradictory beliefs. Hence, we model high-level perception as belief revision. Adopting a classical AGM-style belief revision operator is problematic, since it implies that, as a result of perception, the agent will come to believe everything that follows from its new set of beliefs. We overcome this difficulty in two ways. First, we adopt a belief revision operator based on relevance logic, thus limiting the derived beliefs to those that relevantly follow from the new percept. Second, we focus belief revision on only a subset of the agent's set of beliefs—those that we take to be within the agent's current focus of attention.

1 Introduction

Agent X is about to cross the street. It turns left, and it sees a car approaching fast. X immediately stops and decides to wait until the car had passed and then retry. X's decision to stop is clearly based on reasoning with background practical knowledge. If not for this knowledge, and this reasoning, X would have been in trouble. Now consider agent Y.

Y is about to cross the street. It turns left, and it sees a car approaching fast. Trusting in the veridicality of its percept, Y comes to believe that a car is approaching fast. Unlike X, however, Y never stops. Y is certainly in trouble, but is it to blame? If Y does not have the same practical knowledge that saved X, then whatever it did, though a sign of ignorance, is rational. However, if Y did have the valuable knowledge, but did not use it to derive useful conclusions, then it certainly is to blame. Finally, consider agent Z.

Z is about to cross the street. It turns left, and it sees a car approaching fast. Z immediately stops and decides to wait until the car had passed and then retry. Z stops walking, but it does not stop reasoning about the approaching car. Assuming it has common knowledge of automobiles, Z will infer that *this* car has a steering wheel, an accelerator, and a battery, and that, since it is moving, it must be under the control of a driver. Moreover, it will also infer that the driver is a person, and that, like all people, the driver probably has two hands

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 126–134, 2010.

and two feet, with five digits in each. Z never stops reasoning, and never crosses the street.

Evidently, perception involves some element of reflection on what is perceived. Neither no reflection nor unbounded reflection are appropriate, as the cases of Y and Z attest. Let us refer to this kind of perception-induced reasoning, or reflection, as "high-level perception". In this paper, we present a framework for high-level perception within a grounded, layered agent architecture [1,2]. In this architecture, perception results in the agent's acquiring a new belief of the form "I now perceive s through m", where s is a perceivable state and m is one of the agent's perceptual modalities. As defined above, high-level perception is not the mere addition of such a belief to the agent's belief store; normally, the agent will also come to believe that s and other states (that follow from it) hold. But this might result in the agent's holding contradictory beliefs. Hence, we model high-level perception as belief revision, which is compatible with belief update in our temporal language. Adopting a classical AGM-style belief revision operator satisfying deductive closure is problematic [3], since it implies that, as a result of perception, the agent will come to believe everything that follows from its new set of beliefs. We overcome this difficulty in two ways. First, we adopt a belief revision operator based on relevance logic [4], thus limiting the derived beliefs to those that relevantly follow from the new percept. Second, we focus belief revision on only a subset of the agent's set of beliefs—those that we take to be within the agent's current focus of attention. This allows us to avoid Z-like agents.

Related work on knowledge representation aspects of perception is rather slim. Most of this work presents multi-modal logics of the interactions between perception and belief [5,6,7]. Wooldridge and Lumoscio describe a multi-modal logic $\mathcal{VSK}$ of visibility, perception, and knowledge [8]. The semantics is based on automata-like models of agents and environments rather than possible worlds. All these systems, however, have nothing to say about the issue of high-level perception as we described it above. They also have little to say about the link between perception and belief revision. Our notion of focused belief revision is related, but not identical, to the *local revision* of [9]. Limitations of space prevent us from a careful comparison of the two notions. Suffice it to say that our focused revision seems to subsume local revision and to be more suitable to the treatment of perception.

2 From Sensing to Believing

The theory of agents adopted here is based on the GLAIR agent architecture [1,2]. GLAIR is a layered architecture consisting of three levels. The bottom level, the sensory-actuator level (SAL), is the level controlling the operation of sensors and actuators (being either hardware or simulated). The top level, the knowledge level (KL), is the level at which conscious symbolic reasoning and planning take place. This level is implemented by the SNePS knowledge representation, reasoning, and acting system [10]. Statements in the SNePS knowledge base are taken to represent the beliefs of the agent. The SNePS reasoning system is a natural deduction system of relevance logic [4,11]. The middle level, the perceptuo-motor level (PML), provides an interface between the SAL and the KL for both perception and action. This is the level at which recognition (and cognition) of feature-vector representations of external objects and states takes place. The result of this activity is the formation of a belief to be added to the top KL. (See [2] for details.)

Since the details of the language used to represent the agent's beliefs are orthogonal to the main message of this paper, we minimize the exposition of its syntax and semantics. We assume a first-order language $\mathcal{L}$, with a rich ontology including individuals, time points, acts, and states; states may be thought of as propositional fluents of the situation calculus. The exact structure of time has no bearing on the results we present, but we may assume an unbounded, discrete, linear structure. A sentence of the form $Holds(s, t)$ means that state s holds at time t. A functional term of the form $Prog(a)$ denotes the state that holds whenever act a is in progress. For every perceptual modality m of the agent, we shall have a predicate symbol P_m, where a sentence $P_m(s, t)$ states that the agent has a perceptual experience of state s (or, simply, perceives s) at time t.

In the rest of the paper, by "perception beliefs" we mean beliefs that emanate from the PML into the KL as a result of a perceptual experience. We represent perception beliefs by sentences of the form $P_m(s, ^* \mathsf{NOW})$. Such beliefs are formed at the PML, where s is a perceivable-state term whose sub-terms and function symbols are all grounded in PML representations; $^*\mathsf{NOW}$ denotes the current time. (See [2].) A perception theory is a logical theory in $\mathcal{L}$ that allows us to infer from a perception belief $P_m(s, ^* \mathsf{NOW})$ the belief $Holds(s, ^* \mathsf{NOW})$. Such an inference is clearly defeasible, since perception need not be veridical. Both Musto and Konolige [6] and Bell and Huang [7] present non-monotonic modal logics of perception and belief to account for this defeasibility. Their central claim is that, unless the agent has reason to believe that the perception process is abnormal, it should come to believe in the content of its perception. Abnormalities in perception may be due to faulty senors, illusory environments with abnormal perception conditions, or hallucination. A sophisticated-enough agent will have a theory of its own perceptual apparatus, of the different ways illusions may arise for each of its modalities, and of hallucination. These theories are necessarily causal as argued in [6,7]. (See [7] for an example of a simple theory of the effects of red light on visual perception, and [12] for a theory of noisy sensors.) Assuming such a theory, for each modality m, we will have a perception-belief axiom of the form

PBm. $\forall s, t[P_m(s, t) \wedge \Upsilon_m(s, t) \supset Holds(s, t)]$

where $\Upsilon_m(s, t)$ is a sentence stating that the perception of s at t via modality m is normal. The set of these axioms, together with theories of perceptual apparatus, illusions, and hallucinations form what we call a perception theory. We will, henceforth, refer to this theory as Π.

Now, if the agent does not believe $\Upsilon_m(s, t)$, it will not succumb to believing $Holds(s, t)$, unless it has other reasons. On the other hand, if the agent believes

$\Upsilon_m(s,t)$, it will accept $Holds(s,t)$. This, however, might result in a contradiction, if the agent has independent reasons for rejecting $Holds(s,t)$. In such a case, the agent should revise its beliefs. In neither [6] nor [7] do we find a complete discussion of how this is achieved. (In [6], we find no discussion.) In the following section, we present a belief revision operator, focused belief revision, that we use in Section 4 to model high-level perception.

3 Focused Belief Revision

We assume a proof theory based on Anderson and Belnap's system FR of relevant implication [4]. Cn_R will be henceforth used to denote relevance logic consequence. In FR, the relevant implication operator $\Rightarrow$ is defined in such a way that, where A is a set of formulas, $\phi \Rightarrow \psi \in \text{Cn}_R(A)$ if ϕ is actually instrumental to the derivation of ψ from A. We follow the presentation in [13] of an assumption-based reason maintenance system that implements belief revision in SNePS [11,14]. Incidently, assumption-based reason maintenance requires the recording of hypotheses used in derivations, which is independently motivated by the use of relevance logic.

Definition 1. *A* ***support set*** *of a sentence $\phi \in \mathcal{L}$ is a set $s \subseteq \mathcal{L}$ such that $\phi \in \text{Cn}_R(s)$. s is* ***minimal*** *if, for every $s' \subset s, \phi \notin \text{Cn}_R(s')$.*

Definition 2. *A* ***belief state*** *$\mathbb{S}$ is a quadruple $\langle \mathcal{K}, \mathcal{B}, \sigma, \preccurlyeq \rangle$, where:*

1. *$\mathcal{K} \subseteq \mathcal{L}$ is a* ***belief set****.*
2. *$\mathcal{B} \subseteq \mathcal{K}$, with $\mathcal{K} \subseteq \text{Cn}_R(\mathcal{B})$, is a finite* ***belief base****. If $\phi \in \mathcal{B}$, then ϕ is a* ***base belief****.*
3. *$\sigma : \mathcal{L} \longrightarrow 2^{2^{\mathcal{B}}}$ is a* ***support function****, where each $s \in \sigma(\phi)$ is a minimal support set of ϕ. Further, $\sigma(\phi) \neq \varnothing$ if and only if $\phi \in \mathcal{K}$. In particular, if $\phi \in \mathcal{B}$, then $\{\phi\} \in \sigma(\phi)$.*
4. *$\preccurlyeq \subseteq \mathcal{B} \times \mathcal{B}$ is a total pre-order on base beliefs.*

Base beliefs are beliefs that have independent standing; they are not in the belief state based *solely* on inference. In particular, perception beliefs (in the sense of Section 2) are necessarily base beliefs. The belief set $\mathcal{K}$ is not closed under Cn_R; it represents the set of sentences that are either base beliefs or that were *actually derived* from base beliefs. This is in contrast to the logically-closed $\text{Cn}_R(\mathcal{K})$ which is the set of sentences *derivable* from base beliefs.

The set $\sigma(\phi)$ is the family of minimal support sets that were actually used, or discovered, to derive ϕ. $\mathcal{B}$ may include minimal support sets of ϕ that are, nevertheless, not in $\sigma(\phi)$, if they are not yet discovered to derive ϕ. The total pre-order $\preccurlyeq$ represents a preference ordering over base beliefs. We will refrain from making any commitments about the origins of this ordering; for the purpose of this paper, the ordering is just given. Note that, given the finiteness of $\mathcal{B}$, $\preccurlyeq$ is guaranteed to have minimal and maximal elements.

For brevity, where $\phi \in \mathcal{L}$ and $A \subseteq \mathcal{L}$, let $\text{Cn}_R(A, \phi) = \{\psi \mid \phi \Rightarrow \psi \in \text{Cn}_R(A)\}$. In what follows, $\mathbb{S} = \langle \mathcal{K}, \mathcal{B}, \sigma, \preccurlyeq \rangle$ is a belief state, $\mathcal{F} \subseteq \mathcal{K}$, and $\phi \in \mathcal{L}$.

Definition 3. *A* ***focused expansion*** *with* ***focus set*** $\mathcal{F}$ *of* $\mathbb{S}$ *with* ϕ *is a belief state* $\mathbb{S} +^{\mathcal{F}} \phi = \langle \mathcal{K}_{+^{\mathcal{F}}\phi}, \mathcal{B}_{+^{\mathcal{F}}\phi}, \sigma_{+^{\mathcal{F}}\phi}, \preccurlyeq_{+^{\mathcal{F}}\phi} \rangle$*, satisfying the following properties.*

($A^{+}1$) *Success:* $\mathcal{B}_{+^{\mathcal{F}}\phi} = \mathcal{B} \cup \{\phi\}$.

($A^{+}2$) *Relevant inclusion:* $\mathcal{K}_{+^{\mathcal{F}}\phi} = \mathcal{K} \cup \mathrm{Cn}_R(\mathcal{F}, \phi)$.

($A^{+}3$) *Relevant Support: For every* $\psi \in \mathcal{L}$,

1. $\sigma(\psi) \subseteq \sigma_{+^{\mathcal{F}}\phi}(\psi)$*; and*
2. *if* $\sigma(\psi) \subset \sigma_{+^{\mathcal{F}}\phi}(\psi)$ *then* $\psi \in \mathrm{Cn}_R(\mathcal{F}, \phi)$ *and for every* $s \in \sigma_{+^{\mathcal{F}}\phi}(\psi) \setminus \sigma(\psi)$*, there is an* s'' *such that* $s \in \{s'' \cup s' | s' \in \sigma_{+^{\mathcal{F}}\phi}(\phi)\} \subseteq \sigma_{+^{\mathcal{F}}\phi}(\psi)$.

($A^{+}4$) *Order preservation:* $\preccurlyeq_{+^{\mathcal{F}}\phi}$ *is a total pre-order on* $\mathcal{B}_{+^{\mathcal{F}}\phi}$ *such that, for every* $\psi, \xi \in \mathcal{B}$, $\psi \preccurlyeq_{+^{\mathcal{F}}\phi} \xi$ *if and only if* $\psi \preccurlyeq \xi$.

The belief state resulting from focused expansion by ϕ will include ϕ and anything that follows from it, given the focus set $\mathcal{F}$. That all newly derived sentences indeed follow from ϕ is guaranteed by ($A^{+}2$). In addition, old sentences may acquire new support only as a result of discovered derivations from ϕ (($A^{+}3$)). ($A^{+}4$) makes the simplifying assumption that adding ϕ does not disturb the preference relations already established; ϕ simply gets added in *some* appropriate position in the $\preccurlyeq$-induced chain of equivalence classes (and, hence, the non-uniqueness of focused expansion). We have implemented focused expansion in SNePS as assertion with forward inference, provided that only the sentences in $\mathcal{F}$ are considered for matching.

Definition 4. *A* ***focused revision*** *with focus set* $\mathcal{F}$ *of* $\mathbb{S}$ *with* ϕ *is a belief state* $\mathbb{S} \dotplus^{\mathcal{F}} \phi = \langle \mathcal{K}_{\dotplus^{\mathcal{F}}\phi}, \mathcal{B}_{\dotplus^{\mathcal{F}}\phi}, \sigma_{\dotplus^{\mathcal{F}}\phi}, \preccurlyeq_{\dotplus^{\mathcal{F}}\phi} \rangle$*, satisfying the following properties.*

($A^{\dotplus}1$) *Base inclusion:* $\mathcal{B}_{\dotplus^{\mathcal{F}}\phi} \subseteq \mathcal{B}_{+^{\mathcal{F}}\phi}$.

($A^{\dotplus}2$) *Inclusion:* $\mathcal{K}_{\dotplus^{\mathcal{F}}\phi} \subseteq \mathcal{K}_{+^{\mathcal{F}}\phi}$.

($A^{\dotplus}3$) *Lumping:* $\psi \in \mathcal{K}_{+^{\mathcal{F}}\phi} \backslash \mathcal{K}_{\dotplus^{\mathcal{F}}\phi}$ *if and only if, for every* $s \in \sigma_{+^{\mathcal{F}}\phi}(\psi)$, $s \nsubseteq \mathcal{B}_{\dotplus^{\mathcal{F}}\phi}$.

($A^{\dotplus}4$) *Preferential core-retainment:* $\psi \in \mathcal{B}_{+^{\mathcal{F}}\phi} \backslash \mathcal{B}_{\dotplus^{\mathcal{F}}\phi}$ *if and only if there is* $\chi \in \mathcal{L}$ *such that* $(\chi \wedge \neg\chi) \in \mathrm{Cn}_R(\mathcal{F}, \phi)$ *and there is* $s \in \sigma_{+^{\mathcal{F}}\phi}(\chi \wedge \neg\chi)$ *such that* ψ *is a minimal element of* s *with respect to* $\preccurlyeq_{+^{\mathcal{F}}}$.

($A^{\dotplus}5$) *Support update: For every* $\psi \in \mathcal{L}$, $\sigma_{\dotplus^{\mathcal{F}}\phi}(\psi) = \sigma_{+^{\mathcal{F}}\phi}(\psi) \cap 2^{\mathcal{B}_{\dotplus^{\mathcal{F}}\phi}}$

($A^{\dotplus}6$) *Order preservation:* $\preccurlyeq_{\dotplus^{\mathcal{F}}\phi}$ *is the restriction of* $\preccurlyeq_{+^{\mathcal{F}}\phi}$ *to* $\mathcal{B}_{\dotplus^{\mathcal{F}}\phi}$.

Thus, focused revision is focused assertion with forward inference followed by some kind of focused consolidation. Consolidation may be implemented by removing least-preferred beliefs (as per $\preccurlyeq$) from each support set of a contradiction $(\chi \wedge \neg\chi)$ in the inconsistent belief state resulting from expansion by ϕ. As a result of consolidation, some base beliefs might be retracted in case focused expansion with ϕ results in a contradiction. ($A^{\dotplus}1$) captures this intuition. ($A^{\dotplus}3$) makes sure that only sentences that are still supported are believable. ($A^{\dotplus}4$) guarantees that base beliefs that are evicted to retain (explicit) ϕ-relevant consistency indeed must be evicted. In addition, if a choice is possible, base beliefs that are least preferred are chosen for eviction. Note that, according to the above definition, this selection strategy is *skeptical*; that is, if multiple least preferred beliefs exist, all are evicted. This strategy, however, is only adopted here to simplify the exposition, and nothing relevant depends on it.

The strongest result we can state about post-revision consistency is the following. The proof is eliminated for space limitations.

Theorem 1. *Let $\phi \in \mathcal{L}$ and $\mathbb{S} = \langle \mathcal{K}, \mathcal{B}, \sigma, \preccurlyeq \rangle$ be a belief state with $\mathcal{F} \subseteq \mathcal{K}$. For every, $\chi \in \mathcal{L}$, if $(\chi \wedge \neg\chi) \in \mathrm{Cn}_R(\mathcal{F}, \phi)$, then $(\chi \wedge \neg\chi) \notin \mathcal{K}\dot{+}_{\mathcal{F}}\phi$.*

As a simple corollary, it follows that if the belief set $\mathcal{K}$ is not known to be inconsistent, then, following focused revision, it is not known to be inconsistent.

Corollary 1. *Let $\mathbb{S} = \langle \mathcal{K}, \mathcal{B}, \sigma, \preccurlyeq \rangle$ be a belief state with $\mathcal{F} \subseteq \mathcal{K}$. For every $\phi, \chi \in \mathcal{L}$, if $(\chi \wedge \neg\chi) \notin \mathcal{K}$, then $(\chi \wedge \neg\chi) \notin \mathcal{K}\dot{+}_{\mathcal{F}}\phi$.*

This is as far as focused revision can claim about any notion of consistency restoration. In particular, if $\chi \wedge \neg\chi \in \mathcal{F}$ (not to mention $\mathcal{K}$), it is not necessary that $\chi \wedge \neg\chi \notin \mathcal{K}\dot{+}_{\mathcal{F}}\phi$. This is because focused revision is only guaranteed to treat an inconsistency to which the revising belief ϕ is relevant. Nevertheless, an inconsistency not induced by ϕ might be accidentally removed in case revising by ϕ side-effects the eviction of base beliefs that support the inconsistency.

4 Origins of Focus

To model high-level perception by focused belief revision, we need to consider interpretations of the focus set $\mathcal{F}$ that are suitable for perception. Ultimately, we believe the choice of $\mathcal{F}$ to be an empirical matter that can only be settled in the psychology or in the AI lab. Nevertheless, in this section, we present some general guidelines for the selection of $\mathcal{F}$ in the context of perception.

We believe that the selection of a suitable focus set should be based on (at least) three factors: (i) what is vital for the agent, (ii) what is relevant for the agent, and (iii) how much resources are available for perception-induced reasoning.

For every agent, there are certain things that it cannot afford to not notice. For example, an agent might believe that, whenever there is fire, it should leave the building. (Remember agent Y?) A focus set of such an agent must include beliefs that allow it to conclude the imminence of fire from the perception of signs of fire. Thus, for every agent, there will be a certain set of vital beliefs, defined by fiat, that should be included in any focus set.

At a given time, to decide whether some belief is relevant *for* the agent we need to be explicit about two aspects of being relevant: (i) relevant *to* what, and (ii) relevant in what sense. We take a belief to be relevant for the agent if and only if it is relevant *to what the agent believes it is doing.* We believe that, broadly conceived, the set of acts the agent believes it is performing determines the focus of its attention, and, hence, provide a starting point for deciding on the contents of the focus set. Two acts in this set are either (i) independent concurrent acts, (ii) one of them is performed *by* performing the other, or (iii) one of them is performed as a step in (or towards) performing the other.

Now, what does it mean for a belief to be relevant to what the agent is doing? Intuitively, such a belief would be *about* the act the agent is performing, or about objects, individuals, and states involved in the act. This notion of *aboutness*, however, is notoriously elusive. One possibility is to explicitly associate beliefs to topics in the spirit of [15]. Another is to automatically partition the belief set $\mathcal{K}$ using, for example, the greedy algorithm of Amir and McIlraith [16]. It should be noted, however, that Amir and McIlraith's concerns are very different from ours, and that, hence, their approach might not be suited for the case of perception. In this paper, we adopt a simple syntactic indicator of relevance that we call *n^{th}-degree term sharing.* This is a generalization of the *variable sharing principle* adopted by relevance logicians for propositional languages [4]. The degree of term-sharing between two sentences reflects (in reverse proporition) the degree of relevance between the corresponding beliefs. As it turns out, the notion of degrees of relevance provides a way to tune the construction of the focus set to the amount of resources the agent can spend in the process: the more the resources, the more the beliefs with lower degrees of relevance the agent can consider.

For any $\mathbb{S} = \langle \mathcal{K}, \mathcal{B}, \sigma, \preccurlyeq \rangle$, $\alpha(\mathbb{S})$ (or α when $\mathbb{S}$ is obvious) is the set of all sentences in $\mathcal{K}$ of the form $Holds(Prog(a),^{*}\mathsf{NOW})$. A function $\gamma : 2^{\mathcal{L}} \times \mathcal{L} \longrightarrow 2^{\mathcal{L}}$ is a **relevance filtering function** if $\gamma(A, \phi) \subseteq A$. If $\phi \in \mathcal{L}$, $\tau(\phi)$ is the set of all closed terms occurring in ϕ and $TS(\phi) = \{\psi | \psi \in \mathcal{K} \cup \{\phi\} \text{ and } \tau(\phi) \cap \tau(\psi) \neq \varnothing\}$.

Definition 5. *Let $n \in \mathbb{N}$ and let γ be a relevance filtering function. An n^{th}-**degree term sharing function with filter** γ is a function $\mathfrak{t}_{\gamma}^{n} : \mathcal{L} \longrightarrow 2^{\mathcal{L}}$ defined as follows:*

$$\mathfrak{t}_{\gamma}^{n}(\phi) = \begin{cases} \{\phi\} & \text{if } n = 0 \\ \gamma(TS(\phi), \phi) & \text{if } n = 1 \\ \{\psi | \text{ for some } \xi \in \mathfrak{t}_{\gamma}^{n-1}(\phi), \psi \in \mathfrak{t}_{\gamma}^{1}(\xi)\} & \text{otherwise} \end{cases}$$

$\mathfrak{t}_{\gamma}^{1}(\phi)$ is the result of filtering the set of sentences that share at least one term with ϕ. The filtering function is used to account for the fact that term sharing is not sufficient for relevance. By experimenting with some filtering functions, we have found the following heuristics useful:

1. *If ϕ is of the form $Holds(s, t)$, filter out sentences of the form $Holds(s', t)$, for $\tau(s') \cap \tau(s) = \varnothing$.* The intuition is that merely having two states hold simultaneously does not mean that propositions asserting these facts are relevant to one another.
2. In a language where properties are denoted by terms (which is the case for SNePS-based systems), *if ϕ attributes a property to an individual, filter out sentences that attribute the same property to an unrelated individual.* For example, if the agent sees a dog, what is relevant is not which other individuals are dogs, but general properties of dogs that may be used in reasoning about the perceived instance.

Nevertheless, the exact definition of the filtering function largely depends on pragmatic factors that vary from one agent to another. In what follows, $\mathbb{S} = \langle \mathcal{K}, \mathcal{B}, \sigma, \preccurlyeq \rangle$ and $p = P_m(s,^{*}\mathsf{NOW}) \in \mathcal{L}$.

Definition 6. *A **focus structure** $\mathbb{F}_{\mathbb{S},p}$ is a quadruple $\langle \mathcal{V}, \Gamma, \Delta, \rho \rangle$, where*

- *$\mathcal{V} \subseteq \mathcal{K}$ is a set of **vital** beliefs,*
- *$\Gamma : \alpha \cup \{p\} \longrightarrow [2^{\mathcal{L}} \times \mathcal{L} \longrightarrow 2^{\mathcal{L}}]$ is a mapping that assigns to every member of $\alpha \cup \{p\}$ a relevance filtering function,*
- *$\Delta : \alpha \cup \{p\} \longrightarrow \mathbb{N}$ is a mapping that assigns to every member of $\alpha \cup \{p\}$ a degree of relevance, and*
- *$\rho : 2^{\mathcal{L}} \times 2^{2^{\mathcal{L}}} \longrightarrow 2^{\mathcal{L}}$ is a **relevance choice function**, where*

$$\rho(\mathfrak{t}^{\Delta(p)}_{\Gamma(p)}(p), \{\mathfrak{t}^{\Delta(a)}_{\Gamma(a)}(a)\}_{a \in \alpha}) \subseteq \bigcup_{\phi \in \alpha \cup \{p\}} \mathfrak{t}^{\Delta(\phi)}_{\Gamma(\phi)}(\phi)$$

The above notion of focus structures is an attempt to pinpoint the factors contributing to the construction of focus sets. Nonetheless, the definition is flexible enough to accommodate agent-specific considerations regarding vital beliefs, filtering functions, degrees of relevance to be considered based on available resources, and the construction of the set of relevant beliefs based on a relevance choice function.

Definition 7. *Let $\mathbb{F}_{\mathbb{S},p} = \langle \mathcal{V}, \Gamma, \Delta, \rho \rangle$ be a focus structure. The **high-level perception** of s in $\mathbb{S}$ with focus structure $\mathbb{F}_{\mathbb{S},p}$ is the focused belief revision, $\mathbb{S} \dot{+}^{\mathcal{F}} p$, of $\mathbb{S}$ with p where*

$$\mathcal{F} = \mathcal{V} \cup \Pi \cup \rho(\mathfrak{t}^{\Delta(p)}_{\Gamma(p)}(p), \{\mathfrak{t}^{\Delta(a)}_{\Gamma(a)}(a)\}_{a \in \alpha})$$

and p is a maximal element of $\mathcal{B}_{+^{\mathcal{F}}p}$ with respect to $\preccurlyeq_{+^{\mathcal{F}}p}$.

The set Π appearing in the definition of $\mathcal{F}$ above is the perception theory referred to in Section 2. The requirement that p be a maximally preferred belief reflects the idea (often discussed in the philosophical literature) that having a present perceptual experience of some state s is indefeasible. Given the perception-belief axiom, an agent might not accept the proposition that s currently holds, if it has reason to believe that its sensory apparatus is faulty or that it is having an illusion. Nevertheless, the agent would still believe that it did have a perceptual experience of s. It is reasonable to assume that beliefs in Π about perception being veridical are not maximal elements of $\preccurlyeq_{+^{\mathcal{F}}p}$. If this is indeed the case, high-level perception will satisfy the AGM postulate of success.

5 Conclusion

We have presented a framework for high-level perception as focused belief revision. This simultaneously addresses two issues. On one hand, the defeasibility of perception-induced beliefs is accounted for through the underlying reason maintenance system. On the other hand, bounded reflection on the contents of perception is implemented by two aspects of our system. First, the use of relevance logic guarantees that all perception-induced beliefs follow from the perception belief itself. Second, the definition of the focus set limits reasoning only to what is relevant and vital for the agent, while taking issues of resource boundedness into account in a fairly general way.

We have identified four factors that, we believe, determine the construction of the focus set: the set of vital beliefs; the relevance filtering functions; the degrees of relevance; and the relevance choice function. All these factors depend on the belief state, the percept, and whatever the agent is doing. In the future, more experiments with different focus structures and different agents are needed to further evaluate the proposed framework.

References

1. Hexmoor, H., Lammens, J., Shapiro, S.C.: Embodiment in GLAIR: a grounded layered architecture with integrated reasoning for autonomous agents. In: Proceedings of The Sixth Florida AI Research Symposium (FLAIRS 1993), pp. 325–329 (1993)
2. Shapiro, S.C., Ismail, H.O.: Anchoring in a grounded layered architecture with integrated reasoning. Robotics and Autonomous Systems 43(2-3), 97–108 (2003)
3. Alchourron, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. The Journal of Symbolic Logic 50(2), 510–530 (1985)
4. Anderson, A., Belnap, N.: Entailment, vol. I. Princeton University Press, Princeton (1975)
5. Shoham, Y., del Val, A.: A logic for perception and belief. Technical Report STAN-CS-91-1391, Department of Computer Science, Stanford University (1991)
6. Musto, D., Konolige, K.: Reasoning about perception. In: Papers from the 1993 AAAI Spring Symposium on Reasoning about Mental States, pp. 90–95 (1993)
7. Bell, J., Huang, Z.: Seeing is believing. In: Papers from the 4th Symposium on Logical Formalizations of Commonsense Reasoning, Commonsense 1998 (1998)
8. Wooldridge, M., Lomuscio, A.: A computationally grounded logic of visibility, perception, and knowledge. Logic Journal of the IGPL 9(2), 273–288 (2001)
9. Wassermann, R.: Resource bounded belief revision. Erkenntnis 50(2-3), 429–446 (1999)
10. Shapiro, S.C., Rapaport, W.J.: The SNePS family. Computers and Mathematics with Applications 23(2-5), 243–275 (1992)
11. Martins, J., Shapiro, S.C.: A model for belief revision. Artificial Intelligence 35(1), 25–79 (1988)
12. Bacchus, F., Halpern, J., Levesque, H.: Reasoning about noisy sensors and effectors in the situation calculus. Artificial Intelligence 111(1-2), 171–208 (1999)
13. Ismail, H.O.: A reason maintenance perspective on relevant Ramsey conditionals. Logic Journal of the IGPL 18(4), 508–529 (2010)
14. Johnson, F.L., Shapiro, S.C.: Dependency-directed reconsideration: Belief base optimization for truth maintenance systems. In: Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI 2005), pp. 313–320 (2005)
15. Demolombe, R., Jones, A.J.I.: Reasoning about topics: Towards a formal theory. In: Workshop on Formalizing contexts: Papers from the 1995, AAAI Fall Symposium, Technical Report FS-95-02, Boston, 55–59 (1995)
16. Amir, E., McIlraith, S.: Partition-based logical reasoning for first-order and propositional theories. Artificial Intelligence 162(1-2), 49–88 (2005)

Multi-context Systems with Activation Rules

Stefan Mandl and Bernd Ludwig

Dept. of Computer Science 8 (Artificial Intelligence),
Friedrich-Alexander-Universität Erlangen-Nürnberg,
Haberstraße 2, D-91058 Erlangen, Germany
{Stefan.Mandl,Bernd.Ludwig}@cs.fau.de
http://www8.informatik.uni-erlangen.de

Abstract. Multi-Context Systems provide a formal basis for the integration of knowledge from different knowledge sources. Yet, it is easy to conceive of applications where not all knowledge sources may be used together all the time. We present a natural extension of Multi-Context Systems by adding the notion of activation rules that allows modeling the applicability or relevance of contexts depending on beliefs in the various contexts and their mutual dependencies. We give a short account on possible consequence relations for Multi-Context System with Activation Rules and discuss a potential application in information retrieval.

1 Introduction

Figure 1 shows the by now almost classical motivating example for formal Multi-Context Systems (MCS). Mr.1 and Mr.2 have different perspectives on *the same* real world object. There are queries that either Mr.1 or Mr.2 cannot answer by himself, depending on the configuration of the scenario. MCS provide a basis for formal representation of such settings by using the key concept of *bridge rules* that can be used to describe information flow between the involved parties. For instance, the bridge rule

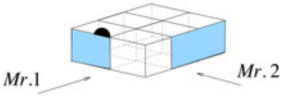


Fig. 1. Magic box (taken from [5])

$$\text{Mr.1:}\texttt{left} \lor \text{Mr.1:}\texttt{right} \leftarrow \text{Mr.2:}\texttt{left}.$$

would transfer enough of Mr.2's knowledge to Mr.1 to enable him to come up with the correct conclusion that from *his point of view* the ball must be hidden behind the left plate of the magic box.

Knowledge integration is not the only possible task imaginable. In [1] the aspect of integration belongs to the *perspective* dimension of context dependence.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 135–142, 2010.

Further dimensions of context dependence like *partiality* and *approximation* are discussed there. In this paper we want to focus on dynamic scenarios like the one depicted in figure 2 where the perspective changes over time. There, the magic box is put on top of a pin and rotates with constant speed. For simplicity, we only consider the cases where a side of the box is fully exposed to the observer.

Now imagine the task to implement behavior similar to that of Mr.1, but now in the dynamic scenario where Mr.1 would stand still while the box is rotating in front of him. When asked if and in which location he sees the ball, he would have to give different answers depending on the orientation of the box at *the time* he is asked. Contrary to the standard magic box scenario, now, only one of the contexts may be used at a time, all other contexts are not relevant or appropriate. Hence, Mr.1 has to decide which context to use and never may incorporate the knowledge contained in contexts currently rendered inappropriate due to the current orientation. We expect that realistic applications be typically not strictly knowledge integration or knowledge separation scenarios. Instead, we think that for real world applications, which involve multiple knowledge bases from various knowledge sources probably using different knowledge representation formalisms, one has to deal with knowledge integration as well as knowledge separation tasks when modeling the systems knowledge. The goal of the paper is to develop a formalism that allows to model knowledge integration and knowledge separation in a uniform way providing a clean semantics of the resulting representations.

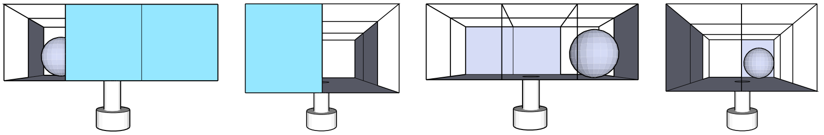

(a) Mr.1/Front (b) Mr.1/Left (c) Mr.1/Back (d) Mr.1/Right

Fig. 2. The four different orientations of the spinning magic box

The paper is structured as follows: Section 2 introduces Multi-Context Systems with Activation Rules (ARMCS) and gives consequence relations for MCS and ARMCS, effectively turning them into knowledge representation and reasoning formalisms. Section 3 shortly discusses the complexity of (AR)MCS reasoning and outlines a straightforward implementation.[1] Section 4 gives a concise example. Section 5 shortly discusses some computational aspects of the computation of equilibrium states of ARMCS, concludes the paper, and outlines directions of future research.

2 MCS with Activation Rules

Brewka and Eiter (see [5]) introduced a theoretical framework for heterogeneous nonmonotonic MCS which—as the by now most refined model of MCS—forms

[1] We write (AR)MCS when we mean both MCS and ARMCS.

the foundation for this paper. Due to space limitations, we only give a short review of the most crucial definitions from this paper (see below).

The basic definitions of heterogeneous MCS from [5]

Definition 1. *A logic $L = (KB_L, BS_L, ACC_L)$ is composed of the following components:*

1. *KB_L is the set of well-formed knowledge bases of L. We assume each element of KB_L is a set.*
2. *BS_L is the set of possible belief sets,*
3. *$ACC_L : KB_L \mapsto 2^{BS_L}$ is a function describing the "semantics" of the logic by assigning to each element of KB_L a set of acceptable sets of beliefs.*

Definition 2. *Let $L = \{L_1, ..., L_n\}$ be a set of logics. An L_k-bridge rule over L ,$1 \leq k \leq n$, is of the form*

$$s \leftarrow (r_1 : p_1), \ldots, (r_j : p_j), \mathbf{not}(r_{j+1} : p_{j+1}), \ldots, \mathbf{not}(r_m : p_m) \tag{1}$$

where $1 \leq r_k \leq n$, p_k is an element of some belief set of L_{rk},and for each $kb \in KB_k : kb \cup \{s\} \in KB_k$.

Definition 3. *A multi-context system $M = (C_1, \ldots, C_n)$ consists of a collection of contexts $C_i = (L_i, kb_i, br_i)$, where $L_i = (KB_i, BS_i, ACC_i)$ is a logic, kb_i a knowledge base (an element of KB_i), and br_i is a set of L_i-bridge rules over $\{L_1, \ldots, L_n\}$.*

Definition 4. *Let $M = (C_1, \ldots, C_n)$ be an MCS. A belief state is a sequence $S = (S_1, \ldots, S_n)$ such that each S_i is an element of BS_i.*

We say a bridge rule r of form (1) is applicable in a belief state $S = (S_1, \ldots, S_n)$ iff for $1 \leq i \leq j : p_i \in S_{r_i}$ and for $j + 1 \leq k \leq m : p_k \notin S_{r_k}$.

Definition 5. *A belief state $S = (S_1, \ldots, S_n)$ of M is an equilibrium iff, for $1 \leq i \leq n$, the following condition holds:*

$$Si \in ACC_i(kb_i \cup \{head(r) | r \in br_i \ applicable \ in \ S\}). \tag{2}$$

Please note that informally an equilibrium of a MCS is a belief state such that the belief set for each context is acceptable and respects the belief sets of the other contexts as 'accessible' via the bridge rules.

Our goal is to introduce means to model knowledge separation, but also to retain the possibility of knowledge integration. To this end, we propose to extend MCS by activation rules. Informally, if an activation rule fires, its context is active and shall be taken into account when computing equilibria. If a context is not activated by any activation rule, it is not considered. The actual definition is more complicated as the applicability of activation rules can depend on the beliefs in the contexts, influenced by bridge rules, and the outcome of other activation rules. Therefore, an activation rule may well be active in a certain equilibrium while it is not active in another equilibrium. Activation rules are similar to bridge rules (definition 2). They target contexts instead of belief set elements and can depend on the activation (or inactivity) of other contexts.

Definition 6. *Let $C = \{C_1, ..., C_n\}$ be a set of contexts. A C_q-activation rule over C, $1 \leq q \leq n$, is of the form*

$$\begin{aligned} C_q \leftarrow C_{r_1}, \ldots, C_{r_j}, \mathbf{not}\, C_{r_{j+1}}, \ldots, \mathbf{not}\, C_{r_k}, \\ r_{k+1} : p_{k+1}, \ldots, r_l : p_l, \mathbf{not}(r_{l+1} : p_{l+1}), \ldots, \mathbf{not}(r_m : p_m) \end{aligned} \tag{3}$$

where $1 \leq r_i \leq n$, p_i is an element of some belief set of L_{r_i}, and C_{r_i} and C_q are contexts.

ARMCS are standard MCS as of definition 3 with a distinct set of activation rules added to each context.

Definition 7. *A multi-context system with activation rules $M = (C_1, \ldots, C_n)$ consists of a collection of contexts $C_i = (L_i, kb_i, br_i, ar_i)$, where $L_i = (KB_i, BS_i, ACC_i)$ is a logic, kb_i a knowledge base (an element of KB_i), br_i is a set of L_i-bridge rules over $\{L_1, \ldots, L_n\}$, and ar_i is a set of C_i-activation rules over $\{C_1, \ldots, C_n\}$.*

Belief states for ARMCS have to respect inactivity of contexts:

Definition 8. *Let $M = (C_1, \ldots, C_n)$ be an ARMCS. A belief state is a sequence $S = (S_1, \ldots, S_n)$ such that each S_i is an element of $BS_i \cup \{*_i\}$, where $*_i$ denotes the fact that i^{th} belief set is not available for inspection.*

We say an activation rule of form (3) is applicable in a belief state $S = (S_1, \ldots, S_n)$ iff for $1 \leq i \leq j$, $S_{r_i} \neq *_i$ and for $j+1 \leq i \leq k$, $S_{r_i} = *_i$ and for $k+1 \leq i \leq l$, $p_i \in S_{r_i}$ and for $l+1 \leq i \leq m$, $p_i \notin S_{r_i}$. Please note that $*_i \neq \emptyset$. The definition for equilibria can easily be adapted to care about the active contexts only.

Definition 9. *A belief state $S = (S_1, \ldots, S_n)$ of an ARMCS M is an equilibrium, iff for each available context S_i in S, there is an activation rule in ar_i that is applicable and S' which is obtained from S by removing all unavailable contexts is an equilibrium as of definition 5 of the multi-context system M' that is obtained from M in the following way:*

- *Delete all bridge rules that* positively *refer to unavailable contexts in S.*
- *In the remaining bridge rules remove all* negative *references to unavailable contexts in S.*
- *Consistently rename all context identifiers in the remaining bridge rules such that they respect the position changes that may have been caused by removing the unavailable contexts when creating S' from S.*
- *Remove all unavailable contexts.*
- *Remove all activation rules.*

Example. Let $M = (C_1, C_2)$ with

$$\begin{aligned} KB &= 2^{\{p,q,r\}} \\ BS &= KB \\ ACC(kb) &= \text{the set of stable models of } kb \in KB \\ L &= (KB, BS, ACC) \\ C_1 &= (L, \emptyset, \{p \leftarrow 2 : q.\}, \{C_1 \leftarrow \mathbf{not}\, 2 : r.\}) \\ C_2 &= (L, \emptyset, \{q \leftarrow 1 : p. \quad r \leftarrow \mathbf{not}\, 1 : p.\}, \{C_2.\}) \end{aligned}$$

Then $(*_1, \{r\})$ and $(\{p\}, \{q\})$ are the two equilibria of M. $(*_1, \{r\})$ is an equilibrium as C_1 is not active and therefore the conditional part of the bridge rule $r \leftarrow \mathbf{not}\, 1 : p.$ is removed, hence r is simply asserted in C_2. $(\{p\}, \{q\})$ is an equilibrium as p in C_1 is justified by the bridge rule $p \leftarrow 2 : q.$ and q in C_2 is justified by $q \leftarrow 1 : p.$ and $r \leftarrow \mathbf{not}\, 1 : p.$ is not applicable. Please note that as of [5] these equilibria are both *minimal*. Furthermore they are also considered *self justified* as they are not grounded in facts in the knowledge bases, which in the example are both empty. In contrast, $(*_1, *_2)$ is not an equilibrium as the activation rules of C_2, namely the single unconditional rule $C_2.$, require the presence of context 2 in every equilibrium.

In order to use (AR)MCS for knowledge representation, we need to define a consequence relation that unifies the different perspectives that the contexts constitute. In order to do so, we need to make the assumption that symbolic names in different contexts are used in a consistent way, hence we assume that $o_1 \in bs_i, o_2 \in bs_j : o_1^{\mathcal{I}} = o_2^{\mathcal{I}} \Rightarrow o_i = o_j$ for all interpretations $\mathcal{I}$ employed with a given (AR)MCS. The following consequence relation(s) are inspired by the consequence relations defined in the field of nonmonotonic reasoning (see for instance [4]).

Definition 10. *Assuming consistent names, a sentence ϕ is bravely (skeptically) entailed by an (AR)MCS M, ($M \vDash \phi$) iff in at least one (every) equilibrium of M there is at least one belief set B such that $\phi \in B$ and no belief set such that $\neg\phi \in B$.*

Hence, in the example above, $\{p, q, r\}$ are bravely entailed and $\emptyset$ is skeptically entailed by M.

3 Computing Equilibria of ARMCS

In [7] it is shown that the problem of checking for existence of equilibria of MCS is in **NP** for local contexts with complexity **P** or **NP** and in **PSPACE** (resp. **EXPTIME**) for local contexts with complexity **PSPACE** (resp. **EXPTIME**). These results are obtained by considering a Turing machine which guesses belief states and tests for equilibria. The introduction of activation rules does not change this picture in a qualitative way. We have realized a simple implementation of ARMCS with finite sets of bridge rules and activation rules, which follows a similar generate and test idea which is based on the observation that for any equilibrium of an (AR)MCS there is exactly one subset of bridge rules (and activation rules) that fire (see [2]). Hence, by considering all those subsets of the bridge rules (and activation rules), and for each subset generating a set of corresponding belief states, which are in turn tested for equilibria, one obtains a straightforward procedure to enumerate all equilibria of an (AR)MCS. While this procedure in general is intractable, many potential applications of (AR)MCS allow for substantial pruning of the involved search space:

- if ACC_i is functional, hence, if there is only one possible belief set in context C_i, the number of candidate belief states per subset of rules is reduced
- if rules can be shown to interact, such that one rule can fire only if the other does not, no subsets of rules with both rules firing have to be considered at all

The task of designing (AR)MCS can be greatly improved by allowing for variables to occur in bridge rules or activation rules. We suggest a simple preprocessing step of grounding an (AR)MCS like the one described in [9] for Answer Set Programs. The only additional step is to consider ground literals that are imported into contexts via bridge rules.

4 Example: ARMCS in Practice

In order to demonstrate the benefit of activation rules, we consider an information retrieval system like the one that was developed in the research project ROSE[2] and tested at the science fair "Lange Nacht der Wissenschaften" which took place in the area of Nürnberg, Fürth, and Erlangen on a Saturday night in September 2009. Visitors could query a database of event descriptions in order to make a selection of events to visit before heading into the evening. As the available events came from different fields of science and domains of discourse and furthermore writers with different background phrased the corresponding descriptions, it is not surprising that the ambiguity of words in the queries lead to unexpected results. For instance, when searching for 'Sprache' (engl.: language), events of the Pattern Recognition Lab—matching the phrase '…gesprochene *Sprache* …' (engl.: '…spoken *language* …')—were returned along with events from the Chair for Data Management —matching the phrase '…in der *Sprache* SQL …' (engl.: '…in the *language* SQL …'). Clearly, users would benefit if the system knew which respective meaning of the query terms is relevant to the user. Currently the ROSE system employs an abstraction step when processing queries. For each (normalized) query term, a set of relevant topics is selected. Currently two kinds of topic spaces are used: The topic space defined by the

[2] ROSE is supported by a grant from the German Federal Ministry of Economics and Technology. A description of ROSE can be found at http://www.rose-mobil.de

Dornseiff lexicon ([6]), which contains 970 hand-moderated topics and an abstract topic space generated by using the LDA algorithm ([3]) on the available textual event descriptions which is based on the statistical properties of terms and documents. Term to topics abstractions can be straightforwardly represented as formal sentences (in this case as binary relations where the first element is the normalized term and the second element is an identifier for a specific topic, like $t('GARDEN', t_{37}).$) and trivially this relation can be thought of as a knowledge representation with an extremely simple consequence relation, namely set membership. Hence the use of knowledge representations in the term to topics abstraction, in future systems there is the opportunity to use ARMCS to enhance the systems performance. For instance, consider the following setup. kb_1 could contain general-purpose topic abstractions that are useful in everyday language as defined by Dornseiff. kb_2 could contain domain specific abstractions, for instance for the domain of natural languages. kb_3 could contain domain specific abstractions for the field of computer science and programming languages. Now, for a given query, the system has to decide which knowledge base to use. The desired behavior is the following: The general-purpose mapping should be available as a fallback. If the system has reasons to believe that the user is interested in one of the specialized domains then the according knowledge base should be used. Those mappings are mutually exclusive. Terms not specified in the specialized domains should be 'imported' from the general purpose one. In order to decide on the user's intentions, the history of queries in one interaction session is observed. We assume a classification context that is able to decide on the type of user interests. Such a context could be built by using classifiers like Hidden Markov Models for local reasoning and the history of query terms as knowledge base. The local consequence relation would contain the value of a class attribute, depending on the values in the query history, eventually triggering the context selection step suggested in [8]. For instance, a sequence of queries like (*"Communication", "Philosophy", "Language"*) is likely to be classified as in the natural language domain while a query sequence like (*"Optimization", "Compilers", "Language"*) is likely to be related to the computer science domain. Skipping the definitions of the involved logics L_i, and assuming a set Q of possible query terms, the ARMCS $M = (C_1, C_2, C_3, C_4)$ with

$$ACC_i = \text{the set of facts in } kb_i \text{ for } i \in \{1, 2, 3\}$$

$$KB_4 = Q^*, \quad kb_4 \in KB_4, \quad ACC_4 : KB_4 \mapsto \{\{class(i)\} | i \in \{unknown, 2, 3\}\}$$

$$C_1 = (L_1, kb_1, \emptyset, \{C_1 \leftarrow \mathbf{not}\, C_2, \mathbf{not}\, C_3\})$$

$$C_2 = (L_2, kb_2, \{t(W,T) \leftarrow 1 : t(W,T), \mathbf{not}\, 2 : t(W, \_).\}, \{C_2 \leftarrow c_4 : class(2).\})$$

$$C_3 = (L_3, kb_3, \{t(W,T) \leftarrow 1 : t(W,T), \mathbf{not}\, 3 : t(W, \_).\}, \{C_3 \leftarrow c_4 : class(3).\})$$

$$C_4 = (L_4, kb_4, \emptyset, \{C_4.\})$$

could be used to achieve the desired behavior under both brave and skeptical consequence as there is only one equilibrium.

5 Conclusion

We introduced activation rules for multi-context systems which allow to model the relevance of contexts given a state of beliefs. The complexity of reasoning with activation rules is in the same class as standard MCS reasoning.

One open research questions about (AR)MCS and multi-context reasoning in general is the following: What intuition and pragmatics can guide the design of bridge rules from contexts with different semantics, e.g. bridge rules from contexts with Closed World Assumption to contexts using the Open World Assumption? Therefore our next steps is to build a stable implementation of (AR)MCS and to provide this implementation to the community, hopefully reaching practitioners in knowledge representation and reasoning whose experience will help us in answering this difficult question.

Acknowedgements

The authors would like to thank the anonymous reviewers for their valuable feedback which helped to improve this paper.

References

1. Benerecetti, M., Bouquet, P., Ghidini, C.: Contextual reasoning distilled. J. Exp. Theor. Artif. Intell. 12(3), 279–305 (2000)
2. Besold, T.R., Mandl, S.: Integrating Logical and Sub-Symbolic Contexts of Reasoning. In: Proceedings of ICAART 2010 - Second International Conference on Agents and Artificial Intelligence. INSTICC Press (2010)
3. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
4. Brewka, G.: Nonmonotonic Reasoning: Logical Foundations of Commonsense. Cambridge University Press, Cambridge (1991)
5. Brewka, G., Eiter, T.: Equilibria in heterogeneous nonmonotonic multi-context systems. In: Proceedings of the National Conference on Artificial Intelligence, vol. 22, p. 385. AAAI Press/MIT Press, Menlo Park/Cambridge (2007)
6. Dornseiff, F., Quasthoff, U.: Der deutsche Wortschatz nach Sachgruppen. Walter de Gruyter (2004)
7. Eiter, T., Fink, M., Schüller, P., Weinzierl, A.: Finding explanations of inconsistency in multi-context systems. In: Knowledge Representation and Reasoning Conference (2010), http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1265
8. Mandl, S., Ludwig, B.: Coping with unconsidered context of formalized knowledge. In: Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L. (eds.) CONTEXT 2007. LNCS (LNAI), vol. 4635, pp. 342–355. Springer, Heidelberg (2007)
9. Syrjänen, T.: Implementation of local grounding for logic programs with stable model semantics. Tech. Rep. B18, Digital Systems Laboratory, Helsinki University of Technology (October 1998)

Pellet-HeaRT – Proposal of an Architecture for Ontology Systems with Rules*

Grzegorz J. Nalepa and Weronika T. Furmańska

Institute of Automatics,
AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
gjn@agh.edu.pl, wtf@agh.edu.pl

Abstract. The ongoing research on integration of rules and ontologies has resulted in multiple solutions, rule languages and systems. They differ in terms of aims and scope, semantics and architecture. The paper describes a proposal of a hybrid system combining a Description Logics reasoner with a forward-chaining rule engine. An integration of a dedicated tool for modularized rule bases, called HeaRT, and a widely-used DL reasoner Pellet, is sketched. An outline of main concepts and architecture of HeaRT-Pellet system is given, explained on an example case. The benefit of this solution is the ability to use a mature rule design, analysis and inference solution together with large fact bases from ontologies.

1 Introduction

Semantic Web research landscape is diverse and changes dynamically. There is an increasing interest in lightweight reasoning over RDF(S) data distributed on the Internet. There exist large RDF Triple Stores with SPARQL query endpoints, various projects work on accumulating and processing RDF data. Applications like semantic wikis or Semantic Web services utilize the technologies presented years ago in the *semantic stack* and put them into action. Variety of semantic techniques are in use, from semantic annotations through ontology querying to reasoning and rules. At the same time, knowledge representation based on Description Logics (DL) has a steadily increasing expressive power. DL-related research includes new versions of ontology language OWL with dedicated profiles. Augmenting ontologies with rules is possible within the OWL 2 RL profile. The rules, however, do not go beyond deductive reasoning.

Research on integrating rules and ontologies has resulted with multiple proposals. They differ in syntax and semantic, as well as the systems architecture. There exist solutions based on combining classical rule engines (expert system shells) with ontologies, as well as new languages designed specifically for OWL (e.g. SWRL.) Apart from using deductive rules to reason about ontologies, there are attempts to run production or reaction rules in ontology-based systems.

* The paper is supported by the BIMLOQ Project funded from 2010–2012 resources for science as a research project.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 143–150, 2010.

Within the HeKatE project (hekate.ia.agh.edu.pl) several solutions for rule-based systems have been developed. They provide means for visual design, formalized analysis and gradual refinement of rule-based systems with structured rule bases using the so-called XTT2 representation [10].

In this paper we propose a system combining the HeKatE rule engine HeaRT [8] with a DL reasoner. Primary goal of the prototype presented here is to run HeaRT inference over ontologies. The system has a hybrid architecture. The HeaRT tool is a control component, responsible for rules handling, selection and execution. In the rule format used by HeaRT the rules preconditions may include complex formulas based on Attributive Logic. These formulas describe relations among system attributes, which are mapped onto DL descriptions. For handling the relations between concepts and individuals, the Pellet DL reasoner is used.

The rest of the paper is organized as follows. In Sect. 2 the motivation is presented. HeaRT is briefly described in Sect. 3. The integration proposal is outlined in Sect. 4. The inference process is shown on an example in Sect. 5. Sect. 6 contains an evaluation of the proposal The paper is summarized in Sect. 7.

2 Motivation

Description Logics provide an effective formal foundation for application based on ontologies described with OWL. They allow for simple inference tasks, e.g. concept classification. Currently the main challenge for DL is the rule formulation. Other problems that so-far have rarely been considered in the Semantic Web research include effective design methodologies for large rule bases, as well as knowledge quality issues.

The XTT2 rule design language is an example of a rule design and analysis framework. It offers flexible knowledge representation for forward-chaining decision rules as well as visual design tools. Thanks to the formal description in ALSV(FD) logic a formalized rule analysis and inference is also possible [10].

The primary goal of this research is to allow the use of the XTT2 rule design framework for the Semantic Web rules. This would open up possibility to use the design tools for XTT2 to design Semantic Web rules, as well as exploit the existing verification and inference solutions. To achieve this goal the logical foundations of XTT2 and ontologies have been analyzed [9]. To summarize: DL enables for complex descriptions of objects in the universe of discourse and the relations between them. The main goal of ALSV(FD) is to provide an expressive notation for dynamic system state transitions in rule-based systems. As of knowledge representation, both logics describe the universe of discourse by identifying certain entities. In ALSV(FD) they are called attributes, in DL – concepts. In DL, the static part of a system is expressed in TBox part of a DL Knowledge Base. The actual state is represented by means of facts asserted in ABox. The knowledge specification with ALSV(FD) is composed of: state specification with facts, and transition specification with formulas building decision rules. Attributes can take single or set values at a moment.

Based on the analysis of knowledge representation with DL and ALSV(FD), a mapping proposal has been formulated. This is the basis for integration of HeKatE tools with DL-based applications. The far-reaching goal is to run HeaRT inference engine in ontology-based systems. This would be a generic approach where rules could be designed visually using the XTT2 representation and analyzed formally. Simultaneously, the rule-based reasoning could exploit facts from OWL ontologies where inference tasks use DL.

3 HeaRT Rule Engine

XTT2 (*eXtended Tabular Trees v2*) knowledge representation incorporates the attributive table format [10]. Similar rules are grouped within separated tables, and the system is split into a network of such tables representing the inference flow. Efficient inference is assured thanks to firing only rules necessary for achieving the goal, by selecting the desired output tables and identifying the tables necessary to be fired first. The visual table-based representation is automatically transformed into HMR (HeKatE Meta Representation), a corresponding textual algebraic notation, suitable for direct execution by the rule engine. An example excerpt of HMR is given below in Sect. 5.

HeKatE RunTime (HeaRT) [8] is a dedicated inference engine for the XTT2 rule bases. The engine is highly modularized. It is composed of the main *inference module* based on ALSV(FD). It supports four types of inference process, Data and Goal Driven, Fixed Order, and Token Driven [10]. HeaRT also provides a *verification module*, also known as HalVA (HeKatE Verification and Analysis). The module implements simple debugging mechanism that allows tracking system trajectory, and logical verification of models (several plugins are available, including completeness, determinism and redundancy checks). The verification plugins can be run from the interpreter or indirectly from the design environment using the communication module. The engine uses HMR as native format.

HeaRT offers a flexible network integration mechanism. It can operate in two modes, stand-alone and as a TCP/IP-based rule logic server. In particular, it allows for integration with a complete rule design and verification environment. The logic server can also be exposed as a network service in the SOA approach.

4 Pellet and HeaRT Integration Proposal

In [9] we proposed a hybrid framework for integrating Attributive Logic (ALSV(FD)) and Description Logic. We introduce a language called *DAAL* (*Description And Attributive Logic*), syntactically based on DL, but trying to capture ALSV(FD) semantics and thus enabling expressing ALSV(FD) models in DL. A novel idea in the DAAL framework is the existence of a *static* TBox with definitions of the domains, *Temporary Rule TBoxes* and *Temporary ABoxes. Temporary Rule TBoxes* express the preconditions of the system rules. During the execution of reasoning they are loaded into and unloaded from a reasoner; they are not a static part of an ontology. *Temporary ABoxes* correspond to the system states. As the system changes its state, new ABoxes replace the previous ones.

In the DAAL approach the attributes are modelled as DL concepts (corresponding to OWL classes). Every attribute in ALSV(FD) has its domain, which constraints the values of the attribute. In DL this kind of specification is done by means of TBox axioms. In order to be able to express a finite domain in DL, a *set of* constructor (denoted by $\mathcal{O}$) is needed. The formulas used in the rule preconditions specify the constraints of the attributes values. They constitute a schema, to which a state of a system in a certain moment of time is matched. Mapping from AL to DL consists in a translation of the AL formulas into TBox-like DL axioms. The basic transitions are fairly clear. For instance, the formula: $A_i = d$ (where A_i is an attribute and d its value) is logically equivalent with $A_i \in \{d\}$ and so we express it in DAAL as: $A_i \equiv \{d\}$ (instances of concept A_i belong to $\{d\}$). Another formula, $A_i \in V_i$ (where V_i is a *set of values*) constraint the set of possible values to the set V_i. This corresponds to the DL axiom: $A_i \equiv V_i$. The actual state of a system under consideration is modelled as a set of DL assertions (individuals in OWL). To express a value of a simple attribute there appears a single assertion. For generalized attributes there appear as many assertions as many values the attribute takes. A formula $A_i := d$ denotes that the attribute A_i *takes value* d at a certain moment. In DL this can be represented as an assertion in ABox, namely: $A_i(d_i)$. In the case of generalized attributes, there is no direct counterpart in the DL for an AL formula: $A_i := V_i$, where V_i is a set. However, the same meaning can be acquired in other way. Based on the assumption that V_i is a finite set of the form: $V_i = \{v_{i_1}, v_{i_2}, \ldots, v_{i_n}\}$ one can add *all* of the following assertions to the DL ABox: $A_i(v_{i_1}).A_i(v_{i_2}).\ldots A_i(v_{i_n})$.. The complete formalization of the translation algorithm is out of scope of this paper.

The *inference scenario* is as follows: At a given moment of time, the state of the system is represented as a conjunction of assertions. In order to check the rules' preconditions satisfiability, appropriate inference tasks have to be executed. For each rule *consistency checking* of the state assertions w.r.t the rule preconditions is performed. If the consistency holds, the rule can be fired.

A practical architecture of such a hybrid reasoning framework consists of a dedicated XTT2 tool – HeaRT, a control component, and a DL reasoner – Pellet. The architecture can be observed in Fig. 1. The inference process is controlled by the HeaRT tool. It takes care of rule handling, including context-based rule selection and execution. Pellet's task is to check the rule preconditions. More specifically, it checks consistency of the ontology built of the actual state of the system and particular rules' preconditions. In a loop appropriate rules are loaded into the DL reasoner together with the actual state of the system. Each rule is temporary joined with existing TBox (in which the definitions of the concepts are stored). The state is a *temporary ABox*. The DL reasoner checks the consistency of the ontology resulted from the TBox and ABox. If the ontology is consistent, then the rule can be fired. The rule axioms are then unloaded and the loop continues. Rules are able to change the knowledge base of the system. Adding

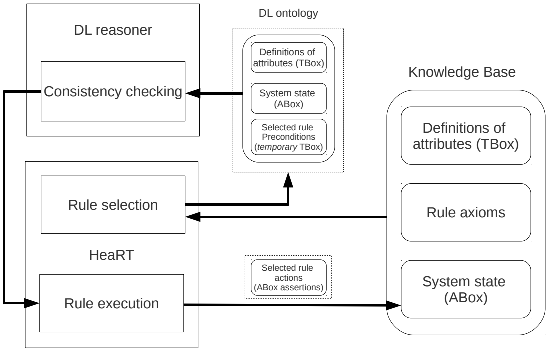

DL reasoner
Consistency checking
DL ontology
Definitions of attributes (TBox)
System state (ABox)
Selected rule Preconditions (*temporary* TBox)
Knowledge Base
Definitions of attributes (TBox)
Rule axioms
System state (ABox)
Rule selection
HeaRT
Rule execution
Selected rule actions (ABox assertions)

Fig. 1. Hybrid system combining HeaRT and Pellet

and removing facts is allowed. These operations generate a new ABox which represents the new state of the system.

Pellet is run as a DIG server and HeaRT works as a DIG client. DIG is a standard interface to Description Logics reasoners. It defines a concept language and a set of basic operations on DL ontologies. DIG defines an XML encoding intended to be used over HTTP protocol. A DIG client sends HTTP POST requests with XML-encoded messages. There are three main sorts of actions that a DIG server can utilize: knowledge base management operations, *tell* operations to make assertions, and *ask* operations to query the knowledge base.

Once appropriate rules are loaded two messages in DIG format are built: 1) `tells` request with definitions of concepts in the rule precondition and assertions about the system state (use of: `defConcept, defIndividual, equivalence`) 2) `asks` request with a question whether the previously given ontology is consistent (`consistentKB`) HeaRT sends messages to Pellet and based on the responses it decides whether execute the rule. Communication is implemented with the SWI-Prolog HTTP library. The prototype can be controlled from command line.

5 Example Case

Let us analyze an example use case which describes a simple diagnosis tool.[1] It provides knowledge and rules that help a doctor to decide whether a patient has tachycardia or bradycardia, based on his blood pressure, heart rate, and general heart condition. The system also tells which drugs are supposed to be given to the patient, and based on the patient's response to those drugs, helps decide on further steps in a treatment. An example rule set in HMR is:

```
xrule 'TachyTreatment2'/1:
    [sinus_rate eq 1, qrs_action eq restore_sin_rythm]
==> [tachy_treatment_finished set observation].
```

[1] See https://ai.ia.agh.edu.pl/wiki/hekate:cases:hekate_case_cardio:start

```
xrule 'TachyTreatment2'/2:
    [sinus_rate eq 0, qrs_action eq restore_sin_rythm]
==> [tachy_treatment_finished set consultation].
xrule 'TachyTreatment2'/3:
    [sinus_rate in [any], qrs_action in [none,amiodaron,beta_blockers]]
==> [tachy_treatment_finished set observation].
```

Attributes are defined in HMR specifying their usage and domains, e.g.:

```
xtype [name: qrs_actions, base: symbolic,
    desc: 'Actions supposed to be taken after QRS complx was examined',
    domain: [restore_sin_rythm,consult_specialist,
             amiodaron,beta_blockers,none] ].
xattr [name: qrs_action, ..., type: qrs_actions ].
```

The attributes' definitions are translated into DIG in a simplified manner:

```
<defClass URI="Qrs_action">
<equivalentClasses>
 <defconcept name="Qrs_action"> <class URI="Qrs_action"/>
 <iset>
   <individual URI="restore_sin_rythm"/> <individual URI="amiodaron"/>
   <individual URI="consult_specialist"/><individual URI="beta_blockers"/>
   <individual URI="none"/> </iset> </equivalentClasses>
```

Due to space limitations we only show example translations of rule preconditions. The first rule is translated into the following DIG requests (some declarations and URI addresses are removed for clarity):

```
<equivalentClasses>
  <defconcept name="Sinus_rate"> <class URI="Sinus_rate"/>
  <iset>
    <individual URI="1"/> </iset> </equivalentClasses> <equivalentClasses>
  <defconcept name="Qrs_action">
  <class URI="Qrs_action"/>
  <iset>
    <individual URI="restore_sin_rythm"/> </iset> </equivalentClasses>
```

The third rule's preconditions with state encoding are translated as follows:

```
<equivalentClasses>
  <defconcept name="Qrs_action"> <class URI="Qrs_action"/>
  <iset>
    <individual URI="amiodaron"/> <individual URI="beta_blockers"/>
    <individual URI="none"/> </iset> </equivalentClasses>

<instanceOf>
  <individual URI="amiodaron"/> <class URI="Qrs_action"/> </instanceOf>
```

The third rule contains two attributes, but only one of them is checked, because the other can take *any* value. Once the connection with Pellet is established, a new knowledge base has to be requested. Then, if the rules are chosen, the above messages are send to the DL reasoner. To check if the rule can be fired HeaRT asks if the knowledge base is consistent. If it is the case Pellet sends a positive response and the rule can be fired.

6 Evaluation and Related Research

The current architecture is a proof-of-concept solution. It shows that the integration of the rule component with the DL reasoner can be achieved. However, due to the HTTP-based interface the performance of the prototype is not satisfactory. This is why alternative implementations, using other APIs are considered.

Combining rules and ontologies in a heterogeneous architecture was realized in several systems, for instance $\mathcal{AL}$-log [3] and CARIN [6]. There were hybrid systems that comprised of a structural part and a deductive one. The former used Description Logic, while the latter included Datalog rules.

Interaction between two knowledge representation or programming paradigm can also be observed in combining DL with Answer Set Programming [5]. It results in a hybrid system with bidirectional communication. The interaction between the subsystems is realized by exchanging ground consequences between the two components. The resulting semantics is incompatible with First Order semantics. Another approach is presented in Hybrid MKNF (Minimal Knowledge with Negation as Failure) knowledge bases [7]. The hybrid solution aims at integrating closed-world reasoning and open-world reasoning. The framework, investigated theoretically, is planned to be implemented in the KAON2 reasoner.

Describing states of a dynamic system using DL constructs implies the problem of updating the state description. Integrating action formalisms and DL may lead to more expressive representation yet with decidable reasoning [2]. The idea of updating ABox over time has been investigated [4] and appropriate DL languages have been defined. In our solution the updated ABoxes are treated by a separate component, so there is no direct requirement for ABox updates support.

The idea presented here has common points with SAIL architecture for situation awareness systems [1]. Similarities lay in representing current states of a conceived world by means of DL ABox assertions and updating them over time. Moreover, a DL reasoner is a component of a system and is invoked by a control component. In [1] the idea of a *sequence of ABoxes* is introduced. The background knowledge consists of an ontology expressed in TBox and static knowledge expressed in ABox. This together with the latest loaded ABox is a base for inferring new facts which are stored in a new ABox.

7 Conclusions and Future Work

In this paper a heterogeneous architecture for Semantic Web rule reasoning was proposed. It is based on using the rule framework for the XTT2 rule representation, together with an ontology framework. The XTT2 inference engine HeaRT is combined with a DL reasoner Pellet. The benefit of this solution is the ability to use a mature rule design, analysis and inference solution together with large fact bases gathered in ontologies. This generic approach could contribute to speeding up the adoption of rule solutions for Semantic Web applications.

Future work includes a plugin for Protegé, as well as alternative mappings from ALSV(FD) into DL (and DIG). In these mappings rule preconditions are built of simple concepts assertions, not TBox-like axioms. The approach is the same as in DL-rules, which regards only existing individuals. In order to check the rule preconditions, alternative `asks` request would be sent: `instances` (of a concept) to check whether given attribute takes the given value. Another idea to be investigated is expressing attributes with Datatype Properties (OWL naming convention). In DIG they are expressed with `attribute`. In this case expressions for domain values could be used in rule preconditions (e.g. `intmin, intmax, intrange`). Further investigation of DIG 2.0 and support for OWL2 constructs is needed.

References

1. Baader, F., Bauer, A., Baumgartner, P., Cregan, A., Gabaldon, A., Ji, K., Lee, K., Rajaratnam, D., Schwitter, R.: A novel architecture for situation awareness systems. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS, vol. 5607, pp. 77–92. Springer, Heidelberg (2009)
2. Baader, F., Lutz, C., Miličic, M., Sattler, U., Wolter, F.: Integrating description logics and action formalisms: first results. In: AAAI 2005: Proceedings of the 20th National Conference on Artificial Intelligence, pp. 572–577. AAAI Press, Menlo Park (2005)
3. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: AL-log: integrating datalog and description logics. J. of Intelligent and Cooperative Information Systems 10, 227–252 (1998)
4. Drescher, C., Liu, H., Baader, F., Guhlemann, S., Petersohn, U., Steinke, P., Thielscher, M.: Putting abox updates into action. In: Ghilardi, S., Sebastiani, R. (eds.) FroCoS 2009. LNCS, vol. 5749, pp. 214–229. Springer, Heidelberg (2009)
5. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. Artificial Intelligence 172(12-13) (2008)
6. Levy, A.Y., Rousset, M.C.: Combining horn rules and description logics in CARIN. Artif. Intell. 104(1-2), 165–209 (1998)
7. Motik, B., Horrocks, I., Rosati, R., Sattler, U.: Can OWL and logic programming live together happily ever after? In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 501–514. Springer, Heidelberg (2006), `http://dx.doi.org/10.1007/11926078_36`
8. Nalepa, G.J.: Architecture of the heart hybrid rule engine. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS (LNAI), vol. 6114, pp. 598–605. Springer, Heidelberg (2010)
9. Nalepa, G.J., Furmańska, W.T.: Integration proposal for description logic and attributive logic– towards semantic web rules. Transactions on Computational Collective Intelligence 1(1) (to be published 2010)
10. Nalepa, G.J., Ligęza, A.: HeKatE methodology, hybrid engineering of intelligent systems. International Journal of Applied Mathematics and Computer Science 20(1), 35–53 (2010)

Putting People's Common Sense into Knowledge Bases of Household Robots

Lars Kunze, Moritz Tenorth, and Michael Beetz

Intelligent Autonomous Systems Group
Department of Informatics
Technische Universität München
{kunzel,tenorth,beetz}@in.tum.de

Abstract. Unlike people, household robots cannot rely on commonsense knowledge when accomplishing everyday tasks. We believe that this is one of the reasons why they perform poorly in comparison to humans. By integrating extensive collections of commonsense knowledge into mobile robot's knowledge bases, the work proposed in this paper enables robots to flexibly infer control decisions under changing environmental conditions. We present a system that converts commonsense knowledge from the large Open Mind Indoor Common Sense database from natural language into a Description Logic representation that allows for automated reasoning and for relating it to other sources of knowledge.

1 Introduction

Household robots are expected to accomplish an open-ended set of everyday tasks. Hence, they need to understand under-specified commands given by humans and find out on their own what to do, what to look out for, where to search for objects and so on. Many of these decisions depend on the context at hand and are difficult to foresee when writing the robot control program. In addition, the sheer number of possible situations makes hard-coding them practically impossible. Human-like commonsense knowledge would give robots the ability to *infer* the right decisions and become more flexible. This raises the questions how to acquire the commonsense knowledge that humans possess, how to represent it, and how to enable robots to make inferences based on that knowledge.

The approach we propose in this paper is to use existing collections of commonsense knowledge, such as the Open Mind Indoor Common Sense database (OMICS, [1]), and convert them into first-order representations the robot can use. The OMICS project collects commonsense knowledge described in natural language from Internet users. This information is contributed by users by completing template sentences in web forms. For example, a template sentence like "When a potted plant is dry, then the ______ becomes ______." is used for capturing information about causal relationships. These fragments cover different areas, like the objects found in different rooms, the correct action to take in a situation, or possible problems that can occur while performing a task. For most humans, such information appears trivial, and they can intuitively answer such questions. For a robot, however, this gives valuable information that can hardly be obtained

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 151–159, 2010.

otherwise. The information entered in the web forms is reviewed by the OMICS project team, post-processed and stored in a relational database.

However, this information cannot directly be used by robots: It is written in colloquial English language, which makes it hard for a robot to interpret it (for example, to find out that “turn on”, “switch on” and “start” actually mean the same thing), to relate it to other sources of information like semantic environment maps, and to perform automated reasoning. Therefore, our system first transforms the statements into a formal representation in Description Logic by resolving the meaning of words and mapping them to ontological concepts. This allows to use well-established reasoning techniques, and further enables the robot to integrate the commonsense knowledge into its knowledge processing system [2].

The rest of the paper is structured as follows: We start with an overview of related work (Section 2) and a description of the system architecture (Section 3), explain the extraction and formalization of knowledge from OMICS (Section 4) and elaborate on the integration into the robot’s knowledge base (Section 5). Section 6 gives examples of how the robot uses the knowledge. Finally, Section 7 concludes this paper by discussing some of the problems encountered and some remaining challenges for further research.

2 Related Work

Equipping computers (or robots) with common sense is not a new endeavor in Artificial Intelligence: Early work has been done by McCarthy already in 1959 [3]. However, the problem is still far from being solved. As Minsky [4] pointed out, computers require large amounts of commonsense knowledge in order to become concerned with human affairs. Furthermore, he points to the problem that much of the commonsense knowledge has never been described because its information always seemed so obvious. This problem is addressed within the knowledge capturing projects Open Mind Common Sense [5] and Open Mind Indoor Common Sense [1] that acquire commonsense knowledge from web users. Several approaches have been proposed to make use of this information (ConceptNet [6], LifeNet [7], and [8]) for textual-information management, modeling of human activities, or robot task planning. Gupta and Pedro [9] use Bayesian networks to infer the most likely response in a given situation based on knowledge from OMICS. Pentney et al. [10] propose to use the knowledge, represented in a probabilistic model, for tracking the state of the world on the basis of sensor data. However, none of these approaches has tackled the problem of converting the knowledge from natural language to well-defined ontological concepts, which is required for automated reasoning.

3 System Overview

Figure 1 illustrates the process of extracting, formalizing, and reasoning about commonsense knowledge from the OMICS project. First, the system applies natural language processing techniques like part-of-speech tagging and syntax parsing to the knowledge. Then, it resolves the word’s meanings to cognitive synonyms (synsets) in WordNet, and exploit mappings between these synsets and concepts in OpenCyc, which already

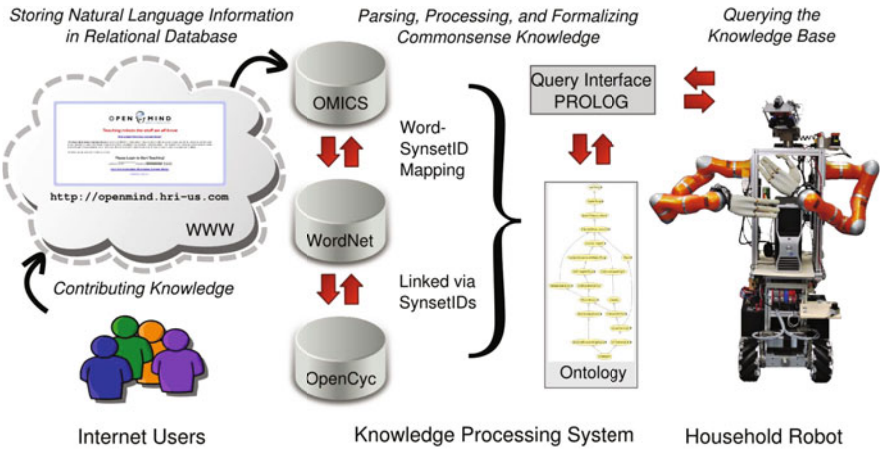

Storing Natural Language Information in Relational Database
Parsing, Processing, and Formalizing Commonsense Knowledge
Querying the Knowledge Base
http://openmind.hri-us.com
WWW
Contributing Knowledge
OMICS
WordNet
OpenCyc
Word-SynsetID Mapping
Linked via SynsetIDs
Query Interface PROLOG
Ontology
Internet Users
Knowledge Processing System
Household Robot

Fig. 1. Overview of the proposed system. Knowledge that is contributed by Internet users is transformed from natural language into a logical representation and can be queried by the robot control program

exist for thousands of concepts. Based on OpenCyc's concept definitions, it generates a formal representation describing the OMICS database relations in Description Logic which becomes part of the robot's knowledge base and can be queried via a Prolog-based interface.

4 Formalizing the Knowledge in OMICS

This section describes how the knowledge is converted from the sentences in natural language in OMICS's semi-structured database tables to meaningful first-order representations.

4.1 Natural Language Processing

In our system, the knowledge is represented as object- or action-centered first-order representations, which we need to extract from the OMICS database. For simple relations, like *locations(Object,Room)*, the database columns already provide this information and describe objects, actions, or properties. More complex relations, however, like *problems(Task,Problem)*, are described by short natural language phrases. Figure 2 (left) lists some examples. These phrases need to be interpreted in order to extract information about objects, actions, and properties. First, each word is tagged with its syntactic category, and phrases like a verb phrase (VP) or a noun phrase (NP) are identified using the Stanford Parser [11], a probabilistic context-free grammar parser. The part-of-speech tags (PoS) describe the role (or syntactic category) an expression has in a sentence. Figure 2 (right) shows the parse trees and the PoS annotations[1] of the words in the command "load the dishwasher" and the related problem "dishwasher is full".

[1] DT - Determiner; NN - Noun, singular or mass; VB - Verb, base form; VBZ - Verb, 3rd person singular present; JJ - Adjective.

Task	**Problem**
load the dishwasher	dishwasher is full
load the dishwasher	cannot open dishwasher
unload the dishwasher	dishes not dry
unload the dishwasher	dishes are dirty

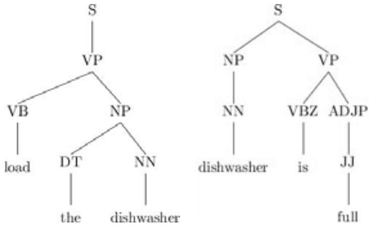


Fig. 2. Left: Examples of tasks and their respective problems taken from the OMICS database. Right: Parse trees of the first task-problem tuple. The part-of-speech information is utilized for extracting object-/action-centered information from the natural language descriptions.

Having assigned the PoS tags, the system knows if a word describes an object (usually a noun), an action (in commands normally verbs), or a property (an adjective). This information is used to further interpret the relations in the database. For instance, in the relation *problems(Task,Problem)*, a task is described by an action and an object, whereas problem specifications consist of an object and a property. The relation can thus be written as

$$problems(\overbrace{Object1, Action}^{Task}, \overbrace{Object2, Property}^{Problem}).$$

and the first example in Figure 2 (left) becomes

$$problems(\overbrace{dishwasher, load}^{load\ the\ dishwasher}, \overbrace{dishwasher, full}^{dishwasher\ is\ full}).$$

In this form, the *problems* relation only comprises actions, objects, and properties and can thus be represented in our knowledge processing system.

4.2 Word Sense Resolution

To make the extracted information usable for automated reasoning, the words need to be transformed from (ambiguous) natural language to well-defined ontological concepts. This transformation is done using the lexical database WordNet, in which words are grouped into sets of cognitive synonyms (synsets). For thousands of these synsets, there are mappings to the OpenCyc ontology, which are used to transform words to ontological concepts.

During this mapping process, the system resolves ambiguities caused by different words that mean the same thing. For example, the expressions “turn on” and “switch on” are both mapped to concept *TurningOnPoweredDevice*. Table 1 shows some examples of synonymous words that are mapped to the same concept. Such a mapping is very important to make use of the full knowledge in OMICS: Without this resolution of the words’ meanings, the system could not detect that e.g. statements about someone mopping or wiping the floor can be combined. Since the knowledge was acquired by many untrained users, the variation in words used is rather high, making the word sense resolution even more important.

Table 1. Mappings for different word types

Type	OMICS	Synset-ID	Cyc Name
Action	turn on	V01468130	*TurningOnPoweredDevice*
	switch on	V01468130	*TurningOnPoweredDevice*
	clean	V01490246	*Cleaning*
	mop	V01352869	*Cleaning*
	sanitize	V01207357	*Cleaning*
	steam clean	V01207630	*Cleaning*
	wipe up	V01352869	*Cleaning*
Property	on	A01599324	*DeviceRunning*
	turned on	A02066470	*DeviceRunning*
	dirty	A00394641	*Dirty*
	unclean	A00394641	*Dirty*
	soiled	A00394641	*Dirty*

A second kind of ambiguity is caused by different meanings of a word: For example, the word "dishwasher" can denote a household appliance or a person cleaning the dishes. Technically, this means to select the right concept the word is to be mapped to. The system uses different sources of information for this task: The parsing and PoS tagging determined the type of a word, so that the algorithm can determine if e.g. "clean" is used as a verb (denoting an action) or as an adjective (describing a property).

If several meanings of a word fall within a single syntactic category, the ambiguities are harder to resolve, for example for the word "dishwasher". However, we know from OMICS that the word denotes an object, and can use the Cyc ontology to select the household appliance and discard the second meaning since a person is not an object. In general, we constrain the resulting ontological concepts of objects and actions to be subclasses of *PartiallyTangible* and *ActionOnObject* respectively. Furthermore, rooms in the *locations(Object,Room)* relation need to be a subclass of *RoomInAConstruction*.

Finally, the following numbers should give the reader a basic idea about word sense resolution. In total, the *locations(Object,Room)* table holds 416 distinct entries for objects that are typically found in a kitchen. From these, we could automatically map 247 object descriptions to 178 distinct ontological concepts. The 169 entries that could not be resolved directly were not found in the WordNet search. At least 18 of these unmapped entries could be mapped after truncating the ending *'s'*, i.e. making plural expressions singular. By looking at the remaining objects most of them are described by two or more words, e.g. "milk bottle", "tea pitcher" or "box of donuts". For improving the performance of our system we will consider more sophisticated techniques like stemming and compound processing.

5 Integration into the Robot's Knowledge Base

For our robot to use the knowledge, we need to represent the relations inside KnowRob [2], its knowledge processing system. KnowRob uses OWL-DL [12], a language based on Description logic (DL) [13], to represent the robot's knowledge. Concepts and relations in KnowRob are the same as in OpenCyc, so that the mappings determined in the previous step can be used here. These concepts are now related according to relations in OMICS and added to the knowledge base. For example, the simple relation *parts(dishwasher,motor)* is transformed to

$$\begin{aligned}&parts(dishwasher, motor)\\&\quad\Rightarrow ((wordnetCyc(dishwasher, Dishwasher) \wedge wordnetCyc(motor, Engine))\\&\qquad\Rightarrow properPhysicalParts(Dishwasher, Engine))\end{aligned}$$

where the words “dishwasher” and “motor” are mapped to the concepts *Dishwasher* and *Engine* respectively. These concepts are related by the role *properPhysicalParts*. A more complex example of a causal relationship is given by

$$\begin{aligned}&causes(dishwasher, used, cup, clean)\\&\quad\Rightarrow ((wordnetCyc(dishwasher, Dishwasher) \wedge wordnetCyc(used, UsedArtifact) \wedge\\&\qquad wordnetCyc(cup, DrinkingMug) \wedge wordnetCyc(clean, Clean))\\&\qquad\Rightarrow \textit{causes-SitSit(}\\&\qquad\quad SituationInWhich(Dishwasher, UsedArtifact),\\&\qquad\quad SituationInWhich(DrinkingMug, Clean)))\end{aligned}$$

where the resulting concepts are related by the role *causes-SitSit*. Since roles in DL can only represent relations that have an arity of exactly two, we represent object-property tuples as sub-concepts of type *Situation* and object-action tuples as sub-concepts of type *ActionOnObject*.

6 Applications

By integrating the OMICS knowledge into an ontology, we can perform reasoning about similar objects and/or sub-classes of objects. Additionally, KnowRob provides methods for accessing sensor data, for instance an semantic environment model and objects detected by the robot’s vision system. These instances can be related to the class knowledge obtained from OMICS for determining in which room one can typically find an

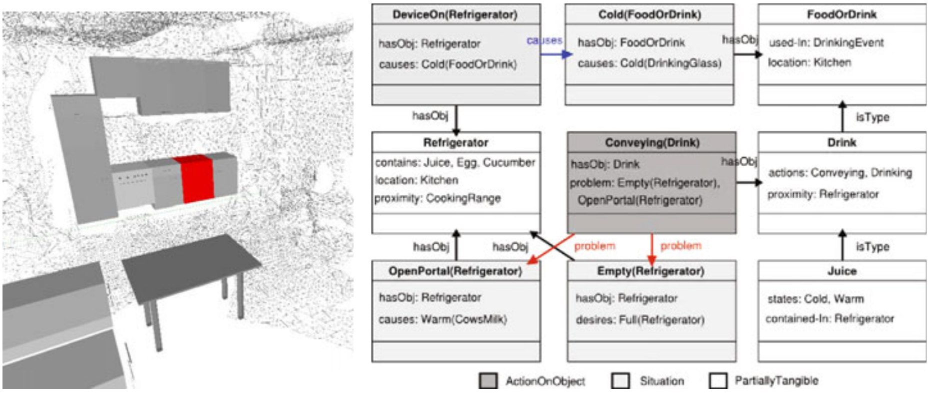


Fig. 3. Left: Refrigerator located in the integrated semantic environment map. Right: Simplified view on the ontology that was generated from the OMICS database.

object, what are possible object states, or what are potential problems when applying a certain action to an object.

The following scenario illustrates what kind of knowledge from the OMICS database can be used by a household robot. We consider a situation where the robot is supposed to serve a cold drink to a person. To accomplish the task the robot has to infer where in its environment it can find a drink that is cold. Some of the relevant knowledge that is described in the text and is referred in the example queries is depicted in Figure 3.

The first task of the robot is to resolve the meanings of "cold" and "drink", which are *Cold* and *Drink* respectively. Then it has to identify causes that have the effect of a cold drink. In the robot's knowledge base there exist no causal rule which explicitly involves a situation of type *Cold(Drink)*, but however, there exist a more general rule that involves the concept *Cold(FoodOrDrink)*. Since the robot knows from its ontology that *Drink* is a specialization (or sub-class) of *FoodOrDrink* it can make use of more general rule, which reads as follows:

$$\mathit{DeviceOn(Refrigerator)} \Rightarrow_{causes} \mathit{Cold(FoodOrDrink)}.$$

From the above rule the robot infers that an instance of type *Refrigerator* with property *DeviceOn*, can cause cold drinks, which is shown by the following query:

```
?- subClassOf('Drink', SuperCls),
   situation(Sit), hasObj(Sit, SuperCls), hasProperty(Sit,'Cold'),
   owl_restriction_on(Cause, restriction(causes,some(Sit))).

   SuperCls  = 'FoodOrDrink';
   Sit       = 'Cold(FoodOrDrink)',
   Cause     = 'DeviceOn(Refrigerator)'.
```

Furthermore, the robot knows from OMICS's (spatial-) relationships that things like *Egg-Foodstuff, Cucumber-Foodstuff,* and *Juice* are typically contained (*in-ContGeneric*) in a *Refrigerator*. Since *Drink* is a super-class of *Juice*, the robot deduces that it could get an instance of type *Juice* from the fridge which, regarding the causal rule, should be cold. From OMICS' *locations* relation the robot knows that a *Refrigerator* is typically found in a *Kitchen*. By accessing the semantic environment map through KnowRob, the robot can precisely locate an instance of type *Refrigerator*. The result of the following query is depicted in Figure 3 (left).

```
?- owl_individual_of(Fridge, 'Refrigerator').

   Fridge = fridge1.
```

However, in the case that the fridge has not been mapped yet, the robot can use OMICS's *proximity* relation and try to find the fridge by localizing objects near-by, e.g. instances of type *CookingRange* or *Freezer*. Finally having located the fridge in the kitchen, the robot can verify with its vision system whether there is some juice inside. While getting the cold drink from the fridge the robot can watch out for potential problems that are related to the *Conveying(Drink)* action, e.g. *Empty(Refrigerator)* or *OpenPortal(Refrigerator)*, which are retrieved by the query:

```
?- actionOnObject(Action,'Drink'),
   owl_restriction_on(Action, restriction(problem,some(Problem))).
```

```
    Action  = 'Conveying(Drink)',
    Problem = 'Empty(Refrigerator)' ;
    Action  = 'Conveying(Drink)',
    Problem = 'OpenPortal(Refrigerator)'.
```

Although the overall reasoning process in this scenario does not work fully automatically, the emerging sub-queries can be answered based on the knowledge retrieved from OMICS. Furthermore, the scenario should point out the importance and omnipresence of commonsense reasoning within everyday tasks.

7 Discussion and Conclusions

In this paper, we described how commonsense knowledge that is acquired from web users and represented in natural language is transformed into first-order representations which enable robots to reason about everyday tasks. We showed how we extract object- and action-centered information from natural language phrases stored in the OMICS database, map it to ontological concepts, integrate it in the knowledge processing system KnowRob, and use it for answering task-relevant queries.

Though the processes described in this work are fully automated, some remaining flaws are best resolved by manual revision: First, the mapping between OMICS and OpenCyc is not complete, meaning that some words or expressions cannot be transformed to ontological concepts. And second, our rather simple word sense disambiguation strategies may fail to select the correct meaning, but this problem is beyond the scope of this work. Regarding the knowledge provided by OMICS, it should be noted that the information, as it was entered from ordinary Internet users, is first not complete, second redundant, and third even sometimes contradictory. By the techniques presented in this paper, we are able to resolve some ambiguities and to assemble all assertions for a topic by linking the words to concepts — otherwise, the robot would see pieces of information using different words as not related. However, completely resolving these issues remains an open challenge.

Acknowledgments

We would like to thank the Honda Research Institute USA Inc. for providing the OMICS data. This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems (CoTeSys)*, see also `www.cotesys.org`.

References

1. Gupta, R., Kochenderfer, M.J.: Common Sense Data Acquisition for Indoor Mobile Robots. In: AAAI, pp. 605–610 (2004)
2. Tenorth, M., Beetz, M.: KnowRob — Knowledge Processing for Autonomous Personal Robots. In: IEEE/RSJ International Conference on Intelligent RObots and Systems (2009)
3. McCarthy, J.: Programs with Common Sense. In: Proceedings of the Teddington Conference on the Mechanization of Thought Processes, London, Her Majesty's Stationary Office (1959)

4. Minsky, M.: Commonsense-based Interfaces. ACM Commun. 43, 66–73 (2000)
5. Singh, P., Lin, T., Mueller, E.T., Lim, G., Perkins, T., Zhu, W.L.: Open mind common sense: Knowledge acquisition from the general public. In: CoopIS/DOA/ODBASE (2002)
6. Liu, H., Singh, P.: ConceptNet: A Practical Commonsense Reasoning Toolkit. BT Technology Journal 22, 211–226 (2004)
7. Singh, P., Williams, W.: LifeNet: A Propositional Model of Ordinary Human Activity. In: Workshop on Distributed and Collaborative Knowledge Capture, DC-KCAP (2003)
8. Shah, C., Gupta, R.: Building Plans for Household Tasks from Distributed Knowledge. In: Workshop on Modeling Natural Action Selection at IJCAI 2005 (2005)
9. Gupta, R., Pedro, V.C.: Knowledge Representation and Bayesian Inference for Response to Situations. In: AAAI 2005 Workshop on Link Analysis (2005)
10. Pentney, W., Popescu, A.M., Wang, S., Kautz, H.A., Philipose, M.: Sensor-Based Understanding of Daily Life via Large-Scale Use of Common Sense. In: AAAI (2006)
11. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA (2003)
12. W3C: OWL Web Ontology Language (2004), http://www.w3.org/TR/owl-ref/
13. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge (2003)

Recognition and Visualization of Music Sequences Using Self-organizing Feature Maps

Tobias Hein[1] and Oliver Kramer[2]

[1] Department of Computer Science, Technische Universität Dortmund,
44227 Dortmund, Germany
[2] International Computer Science Institute,
Berkeley, CA 94704, USA

Abstract. Music consists of sequences, e.g., melodic, rhythmic or harmonic passages. The analysis and automatic discovery of sequences in music has an important part to play in different applications, e.g., intelligent fast-forward to new parts of a song, assisting tools in music composition, or automated spinning of records. In this paper we introduce a method for the automatic discovery of sequences in a song based on self-organizing maps and approximate motif search. In a preprocessing step high-dimensional music feature vectors are extracted on the level of bars, and translated into low-dimensional symbols, i.e., neurons of a self-organizing feature map. We use this quantization of bars for visualization of the song structure and for the recognition of motifs. An experimental analysis on real music data and a comparison to human analysis complements the results.

1 Introduction

The automatic recognition of sequences in music is no easy undertaking. Even worse, the problem itself is not easy to define. What is the definition of a music sequence? We abstain from a precise and mathematical or a musical definition, but settle for a vague definition of a music sequence as a melodic, rhythmic or harmonic passage that can be distinguished from the rest of a song, and may be repeated with small variations, i.e., pitch levels or different vocals. In music information retrieval and everyday tasks sequence analysis depends on the application. More important, this may range from the intelligent fast-forward to new sequences of a song to assisting tools in music composition and automated spinning of songs.

In this paper we introduce an approach for visualization and motif recognition of music sequences based on the discretization with Kohonen's self-organizing feature map (SOM) [5]. This work is structured as follows. In Section 2 we shortly summarize related work in the field of music information retrieval with SOMs. Section 3 describes the basis of our approach, i.e., the discretization of high-dimensional music sequences to sequences of symbols. In Section 4 we describe the subsequent visualization of sequences, while Section 5 describes the recognition of short music sequences in songs.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 160–167, 2010.

2 Related Work

SOMs have been used in music information retrieval for various purposes. Feiten and Günzel [2] use a SOM for the automatic indexing of a sound library. Harford [4] uses a SOM for melody retrieval. His system consists of two nets that represent pitch sequences and rhythms. Both networks are connected via associative maps. Rauber *et al.* [11] create a musical archive based on sound similarity. Frequency based features are extracted and transformed according to psychoacoustic models. A hierarchical SOM is used to create a hierarchical organization. The system offers an interface for interactive exploration as well as music retrieval. Pampalk *et al.* [9] use this psychoacoustic SOM-based model for the exploration of music libraries without the need of a prior manual genre classification. They construct a visual interface that is based on the topographic property of SOMs, i.e., similar genres are neighbored on the map. Recently, Dickerson and Ventura [1] have introduced a user query-by-content recommendation system for consumers based on SOMs. They extend the system by a quasi-supervised design. Our approach has some similarities with string-matching approaches. The history of string-based approaches in music information retrieval began with the query-by-humming system of Ghias *et al.* [3] concentrating only on pitch information. An approach based on MIDI-data has been introduced by Kosugi *et al.* [6]. To the best of our knowledge, the combination of SOM-based quantization for visualization and motif search on contemporary music, and with state-of-the art music features has not been introduced and analyzed before.

3 Feature Quantization with Self-organizing Maps

The basis of our approach is the discretization of a sequence of high-dimensional feature vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ to a sequence of symbols. Mapping and quantization allow the application of string search algorithms, and the assignment of colors for visualization. After feature extraction and normalization, the high-dimensional feature vectors are quantized with the help of a SOM. The idea of using a SOM is motivated by the fact that SOMs allow a *smoother* quantization with regard to the data sample distribution than simply dividing the feature space into grids, or distributing codebook vectors equidistantly. A SOM is able to capture the intrinsic structure of the data. An important property is that neighbored feature vectors in the high-dimensional data space are neighbored on the low-dimensional map as well.

3.1 Short Introduction to SOMs

We summarize the SOM concept and the corresponding learning rule in the following. The SOM by Kohonen [5] distributes K neurons $\mathbf{n}_1, \ldots, \mathbf{n}_K$ in a feature space $\mathcal{F}$. Each neuron possesses a neural weight vector $\mathbf{w}_i \in \mathcal{F}$ and a position on the map, where they are usually arranged as a 1-dimensional chain or a 2-dimensional grid. In the unsupervised training phase, for each data sample $\mathbf{x}_i \in \mathcal{F}$, the closest neural weight vector $\mathbf{w}_*$ is computed and its weights as well as the weights of the neighbored neurons are pulled into the direction of $\mathbf{x}_i$ by $\mathbf{w}'_k = \mathbf{w}_k + \eta \cdot h(\mathbf{w}_*, \mathbf{w}_k) \cdot (\mathbf{x} - \mathbf{w}_k)$, where η is the learning rate and $h(\cdot,\cdot)$ is the neighborhood function that defines the distance to the weight vector $\mathbf{w}_*$ of the winner neuron on the map. The learning rate η is usually decreased during the learning phase, e.g., by $\eta(t) = \eta(0) \cdot \exp(\frac{-t}{c})$ with constant c and iteration t. The neighborhood function h depends on a radius r that is also decreased in the course of the learning process. Often, the Mexcian hat or the Gaussian function is used for h.

3.2 Features and "Symbolification"

All songs are stored in the *Wave*-format, and are recorded in mono, with a sample rate of 22.05 KHz and 16-bit. The first step of the feature extraction is the computation of the song tempo. For this purpose we have used a BPM-tracker tool[1]. The audio signal is subject to segmentation, such that each segment corresponds to one bar of the song. Therefore, the audio-signal is segmented into blocks of length $q = 4 \cdot f_s \cdot \frac{60}{t}$ with sample rate f_s for a 4/4-measure. We have experimented with various lengths for the feature extraction process. A bar-based feature-extraction is motivated by the fact that bars are small, but significant blocks of a song. Melodies develop in the course of short sequences of bars. Motifs can also be described as sequences of bars. Start and end points of motifs in songs, e.g., verse or chorus, begin and end with bars.

Table 1. Survey of energy-based, temporal and spectral audio-features that are basis of the SOM-based sequence analysis approach

Descriptor	**Values**	**Type**
energy envelope	16	energy
energy ratio	16	energy
temporal centroid	4	temporal
crest factor	1	temporal
effective duration	1	temporal
zero crossing rate	1	temporal
spectral centroid	4	spectral
spectral kurtosis	1	spectral
MFCC	20	spectral
spectral rolloff	1	spectral
spectral skewness	1	spectral

After the bar segmentation the feature extraction process begins for each segment. Each feature vector is 66-dimensional, taking into account 11 music features (see Table 1). We use energy-based, temporal, and spectral features. Energy-based features represent a simplified amplitude development. Each segment is again divided into $\hat{q}$ segments. For each subsegment the energy features

[1] http://www.mixmeister.com/bpmanalyzer/bpmanalyzer.asp

are computed and aggregated to a feature vector. The spectral features are computed as the mean value over the short-term spectra of the segments, based on a discrete Fourier transformation of block size 512. A detailed description of the selected audio-features can be found in [10].

The trained SOM translates the sequence of high-dimensional data to a corresponding sequence of symbols, i.e., of winner neurons. Let $\mathbf{x}_1, \ldots, \mathbf{x}_N$ be a sequence of high-dimensional feature vectors. Fed to a trained SOM, this "symbolification" is generated by computing a sequence of symbols $\sigma_1, \ldots, \sigma_N$ with $\sigma_i = \arg\min_{k=1 \ldots K} \|\mathbf{w}_k - \mathbf{x}_i\|_2$.

4 Visualization of Sequence Segmentation

SOMs can be used for visualization of high-dimensional data spaces, see [7], or [8] for the application to music. At first, we use our approach to visualize the structure of songs. In Section 5 we are going to use the same approach for motif recognition.

4.1 Approach

To illustrate the feature quantization induced by the SOM, we use the quantification result for visualization of a sequence. For this sake, each sequence is colorized with regard to the corresponding winner neuron $\mathbf{n}_*$. Each neuron is assigned to a color, while neighbored neurons are assigned to similar colors. A function

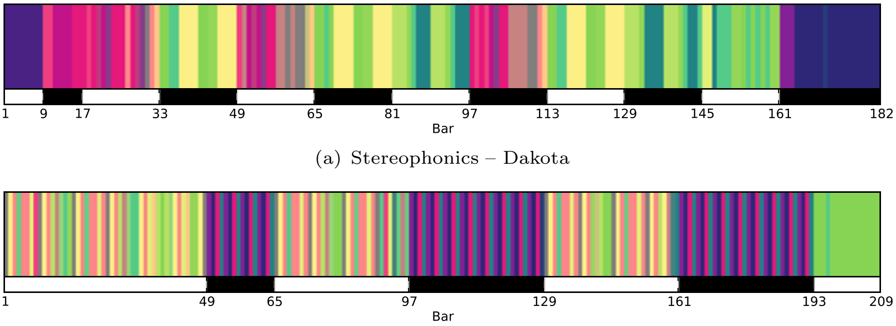
$$f : \mathcal{S} \rightarrow \mathbb{R}^3, \quad \sigma \rightarrow (r_1, r_2, \lambda) \tag{1}$$

maps a symbol $\sigma \in \mathcal{S}$ (with set of symbols $\mathcal{S}$) to a 3-dimensional RGB-value. Values r_1 and r_2 depend on neuron $\mathbf{n}$ with corresponding coordinates on the map, while for the third value λ a constant is chosen. With function f a sequence of symbols corresponding to a sequence of winner neurons of the SOM can be translated into a sequence of RGB-values.

4.2 Experimental Results

In the following experiments of this paper, i.e., in this section and in Section 5.2, we make use of a 5×5-SOM. The limitation to 25 distinct color values offers the best trade-off between the level of detail of a generated visualization and its readability. For each song a new SOM is trained. As initial learning rate we choose $\eta_{\text{init}} = 1.0$. We use the Gaussian function as neighborhood function h with the initial neighborhood radius $r_{\text{init}} = 5.0$. A maximum of $g_{\max} = 1,000$ training cycles is allowed. We have conducted experiments on 14 songs. Due to the limited space we restrict our presentation to the songs *Dakota* by *Stereophonics* and *Escape That* by *4 Hero*.

Figure 1 shows the SOM-based visualization of the two songs. It can be clearly observed that both songs consists of varying, but repeating sequences. For comparison, the black and white lines below each automatic segmentation show the



(a) Stereophonics – Dakota

(b) 4 Hero – Escape That

Fig. 1. Visualization of sequence segmentation for two the songs

manual segmentations regarding the musical similarity of a song's sequences from a human perspective. These segmentations have been chosen by majority decision, i.e., they have been found by at least three of four human evaluations. The results show that the human segmentation is consistent with the automatic, i.e., the markers are located at almost the same areas where the segmentation method visualizes sequences in similar groups of colors.

5 Motif Recognition

The SOM-quantization approach is well appropriate for the recognition of short motifs in songs. Once quantized, the problem becomes a simple string search procedure. In the following, we first introduce the algorithmic approach, and then present two experimental results.

5.1 Approach

We assume that the SOM has been trained with the features of a song, similar to the previous training for visualization, and we seek for a given motif m in a song $\mathcal{Q}$, both translated into sequences of symbols. For the method we propose in the following, we need a distance metric δ determining the distance between motif m and segment s_i with length l. We define the Euclidean distances between two sequences symbol by symbol oriented to the corresponding neurons on the neural map

$$d(m, s_i) = \sum_{j=1}^{l} \|m^j - s_i^j\|_2. \tag{2}$$

Based on $d(\cdot)$ we can define a similarity measure δ for the $k_1 \times k_2$-SOM, i.e.:

$$\delta(m, s_i) = 1 - \frac{d(m, s_i)}{d_{\max}} \tag{3}$$

with the maximal distance between motifs $d_{\max} = l \cdot \sqrt{|k_1|^2 + |k_2|^2}$ that is used for normalizing the similarity. Algorithm 1.1 shows the pseudo-code of our

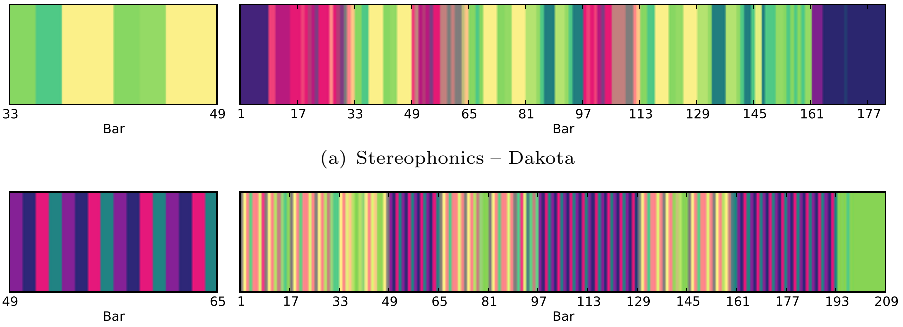
Algorithm 1.1: Motif Recognition

```
    Input: Sequence $\mathcal{Q}$
    Input: Motif $m$
    Output: Output set $\mathcal{M}$
 1  $\mathcal{S} := \{s_1, \ldots, s_{n-l+1}\}$;
 2  $\mathcal{M} := \emptyset$;
 3  forall the $s_i \in \mathcal{S}$ do
 4      Compute similarity $\delta(m, s_i) := 1 - \frac{d(m,s_i)}{d_{\max}}$;
 5      $\mathcal{M} := \mathcal{M} \cup (s_i, \delta(m, s_i))$;
 6  end
 7  Sort $\mathcal{M}$ downwards with regard to $\delta$;
 8  forall the $m \in \mathcal{M}$ do
 9      forall the $m' \in \mathcal{M}$, $m'$ is predecessor of $m$ do
10          if $m$ overlaps with $m'$ then
11              $\mathcal{M} := \mathcal{M} \setminus m$
12          end
13      end
14  end
15  return $\mathcal{M}$;
```

approach. Let $\mathcal{Q}$ be the string of N symbols computed by the SOM and m the motif we are seeking for with length l. First, the approach uses $\mathcal{Q}$ to construct the set $\mathcal{S}$ of $N - l + 1$ segments s_i of length l. For each segment the similarity δ between s_i and m is computed according to Equation 3 in line 4. This procedure results in set $\mathcal{M}$ of sequences (line 5) and corresponding similarities which are sorted downwards (line 7). The set usually consists of overlapping sequences with high similarities, as a slightly shifted sequence often shows a high consensus with itself. For this reason we remove all segments that overlap from $\mathcal{M}$, and keep only the segment with the highest similarity δ (lines 8 - 14).

5.2 Experimental Results

To evaluate the approach we have searched hand-chosen motifs in our song data base, and compared human motif discovery with the algorithmic output. While the term motif describes small repeating melody fragments of a song, we use it in a more general way, describing arbitrary sub-sequences. For convenience, the first instances of a song's chorus and verse have been chosen, since they are easily recognizable by human evaluation. Again, we present the experimental results for the two songs. The motifs we chose for the search are visualized on the left part of Figure 2. The right part of the figure shows a visualization of the corresponding SOM segmentation. From the colored patterns we can already observe that the motif exists at various places with varying similarity. Table 2 shows the result $\mathcal{M}$ of the motif search procedure, i.e., start and end points of the found locations in the song in descending order with regard to similarity δ. Again, we have asked four



(a) Stereophonics – Dakota

(b) 4 Hero – Escape That

Fig. 2. Example for motif search in two songs. The left parts of the figures show the motifs that are searched for in the sequences on the right-hand side.

Table 2. Motif-query results of two search examples in descending order. Values of motifs marked in bold have been found by four test persons.

(a) Stereophonics – Dakota

Sequence	Similarity
33–48	1.0000
113–128	1.0000
65–80	0.9912
133–148	0.9200
85–100	0.9148
149–164	0.8865
49–64	0.8785
5–20	0.8583
165–180	0.8311

(b) 4 Hero – Escape That

Sequence	Similarity
49–64	1.0000
97–112	1.0000
161–176	1.0000
177–192	1.0000
113–128	0.9912
81–96	0.8910
4–19	0.8775
145–160	0.8759
20–35	0.8750
65–80	0.8738
129–144	0.8710

persons to solve the same task, i.e., to find the motifs in the songs. The locations marked in bold show the results of the search of the test persons. We can observe that the segments with the highest similarity correspond to the human search. Our experiments on further 12 songs have shown similar results. Our current feature selection is the result of an extensive search, a systematic description of the feature selection process will be subject to future work.

6 Conclusions

We have introduced an approach for the sequence analysis of high-dimensional music data based on a smooth quantization by SOMs. It was demonstrated that the approach can be used for visualization of sequences, and for motif search.

The experimental results have shown that the success of motif search and of a consistent visualization crucially depends on the chosen musical features. The used features are common in music information retrieval, but do not lead to human-like segmentations in every case. In the future, we plan to find parameters for the automatic segmentation of a song into sequences, with the goal to achieve results that are mostly similar to human segmentations. First experiments have shown that this is no easy undertaking, and we plan to formulate this objective as optimization problem. Furthermore, we plan to extend our approach to the unsupervised discovery of motifs to find over-presented sequences and repetitions, similar to motif discovery in bioinformatics.

References

1. Dickerson, K.B., Ventura, D.: Music recommendation and query-by-content using self-organizing maps. In: IJCNN 2009: Proceedings of the 2009 International Joint Conference on Neural Networks, pp. 2747–2752. IEEE Press, Los Alamitos (2009)
2. Feiten, B., Günzel, S.: Automatic indexing of a sound database using self-organizing neural nets. Computer Music Journal 18(3), 53–65 (1994)
3. Ghias, A., Logan, J., Chamberlin, D., Smith, B.C.: Query by humming: musical information retrieval in an audio database. In: Proceedings of the third ACM International Conference on Multimedia, pp. 231–236. ACM, New York (1995)
4. Harford, S.: Automatic segmentation, learning and retrieval of melodies using a self-organizing neural network. In: Proceedings of International Conference on Music Information Retrieval, MD, Baltimore (2003)
5. Kohonen, T.: The self-organizing map. Proc. IEEE 78(9), 1464–1480 (1990)
6. Kosugi, N., Nishihara, Y., Sakata, T., Yamamuro, M., Kushima, K.: A practical query-by-humming system for a large music database. In: Proceedings of 8. ACM International Conference on Multimedia, pp. 333–342. ACM, New York (2000)
7. Liao, G., Shi, T., Liu, S., Xuan, J.: A novel technique for data visualization based on som. In: ICANN (1), pp. 421–426 (2005)
8. Neumayer, R., Rauber, A.: Multi-modal music information retrieval - visualisation and evaluation of clusterings by both audio and lyrics. In: Proceedings of the 8th Conference Recherche d'Information Assistée par Ordinateur (RIAO 2007). ACM, New York (2007)
9. Pampalk, E., Rauber, A., Merkl, D.: Content-based organization and visualization of music archives. In: Proceedings of the Tenth ACM International Conference on Multimedia, pp. 570–579. ACM, New York (2002)
10. Peeters, G.: A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Tech. rep., IRCAM (2004)
11. Rauber, A., Pampalk, E., Merkl, D.: Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by musical styles. In: ISMIR (2002)

Searching for Locomotion Patterns that Suffer from Imprecise Details

Björn Gottfried

Centre for Computing and Communication Technologies
University of Bremen, Germany

Abstract. Today, a number of positioning technologies exist in order to track moving objects. While GPS devices enable wayfinding in outdoor environments, several techniques have been devised for indoor tracking, to enable smart spaces, for example. But even at the microscopic scale objects are tracked by researchers of the natural sciences with imaging technologies. Regardless of the spatial scale and application at hand, a common problem consists in the ever growing quantities of movement data which are to be managed. One strategy asks for how to simplify the data, such that compact representations save space but do still capture relevant information. Such an abstraction is described in this paper. It is shown how it can be applied to constraint programming techniques in order to search for movement patterns of groups of objects. Instead of exhaustively searching by means of *generate and test*, the representation allows the application of *constraint propagation*. As a consequence, search space can be reduced significantly. Moreover, it is shown how the chosen representation aids the dealing with a specific class of imprecise data. The domain of biological cells is used for illustrating the presented methods. The resulting observations, made by light microscopes, suffer from the addressed class of imprecise data.

1 Introduction

Movement patterns play an essential role in many real live situations as well as in scientific experiments. This concerns in particular the movement of collectives [11]. The ever growing sophistication of tracking methods enables the user to capture large quantities of movement data within a short time [8]. As a consequence, methods are required that manage these data, such as those described by [7]. The search for specific movement events of groups of objects is one of the most fundamental tasks which will be addressed in the current work.

In this paper we will proceed as follows. In Section 2 an application in the context of movement analysis is introduced. A couple of problems are identified within this scenario. Those problems define the conditions which the following representation has to manage. Afterwards, a calculus for representing movement patterns is presented in Section 3. This approach allows the definition of movement patterns at a symbolic abstraction level. Section 4 shows how the representation is employed for solving consistent labeling problems. While a number

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 168–175, 2010.

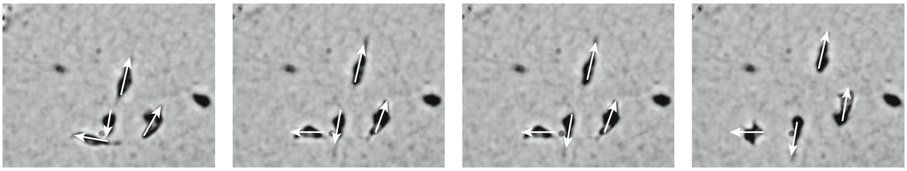


Fig. 1. Four successive video frames: four cells disperse into different directions

of examples demonstrate the application of the discussed representation, Section 5 shows how the approach deals with imperfect data, and hence, how it manages realistic scenarios as described in the application context in Section 2. The conclusion in Section 6 closes this paper.

2 The Search for Locomotion Patterns of Cancer Cells

2.1 Application Scenario

The presented methodology is motivated and illustrated by the following scenario. Cell biologists are interested in investigating the migration of cells, in particular cancer cells, such as MV3 melanoma which move with a speed of about $0.1 - 0.5\mu m/min$ [3]. The idea is to observe cell movements in order to verify pharmacological substances. That is, migration behaviours indicate their influence and enable to learn more about why cancer cells become metastatic.

Those investigations entail more specific research questions which require, among others, the search for specific locomotion patterns among groups of cells. Such patterns include cells that move towards each other, move into different directions, follow each other, and so on; in other words, such patterns are restricted to typical categories which are comprehensible by the scientist. Given the trajectories of moving cells which have been monitored by a microscopic camera over a period of about 2 up to 30 hours [9], it is the aim of the current work to provide a methodology that enables the search for specific locomotion configurations of cells. In order to let those configurations be comprehensible for the scientist, an abstraction of movement events is needed that will be presented in the next section. More details about the analysis and tracking of cells are not within the focus of the presented work and are found elsewhere [9].

2.2 Imprecise Movement Data

The precision with which cell movements are determined is rather low. This is due to the tissue in which the cells migrate, because the tissue shows a structure which introduces edge points. As a consequence, separating cell bodies from the background is difficult. Moreover, cell bodies change from being circular to being elongated while they migrate; this is another reason for difficulties in segmenting cells from the background. Additionally, cells which collide are hardly separable by means of computer vision; boundaries between different cells can frequently not be determined and cell nuclei are not visible within these images which are deliberately taken from a low cost light microscope.

Looking at the state-of-art, it shows that a great many approaches employ fluorescent techniques, like [12]. Such techniques, however, entail serious problems: they require intensive ultraviolet radiation, being destructive for living cells, in particular, when observing cells for longer time periods; since the physiology is affected by fluorescent techniques, so is the movement behaviour we just want to analyse; eventually, cells are to be observed over longer periods of time (24h) – sooner or later fluorescence decays or it is deposited in cell-compartments. As a conclusion, fluorescent techniques are unsuitable.

Besides those difficulties arising from imprecise sensor data, there is yet another reason for preferring a coarse representation of movements. The biologist is interested in investigating how collectives of cells behave. Behaviours the user intends to distinguish include such patterns as *cells moving into similar directions* or *moving towards different directions*. It makes no difference in the current context to introduce finer distinctions. What is relevant here is both the coarse relative position among cells as well as their relative orientation. Therefore, a representation would not help which distinguishes different modes of how objects overlap [10,2]. Instead, different modes of disconnection are to be used in order to capture relative positions among cells.

2.3 Cell Representations

From the previous arguments it follows that movement events carrying relevant information for the human user are found at an abstraction layer that makes such comprehensible distinctions as that a cell is *left of* another cell, *behind*, or *in front of* it. Cells will be represented by means of arrows that indicate their movement direction. They can be extracted out of the data, since an algorithm is employed which approximates each cell body by an ellipse. It gives the position of the cell in each video frame. The direction of the cell's locomotion orientation is determined by the centroids of the cell body of two successive video frames. As a consequence, cells are represented by arrows and collections of cells will be characterised by arrow arrangements. Their change will be described with respect to the calculus introduced below.

3 A Locomotion Calculus

In the following, an abstract representation for cells is presented that employs a relation algebra. The pure representation would be sufficient in order to search for specific configurations of collectives. But in order to avoid the exhaustive search through the entire configuration space, the converse and composition operations of the algebra can be used to enforce arc- and path-consistency in the according constraint net of a specific configuration. To ensure global consistency, search is still needed, but search space is reduced by constraint propagation.

3.1 Movement Calculus $\mathcal{BA}_{23}$

In [4] a relation algebra based on [13] has been introduced with 23 relations among oriented intervals in two dimensions. These relations are shown in Fig. 2. They form a qualitative representation in that they abstract from precise quantitative locations. Having two intervals which are determined by their endpoints we are faced with a four-dimensional configuration space that consists of precise values, namely of the endpoints of the main axes of the cell bodies. The abstraction from precise quantities means to define equivalence classes of locations that are considered to be identical in the chosen representation. For instance, when an object m_1 is *left of* another object m_2, level with it, it is said to be *during left*, $\mathsf{m}_1\mathsf{D}_\mathsf{l}\mathsf{m}_2$ for short (see the relation on the left hand side of the neighbourhood graph in Fig. 2). In this way, each conceivable relation among two linear disconnected objects in two dimensions can be represented. The considerations of sequences of such relations describe relative locomotion patterns among pairs of objects. Sets of such relations describe how whole collections of objects behave.

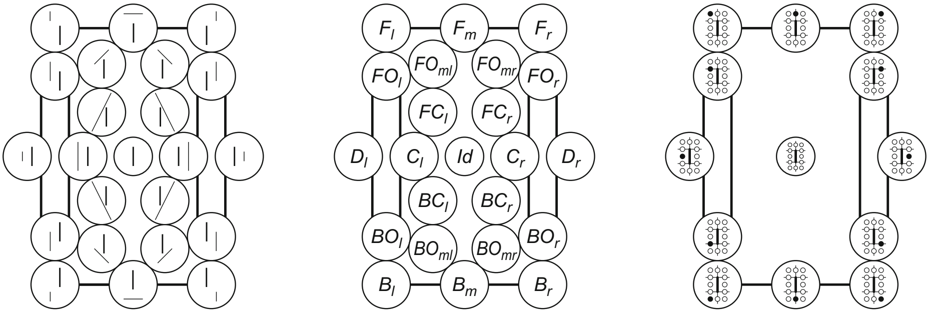


Fig. 2. The 23 $\mathcal{BA}_{23}$ relations that can be distinguished between two line segments in the two-dimensional plane. Left: Example arrangements. Centre: Mnemonic labels. Right: Twelve distinguishable locations around the reference object exist.

4 Searching for Patterns

4.1 Generate and Test

Looking for a configuration of cells which move in a specific constellation amounts to compare this model constellation with single video frames. In a frame, however, there might be a lot of different cell objects. While a specific constellation considers between $k = 2$ and $k = 6$ cells, the number of cells in a frame can be between about $n = 10$ and $n = 60$. As a consequence, there are up to $\frac{n!}{(n-k)!} = \frac{60!}{(60-6)!} = 60 \cdot 59 \cdot 58 \cdot 57 \cdot 56 \cdot 55 = 36.045.979.200$ possible variations how the model objects can be assigned to specific cells in a frame. In order to reduce this huge amount of possibilities, only those constellations are taken into account where the cells move near each other, meaning a radius which restricts the number of cells to approximately a ninth part of the sample. In this way, the investigations of cell movements are restricted to those cells which are close by each other. A number of $\frac{(\frac{60}{9})!}{(\frac{60}{9}-6)!} \simeq 2873$ is still large provided that there are about 200 to 800 frames in a video, making a total of up to $2873 \cdot 800 = 2.298.400$ constellations to be compared. For each of those variations, a *generate and test* algorithm would validate the model constraints in order to decide whether a given variation satisfies the model.

4.2 Enforcing Local Consistency

Provided that a pattern model is not contained in a scene, *generate and test* would have to search through the entire configuration space before terminating. The following example will demonstrate how search space is pruned by enforcing local consistency.

A given example model consists of three objects, m_1, m_2, and m_3. This model describes the relative movements among the objects by a number of three constraints: $m_1\mathsf{F}_\mathsf{l}^\mathsf{F}m_2$, $m_2\mathsf{F}_\mathsf{l}^\mathsf{F}m_3$, and $m_3\mathsf{B}_\mathsf{r}^\mathsf{F}m_1$ (cf. Fig. 2, the superscripts indicating orientations). The configuration of cell movements in Fig. 1 does not satisfy this model. Since there are four cells, each of which could be assigned to the model objects, 24 possible assignments are to be verified by *generate and test*.

Local consistency is enforced as follows. Each of the four cells, c_i could be assigned to each of the three model objects. In other words, each model domain consists of four cells. Then, trying to assign the first cell to one of the cells in the domain of the second model object, only the assignment $m_2 \rightarrow c_2$ satisfies the according constraint between m_1 and m_2. Trying to assign the other cells, c_2, c_3, and c_4 to m_1 is impossible, since those assignments allow no assignments of cells to m_2 without violating the first constraint. As a consequence, the second constraint, $m_2\mathsf{F}_\mathsf{l}^\mathsf{F}m_3$, is only to be verified for $m_2 \rightarrow c_2$. The successive assignments of $c_{i,i=1..4}$ to m_3 shows that no solution exists in order to satisfy the second constraint. Since the first two domains of the model objects are already reduced to one value, no alternative for backtracking exists. That is, there is no object triple in this frame which can be mapped to the model. There are less many steps to be performed until it can be recognised that no solution exists, as when employing *generate and test.* Additionally, only partial assignments need to be tested for enforcing local consistency.

4.3 A Matching Pattern

The previous example has shown how search space is pruned by the enforcement of local consistency. The next example illustrates how the method finds a solution. For this purpose, the chosen model will be mapped to three of the cells in Fig. 1. The model constraints are: $m_1\mathsf{B}_\mathsf{l}^\mathsf{B}m_2$, $m_2\mathsf{B}_\mathsf{l}^\mathsf{B}m_3$, and $m_3\mathsf{F}_\mathsf{l}^\mathsf{F}m_1$. Starting with one of the model objects, for example taking m_3, we reduce its domain to c_1 which is the only value for which an assignment with m_1 exists, namely $m_1 \rightarrow c_2$, so that $m_3\mathsf{F}_\mathsf{l}^\mathsf{F}m_1$ is satisfied. As a consequence, $m_2 \rightarrow c_3$ is the only assignment which can be found in order to satisfy $m_1\mathsf{B}_\mathsf{l}^\mathsf{B}m_2$. Also, this latter assignment is consistent with the last constraint between m_2 and m_3. The constraint net is arc-consistent. In order to ensure global consistency, n-consistency is to be enforced, with n being the number of model objects. 3-consistency can be verified by means of the composition. In this example it shows, that the network is in fact path-consistent: $m_3 \mathsf{F}^{\mathsf{F}}_{\mathsf{l}} m_1 \cdot m_1 \mathsf{B}^{\mathsf{B}}_{\mathsf{l}} m_2 = m_3 \{\mathsf{B}^{\mathsf{B}}_{\mathsf{l}}, \mathsf{B}^{\mathsf{B}}, \mathsf{B}^{\mathsf{B}}_{\mathsf{r}}\} m_2$.[1]

4.4 Constraint Relaxation

Whenever no patterns are found one possibility consists in constraint relaxation. This means to consider similar patterns during the search process, in other words, to allow further relations. The notion of similarity is dealt with specifically by the $\mathcal{BA}_{23}$ representation. In Fig. 2 it is shown how the relations are arranged in their neighbourhood graph. Here, the most similar relations are connected and the neighbourhood of a relation in this graph represents this similarity. For a specific pattern this amounts to include also the search for relations according to this neighbourhood structure.

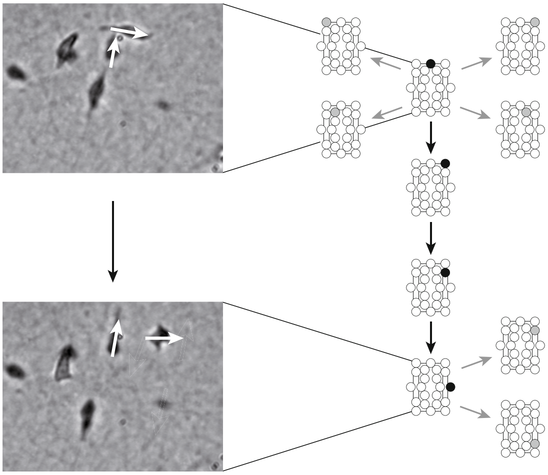


Fig. 3. A specific movement model shown by the relations printed in black. The grey relations show the neighbourhoods for both the first state and the last state. The two intermediate states are not captured by the video. They can be derived by the relations which are necessary in order to get from the first state to the last state.

In particular when looking at successive video frames it shows that relations might slightly change. Fig. 3 shows two successive frames and how the relative positions for a number of two cells change. For reasons of efficiency, a typical video indexing method restricts the analysis of frames. For instance, only each k-th frame is analysed under the assumption that no significant changes occur in the meantime. This makes it even more important to be able to relax constraints. Taking the two frames in Fig. 3, relations change significantly; those

[1] The definition of the composition operation can be found in [4].

changes can be comprehended by the neighbourhood structure. In particular, non-adjacent relations indicate that the used snapshots are in fact incomplete. However, missing relations can be deduced from the neighbourhood graph. This is similar to having different more or less fine grained models which are set into relation by means of what [1] calls neighbourhood expansion.

5 The Dealing with Imprecise Movement Data

Having analysed the problem situation in Section 2.2 it became clear that a number of different reasons exist for the imprecision of movement data. Moreover, what those different circumstances have in common is that the imprecision concerns the level of detail. That is, the first reason concerns the background structure of the tissue: detected edge points might either indicate details of the background structure or of the cell structure. The second reason is the change in morphology: detailed differences of cell bodies are to be dealt with in successive video frames which are due to movement deformations. The third reason concerns cell collisions: details of such cell collisions are not available within the employed experimental setup. In a nutshell, imprecision in the data influences the detection of details of both cell bodies and cell movements.

These observations entail important implications for the chosen representation. Details of the data are imprecise but a more coarser view is still available. But then a representation that abstracts from precise details would have the advantage not to represent imprecise details. This is what the chosen representation with $\mathcal{BA}_{23}$ relations in fact does: it abstracts from precise details and focuses on a coarser view which is pretty well available. Simultaneously, it reflects the level of detail motivated in Section 2.3: at this spatial level distinctions are made that are relevant to the human user for whom the pattern matching algorithms in Section 4 should be comprehensible.

6 Conclusions

Imperfect data are omnipresent in real world applications. This concerns in particular tracking technologies that determine object movements. The imperfection discussed in the current scenario concerns in particular imprecise details. The solution consists in the abstraction from those details and concentrating on coarse movement descriptions. This brings in the advantage that the resulting representation makes distinctions which are easily comprehensible for the user who has to define according movement patterns and who has to understand the matching results made by the computer. That is, the human conceptions on vague patterns are brought into correspondence with the chosen abstract representation.

The chosen representation defines a relation algebra that employs the $\mathcal{BA}_{23}$ relations which combine to 125 relations when considering also orientation variations. Orientation variations are necessary in order to constrain patterns strong enough. In particular, this entails a large composition table which is difficult to verify [4]. A more concise representation would have many advantages from the point of view of implementation. Such a representation is provided by [5]. Future work will look at whether the presented problem domain can be managed with that calculus equally well, though it consists of only 16 movement relations.

A yet more abstract representation is provided by [6] that builds upon topological distinctions and relates object movements to their environmental context. Movements of collectives as well as the relation to their environment could be investigated by looking at how the presented approach combines with [6].

Acknowledgements

The material used in this work has been provided by Peter Friedl from the University Hospital at Würzburg in the context of a project which restricted the analysis of cell movements to their velocities.

References

1. Dylla, F.: Qualitative Spatial Reasoning for Navigating Agents. In: Gottfried, B., et al. (eds.) BMI – Smart Environments, pp. 98–128. IOS Press, Amsterdam (2009)
2. Egenhofer, M., Franzosa, R.: Point-Set Topological Spatial Relations. International Journal of Geographical Information Systems 5(2), 161–174 (1991)
3. Friedl, P., Zänker, K.S., Bröcker, E.-B.: Cell migration strategies in 3-D extracellular matrix: Differences in morphology, cell matrix interactions, and integrin function. Microscopy Research and Technique 43(5), 369–378 (1998)
4. Gottfried, B.: Reasoning about Intervals in Two Dimensions. In: Thissen, W., et al. (eds.) IEEE Int. Conf. on Systems, Man & Cybernetics: The Hague, The Netherlands, October 10-13, pp. 5324–5332. IEEE Press, Los Alamitos (2004)
5. Gottfried, B.: Representing short-term observations of moving objects by a simple visual language. Journal of Visual Languages and Computing 19, 321–342 (2008)
6. Kurata, Y., Egenhofer, M.: Interpret. of Behav. from a Viewpoint of Topology. In: Gottfried, B., et al. (eds.) BMI – Smart Envir., pp. 75–97. IOS Press, Amsterdam (2009)
7. Laube, P.: Progress in Movement Pattern Analysis. In: Gottfried, B., Aghajan, H. (eds.) BMI – Smart Environments, pp. 43–71. IOS Press, Amsterdam (2009)
8. Millonig, A., Brändle, N., Ray, M., Bauer, D., Van Der Spek, S.: Pedestrian Behaviour Monitoring: Methods and Experiences. In: Gottfried, B., et al. (eds.) BMI – Smart Environments, pp. 11–42. IOS Press, Amsterdam (2009)
9. Moeller, J., Gottfried, B., Schlieder, C., Herzog, O., Friedl, P.: Automated Tracking of Cell Movements and Resolution of Cell-Cell Collisions in Three-dimensional Collagen Matrices. In: Keystone-Symposium on Cell-Analysis, Colorado, USA (2003)
10. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: 3rd Int. Conf. on Knowledge Representation and Reasoning, San Mateo, pp. 165–176. Morgan Kaufman, San Francisco (1992)
11. Wood, Z., Galton, A.: Collectives and their Movement Patterns. In: Gottfried, B., Aghajan, H. (eds.) Behaviour Monitoring and Interpretation – Smart Environments, pp. 129–155. IOS Press, Amsterdam (2009)
12. Zhang, B., Zimmer, C., Olivo-Marin, J.-C.: Tracking fluorescent cells with coupled geometric active contours. In: IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 476–479. IEEE Press, Los Alamitos (2004)
13. Zimmermann, K., Freksa, C.: Qualitative Spatial Reasoning Using Orientation, Distance and Path Knowledge. Applied Intelligence 6, 49–58 (1996)

World Modeling for Autonomous Systems

Ioana Gheţa[1], Michael Heizmann[2], Andrey Belkin[1], and Jürgen Beyerer[1,2]

[1]Vision and Fusion Laboratory, Karlsruhe Institute of Technology
Adenauerring 4, 76131 Karlsruhe, Germany
[1]Fraunhofer Institute of Optronics, System Technologies and Image Exploitation
Fraunhoferstraße 1, 76131 Karlsruhe, Germany
{ioana.gheta,andrey.belkin}@kit.edu,
{michael.heizmann,juergen.beyerer}@iosb.fraunhofer.de

Abstract. This contribution proposes a universal, intelligent information storage and management system for autonomous systems, e. g., robots. The proposed system uses a three pillar information architecture consisting of three distinct components: prior knowledge, environment model, and real world. In the center of the architecture, the environment model is situated, which constitutes the fusion target for prior knowledge and sensory information from the real world. The environment model is object oriented and comprehensively models the relevant world of the autonomous system, acting as an information hub for sensors (information sources) and cognitive processes (information sinks). It features mechanisms for information exchange with the other two components. A main characteristic of the system is that it models uncertainties by probabilities, which are handled by a Bayesian framework including instantiation, deletion and update procedures. The information can be accessed on different abstraction levels, as required. For ensuring validity, consistence, relevance and actuality, information check and handling mechanisms are provided.

1 Introduction

Efficient operational autonomous systems require a comprehensive overview on their environment. The present contribution proposes a universal, intelligent system for information storage and management, which is applicable to a wide variety of types of autonomous systems. The primary application of the system is a humanoid robot, designed to help with domestic applications.

The proposed system uses a *three pillar information architecture* and aims at modeling the environment of an autonomous system. The three pillars represent the main components of the architecture: prior knowledge, environment model, and real world. The first two components can be compared to the long and short term memory of the human brain.

Being the central component of the autonomous system, the environment model acts as an information hub, which stores sensory information and prior knowledge, and delivers it to cognitive processes. The information is represented in the environment model as instances of classes with class specific attributes and

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 176–183, 2010.

relations. Instances in the environment model correspond to entities in the real world, their classes map object types of the real world. Classes equate concepts in prior knowledge, their realization are instances.

Besides the pillar architecture, the proposed system is characterized by the following features: object oriented representation of information, use of probability distributions in a Degree-of-Belief (DoB) interpretation, information management and fusion based on a Bayesian framework, and information access on different abstraction levels.

Commonly used approaches for the modeling of information comprise semantic nets, predicate logic or formal languages, see e. g. [1]. Recently presented methods involve ontologies, object oriented and probabilistic approaches [2,3]. Current research combines object oriented with probabilistic approaches [4,5,6,7]. The combination of object oriented approaches and ontologies is also being discussed in literature [8]. Probabilistic ontologies are proposed in [9].

The proposed approaches in literature for modeling the environment of autonomous systems are mostly domain specific and not transferable to other applications [4]. [10] proposes an object oriented world modeling approach with the purpose of creating virtual environments for simulation or engineering and for automation of specific tasks, e. g., financial transactions. Some of the main characteristics of the approach are the separation between real world and system objects and the development of the model using class diagrams. [6] proposes a dynamic approach for cooperative intelligent vehicles, which models the relevant environment for neighboring vehicles. Main characteristics are the incorporation of uncertainties for attributes, the modeling of conceptual objects, the inheritance in the object hierarchy, and the development of check and simple inference mechanisms. In robotics, the approaches proposed are generally simple and task specific. They require large amounts of prior information, e. g., the main task of the robot or his context.

This contribution focusses on describing the structure of the proposed system with accent on information management. The main characteristics and their advantages are presented in Sec. 2. Section 3 discusses the information exchange mechanisms and the construction of the environment model. Inference realizations are not part of the proposed architecture and are thus not within the scope of this contribution.

2 Three Pillar Information Architecture

The three pillar information architecture is mainly characterized through the separation and interaction between prior knowledge, environment model, and real world, see Fig. 1.

In the center of the architecture, the object oriented environment model is situated (middle pillar in Fig. 1). It represents the information the autonomous system has about the entities (objects and persons) in its current relevant environment. In the human memory, its correspondent is the short term memory. The information is represented here in form of instances with attributes and

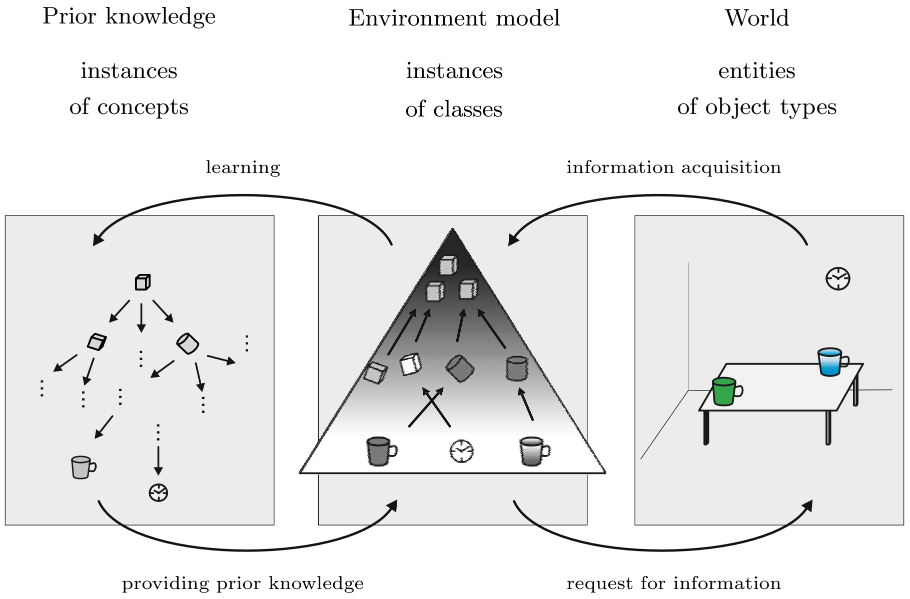


Fig. 1. Three pillar information architecture

relations. The main function of the environment model is to act as information hub, i. e., all relevant information is stored and made available to cognitive processes, e. g., path planning or inference. For modeling the environment model, dynamic progressive mapping of instances and their attributes is employed. The relations are represented by means of multiple semantic networks [11,12].

A second pillar (left in in Fig. 1) of the architecture is established by prior knowledge. It is equivalent to the long term memory of the human brain and contains prior information of two types:

- concepts expressed by ontologies (structural information about classes with attributes, relations, and rules);
- instance related knowledge (e. g., on instances of persons or objects, maps of the environment).

This information is available to the system directly after the instantiation of the environment model. Moreover, the information can be organized in context specific modules, which can be loaded or unloaded on-the-fly, as required. Thus, the environment model is kept lean.

The third information pillar (right in Fig. 1) represents sensory information which is acquired by the autonomous system when operating. The acquired information regards entities, their attributes, and relations. The entities correspond to instances in the other two components and are pooled in object types, which correspond to classes (in the environment model) and concepts (in the prior knowledge).

3 Information Management

In the following, attributes and relations of instances and entities are the considered pieces of information. Each information is characterized by its uncertainty in form of a Degree-of-Belief (DoB) distribution. Details on how uncertainties can be expressed and on the advantages of DoBs are given in [5,7,12,13,14].

Representation of Information in the Environment Model. Attributes can be descriptive or non-descriptive. A typical example of a non-descriptive attribute is the *existence* expressed in the probability that the entity modeled by the instance exists in the real world. Examples for descriptive attributes are *type*, *position*, or *color*. The attribute *type* is in this case also represented by a DoB distribution; it is therefore not deterministic. The advantage of this modeling is that an instance can be created without knowledge on which class it belongs to. The challenge, however, is the mapping of classes to concepts in prior knowledge.

Representation of Information in Prior Knowledge. A part of prior knowledge is the instance related knowledge consisting of instances with attributes and relations that are „important“ to the autonomous system, e. g., persons that often appear in the real world or maps of certain environments. This information is detailed in general, i. e., many attributes are specified, and may be acquired through external sensors or learned by the autonomous system itself.

Another part is the ontology knowledge containing concepts with attributes, relations, and rules modeling contexts. The attributes and relations are represented by DoB distributions over the domains, e. g., according to the observed frequency of possible values in the real world. Thus the DoB distributions of attributes and relations of concepts describe all entities that may appear in the real world, whereas the attributes and relations of instances in the environment model characterize one particular observed entity.

For example, the DoB distribution for the attribute *color* of the concept of a *cup* may be defined as a uniform distribution over all colors, following the assumption that cups may be of any color. In the environment model, the DoB distribution would have a maximum at a certain color according to the entity observed in the world.

Building-up of the Environment Model. The base element is an instance of the concept of the *blanc object*, which only needs to have one attribute: the *existence*. Other attributes are allowed, but not mandatory on instantiation. Starting with blanc objects, the environment model is completed combining top-down and bottom-up strategies, see Fig. 1.

The **top-down** strategy means that the environment model is complemented by mapping new acquired information to attributes of existing instances or by creating new instances [5,7]. In Fig. 1, this strategy is outlined by the arrow “information acquisition”.

The **bottom-up** strategy implies using prior knowledge for the environment model; this is indicated in Fig. 1 by the arrow "providing prior knowledge". Here, the *type* of an instance is determined and/or its attributes are supplemented. Additionally, according to the context of the autonomous system, instance related knowledge may be used to create new instances. For example, in a kitchen scenario, kitchen appliances may be instantiated.

For handling information in the environment model, certain mechanisms are required:

- **Creation of new instances** according to sensory information is performed by means of Bayesian fusion, considering the probability for the existence of the entity. If the posterior probability is higher than an initialization threshold, a new instance is inserted in the environment model. In the case of attributes and relations, a similar calculus is employed [5,7].
- **Information update** with new sensory information or with information from prior knowledge is also accomplished by means of Bayesian fusion using the common prediction-update scheme:

  The prediction step is equivalent to propagating the information in the environment model from one time step to the other, modeling that the knowledge about the real world becomes more uncertain. In consequence, the entropy of the DoB distributions increases [15].

  The update step consists in the fusion of the information in the environment model (interpreted as prior) with the new information (interpreted as likelihood function). If more pieces of information regarding the same attribute or relation are available, a recursive Bayesian fusion is used.

  If no new information is provided from one time step to the other, a function proportional to the Maximum Entropy distribution is used as dummy likelihood function. This is equivalent to assuming the result of the prediction step [5,7].
- **Deletion of instances** If the DoB of the *existence* attribute of an instance drops under the deletion threshold, the instance is deleted from the environment model. In the case of attributes and relations, the deletion is equivalent to setting the DoB distributions to Maximum Entropy distributions over the respective domain.

Information Exchange between the Three Pillars. Information flows to and from the environment model. First, instance related and ontology knowledge are transferred from prior knowledge into the environment model, as part of the bottom-up strategy. This flow is indicated in Fig. 1 by the arrow "providing prior knowledge". The main problem here is the classification, i. e., how to determine the value of the attribute *type* based on known attributes and their DoB distributions.[1] In other words, the challenge is assigning instances in the environment model to concepts of prior knowledge. Moreover, the information in ontologies (especially the rules) can be used for consistency, validity, and relevance checks.

[1] Due to the probabilistic approach, the classification is equivalent to lowering the uncertainty regarding the attribute *type*.

In the opposite direction, the flux of information represents learning of new concepts, attributes, relations, rules, and instance related knowledge. The main difficulty here is inserting new concepts in the ontology at the right place.

Second, sensory information is inserted in the environment model. This information flow is outlined in Fig. 1 through the arrow "information acquisition". The main problem here is the data association between the observed entity and an instance in the environment model.

The information flux in the opposite direction consists in information requests for filling in the gaps, i. e., exploring unknown attributes and relations. The information request is a result of inference processes. For example in the kitchen context, the autonomous system may ask questions regarding the positions of certain appliances known to be existent.

Abstraction Levels. Depending on the task, information with different degree of detail is needed. Accordingly, the information in the environment model can be accessed on different abstraction levels. For example considering a path planning task, only information regarding existence, position, and dimensions of entities is necessary, i. e., information with a low level of detail and a high level of abstraction. On the contrary, for a grasping task, detailed information regarding form, grasp possibilities, and footprint is required, see Fig. 2.

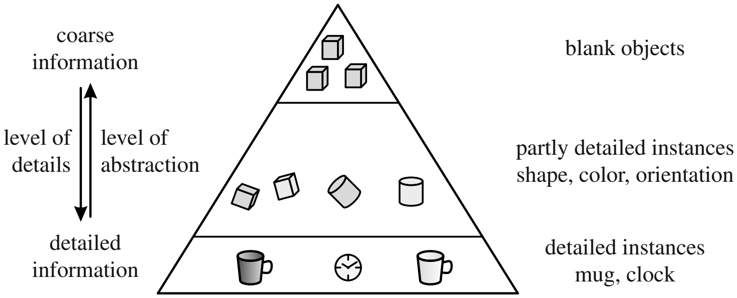


Fig. 2. Abstraction pyramid

Check Mechanisms Check mechanisms are necessary for ensuring constant quality of the environment model:

- The most basic check mechanisms regard the **validity** of the information. They assure that the information incorporated in the environment model fulfills formal correctness restrictions, e. g., DoB distributions are valid.
- **Consistency** checks ensure that basic physical rules are satisfied, e. g., cups cannot fly.
- **Relevance** checks assure that the information is relevant in the present context, e. g., cups are irrelevant when washing the car.
- **Actuality** checks ensure that the information is always up to date, e. g., by triggering exploration requests for transient entities.

4 Realization

The three pillar information architecture has been developed within the DFG project SFB 588 "Humanoid Robots—Learning and Cooperating Multimodal Robots" [16]. Its purpose is to design humanoid robots assisting in household applications. To solve this task, the humanoid robot needs a comprehensive state of knowledge on its environment. To this end, the described information architecture has been employed. Figure 3 shows the humanoid robot in a kitchen scenario. Development and implementation details are given in [11,12].

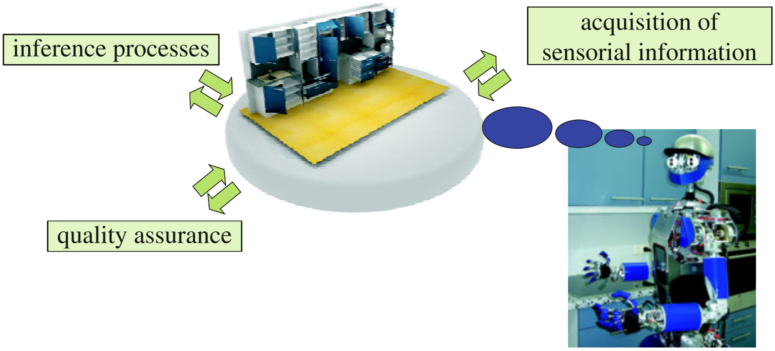


Fig. 3. Humanoid robot in its test environment

5 Conclusions

The present contribution proposes a universal, intelligent information storage and management architecture for autonomous systems. The separation into three components (prior knowledge, environment model, real world) permits an efficient information handling. The central component (the environment model) contains a comprehensive overview on the part of the world which is relevant for the autonomous system. Other components are sensory information and prior knowledge. Thus, the environment model acts as an information hub for all other components of the autonomous system. The environment model can be compared with a Lego landscape, where the Lego bricks compose virtual substitutes (instances) of real objects and persons (entities). The main advantage of the proposed architecture refers to the probabilistic approach, which enables a consistent information management based on the Bayesian framework. This includes standard mechanisms for instantiation, deletion, and update. In addition, information exchange mechanisms between the components of the architecture, along with quality check mechanisms, are provided. An additional powerful feature is the abstraction pyramid, giving the possibility of retrieving information in the desired degree of detail. Together with other cognitive processes, the three pillar information architecture provides autonomous systems with situation awareness.

References

1. Görz, G. (ed.): Handbuch der Künstlichen Intelligenz, 4th edn. Oldenbourg, München (2003)
2. Shlaer, S., Mellor, S.J.: Objekte und ihre Lebensläufe: Modellierung mit Zuständen. Hanser, München (1998)
3. Meystel, A.M., Albus, J.S.: Intelligent systems: architecture, design, control. Wiley series on intelligent systems. Wiley-Interscience Publication, New York (2002)
4. Bauer, A.: Probabilistic reasoning on object occurrence in complex scenes. In: Image and Signal Processing for Remote Sensing XV, Proc. of SPIE, vol. 7477 (2009)
5. Gheţa, I., Heizmann, M., Beyerer, J.: Object oriented environment model for autonomous systems. In: Boström, H., Johansson, R., van Laere, J. (eds.) Proceedings of the Second Skövde Workshop on Information Fusion Topics, Skövde Studies in Informatics, pp. 9–12 (November 2008)
6. Papp, Z., Brown, C., Bartels, C.: World modeling for cooperative intelligent vehicles. In: IEEE Intelligent Vehicles Symposium, pp. 1050–1055 (2008)
7. Heizmann, M., Gheţa, I., Puente León, F., Beyerer, J.: Informationsfusion zur Umgebungsexploration. In: Puente León, F., Sommer, K.D., Heizmann, M. (eds.) Verteilte Messsysteme, pp. 133–152. KIT Scientific Publishing (March 2010)
8. Siricharoen, W.V.: Ontologies and object models in object oriented software engineering. IAENG International Journal of Computer Science 33(1) (2007)
9. da Costa, P.C.G., Laskey, K.B., Laskey, K.J.: PR-OWL: A bayesian ontology language for the semantic web. In: ISWC-URSW, pp. 23–33 (2005)
10. Isoda, S.: Object-oriented world-modeling revisited. Journal of Systems and Software 59(2), 153–162 (2001)
11. Belkin, A.: Object-oriented world modeling for autonomous systems. Technical report, Karlsruhe Institute of Technology KIT (2010)
12. Kühn, B., Belkin, A., Swerdlow, A., Machmer, T., Beyerer, J., Kroschel, K.: Knowledge-driven opto-acoustic scene analysis based on an object-oriented world modelling approach for humanoid robots. In: Proceedings of the 41st International Symposium on Robotics and the 6th German Conference on Robotics. VDE-Verlag (2010)
13. Beyerer, J.: Verfahren zur quantitativen statistischen Bewertung von Zusatzwissen in der Meßtechnik. VDI Verlag, Düsseldorf (1999)
14. Beyerer, J., Heizmann, M., Sander, J., Gheţa, I.: Bayesian Methods for Image Fusion. In: Image Fusion – Algorithms and Applications, pp. 157–192. Academic Press, London (2008)
15. Bernardo, J.M.: Encyclopedia of Life Support Systems (EOLSS). In: Probability and Statistics. UNESCO, Oxford (2003)
16. SFB588: Humanoide Roboter, http://www.sfb588.uni-karlsruhe.de/ (retrieved April 7, 2010)

A Probabilistic MajorClust Variant for the Clustering of Near-Homogeneous Graphs

Oliver Niggemann[1,2], Volker Lohweg[2], and Tim Tack[2]

[1] Fraunhofer IOSB-INA, Competence Center Industrial Automation,
Lemgo, Germany
oliver.niggemann@iosb-ina.fraunhofer.de
[2] inIT – Institute Industrial IT, Lemgo

Abstract. Clustering remains a major topic in machine learning; it is used e.g. for document categorization, for data mining, and for image analysis. In all these application areas, clustering algorithms try to identify groups of related data in large data sets.

In this paper, the established clustering algorithm MajorClust ([12]) is improved; making it applicable to data sets with few structure on the local scale—so called near-homogeneous graphs. This new algorithm MCProb is verified empirically using the problem of image clustering. Furthermore, MCProb is analyzed theoretically. For the applications examined so-far, MCProb outperforms other established clustering techniques.

1 Introduction

MajorClust is a popular clustering algorithm for graphs and high dimensional data—making it especially applicable to the fields of information retrieval and information visualization. MajorClust has several nice features: (*i*) It is simple and therefore easy to implement, (*ii*) it is efficient because it only works on local graph neighborhoods, and (*iii*) it is able to detect the number of clusters automatically. Details to MajorClust can be found in [12,11,6].

For one important class of graphs, MajorClust often fails to detect the optimal clustering: near-homogeneous graphs. These graphs have structures on the large scale but do not display significant structures on the local scale. Typical examples are large, unweighted graphs or graphs comprising large clusters with no significant internal substructures. Because MajorClust only relies on local graph information, it often fails to cluster such graphs satisfactorily.

Figure 1 shows a small example: The graph comprises, obviously, two main clusters; because of its local nature, MajorClust computes further (unwanted) clusters within the two clusters. In section 4, further examples are presented.

In the following, the original MajorClust algorithm is first described using a pseudo-code notation. This notation is then used to outline the ideas behind MajorClust. Based on these introductory elucidations, the new MCProb algorithm is presented in section 2—MCProb extends MajorClust and is especially suited for near-homogeneous graphs. An overview of further clustering

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 184–194, 2010.

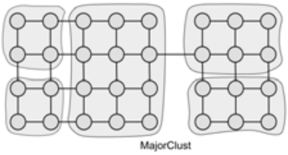


Fig. 1. A suboptimal clustering computed by MajorClust

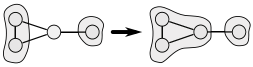


Fig. 2. In each step, MajorClust adds a node to the majority cluster

algorithms can be found in section 3. Empirical results and further comparison to MajorClust and to other established clustering algorithms are given in section 4 and also in [11]. Section 5 analyzes MCProb (and MajorClust) theoretically.

MajorClust.
Input. A graph $G = \langle V, E, w \rangle$ with edge weights $w : E \rightarrow (0, 1]$.
Output. A function $c : V \rightarrow \mathbf{N}$, which assigns a cluster number to each node.

```
(1) $n = 0$
(2) $\forall v \in V$ do $n = n + 1$, $c(v) = n$ end
(3) while clustering not stable do
(4)   $\forall v \in V$ do // random node order
(5)     $c(v) = i, i \in \mathbf{N}_{|V|}$ if $\sum_{\{u,v\} \in E} w(u, v), c(u) = i$ is max.
(6)   end
(7) end
```

MajorClust starts with each node being its own cluster (step 2). MajorClust works then in cycles; one cycle comprises the steps 4 to 6. In each cycle, each node's cluster membership is changed in a random node order: In step 5 each node is added to that cluster to which the (weighted) majority of its neighbors belong to (see also figure 2). MajorClust stops if a stable clustering has been found (step 3, usually defined as an unchanged number of clusters for $k \in \mathbf{N}$ steps). A clustering is stable if within one cycle no significant cluster changes have occurred.

2 A Probabilistic MajorClust Variant

To overcome MajorClust's drawbacks described in section 1, a modified version of MajorClust is introduced here: MCProb. This new clustering algorithm uses probabilistic methods to improve clustering results—especially for near-homongeneuos graphs.

In the following section, MCProb is introduced using a pseudo-code description which is then used to outline the basic ideas behind MCProb.:

MCProb.
Input. A graph $G = \langle V, E, w \rangle$ with edge weights $w : E \rightarrow (0, 1]$.
Output. A function $c : V \rightarrow \mathbf{N}$, which assigns a cluster number to each node.

```
(1) $n = 0$
(2) $\forall v \in V$ do $n = n + 1$, $c(v) = n$ end
(3) while clustering not stable do
(4)   $\forall v \in V$ do // random node order
(5)     $c(v) = i, i \in \mathbf{N}_{|V|}$ with probablity $Z_v \cdot Norm(m_v, \sigma_v, q_v(i))$
        (a) where $Norm(m_v, \sigma_v, q_v(i))$ is the value of the gaussian distribution
            $\frac{1}{2\pi\sigma_v} e^{-(\frac{q_v(i)-m_v}{\sigma_v})^2}$ at place $q_v(i)$ with mean $m_v$ and standard deviation $\sigma_v$
        (b) with $m_v = \max\{q_v(j) | j \in \mathbf{N}\}$ and $\sigma_v = 1$
        (c) and where $q_v(i) = \sum_{\{u,v\} \in E} w(u,v), c(u) = i$ is the (weighted) number of $v$'s neighbors belonging to cluster $i$
        (d) and where $Z_v$ is a normalization factor so that for each $v$ the probabilities sum up to 1
(6)   end
(7) end
```

Just as MajorClust, MCProb also starts off with each node forming its own cluster (step 2). And MCProb also works in cycles; one cycle comprising the steps 4 to 6. MCProb ends if in one cycle no significant cluster improvement has occurred (step 3). A clustering is usually defined as stable if the number of edges within clusters still increases in absolute numbers (i.e. not only compared to the previous step of the algorithm). In each cycle, MCProb–just like MajorClust–iterates over all nodes. But while MajorClust sets a node v's cluster membership always to a cluster to which a (weighted) majority of its neighbors belong, MCProb allows in step 5 v also to become part of a "minority" cluster; this can be seen in figure 3.

MCProb only guarantees that the more neighbors of v belong to a cluster C_i, the more probable it is that v is added to C_i. MCProb does this by computing a probability that v "chooses" cluster C_i ("$c(v) = i$" in step 5). This probability depends on the difference between the (weighted) number of nodes in v's maximum neighbor cluster (m_v) and the number of nodes in C_i (i.e. q_v). This difference is furthermore weighted using the gaussian distribution—generally speaking several weighted schemas are possible.

Figure 3 shows an example: For the central node the most probable result of step 5 is to become part of the left cluster (with a probability of ≈ 63%). But MCProb also allows the choice of the right cluster; but only with a probability of ≈ 37%.

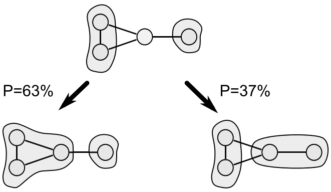


Fig. 3. MCProb can, unlike MajorClust, also choose a non-majority neighbor cluster

3 State of the Art

Clustering is one of oldest fields of computer science; therefore, a huge number of algorithms exists:

Nearest-Neighbor clustering [13] is one of the most established type of clustering algorithm: clusters are created by merging closely related clusters until a predefined criterion–usually the number of clusters–is reached. MinCut clustering ([13]) works the other way around: starting from one large cluster, clusters are created by splitting clusters at a similarity-graph's minimum cut until a predefined stopping criterion is reached. Another class of algorithms, spectral clustering, uses the eigenvalues of the similarity matrix (e.g. in [2]) but is currently mainly used in the field of parallel programming (e.g. for load balancing) and VLSI design. Of course, one can also use optimization algorithms to optimize a given cluster quality criterion; an approach employed e.g. in [10]. In the last years, several density-based algorithms have been developed: Examples are DBSCAN or SUBCLU ([3]), MajorClust, or Chameleon ([5]). Karypis's Chameleon algorithm first splits the similarity-graph into several subgraphs (e.g. using MinCut) and then uses optimization functions to merge these subgraphs into a predefined number of clusters. DBSCAN/SUBCLU algorithms works on vector data whereat dense areas are identified by classifying data points as being (*i*) part of a dense region, (*ii*) close to a dense region, or (*iii*) only noise. Other recent approaches focus on high-dimensional data spaces, e.g. LAC-like methods ([1]) or nCLUSTER ([7]).

4 Empirical Analysis

In the following, the new MCProb algorithm is compared to well-known, established clustering algorithms. Besides MajorClust, MCProb is compared to MinCut clustering and to a Chameleon-like algorithm; details can be found in section 3. For the following experiments, the MinCut and Karypis's Chameleon implementation from [4] is used[1]. MajorClust and MCProb have been implemented here in Lemgo.

As an exemplary application, we apply these algorithms to the clustering of bitmaps. Our target application in Lemgo is the detection of banknote areas; there clustering is used to detect the bitmap's (i.e. the note's) structure, i.e. homogeneous areas of the bitmap. The goal is the detection of faked and dirty notes.

Unlike classical bitmap analysis algorithms such as edge detection (see [14] for an overview), clustering has the advantage of also discovering fuzzy or blurred shapes. Furthermore, bitmap clustering is a thankful testbed for clustering algorithms because the result can be visually inspected and different algorithms can be compared easily.

[1] Cluto's Chameleon-like algorithm is called as follows: "scluster -clmethod=graph -crfun=wslink -agglofrom=n -agglocrfun=g1p"; Cluto's MinCut as "scluster -clmethod=graph -crfun=g1p".

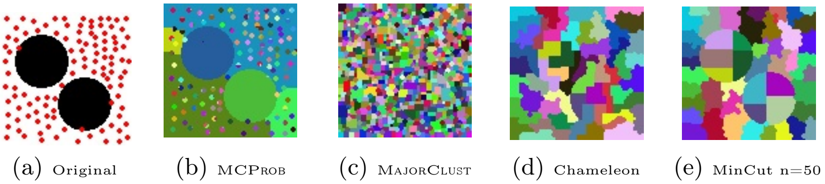

(a) Original (b) MCProb (c) MajorClust (d) Chameleon (e) MinCut n=50

Fig. 4. Test Series 1

For the clustering of bitmaps, pixels are transferred into nodes of a similarity-graph. Neighbored pixels/nodes are connected with edges whose weights correspond to the color differences between the pixels. These graphs are near-homogeneous.

Figure 4 shows a bitmap to-be-clustered: The different sizes of the clusters and the unknown number of clusters make the clustering difficult for Chameleon and MinCut; even if these algorithms would know the correct number of clusters–which they don't–they would still fail to detect the small clusters. In the examples shown here, the number n denotes the number of clusters given to Chameleon and MinCut; both algorithms requiring this number to be given as an input.

MajorClust on the other hand is, as described above, too sensible and detects too many clusters in this near-homogeneous graph. MCProb behaves very nicely; the main drawbacks are caused by its probabilistic behavior: some small clusters have incorrect cluster edges. This is caused by fluctuations due to the probabilistic node membership decision in step 5 of MCProb.

Figure 5 shows another difficult example: Chameleon and MinCut can cope very well with these approximately equally sized, rather large clusters—much better than in example 4. But again, these algorithms suffer from their inability to detect the correct number of clusters. MajorClust on the other hand is very well able to detect the main structures but is too sensitive. The MCProb result is very satisfiable; the main drawbacks are again the fluctuations on the cluster edges. MCProb also overestimates slightly the number of clusters—this being a heritage from MajorClust.

Figure 6 is a good example for the problem of detecting the right number of clusters. Obviously, both Chameleon and MinCut are able to detect the main structures but fail to recognize homogeneous regions, i.e. they fail to detect the

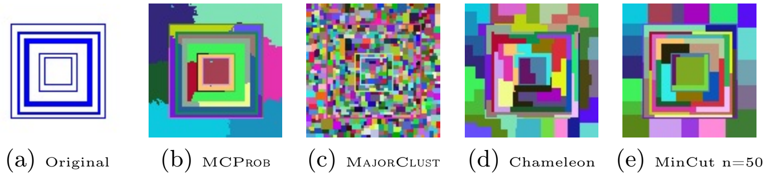

(a) Original (b) MCProb (c) MajorClust (d) Chameleon (e) MinCut n=50

Fig. 5. Test Series 2

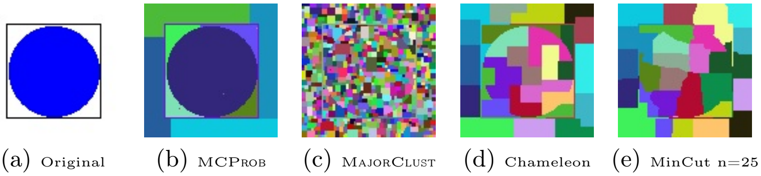

(a) Original (b) MCProb (c) MajorClust (d) Chameleon (e) MinCut n=25

Fig. 6. Test Series 3

correct number of clusters. MajorClust is suffering from its inability to cope with homegeneous regions. Again, MCProb is showing satisfiable results.

On the data sets used in this paper, it is our impression that MCProb outperforms the other examined clustering algorithms.

5 Theoretical Analysis

MCProb (and MajorClust) are probabilistic algorithms, i.e. they may compute different clusters for the same graph if started several times. This section provides a framework for a formal analysis of MCProb (and MajorClust) which allows to determine the probability that a specific clustering is computed, i.e. these probabilities are valid for $n \rightarrow \inf$. This is the first theoretical analysis of MajorClust. Such a theoretical analysis gives valuable insights into the ideas behind the algorithms and may encourage developers to use such probabilistic algorithms. It is here also used to compare the behavior of MCProb and MajorClust.

5.1 General Definitions

First a notion of a state of a graph clustering is needed.

Definition 1. *Clustered Graph. Let $G = \langle V, E, w \rangle$ be a connected undirected graph with nodes $V = \{v_1, \ldots, v_n\}$, edges $E = (e_1, \ldots, e_m)$ and edge weights $w : E \rightarrow (0, 1]$. The clustering is defined by a function $c : V \rightarrow \mathbf{N}$ which assigns a cluster number to each node. Then $\langle V, E, w, c \rangle$ is called a clustered graph. Please note that we assume an arbitrary but fixed ordering of the edges.*

Definition 2. *State of a Clustered Graph. Let $G_c = \langle V, E, w, c \rangle$ be a clustered graph. Then a function $\mu : C \rightarrow \{0, 1\}^{|E|}$ (where C denotes the set of all possible functions c) is defined as follows: $\mu(c) = (b_1, b_2, \ldots, b_{|E|}), b_i = 1$ iff $c(v) = c(u)$ with $e_i = \{u, v\}, e_i \in E$. The result of the function μ for a given clustering is called the current state of a clustered graph. Bits in the state vector mirror the fact whether neighbored nodes belong to the same cluster.*

E.g. a completly connected graph with three nodes whereat all nodes belong to the same cluster is in state $(1, 1, 1)$ since all edges connect nodes within the same cluster.

Definition 3. *Possible States. Let $G_c = \langle V, E, w, c\rangle$ be a clustered graph and $\mu : C \rightarrow \{0,1\}^{|E|}$ be the function defined in definition 2.*

Then PS is defined as the set of states for which a clustering exists: $PS = (\mathbf{s}_1, \ldots, \mathbf{s}_r), \mathbf{s_i} \in \{0,1\}^{|E|}, \exists$ clustering c with $\mu(c) = \mathbf{s}$

5.2 Behavior Analysis

In the following, the behavior of MCProb and MajorClust is described in terms of the well-known markov chain theory. This allows for a formal analysis of the algorithm behavior for specific graphs and also helps to highlight the clustering behavior of both algorithms:

Theorem 1. *Markov Behavior of* MajorClust *and* MCProb*. The behavior of* MajorClust *and* MCProb *on a clustered graph can be described by a series of states $S = (s_1, s_{2,}, \ldots, s_k)$. Each s_i denotes a state of the clustered graph. S forms a homogeneous markov chain.*

Proof: To prove that S forms a markov chain we only have to show that s_{k+1} depends only on s_k. This follows directly from the definitions of MajorClust *and* MCProb *given above: For both algorithms a new clustering c is computed using only the clustering of the previous step. The homogeneity follows from the fact that transition probabilities depend not on time.* □

In the following, we prove that MCProb's and MajorClust's behavior can be described by a special class of markov chain: ergodic markov chains. For these chains analytic means exist to compute the probability that the system is in a given state. This feature can be used to compute the probability that MCProb (and MajorClust) find a specific clustering. Later on, exemplary graphs are analyzed using this approach. The reader may note that this analytic approach is expensive and therefore not suited for computing the clustering for most practical cases.

Definition 4. MCProb *Transition Matrix. Let $G_c = \langle V, E, w, c\rangle$ be a clustered graph and PS as defined in definition 3. Then the matrix $\mathbf{P} = (p_{ij})_{i,j \in \mathbf{N}_{|PS|}}$ is defined as follows:*

1. *$p_{i,j}$ denotes the probability that* MCProb *goes in one step from state $s_i \in PS$ to state $s_j \in PS$. This probability follows directly from the definition of* MCProb *(and* MajorClust*).*
2. *Let be k the column of $\mathbf{P}$ corresponding to the state $(0, 0, \ldots, 0)$. Then we guarantee that the following always holds: $p_{i,k} > \epsilon$ with $0 < \epsilon \ll 1$*
3. *$\sum_j p_{i,j} = 1 \; \forall 1 \leq i \leq |V|$*

Item 2 of definition 4 guarantees that there is a small but positive probability ϵ that state $(0, 0, \ldots, 0)$ can directly be reached from each state. State $(0, 0, \ldots, 0)$ is also called the initial state because MajorClust and MCProb start in this state. Please note that this clustering always exists since with this clustering each node is its own cluster.

ϵ "restarts" the clustering after a some time. This corresponds to several executions of MCProb; please note that this makes sense for a probabilistic algorithm and that in this analysis here we identify the most probable end states, i.e. clusterings. From a markov chain perspective ϵ make the chain recurrent, i.e. without ϵ no stable distribution could be computed. Please note that ϵ must be small enough to give the algorithm sufficient time to reach an end state.

Theorem 2. *Ergodic Markov Chain. Let $G_c = \langle V, E, w, c\rangle$ be a clustered graph and let $\mathbf{P}$ denote its transition matrix as defined in definition 4. Then $\mathbf{P}$ is the transition matrix of an ergodic markov chain.*

Proof: *The show that the homogeneous markov chain S from theorem 1 is ergodic we have to prove that $\mathbf{P}$ is (i) a stochastic matrix with a finite number of states, (ii) that $\mathbf{P}$ has at least one aperiodic state[2], and (iii) that $\mathbf{P}$ is irreducible ([9]).*

To (i) : $\mathbf{P}$ is stochastic because of item 3 of definition 4 and has $|PS|$ states.

To (ii) : Because the initial state can be reached directly from itself, this state is not periodic.

To (iii) : Because it has already been shown that the initial state $(0, 0, \ldots, 0)$ *can directly be reached from all other states, it is here sufficient to show that all states can be reached from the initial state. Let* $\mathbf{s}$ *be an arbitrary state from PS. Because* $\mathbf{P}$ *has been constructed from the behavior of* MCProb *(or* MajorClust*), it is sufficient to show that a clustering corresponding to* $\mathbf{s}$ *can be computed by* MCProb *(or* MajorClust*) starting from the initial state; because of definition 3 such a corresponding clustering c exists.*

c defines q disjunctive clusters $\{C_1, \ldots, C_q\}$. For each cluster C_i there exists a node $v \in C_i$ from which each node in C_i can be reached, e.g. using a depth-first-order. MCProb *(or* MajorClust*) can now construct this cluster by choosing nodes in this order and setting in each step the cluster membership of the node to the cluster of v. Thus because* MCProb *(or* MajorClust*) can compute the corresponding clustering, state* $\mathbf{s}$ *can be reached with a finite number of markov steps from the initial state. I.e. $p^n_{s_i,s} > 0, n \in \mathbf{N}^+$ where $p^n_{i,j}$ denotes an entry in the Matrix $\mathbf{P^n}$ and $\mathbf{s_i}$ denotes the initial state. Therefore $\mathbf{P}$ is irreducible.* □

Theorem 3. *Existence of Stable Distribution. Let $G_c = \langle V, E, w, c\rangle$ be a clustered graph and let $\mathbf{P}$ denote its ergodic transition matrix as defined in definition 4. Then $\mathbf{P}$ has an Eigenvalue of* 1 *and the corresponding Eigenvector $\mathbf{\Pi}$ is the stationary distribution of the markov chain.*

Proof: *Follows directly from theorem 2 and (e.g.) theorem 1.8.3 in [8].* □

A stable distribution is a vector $\mathbf{\Pi} = (\Pi_1, \ldots, \Pi_n)$ for which the equation $\mathbf{\Pi} \cdot \mathbf{P} = \mathbf{\Pi}$ holds. The element $\mathbf{\Pi}_i$ denotes the probability that the markov chain is in state i. Theorem 3 implies therefore that, by computing the first eigenvector, we can compute the probability that the markov chain is in a specific state. Π can be used to compute the probability that MCProb (and MajorClust) finish

[2] Aperiodic means that the number of communication classes necessary to return to this state is 1.

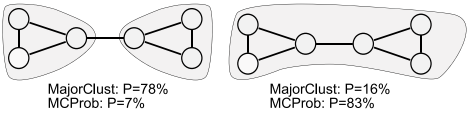


Fig. 7. Clustering probabilities for an exemplary graph

with a specific clustering—times spend in intermediate states can be neglected. Please note that a final state is a state of the markov chain from which mainly only the initial state $(0, \ldots, 0)$ can be reached. I.e. stable clusterings are those states $p_i \in \Pi$ that (a) have a high value (i.e. probability) and that (b) only have a low probability to reach a state different from themselves and from the initial state—i.e. the most probable transition from p_i goes to p_i itself. From a MCProb point of view, this corresponds to a scenario where a stable clustering is reached in which the algorithm remains for some cycles until (with a probability ϵ) MCProb starts again.

Because of their indeterministic behavior, users often distrust probabilistic algorithms such as MCProb and MajorClust. A formal method to analyze the probability, as presented here, helps users to draw confidence in these methods.

In the following the graph in figure 7 is used to illustrate the application of these theorems to analyze MCProb's and MajorClust's clustering behavior: From the graph the corresponding transition matrix $\mathbf{P}$ has been computed. $\mathbf{P}$ is a 50×50 matrix since $|PS| = 50$. Please note that a naive state representation, e.g. using the possible values of the function c, would have resulted in a $6^6 \times 6^6 = 46656 \times 46656$ matrix. So the state representation chosen above reduced the matrix size from $2,176,782,336$ entries to 2500 entries. This complexity reduction is especially important because computing Eigenvectors requires $O(n^3)$ time.

In a next step, the Eigenvector $\mathbf{\Pi}$ corresponding to the Eigenvalue 1 of $\mathbf{P}$ has been computed. Π_i is the probability that the algorithm is in state $\mathbf{i}$. Figure 7 shows the probabilities for two clusterings (i.e. states).

Another example–more relevant for this paper–can be seen in figure 8; the graph does not have edge weights and is therefore a (very small) example for a near-homogeneous graph. MajorClust fails to detect the correct clusters in 51% of all cases, in 32% of all cases the graph is split along a central vertical line in two clusters. MCProb detects the cluster in 93% of all cases. Unlike the examples mentioned in the introduction, these numbers are now based on a formal analysis of the algorithm behavior. Analysis with other near-homogeneuous graphs show similar results. In all cases MCProb outperforms MajorClust.

Figure 9 depicts the probability that MCProb and MajorClust respectively identify a $2 \times N$ grid as *one single* cluster; e.g. figure 8 shows the 2×10 grid. MajorClust reacts to grower graph sizes by identifying additional clusters while MCProb still computes–in most cases–only one cluster.

So the theorem presented here allows for a formal analysis of the behavior of both algorithms for a given graph. Because of the computation costs, this is only

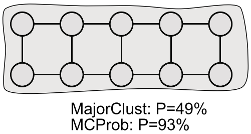


Fig. 8. Clustering probability for an exemplary graph

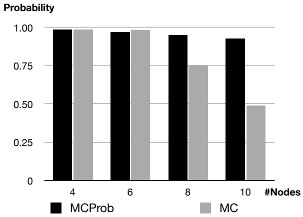


Fig. 9. Probability for correct clustering of a 2xN grid

possible for small graphs. One might even think about MCProb as a sampling algorithm for computing the stationary distribution $\mathbf{\Pi}$ in an efficient way—in the tradition of Monte Carlo or Gibbs samplers (for details refer to [9]).

6 Summary and Future Work

In this paper the new graph clustering algorithm MCProb is introduced. MCProb is based on the established MajorClust algorithm and uses probabilistic methods to improve MajorClust's behavior for an important graph class—near homogeneous graphs, i.e. graphs with global but only few local structures.

MCProb is compared empirically to established clustering algorithms such as Chameleon, MinCut clustering, and, of course, MajorClust. It can be said that MCProb outperforms other established clustering algorithms on the examples presented in this paper. In a next step, the advantages of MCProb compared to MajorClust are shown using theoretical analyses based on markov chain theory.

References

1. Cheng, H., Hua, K.A., Vu, K.: Constrained locally weighted clustering. In: 34nd Internaltional Conference on Very Large Data Bases (2008)
2. Ding, S., Zhang, L., Zhang, Y.: Research on spectral clustering algorithms and prospects. In: 2010 2nd International Conference on Computer Engineering and Technology (ICCET), vol. 6, pp. 16–18 (2010)
3. Kailing, K., Kriegel, H., Kroeger, P.: Density-connected subspace clustering for high-dimensional data. In: 4th SIAM International Conference on Data Mining (2004)
4. Karypis, G.: Cluto - A Clustering Toolkit, http://glaros.dtc.umn.edu/gkhome/views/cluto/
5. Karypis, G., Han, E.-H., Kumar, V.: Chameleon: A hierarchical clustering algorithm using dynamic modeling. Technical Report Paper No. 432, University of Minnesota, Minneapolis (1999)

6. Levner, E., Pinto, D., Rosso, P., Alcaide, D., Sharma, R.R.K.: Fuzzifying clustering algorithms: The case study of majorclust. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 821–830. Springer, Heidelberg (2007)
7. Liu, G., Li, J., Sim, K., Wong, L.: Distance based subspace clustering with flexible dimension partitioning. In: 23th International Conference on Data Engineering (2007)
8. Norris, J.R.: Markov Chains. Cambridge University Press, Cambridge (1997)
9. Bremaud, P.: Markov Chains - Gibbs Fields, Monte Carlo Simulation and Queues. Springer, Heidelberg (1999)
10. Roxborough, T., Sen, A.: Graph Clustering using Multiway Ratio Cut. In: North, S.C. (ed.) GD 1996. LNCS, vol. 1190. Springer, Heidelberg (1997)
11. Stein, B., Meyer zu Eißen, S.: Automatic Document Categorization: Interpreting the Perfomance of Clustering Algorithms. In: Günter, A., Kruse, R., Neumann, B. (eds.) KI 2003. LNCS (LNAI), vol. 2821, pp. 254–266. Springer, Heidelberg (2003)
12. Stein, B., Niggemann, O.: On the Nature of Structure and its Identification. In: 25. Workshop on Graph Theory. LNCS. Springer, Ascona (July 1999)
13. Zadeh, R.B.: Towards a principled theory of clustering. In: Thirteenth International Conference on Artificial Intelligence and Statistics (2010) (under review), http://stanford.edu/~rezab/
14. Ziou, D., Tabbone, S.: Edge detection techniques an overview. International Journal of Pattern Recognition and Image Analysis (1998)

Acceleration of DBSCAN-Based Clustering with Reduced Neighborhood Evaluations

Andreas Thom[1] and Oliver Kramer[2]

[1] Department of Computer Science, Technische Universität Dortmund,
44227 Dortmund, Germany
[2] International Computer Science Institute,
Berkeley CA 94704, USA

Abstract. DBSCAN is a density-based clustering technique, well appropriate to discover clusters of arbitrary shape, and to handle noise. The number of clusters does not have to be known in advance. Its performance is limited by calculating the ϵ-neighborhood of each point of the data set. Besides methods that reduce the query complexity of nearest neighbor search, other approaches concentrate on the reduction of necessary ϵ-neighborhood evaluations. In this paper we propose a heuristic that selects a reduced number of points for the nearest neighborhood search, and uses efficient data structures and algorithms to reduce the runtime significantly. Unlike previous approaches, the number of necessary evaluations is independent of the data space dimensionality. We evaluate the performance of the new approach experimentally on artificial test cases and problems from the UCI machine learning repository.

1 Introduction

Clustering, i.e., the unsupervised detection of groups and connected structures in data sets, has an important part to play in machine learning and data analysis. We assume that a data set $\mathfrak{H} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of samples $\mathbf{x}_i \in \mathbb{R}^d$ is given. Each data sample is a d-dimensional feature vector $\mathbf{x} = (x_1, \ldots, x_d)^T$. A good clustering result is characterized by homogeneity among the elements in the same cluster and heterogeneity of elements in different clusters. Density Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering method introduced by Ester *et al.* [2]. It allows identifying intertwined clusters based on the density of data samples, and does not need pre-knowledge about the number of expected clusters, e.g., like k-means. In Section 2 we review some approaches that have been introduced to improve the runtime of DBSCAN. The new approach to reduce neighborhood evaluations is introduced in Section 3. An experimental analysis concentrating on runtime behavior and clustering error evaluates the new method in Section 4. Finally, Section 5 summarizes the results and provides an outlook to future work.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 195–202, 2010.

2 Related Work

DBSCAN explores clusters in a given data set $\mathfrak{H}$ with the help of nearest neighbor search (NNS). For an introduction to DBSCAN we refer to Ester *et al.* [2]. Here, we restrict ourselves to a short review of the basic concept. For each point $\mathbf{x} \in \mathbb{R}^d$ the ϵ-neighborhood $N_\epsilon(\mathbf{x})$, i.e., the volume within a hypersphere of a given radius ϵ, is inspected. If the ϵ-neighborhood contains at least $|N_\epsilon(\mathbf{x})| \geq \nu$ points, $\mathbf{x}$ is marked as a core point identifying a cluster. Otherwise, $\mathbf{x}$ is either a border point of a cluster or is classified as noise. DBSCAN starts at an arbitrary point $\mathbf{x} \in \mathfrak{H}$ and computes the ϵ-neighborhood $N_\epsilon(\mathbf{x})$. If $\mathbf{x}$ is a core point, for each point $\mathbf{y}$ of set $N_\epsilon(\mathbf{x})$ the set of neighbored points is computed. If $|N_\epsilon(\mathbf{y})|$ is smaller than ν, $\mathbf{y}$ is classified as border point of the found clusters and does not have to be considered anymore. Otherwise, i.e., for $|N_\epsilon(\mathbf{y})| \geq \nu$, all points in $N_\epsilon(\mathbf{y})$ are assigned to the current cluster number $\rho \in \mathbb{N}$, if this has not been done yet. Points that have already been classified as noise are assigned to the cluster. Before a new iteration starts, the current feature vector $\mathbf{x}$ is deleted from set $N_\epsilon(\mathbf{x})$. If $N_\epsilon(\mathbf{x})$ is empty, the method terminates indicating that the cluster has been identified completely.

Efficient data structures are one way to overcome the runtime problems in case of a large data set $\mathfrak{H}$ or high dimensionalities. A bottleneck of the runtime of DBSCAN is the neighborhood request procedure that has to be started for each point. Without an efficient data structure, each point has to be compared with each other point resulting in $\mathcal{O}(n)$ for each neighborhood request, and $\mathcal{O}(n^2)$ requests in total. A suitable data structure to reduce the number of requests is the R-Tree by Guttman [4] that was later modified to R^*-Trees or X-Trees. The expected time complexity of a neighborhood request based on this data structures is $\mathcal{O}(\log n)$. Hence, the runtime of DBSCAN is $\mathcal{O}(n \cdot \log n)$.

Reduction of ϵ-neighborhood evaluations is a further way to reduce the expected time of a neighborhood request. Some approaches try to reduce the number of necessary evaluations of the ϵ-neighborhood. Zhou *et al.* [8] calculate the neighbored points only for $2 \cdot d$ representative points for a d-dimensional data set. For higher dimensions the gain of performance decreases as many representative points have to be evaluated. In contrast, Liu [6] used a kernel function to make the distribution of objects as uniform as possible and reduced the number of neighborhood computations afterwards.

3 Reduced Neighborhood Evaluations

Our Reduced Neighborhood Evaluations (RNE) heuristic is based on the work of Liu [6] and Zhou [7]. But our presented method does not have to transform the data in a preprocessing step, and does not degenerate for higher dimensions d. The idea is to focus the ϵ-neighborhood evaluations on their unexplored area. The goal of any DBSCAN variant must be to significantly reduce the number of ϵ-neighborhood requests, while maintaining the quality of the clustering result.

3.1 Completeness of Clustering Result

One basic idea of Liu [6] and Zhou [7] is to ensure completeness, i.e., to guarantee that DBSCAN detects all clusters, while reducing the necessary number of neighborhood evaluations. This can usually be achieved as follows. In case point $\mathbf{x}$ satisfies the property of a core-point $|N_\epsilon(\mathbf{x})| \geq \nu$, the ϵ-neighborhood of the discovered points $\mathbf{y}_i \in N_\epsilon(\mathbf{x})$ will be evaluated next, and the process starts again. Now, it is essential to choose some of the points of the ϵ-neighborhood of point $\mathbf{x}$ as representative points that will be evaluated next to ensure that an identified cluster is almost completely explored. These representative points should be chosen with respect to the aim of reducing the amount of evaluated points, and to preserve the cluster result of DBSCAN. Zhou *et al.* [7] solve this problem by selecting $2 \cdot d$ points as representative points of the ϵ-neighborhood of the current point $\mathbf{x}$. Hence, in this approach the improvement of performance degenerates in higher dimensions. Only a few representative points might cause a lot of lost points. These points are not enclosed in ϵ-neighborhood of an selected representative data point and its own ϵ-neighborhood includes only none core points. Hence, these points are classified as noise. The question which points are appropriate to be representative points depends on the particular situation, also see [6,7] and the following discussion.

3.2 Rescan of Noisy Data Points for Merging

As not every data point is considered, points may be classified as noise by mistake. A core point may lie in an ϵ-neighborhood that has not been scanned. We assume that the ϵ-neighborhood of noisy points has already been determined. Each point in these neighborhoods that belongs to one cluster is analyzed with respect to the core point property. Finally, for the ϵ-neighborhood of the new core points the intersection with other clusters has to be computed. If core points in the intersection between two clusters are found, the clusters can be merged.

3.3 Chebyshev Metric

An essential element for the reduction of neighborhood evaluations of our approach is the Chebyshev metric $d(\mathbf{x}, \mathbf{y}) = \max_j |x_j - y_j|$. All points $\mathbf{y} \in N_\epsilon(\mathbf{x})$ are sorted in descending order concerning their Chebyshev distance to the current point $\mathbf{x}$. In the following, points $\mathbf{y} \in N_\epsilon(\mathbf{x})$ are considered according to the descending sorting with regard to the current point $\mathbf{x}$. This guarantees that the next evaluated point $\mathbf{y}$ is part of $N_\epsilon(\mathbf{x})$, and cluster C_i is discovered further. The Chebyshev distance ensures that the next considered points are potentially located at the border of ϵ-neighborhood of point $\mathbf{x}$ in one dimension, and these points usually discover an unexplored area of the search space.

Figure 1 illustrates different situations in two dimensions of points located in $N_\epsilon(\mathbf{x})$ with regard to their Chebyshev distance to point $\mathbf{x}$. In part A (upper left) dark blue points have a large Chebyshev distance to core-point $\mathbf{x}$. The light blue points have a close Chebyshev distance to $\mathbf{x}$. Following neighborhood

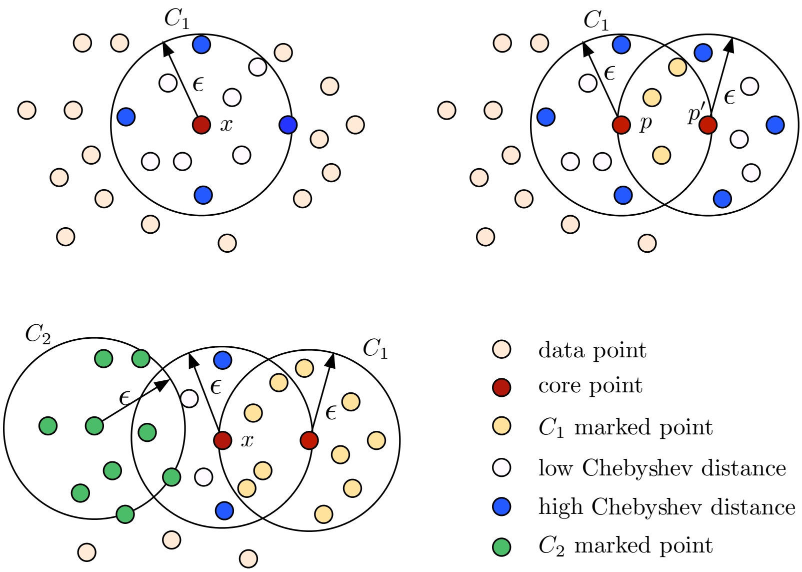


Fig. 1. Various selections of representative data points with regard to the Chebyshev distance of points in $N_\epsilon(\mathbf{x})$

evaluations of one of the dark blue marked points lead to the discovery of a large unexplored area. During this evaluation points that are explored during this neighborhood evaluation and that have already found before, i.e., they have already been representative data points, will be labeled again. In part B (upper right), a large part of the data points is evaluated within the neighborhood evaluation. Bright blue points have been evaluated in the previous step of the neighborhood evaluations of points $\mathbf{x}'$, and do not have to be considered anymore. Part C (bottom left) of Figure 1 shows the selection of new potential data points in case of intersecting ϵ-neighborhoods of different clusters. It is important that every data point is only considered once as representative point. If it is analyzed again, it will be assigned to the clusters of the corresponding core point.

3.4 Changeable Order

Furthermore, our RNE-heuristic is based on the use of efficient data structures and corresponding algorithms. Two aspects are important: the possibility to change the order of points efficiently, and a fast merging of two clusters. The RNE-heuristic clusters the points in an order based on the choice of representatives. This order must be changeable during the clustering process. All points are stored in an array $\mathcal{A}$, and additionally the indices of all points are stored in a double-linked index-list $\mathcal{L}$ that represents the order of points. To reduce the query time of necessary neighborhood evaluations, a kd-Tree is used based on the *sliding-midpoint* construction. Remark that points that stored in the same leaf are located in the same spatial area. This order is used for the base order of the index-list. The chosen representative points are then inserted into $\mathcal{L}$ behind the current point efficiently in $\mathcal{O}(1)$.

3.5 Efficient Merging

We assume cluster C_j has to be merged with cluster C_i, i.e., all points $\mathbf{x} \in C_j$ have to be added to cluster C_i. A union-find data structure that is initially an empty list of lists called U, is used to perform the efficient merging of two or more clusters. For each cluster C_i that has been identified, a double linked list u_i is appended to U. The double linked list u_i with $0 < i \leq |C|$ comprises all indices of points belonging to cluster C_i. To merge two clusters, a double linked list u_j is appended to u_i in $\mathcal{O}(1)$, and all points $\mathbf{x}$ originally belonging to cluster C_j have to be marked with label i of the new cluster, leading to $\mathcal{O}(|C_j|)$ operations. In case of merging m clusters to one, cluster C_i is selected and all other $m-1$ double linked lists are appended to u_i, so that every point is only considered once.

3.6 Handling Noise

Many real-world data sets contain data points with varying density. The *UCI Digits* data set is a good example for varying data densities. In these scenarios it is very difficult to find appropriate parameterizations for ν and ϵ. In order to adjust these parameters, Ester *et al.* [2] propose a heuristic based on a *k-dist*-function. It plots the sorted distances to the k-nearest neighbors of all feature vectors of data set $\mathfrak{H}$. They suggest to use the first break from the left in the *k-dist*-function. But this break cannot easily be identified in every case. Besides parameter tuning, we recommend a post-processing of the DBSCAN / RNE clustering result, i.e., to apply a nearest neighbors assignment of noise. Each noisy data point is assigned to the cluster of the nearest point with a cluster label that has been assigned by the DBSCAN procedure[1].

4 Experimental Evaluation

In the following, we experimentally evaluate the new approach. First, we concentrate on an experimental runtime analysis. Then, we evaluate the clustering error of the new method on typical test problems from the UCI machine learning repository.

4.1 Runtime

Figure 2 shows the experimental comparison between DBSCAN and DBSCAN with RNE-heuristic. Our implementation is based on Python 2.5.4. Although the runtime can be improved by using other programming languages, the relative

[1] Cluster assignments determined during the noise post-processing should be neglected, as the order of processing would induce a bias.

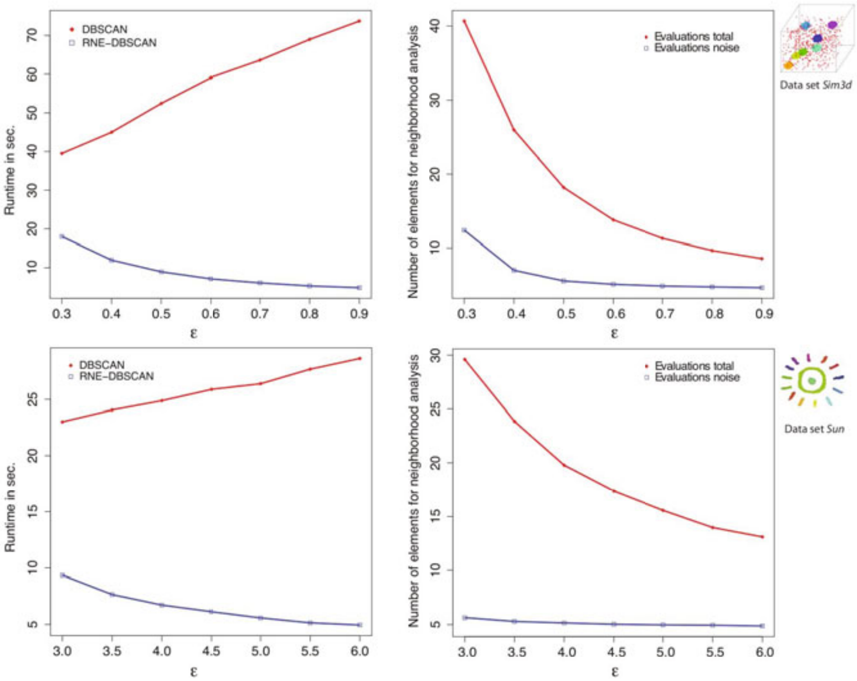


Fig. 2. Comparison of runtime and number of elements for the neighborhood evaluations between standard DBSCAN and DBSCAN with RNE-heuristic on problems *Sun* and *Sim3D*

comparison gives meaningful evidence. The upper part of Figure 2 shows the results for data set *Sim3D*. The left part of the figure shows the runtime in seconds depending on ϵ. The results clearly show that the RNE-variant is much faster than the standard DBSCAN algorithm. Furthermore, RNE becomes faster with increasing ϵ, while the runtime of DBSCAN deteriorates. On the right hand side the corresponding amount of neighborhood evaluations of RNE in total, and in terms of evaluations on noisy points is shown. With increasing ϵ, the total number of evaluations is decreasing. This is the reason for the corresponding acceleration of RNE. The number of noisy evaluations is only decreasing slowly. The lower part of the figure shows the corresponding comparison of DBSCAN and our heuristic on the artificial data set *Sun*. *Sun* consists of 15 clusters, i.e., two circles, and 13 clusters of sunbeams arranged around the inner circles. Also on this data set, the results show that the RNE-heuristic is significantly superior to DBSCAN in terms of runtime. The performance win for higher ϵ in case of the RNE-heuristics is significant, as the runtime of the neighborhood analysis directly depends on the dimension of the data space. Similar to *Sim3D*, the total number of evaluations is decreasing with increasing ϵ. If radius ϵ is chosen too small, e.g. $\epsilon = 0.3$, the compactness of clusters is emphasized. This causes a higher number of noisy points at the border of the clusters. The experiments show that already a slight increase to $\epsilon = 0.4$ leads to a significant reduction of neighborhood evaluations, and less points are classified as noise.

To investigate the dependence of the runtime concerning the size of data set, multiple data sets consists of two separated clouds with a growing number of points are evaluated. The configuration of parameter ν and ϵ are chosen with the *k-dist*-function. In contrast to the runtime of DBSCAN which significantly increases whit growing data size the runtime of RNE only slightly increases due to the fact that RNE depends on the density of data set and the amount of data points which are enclosed in the neighborhood evaluations.

4.2 Comparison on UCI Repository Data

Now, we analyze the RNE-heuristic on a small set of UCI machine learning problems [1]. Table 1 shows a comparison of the RNE-heuristic with three other clustering methods, i.e., k-means [5], EvoMMC, and FastEvoMMC [3]. For the experimental analysis we concentrated on the problems *Digits*, *Ionosphere*, and *Satellite*. Class labels have been removed and the unsupervised techniques have to cluster the unlabeled data. On *Digits*, we cluster four combinations of numbers, in case of *Satellite* we concentrate on the first two of the three available classes. The clustering error is the relative number of false cluster assignments with regard to the class labels from the UCI data set. DBCAN with RNE-heuristic makes use of noise-handling, i.e., a nearest neighbor assignment, see Section 3.6. Table 1 also shows the corresponding RNE-parameterizations for ν and ϵ. The best results are marked in bold.

The experimental results show that the approach is able to show satisfactory results on high-dimensional data. The majority of the data elements have been assigned to the main clusters. Noise has been assigned to these basic clusters with the subsequent nearest neighbor assignment. On *Digits*, the results are competitive to the other clustering techniques, on 3-8 and 8-9 the results are even better. On *Ionosphere*, no approach was able to show satisfying results, but the RNE-heuristic shows worse results than EvoMMC and FastEvoMMC.

Table 1. Comparison of clustering error δ (in %) between k-Means [5], EvoMMC [3], FastEvoMMC [3], and DBSCAN with RNE-heuristic on three UCI data sets, including three variants of *Digits*

	k-Means	EvoMMC	FastEvoMMC	RNE-DBSCAN		
Data set	δ	δ	δ	ϵ	ν	δ
Digits 3-8	5.32 ± 0	2.52 ± 0.5	2.58 ± 0	23.9	12	**0.28** ± 0
Digits 1-7	0.55 ± 0	**0.0** ± 0	**0.0** ± 0	23.0	22	**0.0** ± 0
Digits 2-7	3.09 ± 0	**0.0** ± 0	**0.0** ± 0	25.5	13	**0.0** ± 0
Digits 8-9	9.32 ± 0	3.22 ± 0.40	3.78 ± 4.13	21.0	11	**2.26** ± 0
Ionophere	32 ± 17.9	**17.94** ± 6.84	18.75 ± 3.19	1.3	4	28.21 ± 0
Satellite	4.07 ± 0	1.16 ± 0.45	1.14 ± 0.65	60	11	**0.54** ± 0

5 Conclusions

We have extended DBSCAN to a clustering technique that is able to reduce the number of neighborhood evaluations based on a Chebyshev metric, as well as efficient data structures. Modifications allow changing efficiently the order of the data samples and merging of clusters. A post-processing step allows a reasonable assignment of noise to existing clusters in case of ambiguous data densities. Our experimental analysis on artificial data sets and selected UCI machine learning clustering problems have shown that the approach shows competitive results in terms of runtime and clustering error, in particular in high-dimensional data spaces.

In the future we will conduct further experiments, i.e., on further artificial data sets and also on data from real-world scenarios. Various experiments revealed that density-based clustering techniques are a potentially good choice in many applications, e.g., large astronomical data sets.

References

1. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007) (April 1, 2010)
2. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis, E., Han, J., Fayyad, U. (eds.) 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231. AAAI Press, Menlo Park (1996)
3. Gieseke, F., Pahikkala, T., Kramer, O.: Fast evolutionary maximum margin clustering. In: International Conference on Machine Learning (ICML). ACM Press, New York (2009)
4. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: Yormark, B. (ed.) Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data, Boston, MA, USA, June 18-21, pp. 47–57. ACM Press, New York (1984)
5. Hartigan, J.A., Wong, M.A.: A k-means clustering algorithm. Applied Statistics 28, 100–108 (1979)
6. Liu, B.: A fast density-based clustering algorithm for large databases. In: Proceedings of International Conference on Machine Learning and Cybernetics, pp. 996–1000. IEEE, Los Alamitos (2006)
7. Zhou, S., Zhou, A., Cao, J., Fan, Y., Hu, Y.: Approaches for scaling DBSCAN algorithm to large spatial databases. Journal of Computer Science and Technology 15(6), 509–526 (2000)
8. Zhou, S., Zhou, A., Jin, W., Fan, Y.: ning Qian, W.: FDBSCAN: A fast DBSCAN algorithm. In: Federation, C.C. (ed.) Journal of Software, pp. 735–744. Science Press, Beijing (2000)

Adaptive ε-Greedy Exploration in Reinforcement Learning Based on Value Differences

Michel Tokic[1,2]

[1] Institute of Applied Research, University of Applied Sciences
Ravensburg-Weingarten, 88241 Weingarten, Germany
[2] Institute of Neural Information Processing, University of Ulm, 89069 Ulm, Germany
michel@tokic.com

Abstract. This paper presents "Value-Difference Based Exploration" (VDBE), a method for balancing the exploration/exploitation dilemma inherent to reinforcement learning. The proposed method adapts the exploration parameter of ε-greedy in dependence of the *temporal-difference error* observed from value-function backups, which is considered as a measure of the agent's uncertainty about the environment. VDBE is evaluated on a multi-armed bandit task, which allows for insight into the behavior of the method. Preliminary results indicate that VDBE seems to be more parameter robust than commonly used ad hoc approaches such as ε-greedy or softmax.

1 Introduction

Balancing the ratio of exploration/exploitation is a great challenge in reinforcement learning (RL) that has a great bias on learning time and the quality of learned policies. On the one hand, too much exploration prevents from maximizing the short-term reward because selected "exploration" actions may yield negative reward from the environment. But on the other hand, exploiting uncertain environment knowledge prevents from maximizing the long-term reward because selected actions may not be optimal. For this reason, the described problem is well-known as the *dilemma of exploration and exploitation* [1].

This paper addresses the issue of adaptive exploration in RL and elaborates on a method for controlling the amount of exploration on basis of the agent's uncertainty. For this, the proposed VDBE method extends ε-greedy [2] by adapting a state dependent exploration probability, $\varepsilon(s)$, instead of the classical hand-tuning of this globally used parameter. The key idea is to consider the TD-error observed from value-function backups as a measure of the agent's uncertainty about the environment, which directly affects the exploration probability.

In the following, results are reported from evaluating VDBE and other methods on a multi-armed bandit task, which allows for understanding of VDBE's behavior. Indeed, it is important to mention that the proposed method is not specifically designed for just solving bandit problems, and thus, learning problems with even large state spaces may benefit from VDBE. For this reason, we do

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 203–210, 2010.

not compare the performance with methods that are unpractical in large state spaces because of their memory and computation time requirements.

1.1 Related Work

In the literature, many different approaches exists in order to balance the ratio of exploration/exploitation in RL: many methods utilize counters [3], model learning [4] or reward comparison in a biologically-inspired manner [5]. In practice, however, it turns out that the ε-greedy [2] method is often the method of first choice as reported by Sutton [6]. The reason for this seems to be due to the fact that (1) the method does not require to memorize any exploration specific data and (2) is known to achieve near optimal results in many applications by the hand-tuning of only a single parameter, see e.g. [7].

Even though the ε-greedy method is reported to be widely used, the literature still lacks on methods of adapting the method's exploration rate on basis of the learning progress. Only a few methods such as *ε-first* or *decreasing-ε* [8] consider "time" in order to reduce the exploration probability, but what is known to be less related to the true learning progress. For example, why should an agent be less explorative in unknown parts of a large state space due to a time-decayed exploration rate? In order to propose a possible solution to this problem, this paper introduces a method that takes advantage of the agent's learning progress.

2 Methodology

We consider the RL framework [1] where an agent interacts with a Markovian decision process (MDP). At each discrete time step $t \in \{0, 1, 2, ...\}$ the agent is in a certain state $s_t \in \mathcal{S}$. After the selection of an action, $a_t \in \mathcal{A}(s_t)$, the agent receives a reward signal from the environment, $r_{t+1} \in \mathbb{R}$, and passes into a successor state s'. The decision which action a is chosen in a certain state is characterized by a policy $\pi(s) = a$, which could also be stochastic $\pi(a|s) = Pr\{a_t = a|s_t = s\}$. A policy that maximizes the cumulative reward is denoted as π^*.

In RL, policies are often learned by the use of state-action value functions which denote how "valuable" it is to select action a in state s. Hereby, a state-action value denotes the expected cumulative reward R_t for following π by starting in state s and selecting action a

$$\begin{aligned} Q^\pi(s,a) &= E_\pi\left\{R_t | s_t = s, a_t = a\right\} \\ &= E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right\} \quad , \end{aligned} \tag{1}$$

where γ is a discount factor such that $0 < \gamma \leq 1$ for episodic learning tasks and $0 < \gamma < 1$ for continuous learning tasks.

2.1 Learning the Q Function

Value functions are learned through the agent's interaction with its environment. For this, frequently used algorithms are Q-learning [2] or Sarsa [9] from the *temporal difference* approach, and which are typically used when the environment model is unknown. Other algorithms of the *dynamic programming* approach [10] are used when a model of the environment is available and therefore usually converge faster. In the following, we use a version of *temporal difference* learning with respect to a single-state MDP, which is suitable for experiments with the multi-armed bandit problem. For that reason the discount factor from Equation (1) is set to $\gamma = 0$ which causes the agent to maximize the immediate reward.

A single-state MDP with n different actions is considered where each single action is associated with a stochastic reward distribution. After the selection of action a_t, the environment responds with the reward signal, r_{t+1}, by which the mean reward of action a can be estimated by

$$Q_{t+1}(a) \leftarrow Q_t(a) + \alpha_t \Big[r_{t+1} - Q_t(a) \Big] \ , \tag{2}$$

where α is a positive step-size parameter such that $0 < \alpha \leq 1$. Larger rewards than the so far learned estimate will shift the estimate up into direction of the reward, and lower rewards vice versa. For this, the term in brackets is also called the *temporal-difference error* (TD-error) that indicates in which direction the estimate should be adapted.

Usually, the step-size parameter is a small constant if the reward distribution is assumed to be non-stationary. In contrast, when the reward distribution is assumed to be stationary, then the *sample average* method can be used in order to average the rewards incrementally by

$$\alpha_t(a) = \frac{1}{1 + k_a} \ , \tag{3}$$

where k_a indicates the number of preceding selections of action a. In general, it is important to know that the step-size parameter α is also a key for maximizing the speed of learning since small step-sizes cause long learning times, and however, large step-sizes cause oscillations in the value function. A more detailed overview of these and other step-size methods can be found in [11].

2.2 Exploration/Exploitation Strategies

Two widely used methods for balancing exploration/exploitation are ε-greedy and softmax [1]. With ε-greedy, the agent selects at each time step a random action with a fixed probability, $0 \leq \varepsilon \leq 1$, instead of selecting greedily one of the learned optimal actions with respect to the Q-function:

$$\pi(s) = \begin{cases} \text{random action from } \mathcal{A}(s) & \text{if } \xi < \varepsilon \\ \text{argmax}_{a \in \mathcal{A}(s)} Q(s,a) & \text{otherwise,} \end{cases} \tag{4}$$

where $0 \leq \xi \leq 1$ is a uniform random number drawn at each time step. In contrast, softmax utilizes action-selection probabilities which are determined by ranking the value-function estimates using a Boltzmann distribution:

$$\pi(a|s) = Pr\{a_t = a | s_t = s\} = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_b e^{\frac{Q(s,b)}{\tau}}} \ , \tag{5}$$

where τ is a positive parameter called temperature. High temperatures cause all actions to be nearly equiprobable, whereas low temperatures cause greedy action selection.

In practice, both methods have advantages and disadvantages as described in [1]. Some derivatives of ε-greedy utilize time in order to reduce ε over time [8]. For example, the *decreasing-ε* method starts with a relative high exploration rate, which is reduced at each time step. Another example is the *ε-first* method, where full exploration is performed for a specific amount of time after that full exploitation is performed.

3 ε-Greedy VDBE-Boltzmann

The basic idea of VDBE is to extend the ε-greedy method by controlling a state-dependent exploration probability, $\varepsilon(s)$, in dependence of the value-function error instead of manual tuning. The desired behavior is to have the agent more explorative in situations when the knowledge about the environment is uncertain, i.e. at the beginning of the learning process, which is recognized as large changes in the value function. On the other hand, the exploration rate should be reduced as the agent's knowledge becomes certain about the environment, which can be recognized as very small or no changes in the value function. For this, the following equations adapt such desired behavior according to a (softmax) Boltzmann distribution of the value-function estimates, which is performed after each learning step by

$$\begin{aligned} f(s,a,\sigma) &= \left| \frac{e^{\frac{Q_t(s,a)}{\sigma}}}{e^{\frac{Q_t(s,a)}{\sigma}} + e^{\frac{Q_{t+1}(s,a)}{\sigma}}} - \frac{e^{\frac{Q_{t+1}(s,a)}{\sigma}}}{e^{\frac{Q_t(s,a)}{\sigma}} + e^{\frac{Q_{t+1}(s,a)}{\sigma}}} \right| \\ &= \frac{1 - e^{\frac{-|Q_{t+1}(s,a) - Q_t(s,a)|}{\sigma}}}{1 + e^{\frac{-|Q_{t+1}(s,a) - Q_t(s,a)|}{\sigma}}} \\ &= \frac{1 - e^{\frac{-|\alpha \cdot \mathrm{TD-Error}|}{\sigma}}}{1 + e^{\frac{-|\alpha \cdot \mathrm{TD-Error}|}{\sigma}}} \qquad (6) \\ \varepsilon_{t+1}(s) &= \delta \cdot f(s_t, a_t, \sigma) + (1 - \delta) \cdot \varepsilon_t(s) \ , \qquad (7) \end{aligned}$$

where σ is a positive constant called *inverse sensitivity* and $\delta \in [0,1)$ a parameter determining the influence of the selected action on the exploration rate. An obvious setting for δ may be the inverse of the number of actions in the current state, $\delta = \frac{1}{|\mathcal{A}(s)|}$, which led to good results in our experiments. The resulting effect of σ is depicted in Figure 1. It is shown that low inverse sensitivities cause full exploration even at small value changes. On the other hand, high inverse sensitivities cause a high level of exploration only at large value changes. In the limit, however, the exploration rate converges to zero as the Q-function converges, which results in pure greedy action selection. At the beginning of the learning process, the exploration rate is initialized by $\varepsilon_{t=0}(s) = 1$ for all states.

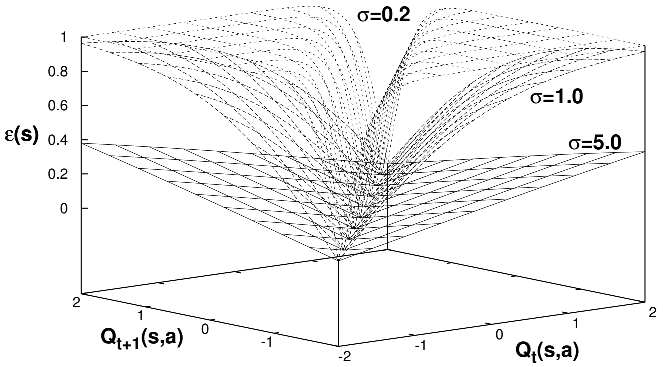


Fig. 1. Graph of $f(s, a, \sigma)$ in dependence of various sensitivities

4 Experiments and Results

A typical scenario for evaluating exploration/exploitation methods is the multi-armed bandit problem [1, 12]. In this example a casino player can choose at each time step among n different levers of an n-armed bandit (slot machine analogy) with the goal of maximizing the cumulative reward within a series of trials. Each pull of a lever returns a numerical reward, i.e. the payoff for hitting the jackpot, which is drawn from a stationary probability distribution dependent on lever a, $a \in \{1, \ldots, n\}$. The player uses the rewards to estimate the "value" of each lever, for example, by averaging the rewards per lever in order to learn which lever optimizes the cumulative reward. During this process, the player has to decide at each time step whether he "exploits" greedily the lever having the highest estimated value or whether he "explores" one of the other levers to improve the estimates.

Some real-world problems analogous to the multi-armed bandit problem are, e.g., adaptive routing in networks with the goal of delay minimization [13] or the economic problem of selecting the best supplier on the basis of incomplete information [14].

4.1 Experiment Setup

The VDBE method is compared and evaluated on a set of 2000 randomly generated 10-armed bandit task as described in [1]. Each selection of a lever returns a stochastic reward drawn from a stationary normal (Gaussian) distribution with mean $Q^*(a)$ and variance 1, where all means are initialized randomly according to a normal distribution with mean 0 and variance 1. The results are averaged over 2000 randomly generated bandit tasks for each exploration parameter, where in each task the player can improve its action selection policy within 1000 trials. Throughout the experiments, learning after each action selection is based on Equation (2) in combination with the sample-average method from Equation (3), which simulates a convergent Q-function in a single-state MDP.

The state-dependent exploration rate of VDBE is immediately recomputed after the value-function backup of the selected action according to Equations (6, 7).

The overall performance is empirically measured and compared against ε-greedy and softmax action selection within a large bandwidth of constant parameter settings. For this, the exploration rate of ε-greedy has been investigated for different values within the interval $[0, 1]$, softmax within $[0.04, 25]$ and VDBE within $[0.04, 25]$, respectively[1]. The δ parameter of VDBE has been set to the inverse of the number of actions, i.e. $\delta = \frac{1}{|\mathcal{A}(s)|} = 0.1$.

4.2 Results

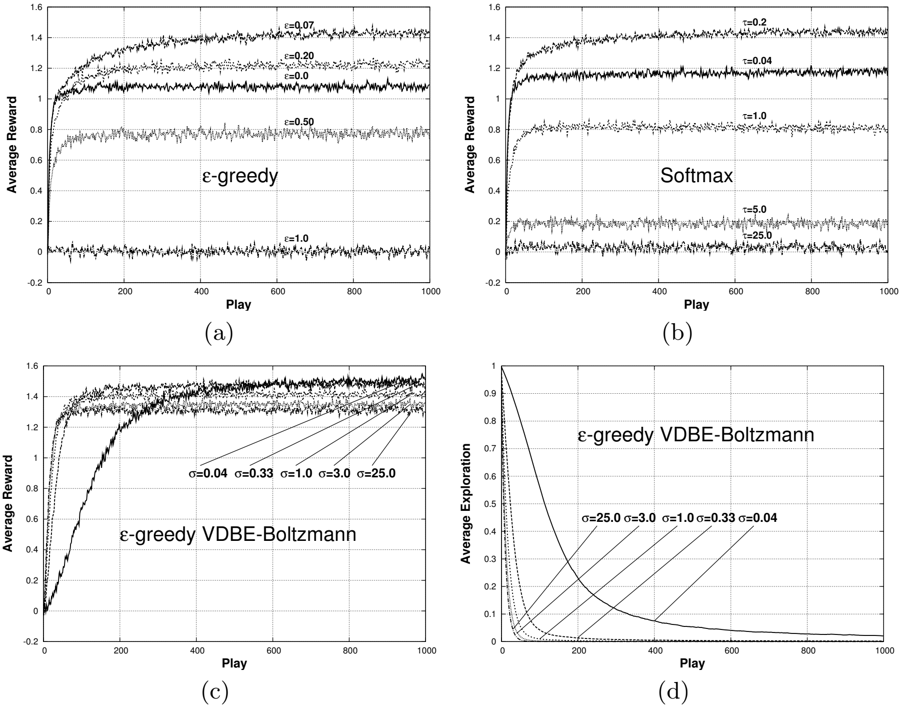
The results depicted in Figure 2 and Table 1 compare the three investigated methods on the multi-armed bandit task. First, from the comparison of ε-greedy and softmax it can be observed that the performance of these methods varies significantly depending on their parameters. The poor performance of both methods is not surprising and due to the fact that a large chosen exploration rate ε causes a large amount of random action selections, whereas the same applies also for high temperatures of softmax. Second, it is also observable that large parameter settings of both methods lead to a relative low level of the mean reward. In contrast, low parameter settings improve the results in the limit (even though very slowly) as the number of plays goes to infinity and when at least a constant bit of exploration is performed. In the case when no exploration is performed at all, the value function will get caught in local minima as it can be observed from the $\varepsilon = 0$ curve, which is equal to softmax and $\tau \rightarrow 0$.

Table 1. Empirical results: $r_{opt} = \frac{r_{t=1} + \cdots + r_{t=1000}}{1000}$ denotes the averaged reward per time step for the parameter that maximizes the cumulative reward within 1000 plays. r_{min} and r_{max} denote the minimum and maximum reward at play $t = 1000$. Numbers in brackets indicate the method's parameter for achieving the results.

Method	r_{opt}	r_{min}	r_{max}
ε-greedy	1.35 ($\varepsilon = 0.07$)	0.00 ($\varepsilon = 1.00$)	1.43 ($\varepsilon = 0.07$)
Softmax	1.38 ($\tau = 0.20$)	0.00 ($\tau = 25.0$)	1.44 ($\tau = 0.20$)
VDBE	1.42 ($\sigma = 0.33$)	1.30 ($\sigma = 25.0$)	1.50 ($\sigma = 0.04$)

In contrast, the advantage of adaptive exploration is shown in the plot of ε-greedy VDBE-Boltzmann. First, it can be observed that the range of the results after 1000 plays is much smaller and in the upper level of the overall reward range than with ε-greedy and softmax. Second, it can also be observed that the exploration rate converges to zero in dependence of the Q-function's convergence and independently of the chosen *inverse sensitivity*.

[1] In detail, the investigated parameters within the intervals have been:
ε-greedy: $\varepsilon \in \{0, 0.01, 0.05, 0.07, 0.08, 0.09, 0.10, 0.20, 0.30, 0.50, 0.80, 1.0\}$
Softmax and VDBE: $\tau, \sigma \in \{0.04, 0.10, 0.20, 0.33, 0.50, 1.0, 2.0, 3.0, 5.0, 25.0\}$



(a)

(b)

(c)

(d)

Fig. 2. Comparison of the average reward on the 10-armed bandit task for (a) ε-greedy, (b) softmax and (c) VDBE. Graph (d) shows the exploration probability of VDBE.

5 Discussion and Conclusion

Based on the results, the VDBE method has been identified to be more robust over a wide range of parameter settings while still achieving acceptable performance results. In case VDBE is used in large state spaces where $\varepsilon(s)$ is approximated as a function (e.g. by a neural network), the method is also robust against errors from generalization since pure exploitation is performed in the limit. Although the method is demonstrated on a single-state MDP, the mathematical principle remains the same in multi-state MDPs where learning is additionally based on neighbor-state information ($\gamma > 0$), e.g. as in Q-learning. The only important assumption for VDBE is the convergence of the Q-function which depends (1) on the learning problem, (2) on the learning algorithm and (3) on the choice of the step-size parameter function. A non-convergent Q-function will cause a constant level of exploration, where the amount is dependent on the chosen inverse sensitivity.

To sum up, the results obtained from the experiments look promising which suggests that balancing the exploration/exploitation ratio based on value differences needs to be further investigated. In order to find out finally whether VDBE outperforms existing exploration strategies—or under which conditions it outperforms other strategies—the application to other more complex learning problems is required.

Acknowledgements. The author gratefully thanks G. Palm, M. Oubbati and F. Schwenker from Ulm University for their valuable comments and discussions on VDBE, which has also been discussed with W. Ertel, P. Ertle, M. Schneider and R. Cubek from UAS Ravensburg-Weingarten. The author also likes to thank P. Ertle, H. Bou Ammar and A. Usadel from UAS Ravensburg-Weingarten for proof reading the paper.

References

[1] Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)

[2] Watkins, C.: Learning from Delayed Rewards. PhD thesis, University of Cambridge, Cambridge, England (1989)

[3] Thrun, S.B.: Efficient exploration in reinforcement learning. Technical Report CMU-CS-92-102, Carnegie Mellon University, Pittsburgh, PA, USA (1992)

[4] Brafman, R.I., Tennenholtz, M.: R-MAX - a general polynomial time algorithm for near-optimal reinforcement learning. Journal of Machine Learning Research 3, 213–231 (2002)

[5] Ishii, S., Yoshida, W., Yoshimoto, J.: Control of exploitation-exploration metaparameter in reinforcement learning. Neural Networks 15(4-6), 665–687 (2002)

[6] Heidrich-Meisner, V.: Interview with Richard S. Sutton. Künstliche Intelligenz 3, 41–43 (2009)

[7] Vermorel, J., Mohri, M.: Multi-armed bandit algorithms and empirical evaluation. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 437–448. Springer, Heidelberg (2005)

[8] Caelen, O., Bontempi, G.: Improving the exploration strategy in bandit algorithms. In: Maniezzo, V., Battiti, R., Watson, J.-P. (eds.) LION 2007 II. LNCS, vol. 5313, pp. 56–68. Springer, Heidelberg (2008)

[9] Rummery, G.A., Niranjan, M.: On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University (1994)

[10] Bertsekas, D.P.: Dynamic Programming: Deterministic and Stochastic Models. Prentice-Hall, Englewood Cliffs (1987)

[11] George, A.P., Powell, W.B.: Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. Machine Learning 65(1), 167–198 (2006)

[12] Robbins, H.: Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society 58, 527–535 (1952)

[13] Awerbuch, B., Kleinberg, R.D.: Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, pp. 45–53. ACM, New York (2004)

[14] Azoulay-Schwartz, R., Kraus, S., Wilkenfeld, J.: Exploitation vs. exploration: Choosing a supplier in an environment of incomplete information. Decision Support Systems 38(1), 1–18 (2004)

Learning the Importance of Latent Topics to Discover Highly Influential News Items

Ralf Krestel[1] and Bhaskar Mehta[2]

[1] L3S Research Center
Leibniz Universität Hannover, Germany
[2] Google Inc.
Zurich, Switzerland

Abstract. Online news is a major source of information for many people. The overwhelming amount of new articles published every day makes it necessary to filter out unimportant ones and detect ground breaking new articles.

In this paper, we propose the use of Latent Dirichlet Allocation (LDA) to find the hidden factors of important news stories. These factors are then used to train a Support Vector Machine (SVM) to classify new news items as they appear. We compare our results with SVMs based on a bag-of-words approach and other language features. The advantage of a LDA processing is not only a better accuracy in predicting important news, but also a better interpretability of the results. The latent topics show directly the important factors of a news story.

1 Introduction

Online news is a major source of information for many people and has been steadily gaining popularity over printed news. Easy access worldwide and a nearly-realtime availability of new breaking news are big advantages over classical paper-based news distribution. The downside of this success is the huge amount of news items generated everyday. For the common user, this situation presents new challenges, since the volume of news makes it difficult – if not impossible – to keep track of all important events.

The current solution is to look at news aggregator sites like Google News[1]. They offer an automatic clustering of news items into broader categories, and collect news from thousands of sources. This is done using information crawled from online news pages. They also offer a ranking based on the information in the Web. One challenge for such aggregators is to pick the most important stories as they break, and to feature them as soon as they are available. Tradional newspapers employ a team of editors to filter the most important news; automatic approaches need smart algorithms to predict such news. We present in this paper an approach to predict the importance of a news item based solely on the content of an article and its inherent topics. In addition, the main contribution of this work is the generation of an easy to understand representation for important news based on LDA topics. Improvements in accuracy over previous approaches show the effectiveness of our approach.

[1] http://news.google.com/

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 211–218, 2010.

2 Related Work

Ranking of news is a rather recent discipline among computer science research. But there is a long tradition trying to explain the importance of some events or news. This research is traditionally conducted by communication theorists or journalists. Starting with [1] who introduced *news value* as a measure of importance followed later by [2] who researched which *factors* make a piece of news important. [3] tried to predict importance (*newsworthiness*) of news based on a manual content analysis of experts. With the access to a lot of news online and the overload of a single user, automatic ranking or filtering of news becomes very important.

Most commercial news sites have some mechanism to rank different news articles. Some have experts ranking the news, some rely on social human filtering. Google News provides an automated news aggreation service which also ranks news stories. [4] suggests the use of collaborative filtering and text clustering. Other features include the source, time of publication, and the size of the cluster.

In [5] the goal is to rank a stream of news. This is done by assigning scores to news sources and articles which can also mutually reinforce each other. Another input feature of the algorithm considers the size of the clusters of similar articles. They argue that this size is a good indicator for importance. Considering the streaming characteristic of news, they treat time as a crucial factor. Old news are less important than fresh news in general. The linear complexity of the algorithm makes it applicable for on-line processing and ranking of news items.

Mutual reinforcement of news sites and news stories is also used in [6]. The authors assume that important news are presented on a visually significant spot on the news page. Visual layout is considered one indicator, the second one is the assumption that important news are covered by many news sites. The relation between news sites, news stories and latent events is represented with a graph model. Sites get scores for credibility, events and stories for their importance. These scores are computed via propagation through the graph.

In [7] this approach is modified by changing the graph structure and only considering news and sources and the corresponding relations between them. The use of a semi-supervised learning algorithm is proposed to predict the recommendation strength of a news site for articles on other news sites, which leads to more edges in the graph and yields a better performance for the algorithm. Similarity between articles is measured using a vector space model and the relation between sources and articles are weighted using visual layout information.

All these systems have one common feature; they use information from news pages on the Internet, either taking the number of similar news articles into account or the internal ranking of articles within news pages. The drawback of these approaches is that they give an overview of what news are there and they rank these news items without regarding their intrinsic importance. Since newspapers and news sites have to publish articles even if nothing really important happened, some news stories might get an inflated score, and thus be highlighed. Further, there is an implied dependence on social feedback, or duplication; however, this information is not necessary available when a news item is reported.

3 Importance Prediction of News Articles

Newspapers or online news providers present news in an unpersonalized way. They have editors who pick the most important news stories for their readers. Good newspapers do this in an unbiased and divers way. The newspaper reader will constantly be confronted with new topics, new opinions, and new views, allowing him to broaden his knowledge and to stay informed on important events. The advantages of an edited newspaper are lost if any kind of personalization is employed. Personalization leads to a limited world view that only covers a focused set of topics or events and in the worst case only a certain opinion about a particular topic. A personalized newspaper does not serve the purpose of a general newspaper anymore. There is no more "surprise" for the user. Important topics are filtered out if they do not fit into the previous reading patterns of a user. Instead of getting controverse view points, articles containting the same topics or opinion as the reader has seen before are prefered. Imagine a user in favour of the democratic party who reads a personalized newspaper. She wants to inform herself for the upcoming elections. In the worst case, the personalization system knows about the pro-democratic attitude of the user and only presents pro democratic articles. Diversity or controversal news coverage is not supported.

Nevertheless, filtering of news articles is essential, solely because of the huge amount published every day. But based on the previous paragraphs we argue that it should not be based on personalization but on importance in an unbiased and highly objective way. Even though it is difficult to draw a sharp line between important news and unimportant ones, it seems easy to identify extremely important news and really unimportant news just by looking at the number of news providers covering a given story. Another alternative is to look at user feedback, e.g. click through rates. While these social features are very strong indications, they are often known only after the story has been around for a while. Our aim is to examine news stories without such signals, so that a reasonably accurate prediction can be made as soon as the article is available electronically. The source of a news story can still be used as signal since this information is available at publishing time. However, the news industry today sources news stories from aggregated news agencies (e.g. the associated Press, or Reuters), and clearly not every story can be deemed important. The filtering algorithm we want to devise relies on plain text, and should make the job of human editors easier by picking the most important stories first. This approach can also be imagined in a TV scenario, where transcribed text from the TV audio is run through a classifier and can recognize important stories as they are made available.

From an historical point of view, certain types of events have triggered the creation of many news articles all over the world. Events like the breakout of a war or big natural catastrophies are important in the sense that they get global news coverage. Our goal is to predict the number of newspapers who will pick up a certain topic and thus estimate the importance of this topic.

4 Features for Importance Prediction

Supervised Machine Learning techniques, in particular Support Vector Machines (SVM), need a set of features for each instance they are supposed to classify. For news prediction, extracting certain language features from news articles and using them as input for a SVM is one solution. We will present this approach together with our approach based on LDA feature reduction.

4.1 Importance Prediction from Language Features

Language features have been studied before in the context of news importance prediction [8]. We implemented the same algorithms to compare the performance with our LDA based approach. We analyzed the effectiveness of part-of-speech information and named entities for importance prediction.

Part-of-Speech Information. Part-of-speech information can be very helpful for various classification tasks. In the area of sentiment analysis, e.g., adjectives have been shown to play a superior role over nouns. For topic classification the opposite is true. In this work we want to find out whether a similar result can be claimed for classifying articles into important and non-important ones. We focused on verbs, nouns, and adjectives as labeled by a part-of-speech tagger.

Named-Entity Information. Since we are interested in building a general classifier named entities seem to be counter productive. They describe well a particular instance of a type of important events but if it comes to abstraction they might not help. We experimented with the most common named entities: Persons, locations, organizations, as well as job titles. The hypothesis is that articles mentioning a “President” or the “NATO” might be important.

4.2 Importance Prediction from Latent Features

In the most general form, we represent news with term frequency vectors (TFV). For each news story, we use text from up to 7 different sources, and then combine the document as a TFV. This representation has a very large number of features and the data is very sparse. [8] explored the effectiveness of SVM based importance classifiers on term frequency vectors. While this approach performed well, generalization was difficult due to the sparseness of features and redundancy. In this work, we propose the use of latent factors derived from dimensionality reduction of text as the features for a classifier.

This dimensionality reduction not only generalizes and smoothens the noise but also decomposes the semantics of the text along different latent dimensions. The latent topics are identified using Latent Dirichlet Allocation (LDA) [9]. LDA models documents as probabilistic combinations of topics i.e. $P(z \mid d)$, with each topic described by terms following another probability distribution i.e. $P(w \mid z)$. This can be formalized as

$$P(w_i) = \sum_{j=1}^{T} P(w_i|z_i = j)P(z_i = j), \tag{1}$$

where $P(w_i)$ is the probability of the ith word for a given document and z_i is the latent topic. $P(w_i|z_i = j)$ is the probability of w_i within topic j. $P(z_i = j)$ is the probability of picking a word from topic j in the document. These probability distributions are specified by LDA using Dirichlet distributions. The number of latent topics T has to be defined in advance and allows to adjust the degree of specialization of the latent topics.

An article about the 2008 Presidential elections in the US would then be represented as a mixture of the “election”-topic (say 40%) and an “Obama & McCain”-topic (maybe 30%) as well as some other latent topics (summing up to 30%). Using LDA, we learn a probabilty distribution over a fixed number of latent topics; for the purpose of classification, we treat the probabilities of latent topics as input features.

5 Predicting Importance Using SVMs

With a representation for each news story and the knowledge about the *importance* for a certain story , we can use supervised learning techniques to train a classifier. Experiments with different machine learning algorithms have shown that Support Vector Machines (SVM) achieve the best results for this task. The input for SVM are the news stories represented as a mixture of probabilities of latent topics, or the tf-idf weights in the bag-of-words approaches.

Evaluating *importance* seems to be at the first glance a very subjective task. To ensure an objective measure and to clearly differentiate our work from what is known under “personalization” of news, we need a measure that is unbiased. In addition, this measure must provide the posibility of a fully automatical evaluation.

We argue that the number of articles published all over the world for a given news story is a good indicator of its importance. If we choose only a fixed number of sources for measuring importance, a certain bias is introduced nevertheless. Instead of a global importance we might therefore only gain a “western” world view of what is important. By selecting different news sources this bias is eliminated. Further, the number of sources for a story can be easily found from Google news service (we call this *cluster size*).

In our last setup we try to predict this number which is a classical *regression problem*. In the particular case of news importance, the actual number of articles might, however, not be the decisive factor. It is generally enough to differentiate between classes, e.g. unimportant news vs. important ones. Thus we can formalize for our first evaluation setting the task as a *classification task* (labeling the stories as important or unimportant). More fine-grained results are achieved using 4 bins: “extremly important, “highly important”, “moderately important”, and “unimportant”. Therefore the corpus is divided not into two equally sized bins but into bins based on cluster size. This acomodates the fact that there is a majority of “unimportant” news stories in the corpus.

In addition to reporting Accuracy, we focus on Receiver Operater Characteristic - Area Under the Curve (ROC-AUC) values [10]. This ensures that we get comparable results even with different sized classes.

For a two-class classification task, where we don’t have an ordering or ranking of the results (e.g. a probability value that an instance belongs to one of the classes) the ROC-AUC value can be computed as: $\textsc{roc-auc} = \frac{1}{2}P_t.P_f + (1 - P_f).P_t + \frac{1}{2}(1 - P_f)(1 - P_t)$ with P_f as the false positive rate and P_t the true positive rate.

6 Experimental Results

For our dataset we collected 3202 stories from the Google news service and crawled 4-7 articles from different sources per story. These stories were collected over a period of one year (2008).

The input for LDA was generated using GATE [11]; we used LingPipe's [12] implementation of LDA and WEKA [13] for the machine learning experiments. All SVM results were obtained using 10-fold cross-validation and a RBF kernel.

To compare our approach to previous approaches for importance prediction [8], we applied the described methods to our corpus. Namely part-of-speech tagging and named entity recognition were used to get an enhanced bag-of-words representation for each news story. With this method, we get ROC-AUC values of up to 0.683 for the two equally sized bins classification compared to 0.728 with our LDA approach. This is an increase of more than 10%.

6.1 Two-Class Classification

In Table 1 (left) the results for filtering the input data making use of part-of-speech information and Named Entity Recognition is shown. The numbers indicate that the preselection of certain word types is decreasing the prediction accuracy. Overall best accuracy is 64.52% achieved using all word types. Nouns tend to have a higher predictive value as e.g. persons. In the following we will compare our results only with the bag-of-words (BOW) approach, since keeping all word types yielded the best results.

Table 1. Results for using different language features (left) and different number of LDA topics T (right) for binary classification on equally sized bins

Feature	Accuracy	ROC-AUC
all types	64.52%	0.683
verbs, nouns, adj.	63.65%	0.677
nouns	62.90%	0.668
named entities	61.84%	0.645
verbs	58.68%	0.606
adjectives	58.34%	0.608
persons	57.78%	0.598
locations	56.62%	0.589
jobtitles	55.56%	0.581
organizations	55.56%	0.573

No. of LDA Topics T	Accuracy	ROC-AUC
50	63.59%	0.682
100	65.58%	0.716
250	64.83%	0.709
500	65.99%	0.720
750	65.15%	0.717
1000	66.27%	0.728
2500	65.87%	0.723

Using LDA to reduce the number of features improves not only efficiency and interpretability but also accuracy. We evaluated the performance of our algorithm varying the number of LDA topics generated out of the news data. The ROC-AUC values are between 0.682 for 50 LDA topics and 0.728 for 1000 (see Table 1 right). The best accuracy is 66.27%. The higher the number of latent topics, the more specific are the LDA topics.

6.2 Regression Setup

The correlation coefficient is 0.47 for using LDA compared to 0.39 when using the bag-of-words approach. Root relative squared error is with 89.14% rather high but still better then using BOW.

Figure 1 show ROC curves for varying the threshold. We therefore do a normal regression and then systematically lower the threshold for a story to be important starting from 1.0. For each threshold we get a false positive rate and a true positive rate. The green line ($f(x) = x$) indicates a random algorithm. Our results are significantly better for both, BOW and LDA.

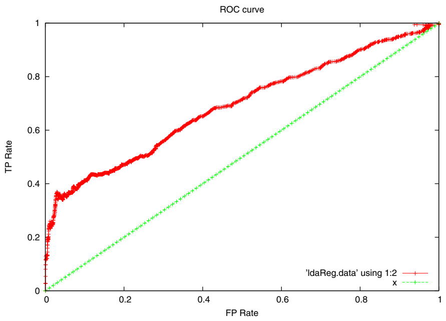


Fig. 1. ROC curve for different thresholds (cluster size) to seperate unimportant and important topics in the regression setup using LDA

6.3 LDA Topics

Table 2 shows the three top ranked LDA topics with respect to information gain. A detailed analysis of the model built by the classifier revealed that the first topic (Topic 128) indicates an unimportant news article whereas the other two indicate important news. Since we did this evaluation using 250 LDA topics to represent our documents, some LDA topics contain actually two “topics” (oil, nigeria, indonesia). Other LDA topics indicating importance are e.g.: “McCain, Obama, campaign” or “gas, Ukraine, Russia, Europe”.

Table 2. Top features based on information gain. First two indicating important news; third one indicating unimportance. For each word also the number of occurances in the corpus is displayed, as well as the probability that the word belongs to the topic.

Topic84			Topic197			Topic128		
Word	Count	Prob	Word	Count	Prob	Word	Count	Prob
afghanistan	5223	0,120	oil	2256	0,135	able	1803	0,069
afghan	1801	0,041	nigeria	363	0,022	browser	1786	0,068
nato	1732	0,040	company	334	0,020	content	1298	0,049
taliban	1288	0,030	militant	302	0,018	view	1275	0,049
troops	1153	0,027	barrel	280	0,017	style	1206	0,046
country	869	0,020	production	263	0,016	enable	1200	0,046
kabul	709	0,016	crude	246	0,015	sheet	1187	0,045
force	665	0,015	niger delta	240	0,014	css	1136	0,043
security	573	0,013	attack	225	0,013	bbc	1017	0,039
fight	569	0,013	pipeline	211	0,013	internet	720	0,027

7 Conclusions and Future Work

The main goal of this work was to make the importance prediction more accessible in the sense of easy interpretable results. We have shown that LDA can achieve this. It is possible to identify general news events, e.g. the war in Afghanistan, and predict the importance of future articles dealing with this topic. Also general events like elections were identified by LDA and help to predict the coverage of other future elections in the media. In conclusion, we explored a new approach to the problem of finding important news as a classification problem using LDA topic mixtures. The results show that accuracy is better than state-of-the-art bag-of-words results. We can find important articles with an accuracy considerably better than random, and around 10% better than previous approaches.

The biggest open issues concerns time. Our current approach has no time dimension. Incorporating the temporal aspects of news articles is decisive to improve accuracy further. Not only because "nothing is older than yesterday's news", but also because of the interdependence of news stories. A news story might not be important only based on its content but also because there was another news story related to it. For future work we try to involve these temporal aspects of news. Extending the LDA implementation to consider a time dimension might be neccessary to achieve this. We also try to incorporate more knowledge from the domain of journalism where research on importance factors of news articles has been carried out. A detailed analysis of mentioned numbers or dates with articles might also improve accuracy.

References

1. Lippmann, W.: Public Opinion. Harcourt, Brace and Company, New York (1922)
2. Østgaard, E.: Factors influencing the flow of news. Journal of Peace Research 2 (1965)
3. Kepplinger, H.M., Ehmig, S.C.: Predicting news decisions. an empirical test of the two-component theory of news selection. Communications 31(1) (April 2006)
4. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google News Personalization: Scalable Online Collaborative Filtering. In: Proc. of the 16th World Wide Web Conference. ACM, New York (2007)
5. Corso, G.D., Gullí, A., Romani, F.: Ranking a stream of news. In: Proc. of the 14th International Conference on World Wide Web. ACM, New York (2005)
6. Yao, J., Wang, J., Li, Z., Li, M., Ma, W.Y.: Ranking web news via homepage visual layout and cross-site voting. In: Lalmas, M., MacFarlane, A., Rüger, S.M., Tombros, A., Tsikrika, T., Yavlinsky, A. (eds.) ECIR 2006. LNCS, vol. 3936, pp. 131–142. Springer, Heidelberg (2006)
7. Hu, Y., Li, M., Li, Z., Ma, W.Y.: Discovering authoritative news sources and top news stories. In: Ng, H.T., Leong, M.-K., Kan, M.-Y., Ji, D. (eds.) AIRS 2006. LNCS, vol. 4182, pp. 230–243. Springer, Heidelberg (2006)
8. Krestel, R., Mehta, B.: Predicting news story importance using language features. In: Proc. of 2008 IEEE / WIC /ACM International Conference on Web Intelligence. IEEE, Los Alamitos (2008)
9. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
10. Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: Proc. of ICML 1998. Morgan Kaufmann, San Francisco (1998)
11. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: ACL 2002 (2002)
12. Alias-i: Lingpipe 3.7.0 (2008), http://alias-i.com/lingpipe
13. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Methods for Automated High-Throughput Toxicity Testing Using Zebrafish Embryos

Rüdiger Alshut, Jessica Legradi, Urban Liebel, Lixin Yang, Jos van Wezel, Uwe Strähle, Ralf Mikut, and Markus Reischl

Karlsruhe Institute of Technology (KIT), P.O. Box 3640, 76021 Karlsruhe, Germany

Abstract. In this paper, an automated process to extract experiment-specific parameters out of microscope images of zebrafish embryos is presented and applied to experiments consisting of toxicological treated zebrafish embryos. The treatments consist of a dilution series of several compounds.

A custom built graphical user interface allows an easy labeling and browsing of the image data. Subsequently image-specific features are extracted for each image based on image processing algorithms. By means of feature selection, the most significant features are determined and a classification divides the images in two classes. Out of the classification results dose-response curves as well as frequently used general indicators of substance's acute toxicity can be automatically calculated. Exemplary the median lethal dose is determined. The presented approach was designed for real high-throughput screening including data handling and the results are stored in a long-time data storage and prepared to be processed on a cluster computing system being build up in the KIT. It provides the possibility to test any amount of chemical substances in high-throughput and is, in combination with new screening microscopes, able to manage ten thousands of risk tests required e.g. in the REACH framework or for drug discovery.

1 Introduction

In 2007, the new EU legislation on chemicals REACH (Registration, Evaluation, and Authorization of Chemicals) has become effective. This means that chemical substances reaching an annual production or import quantity of 1t at least have to be tested and registered with respect to their effects on health and the environment. This causes an immense amount of animal testing. Zebrafish embryos are an ethically acceptable vertebrate model, can be handled easily to manage tens of thousands of risk tests required within the framework of REACH. Also it has been shown in DarT that the zebrafish model organism is sensitive to human hazardous substances [1].

For determining and comparing the acute toxicity of thousands of compounds fish embryo toxicity tests have become a common approach. The zebrafish **(*Danio rerio*)** has several features, for instance, its morphological and physiological similarity to mammals, its fertility, its transparency, and its *ex utero* development, making it a good model organism for these assays [2]. More importantly, standard protocols for conducting the fish embryo test have already been defined [3, 4, 5, 6] that ensure

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 219–226, 2010.

reproducibility and comparability. Zebrafish is the best developed and most promising model for high-throughput *in vivo* screening of intact organ systems [1]. However several challenges remain. On the one hand, the zebrafish model must attain a high level of reliability in order to become widely accepted as a standard model for biological testing. On the other hand, there is a need for standardizing and fully automating the assays since manual inspection of tens of thousands of images is not feasible for high-throughput analysis [7]. The evaluation of the data produced by a screening microscope is one of the most challenging tasks in automation of the whole fish embryo test. Due to a lack of robust solutions this step is currently performed by hand. Commonly a screening is performed 24 or 48 hours after fertilization. At this time, the embryo is already at a level of development that allows him to move within the fish-egg called *chorion* in biological terminology. Thus, neither the alignment of the embryo within the chorion is fixed nor has the chorion a well-defined position in the image. The acquired image can show the embryo in different orientations, for example from dorsal, lateral, or ventral. Figure 1 shows examples of possible alignments of living embryos as well as a coagulated dead egg within the chorion.

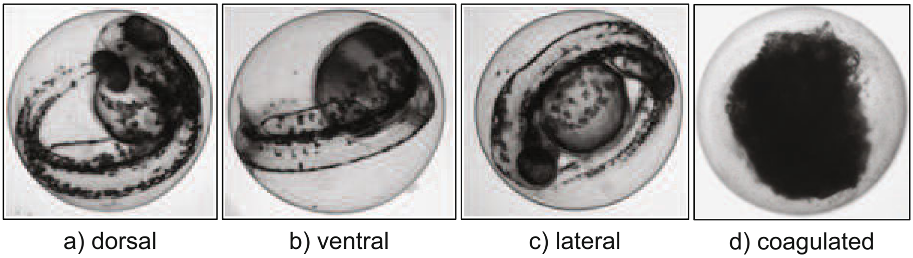


Fig. 1. Examples of alignments of living and dead embryos

Up to now for any automated evaluation, the embryo is manually dechorionated and its orientation is fixed prior to acquiring the images with the microscope [8, 9, 10]. This approach is time-consuming and automation is difficult.

In this paper we present a fully automatic processing chain, which detects the embryo in the well, classifies its status and derives characteristic parameters for the complete embryo cohort. We implemented all algorithms into a graphical user interface and showed its functionality with three datasets.

2 Material and Methods

2.1 Overview

The zebrafish screen is divided in six steps as shown in Figure 2. After defining the toxins to be tested in the assay (1.) the fish is crossed to gather the required fish eggs (2.) which are than treated by the toxins (3.). To find hazardous concentration doses a toxin needs to be tested in a dilution series. For lowest doses every organism should survive whereas beyond a specific dose all of them will show mortality.

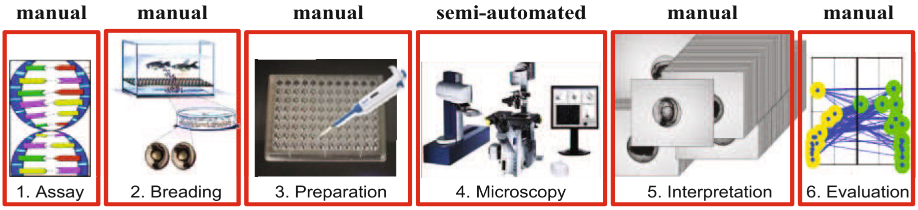


Fig. 2. Workflow in State-of-the-Art toxicological screening assays

At a defined development stage a microscope captures at least one image (4.) of each embryo. These images are classified into the classes *dead* and *alive* (5.). Subsequently the result is evaluated and visualized (6.). Nowadays, all of these processes are done manually as Figure 2 shows.

The interpretation and evaluation (5.-6.) is a time-consuming, tedious and highly subjective step. It consumes up to 70% of the overall processing time and therefore needs to be automated. The essential information to derive is the mortality with respect to a toxin dose. Generally, the microscope images of the zebrafish are observed in terms of so-called endpoints which are defined in standard protocols [4, 6]. The mortality by coagulation of the egg is one of the endpoints which most frequently occurs and furthermore is the most unambiguous. Subsequently the goal is to find or calculate features out of the image that provide a distinction between alive and coagulated. The percentage amount of mortality is then related to the dose exposed to the zebrafish embryo.

We propose a processing automation divided into four basic parts:

- the biological assay and the image acquisition,
- a graphical user interface,
- the image processing algorithm and classification process, and
- the derivation of the dose-response relationships[1].

2.2 Biological Assay and Image Acquisition

Due to the rapid development of the embryos it is necessary to acquire the images within a small time frame. The age difference between zebrafish, measured in hours post fertilization (hpf) should not exceed approximately three hours. For the image taking we used automated high-throughput microscopes equipped with a robot arm form loading 96-well round-bottom microtiter plates. The embryos were exposed to the toxins at an age of 4 hpf and the images were taken at approximately 48 hpf. Imaging of the 96-well plates was carried out on a ”ScanR” high content screening microscope [11] (Olympus Biosystems) with a Hamilton Micro-Lab SWAP plate gripper, a 2.5x objective and an Olympus Biosystems DB-1 (1344x1024 pixels) CCD camera. In an initial step, the central focal plane of the embryo was detected once by

[1] Other common values of interest such as LC50 (lethal concentration 50), LOEL (lowest observed effect level) and NOEL (no observed effect level) may be derived as well.

hand. This position was used as reference for all wells of a plate. The other imaging parameters like e.g. exposure time where also detected by hand and then fixed for one plate. An overview of the dataset is given in Table 1.

Table 1. Dilution series: Used toxins and doses

Toxin	Unit	Dose							
PbCl2	[mg/l]	80	100	120	140	160	180	200	220
Methanol	[%]	3.00	4.00	4.50	5.00	6.00			
As2O3	[mM]	0.80	0.90	1.00	1.20	1.40			

At least 12 embryos were exposed to each concentration and dilution series; 12 negative control embryos in fish water were included. Additionally, a replicate of the whole assay was made. All together, 576 embryos (i.e. 7 plates) were prepared, exposed and screened by the microscopes. For *Methanol* and *As2O3* (arsenic trioxide) one plate and for *PbCl2* two plates were prepared. The second plate increases the range of the dilution series for *PbCl2.*

2.3 Graphical User Interface

To evaluate the mortality status, labeled embryo images are necessary. As the labeling of all images would be a time-intensive step, we propose to train classifier on a small part of the data set (called training data set) and let the classifier label the rest of the data.

However, the image properties differ between each assay, microscope or even between different days because of unequal illumination or microscope setups. To obtain a maximum classifier performance the training set needs to contain examples reflecting all image properties. This makes it necessary to provide a simple method for an expert to label the captured data without being coped with details of the classification procedure. Thus, a graphical user interface was developed. It allows an expert (e.g. a biologist) to label the data to the classes dead, alive or unknown. Additionally, the software provides comfortable browsing through the images filtered to specific toxins or concentrations. Without an adequate browsing and labeling tool the amount of data is hard to handle. Also the risk of probable mistakes by equivocating plates or toxins is big if data is organized in traditional ways such as papers and folder names. The labeling procedure can be progressed by simply clicking at any point in the feature space. Thereupon a popup window with the related image of the zebrafish egg is opened and the users can decide (i.e. label) to which class the content belongs to. Additionally, the user gets a close feeling of the influence of outliers and will be able to easily identify faulty labeled data. The classifier is trained and the image processing routine and the classification process can be applied to the whole dataset as soon as a sufficient amount of data is labeled. All these software features are implemented in the open source toolbox Gait-CAD [12].

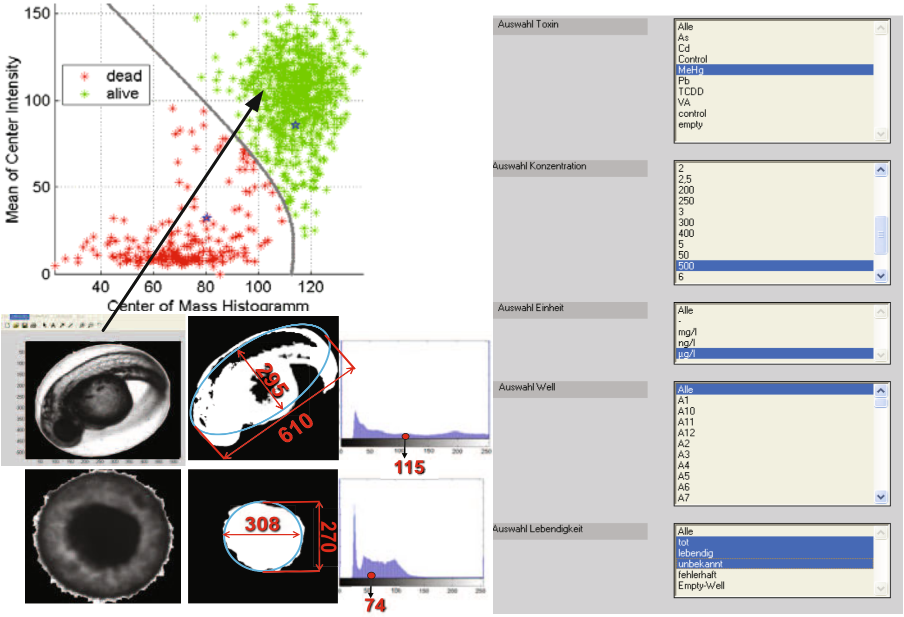


Fig. 3. Graphical User Interface implemented in the Open Source Toolbox Gait-CAD [12]

2.4 Image Processing and Classification

To transform each image to the feature space, an image processing chain was developed. For each image, the contrast is optimized over the available range of intensity levels and then the chorion is distinguished from the background and the artifacts in the image. This is done by a filter chain. It consists of an adaptive threshold and a combination of morphologic operators to calculate a mask. Using this mask the chorion is distinguished from the background. The result is cropped and stored and further processing can be performed. The routine is described in detail in [13].

Once the chorion is isolated from the image, eight robust features for calculating and describing the morphology of dead and living embryos are extracted. Since the zebrafish embryo can appear in different alignments the extracted features are designed to differentiate between the black center of a coagulated egg compared to the developed eggs. Powerful features are the amount of black pixels within the chorion, [1] or the variance of pixel values within the chorion. Eight of these features are extracted (details in [13]) and for reasons of data illustration, interpretability, and classifier robustness, a selection of the two most powerful features by MANOVA (multivariate analysis of variances) is performed. Using these two features, the classification is performed utilizing a Bayesian classifier. However, the visual content and the image properties differ between each assay, microscope or even between different days.

2.5 Derivation of Dose-Response Relationships

After the decision rule is derived all data is labeled automatically. Each concentration is treated as a subset and classified using the trained algorithm. Thus, each embryo is classified as coagulated or alive and each subset is described by the number of classified coagulated embryos. Out of this number a percentage of mortality is calculated. Finally, standard dose-response curves are computed using non linear regression. The described procedure can be applied to any amount of data containing a dilution series.

Figure 4 shows the outcome of the algorithm: After a plate has been chosen by the user in the GUI the automatic classification is performed. Every egg classified as coagulated is marked by a red rectangle (see Figure 4).

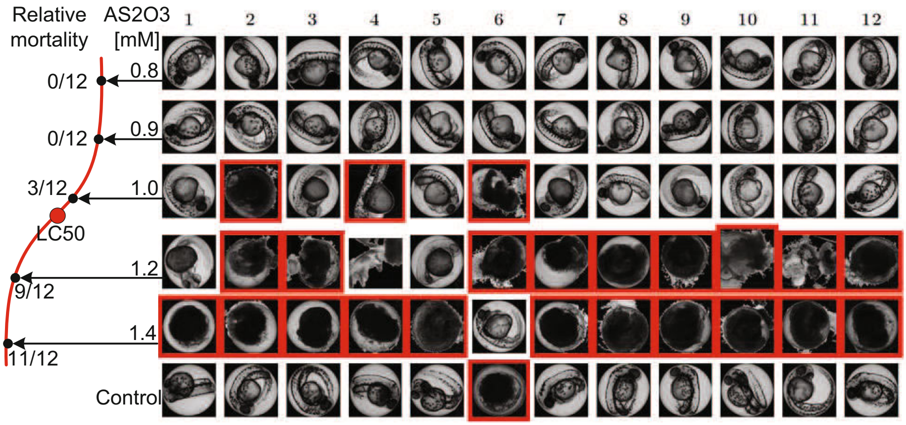


Fig. 4. Automatically generated report to visualize the control a easily probability check over thousands of data

Furthermore, a project report is created containing an overview of the plate. It shows for each toxin a bar plot with the relative mortality. The result of the dose-response regression and the LC50 value calculated out of this can be added. The whole report is saved as a PDF document and provides all important information of the assay, like a laboratory journal. Assay data stored in this manner can easily be verified and archived by the biologist. The report created with the GUI is also suitable if a manual check of the classification result is required.

3 Results

For each toxin a dilution series including 12 negative control embryos was prepared. For evaluation purposes a replica for each series was also tested. This results in an amount of 7 microtiter plates containing 72 zebrafish embryos each and an overall amount of 576 embryos. Out of the data set a training set containing 154 images, the classifier was trained. Based on this training set we let the classifier estimate the rest of the data set. Even though the data set contains for this demonstration only ~600 examples the build classifier could in the same manner be applied to any amount of examples the assay might contain. Using 5-fold cross-validation the average error of ten runs is 1.2±0.2% over the manual labeled training data set. Subsequently the classifier was applied to the unlabeled data and the assay was split into single subsets for every toxin and every concentration. This means 6 classifier runs for each plate and thus 84 for the whole assay. The resulting dose-response curves given in the project report and the LC50 value are shown in Figure 5.

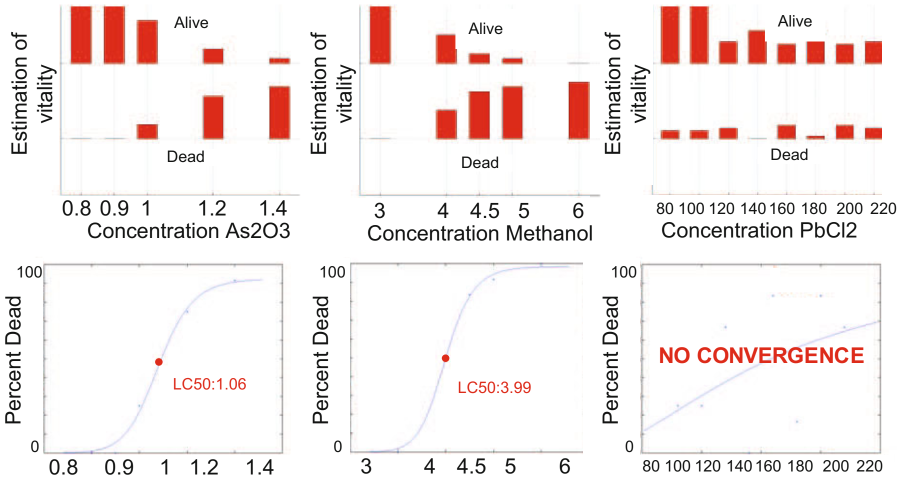


Fig. 5. Dose-response curves and resulting LC50 values for three different toxins

For the toxins *AS2O3* and *Methanol* the typical sigmoid form of the dose-response curve can be fitted well to results of the estimator. Also the bar plots show an increasing amount of mortality for increasing doses, from no effect to almost 100% effect. The calculated LC50 for *AS2O3* is at a dose of 1.08 mM/l and for *Methanol* at a dose of 3.99%. For *PbCl2* obviously the exposed dose was to low so only a few examples showed coagulation. This means that the dose-response regression cannot converge to a sigmoid curve and the LC50 value is not found. In this case the assay needs to be repeated with an increased dose until a successive transition from no effect to 100% effect is shown in the experiment. Although the *PbCl2* experiment was not successful regarding the biological goal to determine the LC50 value the classification and evaluation did work well.

4 Outlook

The increasing number of experiments, concentrations and fishes per experiment will lead to vast amounts of data in the next few months. As a result, we will face problems in generating, saving, processing, and accessing the data. In order to process and to store these large amounts of data a dedicated infrastructure for data intensive computing is being built at KIT. Traditionally, high performance computing (HPC) environments are optimized for computing and for processing moderate amounts of data. The organization of massive data storage, movement, visualization and analysis is concentrated in the Large Scale Data Facility (LSDF) of the Steinbuch Computing Centre at KIT. The LSDF focus is on technologies and services to process large amounts of data, using state-of-the-art hardware. High speed data rates are possible along the whole path from data taking via analysis and process storage to long time storage.

References

[1] Nagel, R.: DarT: the embryo test with the zebrafish (Danio rerio) –a general model in ecotoxicology and toxicology. ALTEX- Alternativen zu Tierexperimenten 19(Suppl. 1), 38–48 (2002)

[2] Zon, L.I., Peterson, R.T.: In vivo drug discovery in the zebrafish. Nature Reviews Drug Discovery 4(1), 35–44 (2005)

[3] Braunbeck, T., Böttcher, M., Hollert, H., Kosmehl, T., Lammer, E., Leist, E., Rudolf, M., Seitz, N.: Towards an alternative for the acute fish LC50 test in chemical assessment: the fish embryo toxicity test goes multi-species – an update. ALTEX- Alternativen zu Tierexperimenten 22(2), 87–102 (2005)

[4] DIN, DIN 38 415-T6, German standard methods for the examination of water, waste water and sludge – Subanimal testing – Part 6: Determination of the non-acute-poisonous effect of waste water to fish eggs by dilution limits. German Standardization Organization, Beuth Vertrieb GmbH, Berlin (2001)

[5] OECD, Background Paper on Fish Embryo Toxicity Assay (May 2006)

[6] OECD, Fish embryo toxicity (FET) test. Draft OECD guideline for the testing of chemicals (May 2006)

[7] Carpenter, A.E.: Image-based chemical screening. Nature Chemical Biology 3(8), 461 (2007)

[8] Gehrig, J., Reischl, M., Kalmar, E., Ferg, M., Hadzhiev, Y., Zaucker, A., Song, C., Schindler, S., Liebel, U., Müller, F.: Automated high throughput mapping of promoter-enhancer interactions in zebrafish embryos. Nature Methods 6(12), 911–916 (2009)

[9] Liu, T., Lu, J., Wang, Y., Campbell, W., Huang, L., Zhu, J., Xia, W., Wong, S.: Computerized image analysis for quantitative neuronal phenotyping in zebrafish. Journal of Neuroscience Methods 153, 190–202 (2006)

[10] Lu, J., Liu, T., Nie, J., Ding, J., Zhu, J., Yang, J., Guo, L., Xia, W., Wong, S.T.: Automated quantitation of zebrafish somites in high-throughput screens. In: Proc. IEEE/NLM Life Science Systems and Applications Workshop, pp. 1–2 (2006)

[11] Liebel, U., Starkuviene, V., Erfle, H., Simpson, J., Poustka, A., Wiemann, S., Pepperkok, R.: A microscope-based screening platform for large-scale functional protein analysis in intact cells. FEBS Letters 554(3), 394–398 (2003)

[12] Mikut, R., Burmeister, O., Braun, S., Reischl, M.: The Open Source Matlab Toolbox Gait-CAD and its Application to Bioelectric Signal Processing. In: Proc. DGBMT-Workshop Biosignalverarbeitung, Potsdam, pp. 109–111 (2008)

[13] Alshut, R., Legradi, J., Mikut, R., Strähle, U., Reischl, M.: Robust identification of coagulated zebrafish eggs using image processing and classification techniques. In: Proc. 19. Workshop Computational Intelligence, pp. 9–21 (2009)

Visualizing Dissimilarity Data Using Generative Topographic Mapping

Andrej Gisbrecht[1], Bassam Mokbel[1],
Alexander Hasenfuss[2], and Barbara Hammer[1]

[1] Cognitive Interaction Technology – Center of Excellence,
Bielefeld University, Germany
[2] Computing Centre, TU Clausthal, Germany

Abstract. The generative topographic mapping (GTM) models data by a mixture of Gaussians induced by a low-dimensional lattice of latent points in low dimensional space. Using back-projection, topographic mapping and visualization can be achieved. The original GTM has been proposed for vectorial data only and, thus, cannot directly be used to visualize data given by pairwise dissimilarities only. In this contribution, we consider an extension of GTM to dissimilarity data. The method can be seen as a direct pendant to GTM if the dissimilarity matrix can be embedded in Euclidean space while constituting a model in pseudo-Euclidean space, otherwise. We compare this visualization method to recent alternative visualization tools.

1 Introduction

Dramatically improved technology to gather and store electronic data has lead to an explosion of electronic data sets available in almost all areas of daily life. Because of the size as well as the complexity, humans can no longer inspect these data sets manually and directly extract relevant information. In consequence, data visualization has become a central issue of data mining in this context since it provides efficient data inspection facilities which allow humans to rely on their astonishing cognitive abilities related to visual perception.

Data visualization and dimensionality reduction for direct display of data on the screen constitute well investigated topics of research with manifold different algorithms and tools readily available, see e.g. [10, 12, 13]. A variety of nonlinear dimensionality reduction techniques relying on different principles has been proposed such as locally linear embedding (LLE), Isomap, Laplacian eigenmaps, or t-stochastic neighbor embedding (t-SNE) [13, 14].

With data sets becoming more and more complex, additional functionalities are of increasing interest such as clustering, inference of the data topology, or an explicit mapping. The self-organizing map (SOM) as introduced by Kohonen provides a very popular method which aims at mapping a lattice of neuron onto a given data set in a topology preserving way, this way achieving simultaneous visualization of data and representation in terms of prototypes [11]. Numerous applications ranging from robotics up to bioinformatics prove the wide applicability of the model [11]. SOM itself constitutes a heuristic for which a

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 227–237, 2010.

mathematical analysis is difficult. The generative topographic mapping (GTM) has been proposed as an alternative which relies on an explicit statistical model [1].

Original SOM and GTM have been proposed for vectorial data. With data sets becoming more and more complex, data are often stored in dedicated formats and a standard Euclidean representation is no longer appropriate for an adequate data inspection. Examples include bioinformatics sequences which are compared using alignment techniques, the compression distance to compare the information included in texts, or dedicated structure kernels to compare strings or graph structures. Various extensions of SOM to more general data structures have recently been proposed such as extensions to recursive models [7] or extensions to general dissimilarity data [2, 4, 5, 6, 16, 18]. A dissimilarity representation of data constitutes a quite general interface to deal with complex data formats by means of an appropriate choice of the underlying metric.

In this contribution, we present an extension of GTM to dissimilarity data, relational GTM (RGTM), and we compare the visualization abilities of RGTM to t-SNE, Isomap, and Laplacian eigenmaps as some of the currently best dimensionality reduction methods as well as GTM applied to MDS embedded data [13, 14]. Apart from visualization, GTM provides additional functionalities by means of a prototype representation of data, topographic mapping, and an explicit prescription of the nonlinear embedding map, while leading to only quadratic effort compared to cubic one for an explicit MDS embedding.

2 Generative Topographic Mapping

The GTM as proposed by Bishop et al. provides a generative stochastic model for data $\mathbf{x} \in \mathbb{R}^D$ which is induced by a constrained mixture of Gaussians [1]. The centers are induced by points which are located on a regular low-dimensional lattice in latent space, this way allowing a prototype representation, clustering, and visualization of data by back-projection to the latent space. $\mathbf{w}$ refers to data in latent space. Typically, $p(\mathbf{w})$ is given by a sum of delta-distributions centered at regular grid nodes. Latent points $\mathbf{w}$ are mapped to target vectors $\mathbf{w} \mapsto \mathbf{t} = y(\mathbf{w}, \mathbf{W})$ in the data space by means of a function y parameterized by $\boldsymbol{W}$ such as a generalized linear regression model $\Phi(\mathbf{w}) \cdot \mathbf{W}$ induced by base functions Φ such as equally spaced Gaussians with bandwidth σ. This way, every latent point induces a Gaussian

$$p(\mathbf{x}|\mathbf{w}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left(-\frac{\beta}{2}\|\mathbf{x} - y(\mathbf{w}, \mathbf{W})\|^2\right) \tag{1}$$

with bandwidth β. Correspondingly, the image of a lattice of points according to $p(\mathbf{w})$ generates a mixture model

$$p(\mathbf{x}|\mathbf{W}, \beta) = \sum_{k=1}^{K} 1/K \cdot p(\mathbf{x}|\mathbf{w}_k, \mathbf{W}, \beta) \tag{2}$$

GTM training optimizes the data log-likelihood

$$\mathcal{L} = \ln\left(\prod_{n=1}^{N}\left(\sum_{k=1}^{K} p(\mathbf{w}_k)p(\mathbf{x}_n|\mathbf{w}_k,\mathbf{W},\beta)\right)\right) \tag{3}$$

with respect to $\mathbf{W}$ and β. Using an EM approach, the responsibilities of the generative mixture components $\mathbf{w}_k$ for the data points $\mathbf{x}_n$ are taken as hidden parameter. EM training in turn computes the responsibilities

$$R_{kn}(\mathbf{W},\beta) = p(\mathbf{w}_k|\mathbf{x}_n,\mathbf{W},\beta) = \frac{p(\mathbf{x}_n|\mathbf{w}_k,\mathbf{W},\beta)p(\mathbf{w}_k)}{\sum_{k'} p(\mathbf{x}_n|\mathbf{w}_{k'},\mathbf{W},\beta)p(\mathbf{w}_{k'})} \tag{4}$$

of component k for point number n (the E step derived from (3)) and the model parameters by means of the formulas

$$\mathbf{\Phi}^t\mathbf{G}_{\text{old}}\mathbf{\Phi}\mathbf{W}^t_{\text{new}} = \mathbf{\Phi}^t\mathbf{R}_{\text{old}}\mathbf{X} \tag{5}$$

for $\mathbf{W}$ and

$$\frac{1}{\beta_{\text{new}}} = \frac{1}{ND}\sum_{k,n} R_{kn}(\mathbf{W}_{\text{old}},\beta_{\text{old}})\|\mathbf{\Phi}(\mathbf{w}_k)\mathbf{W}_{\text{new}} - \mathbf{x}_n\|^2 \tag{6}$$

for the bandwith β (the M step derived from (3) by maximizing w.r.t. the parameters $\mathbf{W}$ and β, afterwards.) The subscript old/new refers to the values before/after the M step, respectively. $\mathbf{\Phi}$ refers to the matrix of base functions, $\mathbf{X}$ to the data points, $\mathbf{R}$ to the responsibilities, and $\mathbf{G}$ is a diagonal matrix with accumulated responsibilities $G_{nn} = \sum_n R_{kn}(\mathbf{W},\beta)$.

3 Relational GTM

We assume that data $\mathbf{x}$ are not directly represented as vectors. Rather pairwise dissimilarities d_{ij} are available. Thus, it is no longer possible to directly define target vectors $\mathbf{t}$ since no embedding space is known, nor is it possible to directly define a mixture of Gaussians centered in the targets $\mathbf{t}$.

GTM for Euclidean Dissimilarities

For the moment, we assume that an embedding exists, i.e. $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$, albeit it is not explicitly given. In [6, 9], the following fundamental observation leads to an extension of popular vectorial clustering methods towards data which indirectly given by a dissimilarity matrix. If targets are restricted to linear combinations of data points $\mathbf{t}_k = \sum_{n=1}^{N}\alpha_{kn}\mathbf{x}_n$ where $\sum_{n=1}^{N}\alpha_{kn} = 1$ distances of data points and targets can be computed by means of the formula

$$\|\mathbf{x}_n - \mathbf{t}_k\|^2 = [\mathbf{D}\alpha_k]_n - \frac{1}{2}\cdot\alpha_k^t\mathbf{D}\alpha_k \tag{7}$$

where $\mathbf{D}$ refers to the matrix of pairwise dissimilarities of data points and $[\cdot]_i$ denotes component i of a vector.

This observation has been used in [6] to derive a relational variant of SOM. We use the same principle to generalize GTM to relational data described by a dissimilarity matrix $\mathbf{D}$. The principled idea is to perform original GTM in the (unknown) embedding space only indirectly by referring to targets $\mathbf{t}$ via the coefficient vectors α of their linear combination, and by using (7) to compute the distance of targets and data indirectly based on these coefficients and $\mathbf{D}$. We represent target vectors $\mathbf{t}_k$ indirectly in terms of coefficient vectors α_k of the linear combination. The mapping y maps latent points to coefficient vectors

$$y : \mathbf{w}_k \mapsto \alpha_k = \Phi(\mathbf{w}_k) \cdot \mathbf{W} \tag{8}$$

where Φ refers to base functions such as equally spaced Gaussians with bandwidth σ in the latent space as beforehand. To apply (7), we pose the restriction

$$\sum_n [\Phi(\mathbf{w}_k) \cdot \mathbf{W}]_n = 1 \tag{9}$$

i.e. coefficient vectors sum up to 1. It is possible to compute Gaussians based on the coefficient vectors $\alpha_k = \Phi(\mathbf{w}_k) \cdot \mathbf{W}$ as

$$p(\mathbf{x}_n|\mathbf{w}_k, \mathbf{W}, \beta) = \left(\frac{\beta}{2}\right)^{D/2} \exp\left(-\frac{\beta}{2}\left([\mathbf{D}\alpha_k]_n - \frac{1}{2} \cdot \alpha_k^t \mathbf{D} \alpha_k\right)\right) \tag{10}$$

The data log likelihood can be computed in analogy to (3) based thereof. Note that this quantity is identical to GTM in the unknown Euclidean data space such that a valid probability results thereof.

As before, an EM optimization scheme with hidden variables given by the mode $\mathbf{w}_k$ responsible for data point $\mathbf{x}_n$ can be used. After some algebraic computations, one can see that an EM algorithm in turn computes the responsibilities (4) using (10), and it optimizes

$$\sum_{k,n} R_{kn}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \ln p(\mathbf{x}_n|\mathbf{w}_k, \mathbf{W}_{\text{new}}, \beta_{\text{new}}) \tag{11}$$

with respect to $\mathbf{W}$ and β under the constraint (9). Lagrange optimization with Lagrange multiplier μ_k for component $\mathbf{t}_k$ leads to vanishing Lagrange multipliers, i.e. the constraint (9) is automatically fulfilled for GTM. Hence the model parameters can be determined in analogy to (5,6). We refer to this method, the repeated computation of the responsibilities using (4) and (10) and the determination of the parameters using (5,6) as relational GTM (RGTM).

Like GTM, a back-projection of the data to the grid of latent points in latent space allows a clustering, topographic mapping, and visualization of the data with explicit mapping by means of the winner-takes-all function.

Non-euclidean Data

We would like to point out that, for dissimilarities which stem from Euclidean points $d_{ij} = \|\mathbf{x} - \mathbf{x}_j\|^2$ RGTM is equivalent to GTM. A linear relationship of targets $\mathbf{t}$ and coefficients α holds, which allows to map coefficient vectors to targets if an embedding of datas point is known. Note that RGTM, while leading to the same result, saves computation time: it requires only squared complexity as compared to cubic complexity for an explicit embedding of data.

RGTM can in principle be applied to every given dissimilarity matrix $\mathbf{D}$. An interpretation of this procedure in terms of vectorial data is also possible: For every symmetric dissimilarity matrix with zero diagonal there exists an embedding into so-called pseudo-Euclidean space, i.e. a real vector space with signature $(p, q, N - p - q)$ where $p + q \leq N$ and possibly $q > 0$, corresponding to the symmetric bilinear form $\langle \mathbf{x}, \mathbf{y} \rangle_{p,q} = \sum_{i=1}^{p} x_i y_i - \sum_{i=p+1}^{p+q} x_i y_i$ which induces the given dissimilarity matrix [15]. The data points $\mathbf{X}$ can be embedded in pseudo-Euclidean space by means of the eigenvectors associated to the gram matrix connected to the dissimilarity matrix, [6, 15]. The equality (7) holds for every symmetric bilinear form, hence RGTM can be understood as the standard GTM applied in pseudo-Euclidean space. There are, however, a few problems:

- Distances can become negative in pseudo-Euclidean space, such that Gaussian functions do not yield to a valid probability. This occurs in practice, but does not seem to harm the results. One of the major problems in this context is that a probability model for pseudo-Euclidean spaces has not yet been proposed [15].
- The bandwidth β can become negative, such that numeric problems occur. Although this behavior is possible in theory we never observed it in practice.
- It is not clear that RGTM optimizes the data likelihood and converges. In fact, it can be seen in analogy to [6] that the M step might find a saddle point instead of an optimum, and convergence is not guaranteed. However, we never observed divergence in practice.

In general, it seems that RGTM works well for real-life non-Euclidean data sets.

Initialization

Original GTM uses an initialization of the target vectors by means of the main principal components of the data. This ideas can be transferred to the case of dissimilarity data as follows. We use a MDS projection of the dissimilarities to two dimensional points $\mathbf{A}$. This projection induces the two primary coefficients of the unit vectors in the space of linear combinations in $\mathbb{R}^N$ with coefficient sum 1. The weights $\mathbf{W}$ should be initialized such that the latent grid is mapped to the two-dimensional manifold spanned by these components. Since this hyperplane is going through the origin, $\mathbf{1}/N$ should be added to the linear component of $\mathbf{W}$. Normalizing the matrix such that there are no negative coefficients, we compute $\mathbf{\Phi W} = \mathbf{A}^T / (N \max_{ij}(|[\mathbf{A}^T]_{ij}|))$, and add $\mathbf{1}/N$ afterwards.

4 Experiments

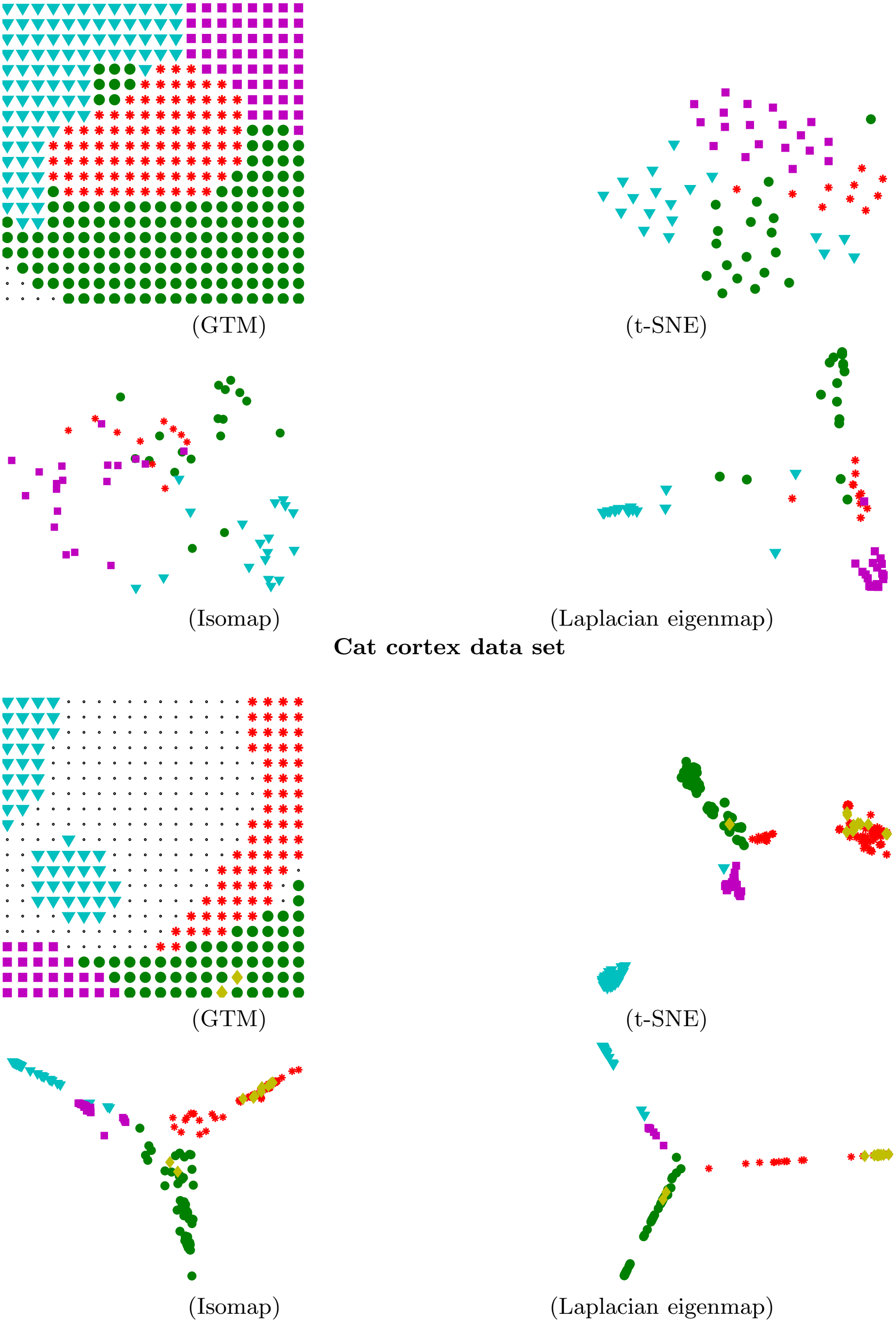
We test the capability of RGTM to visualize dissimilarity data on a couple of benchmark problems as introduced in [3, 6]:

- **Protein data:** The protein data set [3, 8] consists of 226 globin proteins compared based on their evolutionary distance. The samples originate from different protein families, here we distinguish five classes as proposed in [8]. Unlike the other data sets considered here, the protein data set has a highly unbalanced class structure, with class distribution HA (31.86%), HB (31.86%), MY (17.26%), GG/GP (13.27%), and others (5.75%). The signature is $(218, 4, 4)$, i.e. data are almost Euclidean.
- **Aural sonar data:** The aural sonar data set as described in [3] consists of 100 returns from a broadband active sonar system, which are labeled in two classes, target-of-interest versus clutter. The dissimilarity is scored by two independent human subjects each resulting in a dissimilarity score in $\{0, 0.1, \ldots, 1\}$. The signature of the data set is $(54, 45, 1)$.
- **Voting data:** The voting data set describes a two-class classification problem incorporating 435 samples which are given by 16 categorical features with 3 different possible values each. The dissimilarity is determined based on the value difference metric, see [3]. The signature of the data set is $(105, 235, 95)$.
- **Cat cortex:** The cat cortex data originates from anatomic studies of cats' brains. The dissimilarity matrix displays the connection strength between 65 cortical areas [5]. A preprocessed version as presented in [8] was used. The matrix is symmetric with zero diagonal, but the triangle inequality does not hold. The signature of the related pseudo-Euclidean space is $(41, 23, 1)$.

For all data sets, we show the result of RGTM in comparison to Isomap, t-SNE, and Laplacian eigenmaps as some of the most popular nonlinear dimensionality reduction methods. Further, we consider the classical GTM as applied to the data set embedded in Euclidean space. Since data are non-Euclidean, we use two principles to make data Euclidean after pseudo-Euclicean embedding: we either clip negative eigenvalues, or we flip the corresponding eigenvalues.

The parameter setting is as follows: for t-SNE, data are projected to dimensionality $N/4$ by MDS and projected using t-SNE with perplexity 30 (42 for voting), afterwards. For Laplacian eigenmaps and Isomap, the number of neighbors is 12 for all data sets. RGTM uses a regular lattice of 20×20 latent points, and the mapping to the feature space spanned by the coefficients α is based on 2×2 base functions with bandwidth 1. 30 epochs were used for training. In addition, we show the result of a standard Euclidean GTM using the same parameters and data embedded into Euclidean space.

Note that Isomap, t-SNE, and Laplacian eigenmaps provide projections of the points to the two-dimensional plane while GTM allows a visualization of data in latent space at the latent point which corresponds to the closest target. This way, GTM provides a clustering of the data and an explicit mapping of points. This fact also holds for RGTM based on the pairwise dissimilarities of a new point to all given data using the out-of-sample extension as shown in [6]. Latent points for RGTM can be labeled according to the responsibilities for data points of the corresponding label, setting a threshold value, which we pick as 10^{-3} in our experiments. This gives only a hint about the visualization quality, since the labeling might not correspond to the inherent data structure and the goal



(GTM)

(t-SNE)

(Isomap)

(Laplacian eigenmap)

Cat cortex data set

(GTM)

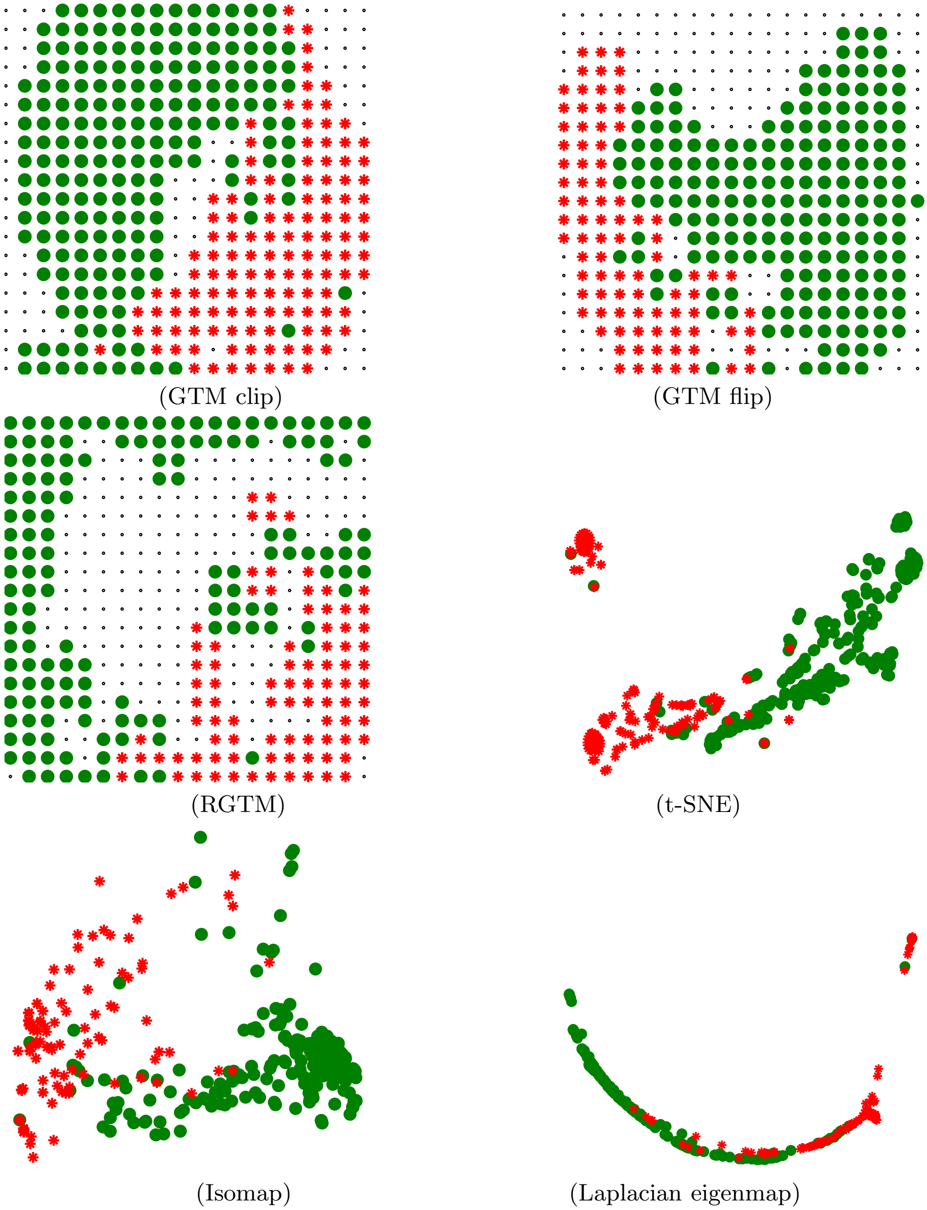
(t-SNE)

(Isomap)

(Laplacian eigenmap)

Protein data set

Fig. 1. Visualization of benchmark dissimilarity data sets



(GTM clip)

(GTM flip)

(RGTM)

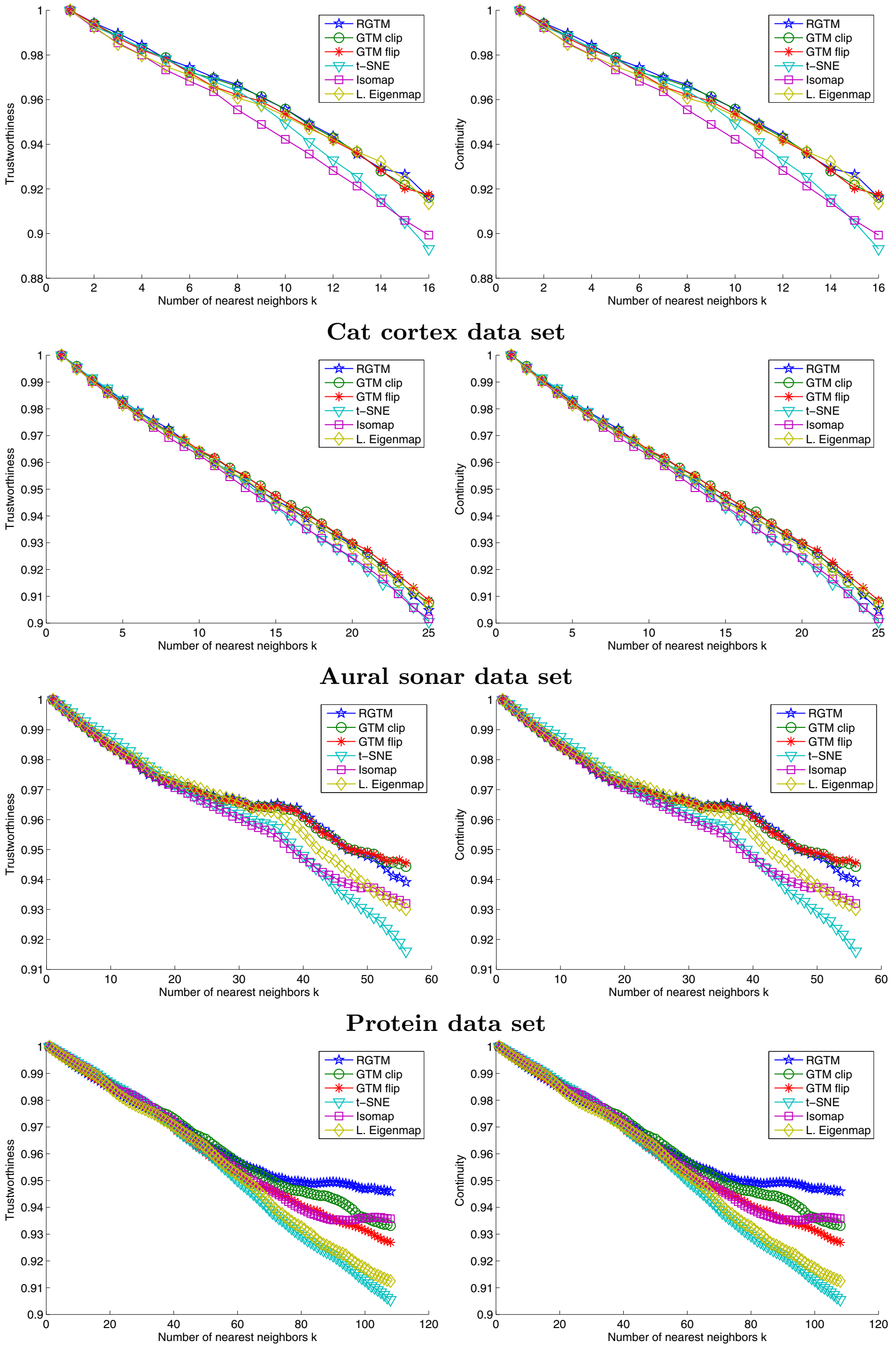
(t-SNE)

(Isomap)

(Laplacian eigenmap)

Voting data set

Fig. 2. Visualization of benchmark dissimilarity data sets



Cat cortex data set

Aural sonar data set

Protein data set

Voting data set

Fig. 3. Trustworthiness and continuity of the data projections

is to visualize data, not to classify the results which could better be achieved by a supervised method such as SVM. Figs. 1,2 show the corresponding results. We formally evaluate the visual impression by computing the trustworthiness and continuity of the mappings as introduced in [17]. For a fixed number of neighbors, this measure quantifies the amount of data points which are not trustworthy since they are displayed too close in the projection space, and the amount of data where discontinuity takes place since data are located too far away in the projection, respectively. The explicit formulas for the trustworthiness and continuity just counts the number of data which are among the k-nearest neighbors in projection space but not in the original space and average and normalize this number (resp. vice versa). Fig. 3 displays trustworthiness and continuity of the projections in dependence of the number of neighbors.

Obviously, RGTM well separates the classes and arranges the data on the latent space such that the visualization space is used as much as possible. Interestingly, RGTM does hardly differ from its Euclidean counterparts after embedding and clip/flip for all data sets but voting, for the latter providing more details and a better trustworthiness/continuity for large k. This can be expected for the proteins data which are almost Euclidean, it also holds for two of the other data sets, whereby RGTM requires only squared complexity compared to cubic complexity for an explicit Euclidean embedding. For all data sets, RGTM shows comparable or superior results compared to alternative visualization methods as also mirrored by the trustworthiness/continuity.

5 Discussion

We have proposed a relational variant of the generative topographic mapping for general dissimilarity datasets and we evaluated its visualization capability in comparison to recent popular nonlinear dimensionality reduction methods. Both, the visual inspection as well as an exact evaluation in terms of trustworthiness and continuity show that RGTM provides a reliable display on a couple of benchmark data, while providing additional functionality such as an explicit projection function, topographic mapping, and clustering, and, compared to standard GTM for embedded data, improved computational complexity.

References

1. Bishop, C.M., Svensen, M., Williams, C.K.I.: GTM: the generative topographic mapping. Neural Computation 10(1), 215–234 (1998)
2. Boulet, R., Jouve, B., Rossi, F., Villa, N.: Batch kernel SOM and related Laplacian methods for social network analysis. Neurocomputing 71(7-9), 1257–1273 (2008)
3. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzani, L.: Similarity-based classification: concepts and algorithms. JMLR 10, 747–776 (2009)
4. Cottrell, M., Hammer, B., Hasenfuss, A., Villmann, T.: Batch and median neural gas. Neural Networks 19, 762–771 (2006)
5. Graepel, T., Obermayer, K.: A stochastic self-organizing map for proximity data. Neural Computation 11, 139–155 (1999)

6. Hammer, B., Hasenfuss, A.: Topographic mapping of large dissimilarity data sets. Neural Computation (to appear)
7. Hammer, B., Micheli, A., Sperduti, A., Strickert, M.: Recursive self-organizing network models. Neural Networks 17(8-9), 1061–1086 (2004)
8. Haasdonk, B., Bahlmann, C.: Learning with distance substitution kernels. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 220–227. Springer, Heidelberg (2004)
9. Hathaway, R.J., Bezdek, J.C.: Nerf c-means: Non-Euclidean relational fuzzy clustering. Pattern Recognition 27(3), 429–437 (1994)
10. Keim, D.A., Schneidewind, J.: Scalable Visual Data Exploration of Large Data Sets via MultiResolution. Journ. of Universal Comp. Sci. 11(11), 1766–1779 (2005)
11. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (1995)
12. Lee, J., Verleysen, M.: Nonlinear dimensionality reduction. Springer, Heidelberg (2007)
13. van der Maaten, L.J.P., Postma, E.O., van den Herik, H.J.: Dimensionality Reduction: A Comparative Review. Tilburg Univ. Tech. Rep., TiCC-TR 2009-005 (2009)
14. van der Maaten, L.J.P., Hinton, G.E.: Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9, 2579–2605 (2008)
15. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition, Foundations and Applications. World Scientific, Singapore (2005)
16. Seo, S., Obermayer, K.: Self-organizing maps and clustering methods for matrix data. Neural Networks 17, 1211–1230 (2004)
17. Venna, J., Kaski, S.: Local multidimensional scaling. Neural Networks 19(6-7), 889–899 (2006)
18. Yin, H.: On the equivalence between kernel self-organising maps and self-organising mixture density network. Neural Networks 19(6), 780–784 (2006)

An Empirical Comparison of Some Multiobjective Graph Search Algorithms[⋆]

Enrique Machuca, Lorenzo Mandow,
Jose L. Pérez de la Cruz, and Amparo Ruiz-Sepulveda

Dpto. Lenguajes y Ciencias de la Computación
Universidad de Málaga 29071 - Málaga, Spain
{machuca,lawrence,perez,amparo}@lcc.uma.es

Abstract. This paper compares empirically the performance in time and space of two multiobjective graph search algorithms, MOA* and NAMOA*. Previous theoretical work has shown that NAMOA* is never worse than MOA*. Now, a statistical analysis is presented on the relative performance of both algorithms in space and time over sets of randomly generated problems.

1 Introduction

The Multiobjective Search Problem is an extension of the Shortest Path Problem where arcs are labeled with vector costs. Each component in a cost vector stands for a different relevant attribute, e.g. distance, time, or monetary cost. Recent works in multicriteria optimization include [1] [2].

This paper deals with two multiobjective counterparts of the A* algorithm, namely MOA* (Multi-Objective A*) [3] and NAMOA* (New Approach to Multi-Objective A*) [4]. Formal developments [5] show that NAMOA* is optimal over the class of admissible multiobjective search algorithms when heuristics are monotone. In other words, no algorithm in this class equipped with the same heuristic information can skip the expansion of a path expanded by NAMOA* without compromising its admissibility.

Little experimental evaluation has been performed to date to compare the actual performance of multiobjective algorithms. Raith & Ehrgott [6] compared several classes of blind multiobjective search algorithms, including label-setting (best-first), label-correcting, and some variants. Brumbaugh & Shier [7] analyzed the performance of several instances of label-correcting algorithms. However, with the exception of the limited analysis reported in [4], no empirical comparison on MOA* and NAMOA* has been performed.

This paper presents a detailed empirical comparison of NAMOA* and MOA*, in absence of heuristic information, on sets of randomly generated bicriterion grids. The experimental setup allows the controlled evaluation of performance with respect to solution depth and correlation between objectives.

The paper is organized as follows: next section summarizes the main differences between MOA* and NAMOA*. Then, the experimental setup is described and relevant

[⋆] This work is partially funded by/Este trabajo está parcialmente financiado por: Consejería de Innovación, Ciencia y Empresa. Junta de Andalucía (España), P07-TIC-03018.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 238–245, 2010.

results are reported. Section 5 discusses the results and characterizes statistically the relative performance of both algorithms. Finally, some conclusions and future work are outlined.

2 Algorithms MOA* and NAMOA*

A detailed description of algorithms MOA* and NAMOA* can be found in [3] and [4] respectively. From a theoretical point of view, NAMOA* has been found to be optimal over the class of admissible multiobjective search algorithms when heuristics are monotone and, particularly, to strictly dominate MOA* [5]. In other words, no algorithm in this class equiped with the same heuristic information can skip the expansion of a path expanded by NAMOA* without compromising its admissibility.

Multiobjective search algorithms differ from their scalar counterparts in several ways. First of all, the use of cost vectors in multiobjective problems induces a partial order relation called dominance, $\forall \boldsymbol{v}, \boldsymbol{v}' \in \mathbb{R}^q, \quad \boldsymbol{v} \prec \boldsymbol{v}' \quad \Leftrightarrow \quad \forall i(1 \leq \ i \leq \ q), \quad v_i \ \leq \ v'_i \ \wedge \ \boldsymbol{v} \neq \boldsymbol{v}'$, where v_i denotes the i-th component of vector $\boldsymbol{v}$. Now, a vector $\boldsymbol{v}$ can not always be ranked as better than another different vector $\boldsymbol{v}'$, e.g. no dominance relation exists between $(2, 3)$ and $(3, 2)$. As a consequence, many different non-dominated paths may reach every node. In particular, multiobjective search algorithms try to find the set of *all* non-dominated solution paths, rather than a single optimal solution.

Both algorithms follow a best-first scheme and use an acyclic search graph SG to record interesting partial solution paths. They apply the principle of optimality to discard unpromising paths, i.e. if the cost of a path to some node n is found to be dominated by the cost other known paths to that node, then the new path can be safely discarded.

The main difference between both algorithms is that MOA* is built around the idea of *node* selection and expansion, while NAMOA* is built around the idea of *path* selection and expansion. MOA* keeps a list of open nodes, and each time a node n is selected for expansion, the extensions of all known non-dominated paths reaching n are simultaneously considered for inclusion in the search graph. On the other hand, NAMOA* keeps a list of open paths. Each time a path is selected for expansion, only its extension is considered for inclusion in the graph.

The selection and expansion policy of MOA* may seem more efficient at first sight, however it has major drawback, since each time a new non-dominated path is found to a closed node, the whole node needs to be put back into the open list.

For each node n in SG, NAMOA* keeps two sets $G_{cl}(n)$ and $G_{op}(n)$, which denote the sets of non-dominated cost vectors of paths reaching n that have or have not been explored yet respectively (i.e. open or closed). On the other hand, MOA* makes no such distinction and a single set $G(n)$ is kept for each node.

It is important to note that in multiobjective search algorithms, the number of nodes considered is not a significant performance measure. Time requirements are dominated by the number of distinct paths considered for expansion (individually in NAMOA*, and collectively in MOA* at each iteration). Space requirements are dominated by the number of cost vectors stored at each node in the graph (in the $G_{op}(n)$ and $G_{cl}(n)$ sets in NAMOA*, and in the $G(m)$ sets in MOA*). The number of nondominated paths in multiobjective problems is known to grow exponentially with solution depth in the worst case [8] even for the two-objective case. In polynomial state spaces with bounded integer costs, this number is known to grow only polynomially with solution depth, although still exponentially with the number of objectives in the worst case [9].

3 Experimental Setup

The experimental setup is designed to evaluate the relative performance of the algorithms as a function of both solution depth, and correlation between costs. For this purpose, square grids of varying size have been randomly generated. A vicinity of four neighbours was used. Only two costs (c^1_{ij}, c^2_{ij}) were considered for each arc from i to j. The values of these objectives were calculated in the range $[1, 10]$ using the formula proposed by Mote et al. [10]. In particular, this formula lets us introduce to the problems a positive association between the arc costs using a correlation multiplier, $0 \leq \rho \leq 1$. The first arc cost c^1_{ij} is randomly generated using a uniform distribution in the range specified. The second arc cost is generated in the same range as $c^2_{ij} = \rho * c^1_{ij} + (1 - \rho) * c^{2*}_{ij}$ where c^{2*}_{ij} is randomly generated using a uniform distribution in the same range. The formula proposed by Mote et al. does not cover the cases where there exists a negative association between arc costs, i.e. $-1 \leq \rho \leq 0$. In these cases, the following formula has been applied, $c^2_{ij} = 1 + (c^2_{max} - (|\rho| * c^1_{ij} + (1 - |\rho|) * c^{2*}_{ij}))$, where c^2_{max} is the maximum value in the range used (10, in this case).

Correlation values considered are $0.8, 0.4, 0, -0.4, -0.8$. Notice that correlation 1 implies that $c^1_{ij} = c^2_{ij}$, i.e. a single objective problem. Decreasing values of correlation yield progressively more difficult problems, where c^1_{ij} differs more and more from c^2_{ij}.

Two different classes of problem instances were used. Let s be the number of nodes in each of the dimensions in the grid. In the first class (I), problems were generated as in [11], i.e. searching from one corner of the grid $(0, 0)$, to the opposite $(s - 1, s - 1)$. Solution depth is $d = 2s - 2$ in this case. In the second class (II), the size was set to $s = 2d + 1$, with the start node at (d, d), and the goal node at $(d/2, d/2)$, i.e. placing the goal node at depth d.

For the first class, a problem set was generated with sizes s varying from 10 to 100 in steps of 10, and 10 problems for each size. For the second class, d varies from 10 to 100 in steps of 10 with 10 problems for each size.

The algorithms were implemented in Lisp using LispWorks Professional 5.01, and run on a HP Proliant D160 G5 server with 2 Intel Xeon QuadCore 4572 @ 3GHz processors and 18 Gb of RAM under Windows Server 2008 (32-bit). The algorithms were implemented to share as much code as possible. The $OPEN$ list was implemented as a binary heap, and lexicographic order was used to break ties in all cases.

4 Results

Let t_{MOA} and t_{NAMOA} be the time taken to solve a problem by MOA* and NAMOA* respectively. Let also v_{MOA} and v_{NAMOA} be the maximum number of cost vectors stored by both algorithms when solving a problem.

For class I problems, no significant difference was found regarding memory requirements, in all correlation values. The hardests problems took about $3,35 \times 10^6$ cost vectors. The relative performance in time t_{MOA}/t_{NAMOA} is presented in figure 1(a) as a function of solution depth d. The hardest problems took about 2303 s (MOA* with $d = 160$) and 1590 s (NAMOA* with $d = 200$). Notice that a time-out of 3600 s was set for the resolution of each problem.

Values are shown for all correlation ratios and averaged over the ten problems generated for each depth, in all reported results in this section.

Figure 1(b) presents the relative space requirements v_{MOA}/v_{NAMOA} against solution depth d, for class II problems. The hardests problems took about $2,99 \times 10^6$ cost vectors (NAMOA*) and $3,84 \times 10^6$ cost vectors (MOA*). Relative time performance t_{MOA}/t_{NAMOA} of the algorithms is presented in figure 1(c) as a function of solution depth d. The hardest problems took about 2531 s (MOA* with $d = 90$) and 1496 s (NAMOA* with $d = 100$). The same time-out of 3600 s was set for the resolution of each problem.

5 Analysis

5.1 Class I Problems

In this class, memory requirements of both algorithms were very similar, in fact indistinguishable in a graphical plot. This is due to the fact that both algorithms need to search virtually all nodes in this class. However, important differences in time requirements can be appreciated in figure 1(a).

In order to characterize the evolution of the relative performance of the algorithms, a statistical regression analysis has been carried out. Let us hypothesize by the observation of figure 1(a), that the relative difference of performance decreases following the rule[1] , $r_p = \alpha * s^\beta$, where $r_p = t_{MOA}/t_{NAMOA}$ is the dependent variable to adjust, denoting the relative performance of the algorithms, and s is the independent variable, representing the size of the problem[2].

Applying logarithms on both sides, the formula would be transformed into $\log r_p = \log \alpha + \beta * \log s$. With this model we can calculate a lineal regression adjustment, assuming by dependent/independent variables the logarithms of the original variables.

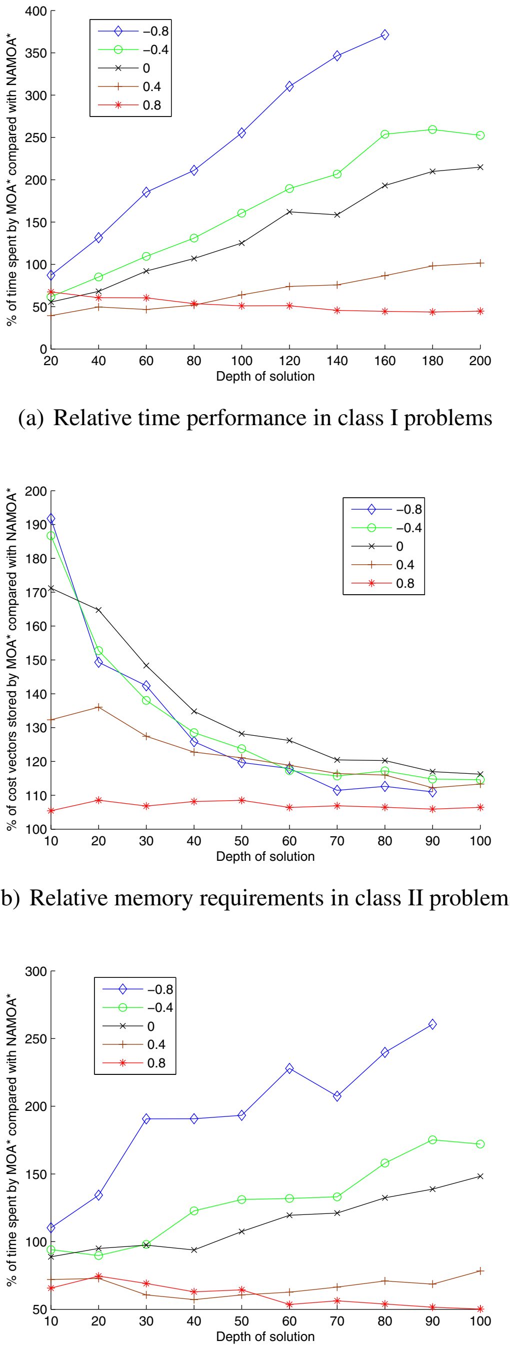
Several indicators can be presented to test the precision of this model. Table 1(a) summarizes some of these indicators for each correlation. The R values represent the gain that can be obtained when predicting the dependent variable from the knowledge of the value from the independent one. Values of R close to 1 indicate the model is a good approximation. The Durbin-Watson (D-W) indicator measures the independence of consecutive errors in the adjustment. Values close to 2 indicate errors are uncorrelated, and therefore, a good adjustment of the model.

Values presented in table 1(a) indicate that the adjustment to the model is good enough to explain the behaviour, specially for correlation 0 and below. The more complex the problem (i.e. negative correlation) the better the adjustment is.

Bad results on correlation 0.4 are related to the change of tendency in the difference in relative performance. MOA* is clearly faster for problems with correlations 0.8 and

[1] Several rules have been tested, but they are not shown due to limitations of space.

[2] Notice that depth of the solution could have been used as independent variable, and the results would be only affected by a constant factor.



(a) Relative time performance in class I problems

(b) Relative memory requirements in class II problems

(c) Relative time performance in class II problems

Fig. 1. Relative performance between MOA* and NAMOA*

Table 1. Estimated parameter values and indicators in the statistical analysis for class I/II problems

(a) Indicators for time, class I

Correlation	α	β	R	R^2	D-W
0.8	0.265	-0.319	0.705	0.497	1.784
0.4	-0.405	0.153	0.352	0.124	0.430
0	-0.868	0.588	0.913	0.833	1.522
-0.4	-0.938	0.678	0.947	0.897	2.199
-0.8	-0.815	0.726	0.973	0.947	2.068

(b) Indicators for memory, class II

Correlation	α	β	R	R^2	D-W
0.8	0.032	-0.001	0.035	0.001	1.948
0.4	0.253	-0.086	0.833	0.694	1.762
0	0.493	-0.188	0.936	0.877	1.963
-0.4	0.518	-0.208	0.930	0.864	1.316
-0.8	0.580	-0.245	0.922	0.849	2.065

(c) Indicators for time, class II

Correlation	α	β	R	R^2	D-W
0.8	0.321	-0.269	0.822	0.676	2.230
0.4	-0.161	-0.007	0.033	0.001	1.114
0	-0.376	0.218	0.692	0.479	1.777
-0.4	-0.509	0.311	0.801	0.642	1.361
-0.8	-0.487	0.394	0.804	0.647	2.108

0.4. However, for problems with lower correlation MOA* is faster only with smaller solution depth. As solution depth increases, NAMOA* is clearly better than MOA*.

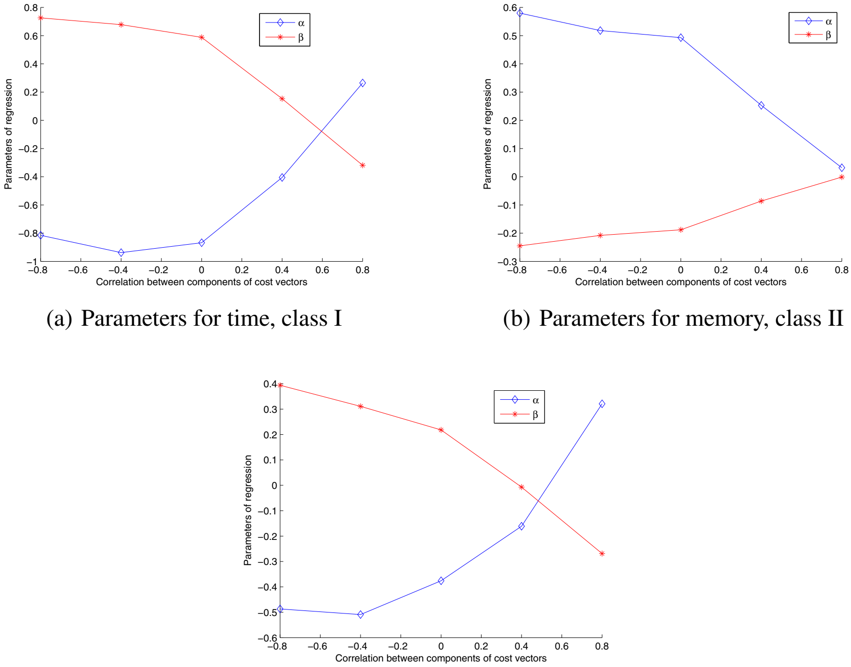
The estimated parameter values α and β are shown[3] in figure 2(a), for different correlations. Notice that horizontal axis present correlation values from lower to higher, i.e. the average number of non-dominated paths to node (difficulty) follow an inverse role. The lower the correlation, the more complex the problem is and more the value of α increases and β decreases. This means that MOA* behaves better for positive correlations, though NAMOA* is better for 0 or negative correlations.

5.2 Class II Problems

For class II problems, a different result can be observed. Figure 1(b) shows that relative space requirements v_{MOA}/v_{NAMOA} are reduced as solution depth d increases. For easier problems MOA* behaves even nearly two times worse. But additional effort tends to only a 15%-25% approximately in the more difficult problems. This is actually smaller than originally expected, given that MOA* can consider for expansion many dominated paths during search.

An analogous statistical analysis has been carried out for this case. The parameters and indicators can be found in table 1(b). The adjustment is good enough to say that MOA* relative difference in space requirements, with respect to NAMOA*, decreases in an inverse rule to that shown before. The value of R for 0.8 can not be representative because differences between algorithms are minimal due to simplicity of problems. The evolution of parameters α and β can be found in figure 2(b). Now, the value of α

[3] Logarithm of α should be applied over all values reported. However, results would only be affected by a constant factor.



(a) Parameters for time, class I (b) Parameters for memory, class II

(c) Parameters for time, class II (d)

Fig. 2. Estimated parameter values in the statistical analysis for class I/II problems and various correlation values

increases with the difficulty of the problem (correlation) and the value of β decreases. This means that the relative space overhead of MOA* approaches a constant ratio for difficult problems (i.e. higher solution depth) in all correlations considered.

In the case of time requirements, the analysis is similar to that found for class I problems. Figure 1(c) shows the same behaviour, though difference is not as big for complex problems. Notice that results presented are from half depth from class I, though the search in class II problems is more complex, because it is not limited by boundaries of grid. The adjustment to the same rule used in section 5.1 is not as good as that observed for class I problems (see table 1(c)). However, it is sufficient to draw the same conclusions as in the corner to corner grids. The same observation can be applied to R from correlation 0.4 as in that case.

6 Conclusions and Future Work

This paper analyses the relative performance of MOA* and NAMOA* in a controlled environment, a polynomial state space, as a function of solution depth and correlation between objectives. The goal is to characterise this performance, and to determine in which cases each algorithm is the best option. Blind search was considered over sets of randomly generated square grids with two objectives and bounded integer cost values.

As expected from previous formal analyses, NAMOA* is never beaten by MOA* in space requirements. In class I problems both algorithms have very similar space requirements, while in class II problems, the overhead of MOA* is found (counterintuitively) to be relatively small, and is well approximated by a polynomial law for uncorrelated objectives.

In problems with high correlation between objectives the selection and expansion scheme of MOA* results in a faster algorithm. In class II problems there is a speed/ memory trade-off between MOA* and NAMOA* for high correlations. In the most difficult problems (uncorrelated objectives and deep solutions) NAMOA* clearly outperforms MOA* in time though there is not as much difference in memory. The time overhead of MOA* for uncorrelated objectives is also well approximated by a polynomial law.

Future work includes the extension of this analysis to other classes of problems and heuristic functions.

References

1. Berger, A., Grimmer, M., Mueller-Hannemann, M.: Fully dynamic speed-up techniques for multi-criteria shortest path searches in time-dependent networks. In: Festa, P. (ed.) Experimental Algorithms. LNCS, vol. 6049, pp. 35–46. Springer, Heidelberg (2010)
2. Delort, C., Spanjaard, O.: Using bound sets in multiobjective optimization: Application to the biobjective binary knapsack problem. In: Festa, P. (ed.) Experimental Algorithms. LNCS, vol. 6049, pp. 253–265. Springer, Heidelberg (2010)
3. Stewart, B.S., White, C.C.: Multiobjective A*. Journal of the ACM 38(4), 775–814 (1991)
4. Mandow, L., Pérez de la Cruz, J.L.: A new approach to multiobjective A* search. In: Proc. of the XIX Int. Joint Conf. on Artificial Intelligence (IJCAI 2005), pp. 218–223 (2005)
5. Mandow, L., Pérez de la Cruz, J.L.: Multiobjective A* search with consistent heuristics. Journal of the ACM 57(5), 27:1–27:25 (2010)
6. Raith, A., Ehrgott, M.: A comparison of solution strategies for biobjective shortest path problems. Computers & Operations Research 36(4), 1299–1331 (2009)
7. Brumbaugh-Smith, J., Shier, D.: An empirical investigation of some bicriterion shortest path problems. European Journal of Operational Research 43, 216–224 (1989)
8. Hansen, P.: Bicriterion path problems. Lecture Notes in Economics and Mathematical Systems, vol. 177, pp. 109–127. Springer, Heidelberg (1979)
9. Mandow, L., Pérez de la Cruz, J.L.: A Memory-Efficient search strategy for multiobjective shortest path problems. In: Mertsching, B., Hund, M., Aziz, Z. (eds.) KI 2009. LNCS (LNAI), vol. 5803, pp. 25–32. Springer, Heidelberg (2009)
10. Mote, J., Murthy, I., Olson, D.L.: A parametric approach to solving bicriterion shortest path problems. European Journal of Operational Research 53(1), 81–92 (1991)
11. Machuca, E., Mandow, L., Pérez de la Cruz, J.L.: An evaluation of heuristic functions for bicriterion shortest path problems. In: Seabra Lopes, L., Lau, N., Mariano, P., Rocha, L. (eds.) New Trends in Artificial Intelligence. Proceedings of the XIV Portuguese Conference on Artificial Intelligence (EPIA 2009). Universidade de Aveiro, Portugal (2009)

Completeness for Generalized First-Order LTL

Norihiro Kamide

Waseda Institute for Advanced Study, Waseda University,
1-6-1 Nishi Waseda, Shinjuku-ku, Tokyo 169-8050, Japan
logician-kamide@aoni.waseda.jp

Abstract. A new first-order fixpoint logic, FL, is introduced as a Gentzen-type sequent calculus. FL is regarded as a generalization of the first-order linear-time temporal logic. The completeness and cut-elimination theorems for FL are proved using some theorems for embedding FL into infinitary logic.

1 Introduction

Fixpoint logics (or *fixed point logics*) are regarded as logics with a *fixpoint* (or *fixed point*) operator. Typical examples of fixpoint logics are *propositional μ-calculus* [7], which is more expressive than temporal logics, and *common knowledge logic* [3], which is an extension of multi-agent epistemic logic. It is known that fixpoint logics are useful for representing temporal and knowledge-based reasoning in Computer Science. A cut-free and complete Gentzen-type sequent calculus for such a fixpoint logic has been required for providing a theoretical basis for automated temporal and knowledge-based theorem proving.

However, a Gentzen-type sequent calculus for a first-order fixpoint logic has not yet been studied. A reason may be that proving the cut-elimination and completeness theorems for such a fixpoint logic is difficult since the traditional formulations of fixpoint operators are rather complex. This paper tries to overcome such a difficulty by introducing a new simple formulation of a fixpoint operator and by using an embedding-based proof method.

We now roughly explain the proposed formulation of the fixpoint operator. The symbol K is used to represent the set $\{\heartsuit_i \mid i \in \omega\}$ of modal operators, and the symbol K^* is used to represent the set of all words of finite length of the alphabet K. Greek lower-case letters ι and κ are used to represent any members of K^*. The characteristic inference rules for a fixpoint operator $\heartsuit_{\mathrm{F}}$ are as follows:

$$\frac{\iota\kappa\alpha, \Gamma \Rightarrow \Delta}{\iota\heartsuit_{\mathrm{F}}\alpha, \Gamma \Rightarrow \Delta}\ (\heartsuit_{\mathrm{F}}\text{left}) \qquad \frac{\{\ \Gamma \Rightarrow \Delta, \iota\kappa\alpha \mid \kappa \in K^*\ \}}{\Gamma \Rightarrow \Delta, \iota\heartsuit_{\mathrm{F}}\alpha}\ (\heartsuit_{\mathrm{F}}\text{right}).$$

These inference rules are intended to imply the axiom scheme $\heartsuit_{\mathrm{F}}\alpha \leftrightarrow \bigwedge\{\iota\alpha \mid \iota \in K^*\}$ where $\bigwedge$ represents infinitary conjunction. Suppose that for any formula α, f_α is a mapping on the set of formulas such that $f_\alpha(x) := \bigwedge\{\heartsuit_i(x \wedge \alpha) \mid i \in \omega\}$. Then, $\heartsuit_{\mathrm{F}}\alpha$ becomes a fixpoint of f_α.

The axiom scheme presented above just corresponds to the so-called iterative interpretation of common knowledge. On the other hand, if we take $K := \{\heartsuit_1\}$,

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 246–254, 2010.

then we can understand $\heartsuit_1$ and $\heartsuit_{\mathrm{F}}$ as the temporal operators X (next-time) and G (any-time), respectively, in *linear-time temporal logic* (LTL) [8]. The corresponding axiom scheme for the singleton case represents the LTL-axiom scheme $\mathrm{G}\alpha \leftrightarrow \bigwedge\{\mathrm{X}^i\alpha \mid i \in \omega\}$ where $\mathrm{X}^i\alpha$ is defined inductively by $\mathrm{X}^0\alpha := \alpha$ and $\mathrm{X}^{i+1}\alpha := \mathrm{XX}^i\alpha$. The fixpoint operator $\heartsuit_{\mathrm{F}}$ is thus regarded as a natural generalization of G.

The contents of this paper are then summarized as follows. In Section 2, a first-order fixpoint logic, FL, is introduced as a Gentzen-type sequent calculus, and the cut-elimination theorem for FL is shown by using a theorem for *syntactically* embedding FL into a sequent calculus LK_ω for first-order infinitary logic. In Section 3, a semantics for FL is introduced, and the completeness theorem with respect to this semantics is proved by using two theorems for *semantically* and *syntactically* embedding FL into LK_ω.

We remark that an embedding-based proof of the cut-elimination theorem for Kawai's sequent calculus LT_ω [6] for LTL was given in [5] by using a theorem for syntactically embedding LT_ω into LK_ω.[1] The proof of the cut-elimination theorem for FL is regarded as a modified extension of the embedding-based cut-elimination proof proposed in [5]. The proof of the completeness theorem for FL (and its special case, LT_ω), which uses both the syntactical and semantical embedding theorems, is a new contribution of the present paper.

There are many papers concerned with LTL, infinitary logic, μ-calculus and common knowledge logic. For more information on these logics, see e.g. [3,5,7] and the references therein. The definitions and terminologies which are used in this paper are mainly based on those of [5]. The terminology "completeness (theorem)" used in this paper implicitly includes the notion of "soundness (theorem)".

Finally in this section, some previous works on cut-free sequent calculi for some common knowledge logics are reviewed below. A *tree sequent calculus* for a first-order common knowledge logic based on the modal logic K with Barcan axiom was introduced by Tanaka, and the cut-elimination theorem for this calculus was proved via the completeness theorem [11]. Some *Tite-style (one-sided) sequent calculi* for propositional common knowledge logic based on some modal logics (esp. K) were introduced by Alberucci and Jäger, and the cut-elimination theorems for these calculi were proved via the completeness theorems [1]. A finite version of a Tait-style system proposed in [1] was studied by Jäger, Kretz and Studer [4]. A syntactic cut-elimination procedure for a calculus proposed in [1] and a *deep sequent calculus* was studied by Brünnler and Studer [2]. The most simpler sequent calculus is the system $\mathrm{K}_n^\omega(C)$ by Alberucci et al. [1,4], but $\mathrm{K}_n^\omega(C)$ is only for a propositional common knowledge logic, i.e., the completeness theorem for a first-order predicate version of $\mathrm{K}_n^\omega(C)$ has not yet been obtained. The proposed system FL is not simpler than $\mathrm{K}_n^\omega(C)$, but FL can straightforwardly be dealt with the first-order case by a simple semantics. The finite system $\mathrm{K}_n^{<\omega}(C)$, which was introduced by Jäger et al. [4], has a finitization property, where the infinitely many premises rule of $\mathrm{K}_n^\omega(C)$ can be replaced by a finite premises rule

[1] The proof of the embedding theorem in [5] has an error. This error is corrected here.

in $\mathrm{K}_n^{<\omega}(C)$. Compared with $\mathrm{K}_n^{<\omega}(C)$, the system FL has not yet been obtained the finitization property as presented in $\mathrm{K}_n^{<\omega}(C)$.

2 Sequent Calculus and Cut-Elimination

Let n be a fixed positive integer. Then, the symbol N is used to represent the set $\{1, 2, ..., n\}$ of indexes of modal operators. The following list of symbols is adopted for the language $\mathcal{L}$ of the underlying logic: free variables $a_0, a_1, ...$, bound variables $x_0, x_1, ...$, functions $f_0, f_1, ...$, predicates $p_0, p_1, ...$, logical connectives $\rightarrow$ (implication), $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\forall$ (any), $\exists$ (exists) and modal operators $\heartsuit_i$ $(i \in N)$, $\heartsuit_{\mathrm{F}}$ (fixpoint) and $\heartsuit_{\mathrm{D}}$ (co-fixpoint). The numbers of free and bound variables are assumed to be countable, and the numbers of functions and predicates are also assumed to be countable. Also assume that there is at least one predicate. A 0-ary function is an individual constant, and a 0-ary predicate is a propositional variable. Small letters $p, q, ...$ are used to denote atomic formulas, Greek lower-case letters $\alpha, \beta, ...$ are used to denote formulas, and Greek capital letters $\Gamma, \Delta, ...$ are used to represent finite (possibly empty) sets of formulas. An expression $\heartsuit\Gamma$ where $\heartsuit \in \{\heartsuit_i \mid i \in N\} \cup \{\heartsuit_{\mathrm{F}}, \heartsuit_{\mathrm{D}}\}$ is used to denote the set $\{\heartsuit\gamma \mid \gamma \in \Gamma\}$. The symbol ω is used to represent the set of natural numbers. The symbol K is used to represent the set $\{\heartsuit_i \mid i \in N\}$, and the symbol K^* is used to represent the set of all words of finite length of the alphabet K. Remark that K^* includes $\emptyset$ and hence $\{\iota\alpha \mid \iota \in K^*\}$ includes α. Greek lower-case letters ι and κ are used to denote any members of K^*. An expression of the form $\Gamma \Rightarrow \Delta$ is called a *sequent*. An expression $L \vdash S$ is used to denote the fact that a sequent S is provable in a sequent calculus L. A rule R of inference is said to be *admissible* in a sequent calculus L if the following condition is satisfied: for any instance

$$\frac{S_1 \cdots S_n}{S}$$

of R, if $L \vdash S_i$ for all i, then $L \vdash S$.

A sequent calculus FL for a first-order fixpoint logic is then introduced below.

Definition 1. *The initial sequents of* FL *are of the form:* $\iota p \Rightarrow \iota p$ *for any atomic formula* p.

The structural rules of FL *are of the form:*

$$\frac{\Gamma \Rightarrow \Delta, \alpha \quad \alpha, \Sigma \Rightarrow \Pi}{\Gamma, \Sigma \Rightarrow \Delta, \Pi}\ (\text{cut}) \qquad \frac{\Gamma \Rightarrow \Delta}{\Pi, \Gamma \Rightarrow \Delta, \Sigma}\ (\text{we}).$$

The logical inference rules of FL *are of the form:*

$$\frac{\Gamma \Rightarrow \Sigma, \iota\alpha \quad \iota\beta, \Delta \Rightarrow \Pi}{\iota(\alpha{\rightarrow}\beta), \Gamma, \Delta \Rightarrow \Sigma, \Pi}\ (\rightarrow\text{left}) \qquad \frac{\iota\alpha, \Gamma \Rightarrow \Delta, \iota\beta}{\Gamma \Rightarrow \Delta, \iota(\alpha{\rightarrow}\beta)}\ (\rightarrow\text{right})$$

$$\frac{\iota\alpha, \Gamma \Rightarrow \Delta}{\iota(\alpha \wedge \beta), \Gamma \Rightarrow \Delta}\ (\wedge\text{left1}) \qquad \frac{\iota\beta, \Gamma \Rightarrow \Delta}{\iota(\alpha \wedge \beta), \Gamma \Rightarrow \Delta}\ (\wedge\text{left2})$$

$$\frac{\Gamma \Rightarrow \Delta, \iota\alpha \quad \Gamma \Rightarrow \Delta, \iota\beta}{\Gamma \Rightarrow \Delta, \iota(\alpha \wedge \beta)}\ (\wedge\text{right}) \qquad \frac{\iota\alpha, \Gamma \Rightarrow \Delta \quad \iota\beta, \Gamma \Rightarrow \Delta}{\iota(\alpha \vee \beta), \Gamma \Rightarrow \Delta}\ (\vee\text{left})$$

$$\frac{\Gamma \Rightarrow \Delta, \iota\alpha}{\Gamma \Rightarrow \Delta, \iota(\alpha \vee \beta)}\ (\vee\text{right1}) \qquad \frac{\Gamma \Rightarrow \Delta, \iota\beta}{\Gamma \Rightarrow \Delta, \iota(\alpha \vee \beta)}\ (\vee\text{right2})$$

$$\frac{\Gamma \Rightarrow \Delta, \iota\alpha}{\iota\neg\alpha, \Gamma \Rightarrow \Delta}\ (\neg\text{left}) \qquad \frac{\iota\alpha, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \iota\neg\alpha}\ (\neg\text{right})$$

$$\frac{\iota\alpha(t), \Gamma \Rightarrow \Delta}{\iota\forall x\alpha(x), \Gamma \Rightarrow \Delta}\ (\forall\text{left}) \qquad \frac{\Gamma \Rightarrow \Delta, \iota\alpha(a)}{\Gamma \Rightarrow \Delta, \iota\forall x\alpha(x)}\ (\forall\text{right})$$

$$\frac{\iota\alpha(a), \Gamma \Rightarrow \Delta}{\iota\exists x\alpha(x), \Gamma \Rightarrow \Delta}\ (\exists\text{left}) \qquad \frac{\Gamma \Rightarrow \Delta, \iota\alpha(t)}{\Gamma \Rightarrow \Delta, \iota\exists x\alpha(x)}\ (\exists\text{right})$$

where a is a free variable which must not occur in the lower sequents of ($\forall$right) *and* ($\exists$left)*, and t is an arbitrary term,*

$$\frac{\iota\kappa\alpha, \Gamma \Rightarrow \Delta \quad (\kappa \in K^*)}{\iota\heartsuit_{\mathrm{F}}\alpha, \Gamma \Rightarrow \Delta}\ (\heartsuit_{\mathrm{F}}\text{left}) \qquad \frac{\{\ \Gamma \Rightarrow \Delta, \iota\kappa\alpha \mid \kappa \in K^*\ \}}{\Gamma \Rightarrow \Delta, \iota\heartsuit_{\mathrm{F}}\alpha}\ (\heartsuit_{\mathrm{F}}\text{right})$$

$$\frac{\{\ \iota\kappa\alpha, \Gamma \Rightarrow \Delta \mid \kappa \in K^*\ \}}{\iota\heartsuit_{\mathrm{D}}\alpha, \Gamma \Rightarrow \Delta}\ (\heartsuit_{\mathrm{D}}\text{left}) \qquad \frac{\Gamma \Rightarrow \Delta, \iota\kappa\alpha \quad (\kappa \in K^*)}{\Gamma \Rightarrow \Delta, \iota\heartsuit_{\mathrm{D}}\alpha}\ (\heartsuit_{\mathrm{D}}\text{right}).$$

Note that ($\heartsuit_{\mathrm{F}}$right) and ($\heartsuit_{\mathrm{D}}$left) have infinite premises. The sequents of the form: $\iota\alpha \Rightarrow \iota\alpha$ for any formula α are provable in cut-free FL. This fact can be proved by induction on the complexity of α. Remark that FL includes Kawai's LT_ω [6] as a special case. Remark also that Gentzen's LK for first-order classical logic is a subsystem of FL.

A language of infinitary logic is obtained from $\mathcal{L}$ by deleting $\{\heartsuit_i \mid i \in N\} \cup \{\wedge, \vee, \heartsuit_{\mathrm{F}}, \heartsuit_{\mathrm{D}}\}$ and adding $\bigwedge$ (infinitary conjunction) and $\bigvee$ (infinitary disjunction). $\bigwedge\{\alpha\}$ and $\bigvee\{\alpha\}$ are equivalent to α. $\wedge$ and $\vee$ are regarded as special cases of $\bigwedge$ and $\bigvee$, respectively.

Definition 2. *Assume that the notion of* term *is defined as usual. Let F_0 be the set of all formulas generated by the standard finitely inductive definition with respect to $\{\rightarrow, \neg, \forall, \exists\}$ from the set of atomic formulas. Suppose that F_t is already defined with respect to $t = 0, 1, 2, ...$ A non-empty countable subset Θ_t of F_t is called an* allowable set *if it contains a finite number of free variables. The expressions $\bigwedge\Theta$ and $\bigvee\Theta$ for an allowable set Θ are considered below. We define F_{t+1} from $F_t \cup \{\bigwedge\Theta, \bigvee\Theta \mid \Theta$ is an allowable set in $F_t\}$ by the standard finitely inductive definition with respect to $\{\rightarrow, \neg, \forall, \exists\}$. F_ω, which is called the set of formulas, is defined by $\bigcup_{t<\omega} F_t$, and an expression in F_ω is called a* formula.

A sequent calculus LK_ω for infinitary logic [9,10] is then presented below.

Definition 3. *The initial sequents of* LK_ω *are of the form: $p \Rightarrow p$ for any atomic formula p.*

The structural rules of LK_ω *are* (cut) *and* (we) *presented in Definition 1.*

The logical inference rules of LK_ω *are of the form:*

$$\frac{\Gamma \Rightarrow \Sigma, \alpha \quad \beta, \Delta \Rightarrow \Pi}{\alpha{\rightarrow}\beta, \Gamma, \Delta \Rightarrow \Sigma, \Pi}\ (\rightarrow\text{left}^{\emptyset}) \qquad \frac{\alpha, \Gamma \Rightarrow \Delta, \beta}{\Gamma \Rightarrow \Delta, \alpha{\rightarrow}\beta}\ (\rightarrow\text{right}^{\emptyset})$$

$$\frac{\Gamma \Rightarrow \Delta, \alpha}{\neg\alpha, \Gamma \Rightarrow \Delta}\ (\neg\text{left}^{\emptyset}) \qquad \frac{\alpha, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg\alpha}\ (\neg\text{right}^{\emptyset})$$

$$\frac{\alpha(t), \Gamma \Rightarrow \Delta}{\forall x\alpha(x), \Gamma \Rightarrow \Delta}\ (\forall\text{left}^{\emptyset}) \qquad \frac{\Gamma \Rightarrow \Delta, \alpha(a)}{\Gamma \Rightarrow \Delta, \forall x\alpha(x)}\ (\forall\text{right}^{\emptyset})$$

$$\frac{\alpha(a), \Gamma \Rightarrow \Delta}{\exists x\alpha(x), \Gamma \Rightarrow \Delta}\ (\exists\text{left}^{\emptyset}) \qquad \frac{\Gamma \Rightarrow \Delta, \alpha(t)}{\Gamma \Rightarrow \Delta, \exists x\alpha(x)}\ (\exists\text{right}^{\emptyset})$$

where a *is a free variable which must not occur in the lower sequents of* $(\forall\text{right}^{\emptyset})$ *and* $(\exists\text{left}^{\emptyset})$*, and* t *is an arbitrary term,*

$$\frac{\alpha, \Gamma \Rightarrow \Delta \quad (\alpha \in \Theta)}{\bigwedge \Theta, \Gamma \Rightarrow \Delta}\ (\textstyle\bigwedge\text{left}) \qquad \frac{\{\ \Gamma \Rightarrow \Delta, \alpha \mid \alpha \in \Theta\ \}}{\Gamma \Rightarrow \Delta, \bigwedge \Theta}\ (\textstyle\bigwedge\text{right})$$

$$\frac{\{\ \alpha, \Gamma \Rightarrow \Delta \mid \alpha \in \Theta\ \}}{\bigvee \Theta, \Gamma \Rightarrow \Delta}\ (\textstyle\bigvee\text{left}) \qquad \frac{\Gamma \Rightarrow \Delta, \alpha \quad (\alpha \in \Theta)}{\Gamma \Rightarrow \Delta, \bigvee \Theta}\ (\textstyle\bigvee\text{right})$$

where Θ *is an allowable set.*

The superscript "$\emptyset$" in the rule names in LK_ω means that these rules are the special cases of the corresponding rules of FL, i.e., the case that ι is $\emptyset$. Remark that the sequents of the form $\alpha \Rightarrow \alpha$ for any formula α are provable in cut-free LK_ω. As well-known, LK_ω enjoys cut-elimination (see e.g., [9,10]).

Definition 4. *Fix a countable non-empty set* Φ *of atomic formulas, and define the sets* $\Phi_\kappa := \{p_\kappa \mid p \in \Phi\}$ $(\kappa \in K^*)$ *of atomic formulas where* $p_\emptyset = p$ *(i.e.,* $\Phi_\emptyset := \Phi$*). The language* $\mathcal{L}^{\mathrm{FL}}$ *(or the set of formulas) of* FL *is defined using* Φ*,* $\rightarrow, \neg, \wedge, \vee, \forall, \exists$*,* $\heartsuit_i$ $(i \in N)$*,* $\heartsuit_{\mathrm{F}}$ *and* $\heartsuit_{\mathrm{D}}$*. The language* $\mathcal{L}^{\mathrm{IL}}$ *of* LK_ω *is defined using* $\bigcup_{\kappa \in K^*} \Phi_\kappa$*,* $\rightarrow, \neg$*,* $\bigwedge, \bigvee, \forall$ *and* $\exists$ *in a similar way as in Definition 2. The binary versions of* $\bigwedge$ *and* $\bigvee$ *are also denoted as* $\wedge$ *and* $\vee$*, respectively, and these binary symbols are assumed to be included in* $\mathcal{L}^{\mathrm{IL}}$*.*

A mapping f *from* $\mathcal{L}^{\mathrm{FL}}$ *to* $\mathcal{L}^{\mathrm{IL}}$ *is defined as follows.*

1. $f(\iota p) := p_\iota \in \Phi_\iota$ $(\iota \in K^*)$ *for any* $p \in \Phi$ *(especially,* $f(p) := p \in \Phi_\emptyset$*),*
2. $f(\iota(\alpha \circ \beta)) := f(\iota\alpha) \circ f(\iota\beta)$ *where* $\circ \in \{\rightarrow, \wedge, \vee\}$*,*
3. $f(\iota\neg\alpha) := \neg f(\iota\alpha)$*,*
4. $f(\iota Qx\alpha(x)) := Qxf(\iota\alpha(x))$ *where* $Q \in \{\forall, \exists\}$*.*
5. $f(\iota\heartsuit_{\mathrm{F}}\alpha) := \bigwedge\{f(\iota\kappa\alpha) \mid \kappa \in K^*\}$*,*
6. $f(\iota\heartsuit_{\mathrm{D}}\alpha) := \bigvee\{f(\iota\kappa\alpha) \mid \kappa \in K^*\}$*.*

An expression $f(\Gamma)$ denotes the result of replacing every occurrence of a formula α in Γ by an occurrence of $f(\alpha)$.

Theorem 5 (Syntactical embedding) *Let* Γ *and* Δ *be sets of formulas in* $\mathcal{L}^{\mathrm{FL}}$*, and* f *be the mapping defined in Definition 4. Then:*

1. *if* FL $\vdash \Gamma \Rightarrow \Delta$, *then* $\mathrm{LK}_\omega \vdash f(\Gamma) \Rightarrow f(\Delta)$.
2. *if* $\mathrm{LK}_\omega -$ (cut) $\vdash f(\Gamma) \Rightarrow f(\Delta)$, *then* FL $-$ (cut) $\vdash \Gamma \Rightarrow \Delta$.

Proof. We can show (1) by induction on the proofs P of $\Gamma \Rightarrow \Delta$ in FL. We can show (2) by induction on the proofs Q of $f(\Gamma) \Rightarrow f(\Delta)$ in $\mathrm{LK}_\omega -$ (cut). In the following, we show only the following cases for (1). The last inference of P is of the form:

$$\frac{\{\ \Gamma \Rightarrow \Delta, \iota\kappa\alpha \mid \kappa \in K^*\ \}}{\Gamma \Rightarrow \Delta, \iota\heartsuit_{\mathrm{F}}\alpha}\ (\heartsuit_{\mathrm{F}}\text{right}).$$

By induction hypothesis, we have $\mathrm{LK}_\omega \vdash f(\Gamma) \Rightarrow f(\Delta), f(\iota\kappa\alpha)$ for all $\kappa \in K^*$. Let Φ be $\{f(\iota\kappa\alpha) \mid \kappa \in K^*\}$. We obtain the required fact:

$$\frac{\begin{matrix}\vdots \\ \{\ f(\Gamma) \Rightarrow f(\Delta), f(\iota\kappa\alpha) \mid f(\iota\kappa\alpha) \in \Phi\ \}\end{matrix}}{f(\Gamma) \Rightarrow f(\Delta), \bigwedge \Phi}\ (\textstyle\bigwedge\text{right})$$

where $\bigwedge \Phi$ coincides with $f(\iota\heartsuit_{\mathrm{F}}\alpha)$ by the definition of f. **Q.E.D.**

Theorem 6 (Cut-elimination) *The rule* (cut) *is admissible in cut-free* FL.

Proof. Suppose FL $\vdash \Gamma \Rightarrow \Delta$. Then, we have $\mathrm{LK}_\omega \vdash f(\Gamma) \Rightarrow f(\Delta)$ by Theorem 5 (1), and hence $\mathrm{LK}_\omega -$ (cut) $\vdash f(\Gamma) \Rightarrow f(\Delta)$ by the cut-elimination theorem for LK_ω. By Theorem 5 (2), we obtain FL $-$ (cut) $\vdash \Gamma \Rightarrow \Delta$. **Q.E.D.**

Remark that by Theorem 6, we can strengthen the statements of Theorem 5 by replacing "if then" with "iff". This fact will be used to prove the completeness theorem for FL.

3 Semantics and Completeness

An expression $\alpha[y/x]$ is used to denote the formula which is obtained from a formula α by replacing all free occurrences of an individual variable x in α by an arbitrary individual variable y, but avoiding the clash of variables. Let Γ be a set $\{\alpha_1, ..., \alpha_m\}$ $(m \geq 0)$ of formulas. Then, Γ^* means $\alpha_1 \vee \cdots \vee \alpha_m$ if $m \geq 1$, and otherwise $\neg(p \rightarrow p)$ where p is a fixed atomic formula. Also Γ_* means $\alpha_1 \wedge \cdots \wedge \alpha_m$ if $m \geq 1$, and otherwise $p \rightarrow p$ where p is a fixed atomic formula.

For the sake of simplicity, a first-order language $\mathcal{L}^{\mathrm{FL}}$ without individual constants and function symbols is adopted for FL. An expression $\hat{\iota}$ is used to express $i_1 i_2 \cdots i_k$ if $\iota = \heartsuit_{i_1}\heartsuit_{i_2}\cdots\heartsuit_{i_k}$ and $\emptyset$ if $\iota = \emptyset$.

Definition 7. *A structure* $A := \langle U, \{I^{\hat{\iota}}\}_{\iota \in K^*}\rangle$ *is called an* FL-model *if the following conditions hold:*

1. *U is a non-empty set,*
2. *$I^{\hat{\iota}}$ $(\iota \in K^*)$ are mappings such that $p^{I^{\hat{\iota}}} \subseteq U^n$ (i.e., $p^{I^{\hat{\iota}}}$ are n-ary relations on U) for each n-ary predicate symbol p.*

We introduce the notation $\underline{u}$ *for the name of* $u \in U$*, and denote* $\mathcal{L}^{\mathrm{FL}}[A]$ *for the language obtained from* $\mathcal{L}^{\mathrm{FL}}$ *by adding the names of all the elements of* U*. A formula* α *is called a* closed formula *if* α *has no free individual variable. A formula of the form* $\forall x_1 \cdots \forall x_m \alpha$ *is called the* universal closure *of* α *if the free variables of* α *are* $x_1, ..., x_m$*. We write* $cl(\alpha)$ *for the universal closure of* α*.*

Definition 8. *Let* $A := \langle U, \{I^{\hat{\iota}}\}_{\iota \in K^*} \rangle$ *be an FL-model. The satisfaction relations* $A \models_{\hat{\iota}} \alpha$ $(\iota \in K^*)$ *for any closed formula* α *of* $\mathcal{L}^{\mathrm{FL}}[A]$ *are defined inductively by:*

1. $A \models_{\hat{\iota}} p(\underline{u}_1, ..., \underline{u}_n)$ *iff* $(u_1, ..., u_n) \in p^{I^{\hat{\iota}}}$ *for each n-ary atomic formula* $p(\underline{u}_1, ..., \underline{u}_n)$,
2. $A \models_{\hat{\iota}} \alpha \wedge \beta$ *iff* $A \models_{\hat{\iota}} \alpha$ *and* $A \models_{\hat{\iota}} \beta$,
3. $A \models_{\hat{\iota}} \alpha \vee \beta$ *iff* $A \models_{\hat{\iota}} \alpha$ *or* $A \models_{\hat{\iota}} \beta$,
4. $A \models_{\hat{\iota}} \alpha \rightarrow \beta$ *iff not-*$(A \models_{\hat{\iota}} \alpha)$ *or* $A \models_{\hat{\iota}} \beta$,
5. $A \models_{\hat{\iota}} \neg\alpha$ *iff not-*$(A \models_{\hat{\iota}} \alpha)$,
6. $A \models_{\hat{\iota}} \forall x \alpha$ *iff* $A \models_{\hat{\iota}} \alpha[\underline{u}/x]$ *for all* $u \in U$,
7. $A \models_{\hat{\iota}} \exists x \alpha$ *iff* $A \models_{\hat{\iota}} \alpha[\underline{u}/x]$ *for some* $u \in U$,
8. *for any* $k \in N$, $A \models_{\hat{\iota}} \heartsuit_k \alpha$ *iff* $A \models_{\hat{\iota} k} \alpha$,
9. $A \models_{\hat{\iota}} \heartsuit_{\mathrm{F}} \alpha$ *iff* $A \models_{\hat{\iota}\hat{\kappa}} \alpha$ *for all* $\kappa \in K^*$,
10. $A \models_{\hat{\iota}} \heartsuit_{\mathrm{D}} \alpha$ *iff* $A \models_{\hat{\iota}\hat{\kappa}} \alpha$ *for some* $\kappa \in K^*$.

The satisfaction relations $A \models_{\hat{\iota}} \alpha$ $(\iota \in K^*)$ *for any formula* α *of* $\mathcal{L}^{\mathrm{FL}}$ *are defined by* $(A \models_{\hat{\iota}} \alpha$ *iff* $A \models_{\hat{\iota}} cl(\alpha))$*. A formula* α *of* $\mathcal{L}^{\mathrm{FL}}$ *is called* FL-valid *if* $A \models_{\emptyset} \alpha$ *holds for each model* A*. A sequent* $\Gamma \Rightarrow \Delta$ *of* $\mathcal{L}^{\mathrm{FL}}$ *is called FL-valid if so is the formula* $\Gamma_* \rightarrow \Delta^*$*.*

Remark that the following clause holds for any satisfaction relation $\models_{\hat{\iota}}$, any formula α and any $\kappa \in K^*$: $A \models_{\hat{\iota}} \kappa\alpha$ iff $A \models_{\hat{\iota}\hat{\kappa}} \alpha$.

In the following, a first-order language $\mathcal{L}^{\mathrm{IL}}$ without individual constants and function symbols is adopted for LK_{ω}. Also, $\mathcal{L}^{\mathrm{IL}}$ is assumed to have uncountably many individual variables. This assumption is known to be necessary to get a completeness theorem for first-order infinitary logic. The assumption is used to rename bound variables in the completeness proof.

Definition 9. *A structure* $B := \langle U, I \rangle$ *is called an* IL-model *if the following conditions hold:*

1. U *is a non-empty set,*
2. I *is a mapping such that* $p^I \subseteq U^n$ *(i.e.,* p^I *is a n-ary relation on* U*) for each n-ary predicate symbol* p*.*

We denote $\mathcal{L}^{\mathrm{IL}}[B]$ *for the language obtained from* $\mathcal{L}^{\mathrm{IL}}$ *by adding the names of all the elements of* U*.*

Definition 10. *Let* Θ *be an allowable set. Let* $B := \langle U, I \rangle$ *be an IL-model. The satisfaction relation* $B \models \alpha$ *for any closed formula* α *of* $\mathcal{L}^{\mathrm{IL}}[B]$ *is defined inductively by:*

1. *$B \models p(\underline{u}_1, ..., \underline{u}_n)$ iff $(u_1, ..., u_n) \in p^I$ for each n-ary atomic formula $p(\underline{u}_1, ..., \underline{u}_n)$,*
2. *$B \models \bigwedge \Theta$ iff $B \models \alpha$ for all $\alpha \in \Theta$,*
3. *$B \models \bigvee \Theta$ iff $B \models \alpha$ for some $\alpha \in \Theta$,*
4. *$B \models \alpha \rightarrow \beta$ iff not-($B \models \alpha$) or $B \models \beta$,*
5. *$B \models \neg\alpha$ iff not-($B \models \alpha$),*
6. *$B \models \forall x\alpha$ iff $B \models \alpha[\underline{u}/x]$ for all $u \in U$,*
7. *$B \models \exists x\alpha$ iff $B \models \alpha[\underline{u}/x]$ for some $u \in U$.*

The satisfaction relation $B \models \alpha$ for any formula α of $\mathcal{L}^{\mathrm{IL}}$ are defined by ($B \models \alpha$ iff $B \models cl(\alpha)$). A formula α of $\mathcal{L}^{\mathrm{IL}}$ is called IL-valid *if $B \models \alpha$ holds for each IL-model B. A sequent $\Gamma \Rightarrow \Delta$ of $\mathcal{L}^{\mathrm{IL}}$ is called IL-valid if so is the formula $\Gamma_* \rightarrow \Delta^*$.*

As well known, the completeness theorem with respect to the IL-model holds for LK_ω: for any sequent S, $\mathrm{LK}_\omega \vdash S$ iff S is IL-valid. See e.g. [9,10].

In the following, we use the same languages $\mathcal{L}^{\mathrm{FL}}$ (for FL) and $\mathcal{L}^{\mathrm{IL}}$ (for LK_ω) as in the previous discussion (i.e., they have no individual constants or function symbols), and also assume that $\mathcal{L}^{\mathrm{FL}}$ has uncountably many individual variables for the sake of compatibility between $\mathcal{L}^{\mathrm{IL}}$ and $\mathcal{L}^{\mathrm{FL}}$. In order to apply the mapping f in Definition 4, we assume the languages $\mathcal{L}^{\mathrm{FL}}$ and $\mathcal{L}^{\mathrm{IL}}$ based on Φ and $\bigcup_{\kappa \in K^*} \Phi_\kappa$, respectively.

Lemma 11. *Let f be the mapping defined in Definition 4. For any FL-model $A = \langle U, \{I^{\hat\iota}\}_{\iota \in K^*}\rangle$, we can construct an IL-model $B = \langle U, I\rangle$ such that for any formula α in $\mathcal{L}^{\mathrm{FL}}$, $A \models_{\hat\iota} \alpha$ iff $B \models f(\iota\alpha)$.*

Proof. Let Φ be a set of atomic formulas and Φ_κ be the set $\{p_\kappa \mid p \in \Phi\}$ of atomic formulas with $p_\emptyset := p$. Let U be given (i.e., U is common in A and B). Suppose that A is an FL-model $\langle U, \{I^{\hat\iota}\}_{\iota \in K^*}\rangle$ such that $I^{\hat\iota}$ ($\iota \in K^*$) are mappings satisfying $p^{I^{\hat\iota}} \subseteq U^n$ for all $p \in \Phi$. Let B be an IL-model $\langle U, I\rangle$ such that I is a mapping satisfying $p^I \subseteq U^n$ for all $p \in \displaystyle\bigcup_{\kappa \in K^*} \Phi_\kappa$ and that $(x_1, x_2, ..., x_n) \in p^{I^{\hat\iota}}$ iff $(x_1, x_2, ..., x_n) \in p_\iota^I$. Then, the claim follows by induction on the complexity of α.

- Base step:

Case ($\alpha \equiv p(\underline{x}_1, ..., \underline{x}_n) \in \Phi$): $A \models_{\hat\iota} p(\underline{x}_1, ..., \underline{x}_n)$ iff $(x_1, x_2, ..., x_n) \in p^{I^{\hat\iota}}$ iff $(x_1, x_2, ..., x_n) \in p_\iota^I$ iff $B \models p_\iota$ iff $B \models f(\iota p(\underline{x}_1, ..., \underline{x}_n))$ (by the definition of f).

- Induction step: We show only the following cases.

Case ($\alpha \equiv \heartsuit_i\beta$): $A \models_{\hat\iota} \heartsuit_i\beta$ iff $A \models_{\hat\iota i} \beta$ iff $B \models f(\iota\heartsuit_i\beta)$ (by induction hypothesis).

Case ($\alpha \equiv \heartsuit_{\mathrm{F}}\beta$): $A \models_{\hat\iota} \heartsuit_{\mathrm{F}}\beta$ iff $A \models_{\hat\iota\hat\kappa} \beta$ for all $\kappa \in K^*$ iff $B \models f(\iota\kappa\beta)$ for all $\kappa \in K^*$ (by induction hypothesis) iff $B \models \bigwedge\{f(\iota\kappa\beta) \mid \kappa \in K^*\}$ iff $B \models f(\iota\heartsuit_{\mathrm{F}}\beta)$ (by the definition of f). **Q.E.D.**

Lemma 12. *Let f be the mapping defined in Definition 4. For any IL-model $B = \langle U, I\rangle$, we can construct an FL-model $A = \langle U, \{I^{\hat\iota}\}_{\iota \in K^*}\rangle$ such that for any formula α in $\mathcal{L}^{\mathrm{FL}}$, $B \models f(\iota\alpha)$ iff $A \models_{\hat\iota} \alpha$.*

Proof. Similar to the proof of Lemma 11. **Q.E.D.**

Theorem 13 (Semantical embedding) *Let f be the mapping defined in Definition 4. For any formula α in $\mathcal{L}^{\mathrm{FL}}$, α is FL-valid iff $f(\alpha)$ is IL-valid.*

Proof. By Lemmas 11 and 12. **Q.E.D.**

Theorem 14 (Completeness) *For any sequent S,* $\mathrm{FL} \vdash S$ *iff S is FL-valid.*

Proof. Let S be $\Gamma \Rightarrow \Delta$ and α be $\Gamma_{*} \rightarrow \Delta^{*}$. $\mathrm{FL} \vdash \alpha$ iff $\mathrm{LK}_{\omega} \vdash f(\alpha)$ (by Theorem 5 and Theorem 6) iff $f(\alpha)$ is IL-valid (by the completeness theorem for LK_{ω}) iff α is FL-valid (by Theorem 13). **Q.E.D.**

Acknowledgments. I would like to thank the KI2010 referees for their valuable comments. This work was partially supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Young Scientists (B) 20700015.

References

1. Alberucci, L., Jäger, G.: About cut-elimination for logics of common knowledge. Annals of Pure and Applied Logic 133(1-3), 73–99 (2005)
2. Brünnler, K., Studer, T.: Syntactic cut-elimination for common knowledge. In: Proceedings of Methods for Modalities 5. Electronic Notes in Computer Science, vol. 231, pp. 227–246 (2009)
3. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about knowledge. MIT Press, Cambridge (1995)
4. Jäger, G., Kretz, M., Studer, T.: Cut-free common knowledge. Journal of Applied Logic 5(4), 681–689 (2007)
5. Kamide, N.: Embedding linear-time temporal logic into infinitary logic: Application to cut-elimination for multi-agent infinitary epistemic linear-time temporal logic. In: Fisher, M., Sadri, F., Thielscher, M. (eds.) CLIMA 2009. LNCS (LNAI), vol. 5405, pp. 57–76. Springer, Heidelberg (2009)
6. Kawai, H.: Sequential calculus for a first order infinitary temporal logic. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 33, 423–432 (1987)
7. Kozen, D.: Results on the propositional mu-calculus. Theoretical Computer Science 27, 333–354 (1983)
8. Pnueli, A.: The temporal logic of programs. In: Proceedings of the 18th IEEE Symposium on Foundations of Computer Science, pp. 46–57 (1977)
9. Takeuti, G.: Proof theory. North-Holland Pub. Co., Amsterdam (1975)
10. Tanaka, Y.: Representations of algebras and Kripke completeness of infinitary and predicate logics, Doctor thesis, Japan Advanced Institute of Science and Technology (1999)
11. Tanaka, Y.: Some proof systems for predicate common knowledge logic. Reports on Mathematical Logic 37, 79–100 (2003)

Instantiating General Games Using Prolog or Dependency Graphs

Peter Kissmann and Stefan Edelkamp

TZI Universität Bremen, Germany
kissmann,edelkamp@tzi.de

Abstract. This paper proposes two ways to instantiate general games specified in the game description language GDL to enhance exploration efficiencies of existing players. One uses Prolog's inference mechanism to find supersets of reachable atoms and moves; the other one utilizes dependency graphs, a datastructure that can calculate the dependencies of the arguments of predicates by evaluating the various formulas from the game's description.

1 Introduction

General game playing (GGP) is concerned with the playing of games whose rules are not known beforehand. Thus, in contrast to classical game playing it is not possible to write a highly specialized game player, such as DEEP BLUE [3] for Chess, or CHINOOK [15] for American Checkers. Instead, the players are supposed to perform well on a much larger variety of games.

Most of the successful players of the last years (e. g., CADIAPLAYER [5] or ARY [14]) use the Monte-Carlo based approach UCT [11] to play general games. So far, they still use the uninstantiated input specified in the game description language GDL [13] and use Prolog to infer knowledge about possible moves and successor states.

Initial experiments with pure Monte-Carlo runs on a number of games indicate that using Prolog on the uninstantiated input usually leads to a significantly worse runtime behavior in the number of expanded nodes per second compared to the performance on instantiated input (see Table 1). Thus, transforming the input to instantiated form, i. e., one without any variables, should be of high priority.

Moreover, in the domain of action planning almost all recent planners (including, e. g., FF [7], GRAPHPLAN [1], and SATPLAN [8]) infer such an instantiated input. The typical input comes in uninstantiated form, so there exists the same problem as in general game playing. Therefore, we expect that the next generation of efficient general game players will avoid unification for binding variables during play via a static analysis prior to the search that yields an instantiated version of the input.

To come up with a superset for all reachable atoms, moves and axioms (as needed for instantiating the problem) we use either a fixpoint computation via unification by implementing an inference interface to the logical programming language Prolog or the knowledge we gain by evaluating the dependency graphs for the several structures in the game's description.

This paper, which provides an extension and improvement to a precursing workshop paper [9], is structured as follows: In Section 2, we give a brief introduction to general

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 255–262, 2010.

Table 1. Number of expanded nodes using pure Monte-Carlo search with Prolog and with instantiated input. The timeout was set to 10 seconds.

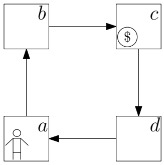
Game	Exp_{Prolog}	$\text{Exp}_{Inst.}$	Factor	Game	Exp_{Prolog}	$\text{Exp}_{Inst.}$	Factor
asteroidsserial	59,364	219,575	3.70	lightsout	28,800	7,230,080	251.04
beatmania	28,680	3,129,300	109.11	pancakes6	154,219	2,092,308	13.57
chomp	22,020	1,526,445	69.32	peg_bugfixed	19,951	1,966,075	98.55
connectfour	44,449	2,020,006	45.45	sheep_and_wolf	20,448	882,738	43.17
hanoi	84,785	7,927,847	93.51	tictactoe	65,864	5,654,553	85.85

game playing and the game description language. In Section 3, we describe the instantiation process. In Section 4, we present our experimental results. Finally, we draw some conclusions in Section 5 and show ideas for further improving the instantiation process.

2 The Game Description Language GDL

In recent years the game description language GDL [13] has become the language of choice for modeling general games. It is a logic-based language, which describes games in an uninstantiated form, i. e., the game description typically incorporates variables.

In the following we will explain the important keywords of GDL and give examples according to the description of the game Maze, which is a single-player game containing four cells with the robot starting in a and a stack of gold lying in c. The goal is to move the gold to a. If this goal is not reached, the game ends after at most nine moves. The cells are connected as shown in the figure to the right.



All the formulas consist of a body (an arbitrary formula, which might be empty) and a head, which is a positive atom.

role: Specifies the names of the players defined in the game.

```
(role robot)
```

init: Specifies the initial state.

```
(init (cell a)) (init (gold c)) (init (step 1))
```

true: Denotes the atoms true in the current state (appears only in bodies of formulas).

state axioms: Help to shorten the description. The operator <= denotes the implication, so that some of the axioms can be considered as derived predicates.

```
(adjacent a b) (adjacent b c) (adjacent c d) (adjacent d a)
(<= timeout (true (step 10)))
```

variables: Variables are denoted with prefix ?.

legal: Describes the moves possible in a specific state for one of the players. For each reachable non-terminal state and each player at least one move has to be possible.

```
(<= (legal robot grab) (true (cell ?x)) (true (gold ?x)))
(<= (legal robot move))
```

does: Specifies the moves chosen by the players (appears only in the formulas' bodies).

next: Determines the successor state, mostly dependent on the moves chosen by all the players. The frame is modeled explicitly in GDL, i. e., any state atom not satisfied by any of the `next` formulas is supposed to be false.

```
(<= (next (cell ?y))
    (does robot move) (true (cell ?x)) (adjacent ?x ?y))
```

terminal: Provides a description of the terminal states. The game ends once one of these states is reached.

```
(<= terminal timeout)
(<= terminal (true (gold a)))
```

distinct: States that the two arguments must differ (appears only in formulas' bodies).

goal: Specifies the rewards (within $[0,\ldots,100]$) for the players for each possible terminal state. Higher rewards are to be prefered.

```
(<= (goal robot 100) (true (gold a)))
(<= (goal robot 0) (true (gold ?x)) (distinct ?x a))
```

The game starts at the initial state. Each player has to check which moves it can perform and select one of these. Once all players have chosen a move, the successor state is calculated and the process starts over. When a terminal state is reached, the game ends and the players get the corresponding rewards.

3 Instantiation Process

There are several cases where it might be important to have the games' descriptions in an instantiated form. One is our general game solver [10], which uses binary decision diagrams (BDDs) [2] to represent the states. To create a BDD, it is important to minimize the number of variables needed for representing any state. This number depends on the total number of reachable state atoms (it might be reduced if groups of mutually exclusive atoms can be found). The complete instantiation process works similar to the ideas in [4,6] (see Algorithm 1). Apart from some more subtle differences, GDL differs from PDDL mainly by modeling the frame explicitly and by splitting the moves into two sets of formulas, namely the `legal` formulas, which act as a move's precondition, and the `next` formulas, which specify the effects, but the moves for which they are relevant are denoted by the `does` terms and thus can appear deep inside the body's formula. Thus, the various steps cannot be performed in the same way as in planning. In the following, we will present the critical steps in more detail.

3.1 Calculating Supersets of Reachable State Atoms, Moves and Axioms

Here, the two approaches (using Prolog and dependency graphs) differ.

Algorithm 1. Instantiation

```
1 Parse the GDL input.
2 Create the disjunctive normal form of the bodies of all formulas.
3 Calculate the supersets of all reachable atoms, moves and axioms (see Section 3.1).
4 Instantiate all formulas (see Section 3.2).
5 Find groups of mutually exclusive atoms (see Section 3.3).
6 Remove the axioms (by applying them in topological order).
7 Generate the instantiated GDL output.
```

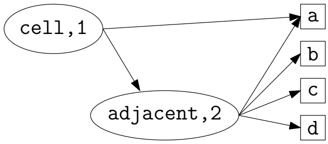
Prolog. We generate a Prolog description of the game but remove all negated atoms (in disjunctive normal form, any negation appears only right in front of an atom).

We repeat the following two steps, until no new moves or state atoms are found. The knowledge base contains all atoms and moves reached so far. Given these, we check for possible moves by querying Prolog for `legal(P, M)`[1]. For each returned player-move pair we add a corresponding move into the knowledge base by using Prolog's assert mechanism (`assert(does(player, move))`). Now, we determine the atoms true in the successor states by querying `next(Atom)` and insert them into the knowledge base as well (`assert(true(atom))`).

When a fixpoint is reached, the knowledge base contains a superset of all reachable state atoms and moves. To also come up with a superset of reachable axioms, we iterate through all the axioms and send queries to the Prolog program (by calling `ax(V1, ..., Vn)` with `ax` representing an axiom with n parameters).

Dependency Graph. We found out that in several games Prolog is rather slow as it finds a large amount of duplicates. In these cases, our second approach using a dependency graph often is superior to the Prolog approach.

We calculate the dependencies of any formula, i. e., for each head we connect its arguments to the arguments of predicates in the body containing the same variables (see the figure to the right displaying a part of the dependencies for the game Maze[2], esp. those for the `next` formula and the axioms in Section 2), similar to [16].



When resolving these dependencies, we come up with possible assignments for each of the parameters of each predicate. Unfortunately, we lose all information about the interaction between the different parameters, e. g., in a chess-game where a rook can move only along rows or columns, we may lose this information, so that a rook's move might be assumed to be possible from any cell to any other cell (depending on the quality of the game's description). To remedy this we perform several post-processing steps. We start by calculating all the possible state atoms, moves and axioms that the dependency graph suggests. Then we evaluate each formula for a given head to check if it might be applicable. This way, we can erase several of the unreachable state atoms, moves and axioms, so that the next stepwill take significantly less time.

[1] For Prolog we denote variables by starting with a capital letter.

[2] Note, that we use `pred,n` to denote the nth argument of the predicate `pred`.

```
(<= (next (cell b))   (<= (next (cell c))
    (does robot move)     (does robot move)
    (true (cell a))       (true (cell b))
    (adjacent a b))       (adjacent b c))

(<= (next (cell d))   (<= (next (cell a))
    (does robot move)     (true (cell d))
    (true (cell c))       (does robot move)
    (adjacent c d))       (adjacent d a))
```

Fig. 1. Instantiated next formula from the Maze example

3.2 Instantiating Formulas

Once the supersets of all reachable state atoms, moves and axioms are found we can instantiate all the given formulas. A naïve idea would be to test each possible instantiation of each predicate of each formula and keep only those that are not conflicting (due to multiple assignments to variables), but this results in a big overhead concerning memory and runtime.

As we already calculated the disjunctive normal form and split those formulas whose base operator is a disjunction, all formulas incorporate only conjunctions. For these, we determine the number of different variables within the formula and create a matrix. The columns of this matrix represent the different variables, the rows their possible assignments. To get those we consider each predicate appearing in the conjunction and find, for each predicate, the set of reachable instantiated atoms (state atoms, moves or axioms), no matter how the other predicates might be instantiated.

To come up with the full instantiations we combine the assignments of the variables. For each predicate of the conjunction we need to check if the chosen instantiation is indeed possible. In case of some variables being distinct we additionally check if they are still distinct after instantiation (if they are not, we forget about this instantiation). When we are done, we have generated a superset of all reachable instantiations of the formulas.

In the dependency graph approach, we found that these supersets still are too large in several cases. Thus, we perform another post-processing step. Right after having calculated all the instantiated formulas, we perform a reachability analysis using these formulas but omit the negations, which is similar to what we did using Prolog with the uninstantiated formulas. In contrast to that approach, here we do not suffer from predicates being satisfied by several formulas and thus a large number of duplicates, so that the overhead in the runtime is small compared to the previous calculations.

The next formula from Section 2 would be represented by the matrix displayed to the right resulting in the instantiated formulas shown in Figure 1.

	?y	?x
(next(cell ?y))	a	
	b	
	c	
	d	
(true(cell ?x))		a
		b
		c
		d
(adjacent ?x ?y)	b	a
	c	b
	d	c
	a	d

3.3 Finding Mutex Groups

In principle, at this point the instantiation is finished. To be prepared for a better evaluation of the games, we next calculate groups of mutually exclusive state atoms. A group of n mutually exclusive atoms can be encoded using $\lceil \log n \rceil$ bits.

To calculate these mutexes we proceed similar to the ideas of [12,16]. For each predicate they try to find its input and output parameters. In most cases the first denote some position on a game board, whereas the latter denote the tokens placed there. For one predicate, all instantiated atoms with the same input parameters but different output parameters are mutually exclusive. To calculate these mutex groups we start by creating several hypotheses for each predicate specifying which of the parameters might be input parameters and check these in a number of states. Finally, we retain that hypothesis that holds in all visited states and produces the smallest state encoding. Using this, for a predicate we group those instantiated atoms together that share the same input parameters. We add an additional atom to each group denoting that there might be states where none of the group's atoms hold. Only for predicates with one parameter we check, if they appear in each visited state. If that is the case, we omit the additional atom.

To find the input and output parameters we write a Prolog program representing the game. With this we perform several simulation steps, where we calculate the `legal` moves for each player and choose one randomly. After each step we check, if the hypotheses hold by evaluating the corresponding predicates true in the current state. For these, we analyze their supposed input parameters by sorting and scanning them. If we find some duplicates, we know for certain that this set of parameters does not form a set of input parameters and discard the corresponding hypothesis. In the worst case, we have to check a number of hypotheses exponential in the number of parameters for each predicate, but often checking the initial state already discards lots of these hypotheses. Also, most games have only predicates with three or four parameters, which would result in at most three or six hypotheses, respectively.

In our experiments we observed no significant slowdown. Also, we found that in many cases already the analysis of the initial state yielded the final hypotheses. Only the games where the game board is not completely specified beforehand, e. g., in ConnectFour or TicTacToe, we need a number of steps to find the final hypotheses. Opposed to [12,16], for us two or three steps might not be enough. Especially in ConnectFour some more steps are required to reliably get correct mutexes. So, we chose to perform 50 steps, but still there is no guarantee that the final hypotheses are correct.

4 Experimental Results

We have implemented the instantiator in C++ using SWI-Prolog[3] and performed some experiments on our machine (one core of an Intel i7 920 with 2.67 GHz and 12 GB RAM). Using a timeout of one minute, which is close to the startup time during a competition, with this implementation we can instantiate 96 of the 171 enabled games from the website of Dresden's GGP server[4] using the Prolog based approach and 90

[3] `http://www.swi-prolog.org`

[4] `http://euklid.inf.tu-dresden.de:8180/ggpserver`

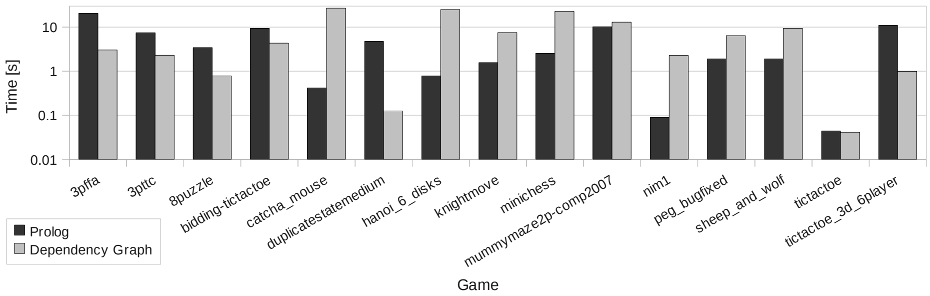


Fig. 2. Results for some of the instantiated games

games using the dependency graphs. Of all the games instantiated, the Prolog based approach instantiated 11 games the dependency graph based one cannot, while the latter can instantiated 4 we cannot when we use Prolog.

A number of the games on the server (20) uses GDL features we do not yet support, so that we cannot instantiate them for now; mostly these are encapsulated formulas. For the remaining games, we often run out of time (typically in the step of finding the supersets, or the one where we instantiate the formulas) or out of memory (either during the instantiation step or the removal of the axioms).

Note that, compared to our workshop paper [9], we used another output. While the GDDL output we previously generated is better for our solver and in most cases also our player, here we output instantiated GDL (or KIF), so that our output can also be processed by all existing players. Also, especially for multi-player games, the generation of GDDL often dominates the runtime. So, we decided to generate the simpler output to better find the differences between the two instantiation approaches.

For a large number of the games the runtimes are similar with both approaches. For some games such as 3pffa, duplicatestate, or tictactoe_3d_6player we found that the Prolog based approach is a lot slower, because the inference mechanism returns too many duplicates, while for others such as catcha_mouse, hanoi, minichess, peg, or sheep_and_wolf the dependency graph based approach initially generates supersets that are a lot larger than the final ones, so that the instantiation of the formulas is more time-consuming and also the post-processing steps take a while. Some results can be found in Figure 2.

5 Conclusion and Future Work

In this paper we presented a way to instantiate general games specified in GDL, where we followed two different approaches for finding the supersets of reachable state atoms, moves and axioms, i. e., Prolog and dependency graphs.

The instantiator has been used to instantiate several games, though there are some for which the process either takes too much time or the memory runs out.

Especially for games containing lots of duplicates (or different ways to satisfy a formula), dependency graphs can outperform Prolog, while Prolog is faster if the supersets found by the dependency graph are too large. Thus, one might try to find out automatically which algorithm to use for instantiation – or to run both instantiations in parallel on two cores. Alternatively, if it is possible, decreasing the sizes of the supersets found by the dependency graph earlier should positively influence that algorithm's runtime.

Another aspect for future work might be the parallelization of the instantiation process. At least for some steps, e. g., the calculation of the disjunctive normal form (step 2) as well as the instantiation of the formulas (step 4), a parallel calculation is possible.

Acknowledgments. Thanks to DFG for support in project ED 74/11-1 and to the anonymous reviewers for their helpful comments.

References

1. Blum, A.L., Furst, M.L.: Fast planning through planning graph analysis. In: IJCAI, pp. 1636–1642 (1995)
2. Bryant, R.E.: Graph-based algorithms for boolean function manipulation. IEEE Transactions on Computers 35(8), 677–691 (1986)
3. Campbell, M., Hoane Jr., A.J., Hsu, F.-H.: Deep Blue. Artificial Intelligence 134(1-2), 57–83 (2002)
4. Edelkamp, S., Helmert, M.: Exhibiting knowledge in planning problems to minimize state encoding length. In: Biundo, S., Fox, M. (eds.) ECP 1999. LNCS, vol. 1809, pp. 135–147. Springer, Heidelberg (2000)
5. Finnsson, H., Björnsson, Y.: Simulation-based approach to general game playing. In: AAAI, pp. 259–264 (2008)
6. Helmert, M.: Understanding Planning Tasks: Domain Complexity and Heuristic Decomposition. LNCS (LNAI), vol. 4929. Springer, Heidelberg (2008)
7. Hoffmann, J., Nebel, B.: The FF planning system: Fast plan generation through heuristic search. JAIR 14, 253–302 (2001)
8. Kautz, H., Selman, B.: Pushing the envelope: Planning, propositional logic and stochastic search. In: AAAI, pp. 1194–1201 (1996)
9. Kissmann, P., Edelkamp, S.: Instantiating general games. In: IJCAI-Workshop on General Game Playing, pp. 43–50 (2009)
10. Kissmann, P., Edelkamp, S.: Layer-abstraction for symbolically solving general two-player games. In: SoCS (2010)
11. Kocsis, L., Szepesvári, C.: Bandit based Monte-Carlo planning. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 282–293. Springer, Heidelberg (2006)
12. Kuhlmann, G., Dresner, K., Stone, P.: Automatic heuristic construction in a complete general game player. In: AAAI, pp. 1457–1462 (2006)
13. Love, N.C., Hinrichs, T.L., Genesereth, M.R.: General game playing: Game description language specification. Technical Report LG-2006-01, Stanford Logic Group (April 2006)
14. Méhat, J., Cazenave, T.: Ary, a general game playing program. In: 13th Board Game Studies Colloquium (2010)
15. Schaeffer, J.: One Jump Ahead: Computer Perfection at Checkers. Springer, Heidelberg (2009)
16. Schiffel, S., Thielscher, M.: Fluxplayer: A successful general game player. In: AAAI, pp. 1191–1196 (2007)

Plan Assessment for Autonomous Manufacturing as Bayesian Inference

Paul Maier, Dominik Jain, Stefan Waldherr, and Martin Sachenbacher

Technische Universität München, Department of Informatics
Boltzmanstraße 3, 85748 Garching, Germany
{maierpa,jain,waldherr,sachenba}@in.tum.de

Abstract. Next-generation autonomous manufacturing plants create individualized products by automatically deriving manufacturing schedules from design specifications. However, because planning and scheduling are computationally hard, they must typically be done offline using a simplified system model, meaning that online observations and potential component faults cannot be considered. This leads to the problem of *plan assessment*: Given behavior models and current observations of the plant's (possibly faulty) behavior, what is the probability of a partially executed manufacturing plan succeeding? In this work, we propose 1) a statistical relational behavior model for a class of manufacturing scenarios and 2) a method to derive statistical bounds on plan success probabilities for each product from confidence intervals based on sampled system behaviors. Experimental results are presented for three hypothetical yet realistic manufacturing scenarios.

1 Introduction

In a scenario of mass customization using autonomous manufacturing, a factory is envisaged that generates, during the night, the manufacturing plans for numerous individualized products to be produced the next day. It employs model-based planning and scheduling capabilities, which use very abstract models to keep planning/scheduling tractable, omitting e.g. behavioral knowledge about potential failures of factory stations. In addition, observations made at execution time are not available at planning/scheduling time. In the light of such partial observations, it may become clear that certain plans will fail, e.g. if a plan operates a component that is now likely to be faulty. This leads to a problem of evaluating manufacturing plans with respect to online observations, based on models focussed on station behavior. It is especially interesting from the point of view of autonomous manufacturing control, where systems are rigid enough to allow automated advance planning/scheduling (rather than online planning), yet bear inherent uncertainties such as station failures.

We call this evaluation *plan assessment* [1]. The idea is to compute, for each product, bounds on the respective success probability. This allows to decide whether to 1) continue with a plan, 2) stop the plan because it probably will not succeed or 3) gather more information. It requires a) models of the complex, uncertain interactions among products and factory stations and b) efficient reasoning. In [1] we proposed using probabilistic automata models and a solution based on constraint optimization,

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 263–271, 2010.

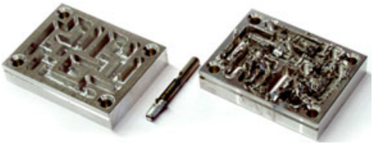


Fig. 1. Effects of cutter deterioration until breakage in machining. Image © Prof. Shea TUM PE.

which enumerates the k most probable system behaviors to estimate success probabilities. However, computing bounds is not yet possible with this approach. In this work we choose a different approach, where we a) model entire *classes of manufacturing systems* as Bayesian Logic Networks (BLNs) [2] and b) use sampling algorithms for efficient computation of *statistical* bounds on success probabilities based on *confidence intervals*. The contribution of this paper is 1) to present a BLN model for a class of manufacturing systems, 2) to propose a method to obtain said bounds from Clopper-Pearson confidence intervals [3] computed during inference and 3) to demonstrate the feasibility of this approach through experimental results.

Closest to our work are verification methods such as probabilistic model checking [4], online verification [5], or probabilistic verification [6]. However, [4] don't regard online observations, and [6] deal only with single most likely behaviors, whereas we have to consider many goal-violating (and achieving) behaviors, and all of them usually focus on models of single systems such as cars [5]. In contrast, we model a manufacturing facility and the products it processes. Other work addressed *automated manufacturability analysis*. They ask whether machining plans violate design tolerances or cost constraints [7,8], evaluating them against static constraints. In contrast, we are interested in dynamic machine behavior (nominal and off-nominal) induced by plan execution.

Assembly and Metal Machining Example. Our factory test-bed – an iCim3000-based Festo Flexible Manufacturing System – consists of conveyor transports, storage, machining and assembly. It serves as the basis for hypothetical example scenarios, where a scheduler schedules the manufacturing of toy mazes (Fig. 1). A maze consists of an alloy base plate, a small metal ball and an acrylic glass cover fixed by metal pins. It is manufactured by first cutting the labyrinth groove into the maze base-plate, then drilling the fixation holes, putting the ball into the labyrinth, putting the glass cover onto the base plate and finally pushing the pins in place to fixate it.[1] While pushing, the assembly station measures the force to prevent applying too much of it. On its route through the factory the product might get flawed as a result of being worked on by faulty stations. Machining stations are suspicious candidates, because their cutter might break during operation. A broken cutter severely damages maze products (see Fig. 1). Since machining stations not only cut grooves but also drill the holes for the pins, broken cutters might also damage these holes. The damage, however, can only be detected later on: If an assembly station tries to push pins into damaged holes, too much force is applied and an alarm is triggered. The same alarm might be triggered if the

[1] In our abstract scenarios, we disregard transportation processes.

assembly station's calibration is off, leading to a misalignment of the gripper holding the pin and the base-plate's hole. This means that one cannot infer from the alarm whether the assembly station or the machining station is at fault. The question now is: Is the alarm an indicator for a broken cutter, and how does this possibility affect the different manufacturing plans?

2 Plan Assessment with Predicted System Behaviors

We address the following *plan assessment problem*: Given a model M_{assess} and observations $o_{0:t}$ obtained up to time point t, compute good lower and upper bounds p_l and p_u on the probability $Pr(\mathcal{G}_i \mid o_{0:t})$ of a manufacturing plan $\mathcal{P}_i$ succeeding: $p_l \leq Pr(\mathcal{G}_i \mid o_{0:t}) \leq p_u$.

We assume a given schedule $\mathcal{S}$ of manufacturing operations, i.e. a sequence of N tuples $\langle(p_{id}, c_{id}, t, a)\rangle_j$, where a tuple specifies that component c_{id} performs action a on product p_{id} at time t. It can be seen as a composition of the individual plans $\mathcal{P}_i$ for each product, which are sequences of actions a. M_{assess} is a probabilistic state space model derived from $\mathcal{S}$, which encodes factory station and product behavior, as well as possible observations. It defines a distribution describing the possible state evolutions for all modeled factory stations and products over time and the influence of observations on this evolution. Variables X^t encode possible states at time t, where $St^t(M_{\text{assess}})$ is the set of all possible (atomic) assignment vectors for X^t, and $St(M_{\text{assess}})$ is the set of all assignment vectors over all N time steps, i.e. the set of all possible *system trajectories*. These trajectories go beyond current time t, thereby *predicting system behavior*. $\mathcal{G}_i$ represents the event that a manufacturing plan $\mathcal{P}_i$ succeeds, i.e. generates a product according to its specification (e.g. a CAD/CAM model or a Bill of Materials (BOM)). Our basic assumption is that $\mathcal{P}_i$ succeeds *as long as no component of the factory fails*. Therefore, we model products as Boolean variables $G_i^{t_{\text{end}}}$ (with True/False for "product ok/flawed"). *Success* of $\mathcal{P}_i$ means that $G_i^{t_{\text{end}}} = \mathsf{True}$ at the future finishing time point t_{end} of the product. This simple modeling could be extended to cover multiple intermediate product states by using richer (finite) domains than $\{\mathsf{True}, \mathsf{False}\}$. We define $\mathcal{G}_i$ as the set of all *goal-achieving* trajectories $\mathcal{G}_i = \{\theta \in St(M_{\text{assess}}) \mid \theta \models G_i^{t_{\text{end}}} = \mathsf{True}\}$. We now define the success probability in terms of goal-achieving system trajectories:

Definition 1. Plan Success Probability. *Given a model M_{assess}, observations $o_{0:t}$ and a manufacturing plan $\mathcal{P}_i$, we define the probability that $\mathcal{P}_i$ will succeed as*

$$Pr(\mathcal{G}_i \mid o_{0:t}) = \sum_{\theta \in \mathcal{G}_i} Pr(\theta \mid o_{0:t})$$

In most cases, it is infeasible to compute $Pr(\mathcal{G}_i \mid o_{0:t})$ exactly as it requires enumerating all trajectories to generate the complete distribution. Approximations can be computed based on a reduced set of trajectories $\Theta^* \subset St(M_{\text{assess}})$. In [1] we introduced an approach that enumerates only the k most probable trajectories. Even better is to compute hard bounds p_l and p_u defined as sum over conditional probabilities $Pr(\theta \mid o_{0:t}) = \frac{Pr(\theta, o_{0:t})}{Pr(o_{0:t})}$ of goal-achieving and goal-violating trajectories (entailing $G_i^{t_{\text{end}}} = \mathsf{False}$) $\theta \in \Theta^*$. Unfortunately, these bounds require to exactly compute

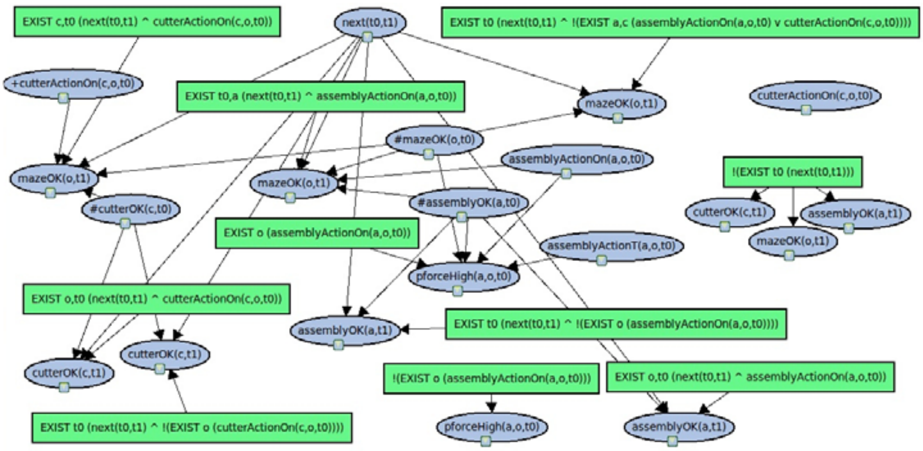

EXIST c,t0 (next(t0,t1) ^ cutterActionOn(c,o,t0))
next(t0,t1)
EXIST t0 (next(t0,t1) ^ !(EXIST a,c (assemblyActionOn(a,o,t0) v cutterActionOn(c,o,t0))))
+cutterActionOn(c,o,t0)
EXIST t0,a (next(t0,t1) ^ assemblyActionOn(a,o,t0))
mazeOK(o,t1)
cutterActionOn(c,o,t0)
#mazeOK(o,t0)
assemblyActionOn(a,o,t0)
mazeOK(o,t1)
mazeOK(o,t1)
!(EXIST t0 (next(t0,t1)))
#cutterOK(c,t0)
#assemblyOK(a,t0)
cutterOK(c,t1)
assemblyOK(a,t1)
mazeOK(o,t1)
EXIST o (assemblyActionOn(a,o,t0))
assemblyActionT(a,o,t0)
pforceHigh(a,o,t0)
EXIST o,t0 (next(t0,t1) ^ cutterActionOn(c,o,t0))
assemblyOK(a,t1)
EXIST t0 (next(t0,t1) ^ !(EXIST o (assemblyActionOn(a,o,t0))))
cutterOK(c,t1)
cutterOK(c,t1)
!(EXIST o (assemblyActionOn(a,o,t0)))
EXIST o,t0 (next(t0,t1) ^ assemblyActionOn(a,o,t0))
EXIST t0 (next(t0,t1) ^ !(EXIST o (cutterActionOn(c,o,t0))))
pforceHigh(a,o,t0)
assemblyOK(a,t1)

Fig. 2. Bayesian logic network that models a class of manufacturing scenarios with machining (called cutter) and assembly stations, and their interaction with maze products

$Pr(o_{0:t})$, which again requires the complete distribution. Therefore, in this work, we use sampling, which allows us to derive *statistical* bounds p_u^* and p_l^* from *confidence intervals* that we compute from the samples. Then we can apply a decision procedure we described in [1], with two modifications: 1) p_l^* is compared against a threshold ω_{success} and p_u^* against a threshold ω_{fail} (which we assume as given) and 2) approximation with k most probable trajectories is replaced by sampling.

3 A Bayesian Logic Network Model of Manufacturing Scenarios

We modeled the behavior of factory machining and assembly stations as well as products as a Bayesian logic network (BLN) [2] $\mathcal{B}_{\text{assess}}$ (see Fig. 2). BLNs combine first-order logic (FOL) with probabilistic modeling, allowing for compact representations of typical manufacturing interactions as well as uncertain events for classes of manufacturing systems, and they are geared towards practical application of many inference algorithms. A BLN is a *template* for the construction of a mixed deterministic/probabilistic network [9], which This means $\mathcal{B}_{\text{assess}}$ models a class of manufacturing systems, capturing general relations between stations and products, from which models M_{assess} are instantiated for concrete factories and schedules. We then either convert M_{assess} to a standard Bayesian net, so we can apply the large body of Bayesian inference techniques, or do inference in mixed networks directly [10].[2] BLNs generalize the well-known formalisms of hidden Markov models (HMM) and dynamic Bayesian networks (DBN).

Key elements of a BLN $\mathcal{B}_{\text{assess}}$ are abstract random variables (ARVs), entity types, fragments and logical formulas. ARVs correspond to logical predicates evaluating to true or false and model states of stations and products as well as relations such as stations working specific products. Placeholders are used within ARVs to refer to abstract

[2] We refer to both Bayesian and mixed nets with M_{assess}

typed entities. Fragments are associated with conditional probability tables. Distributions over instantiations of ARVs are defined through multiple fragments (ellipses in Fig. 2) with mutually exclusive first-order logic preconditions (boxes in Fig. 2).

Our model $\mathcal{B}_{\text{assess}}$ realizes state evolution over time with two abstract entities $t0$ and $t1$, representing successive time points, and ARVs relating to them for successive actions, states, etc. A time line is enforced through ARV $next(t0, t1)$, which encodes that $t0$ precedes $t1$. When instanciating, successiveness of time points $T0, T1, \ldots$ is ensured by clamping $next(T0, T1)$, $next(T1, T2)$, and so on to True. Uncertain station evolution is modeled using two ARVs: $\mathsf{assemblyOK}(a, t0)$, $\mathsf{assemblyOK}(a, t1)$ and $\mathsf{cutterOk}(c, t0)$, $\mathsf{cutterOk}(c, t1)$ for assembly and machining stations. The failure probabilities 0.03 and 0.01 for machining and assembly stations are encoded in the fragments for these ARVs. Product state evolution is modeled in a similar way, i.e. we have the ARVs $\mathsf{mazeOK}(o, t0)$, $\mathsf{mazeOK}(o, t1)$. Their fragments are different in that they currently don't encode any uncertainty. The force alarm observations are encoded as evidence ARVs: $\mathsf{pforceHigh}(a, t)$ encodes that the force measured at the assembly station a at time t was too high (if True). To encode actions, $\mathcal{B}_{\text{assess}}$ consists of relations for assembly and machining stations working mazes at a certain time, i.e. ARVs $\mathsf{assemblyActionOn}(a, o, t)$ and $\mathsf{cutterActionOn}(c, o, t)$. The complex relation that a force alarm can be triggered by cutter-damaged holes as well as a miscalibrated assembly (section 1) can be expressed as a FOL formula: $\mathsf{assemblyActionOn}(a, o, t) \Rightarrow (\mathsf{pforceHigh}(a, t) \Leftrightarrow (\neg\mathsf{assemblyOK}(a, t) \vee \neg\mathsf{mazeOK}(o, t)))$[3].

To perform plan assessment for a specific scenario, $\mathcal{B}_{\text{assess}}$ is instantiated to a concrete model M_{assess}. ARVs are thereby compiled to a set of random variables (e.g. $\mathsf{mazeOK}(\mathsf{Maze0}, \mathsf{T0})$), fragments to conditional probability distributions over them, and logical formulas to propositional logical constraints. In particular, instances of ARVs representing product states when the product should be finished encode product success, i.e. the goal variables $G_i^{t_{\text{end}}}$. In our scenarios we use goals $\mathsf{mazeOK}(\mathsf{Maze0}, \mathsf{T5})$, $\mathsf{mazeOK}(\mathsf{Maze1}, \mathsf{T9})$, $\mathsf{mazeOK}(\mathsf{Maze2}, \mathsf{T7})$ and $\mathsf{mazeOK}(\mathsf{Maze3}, \mathsf{T12})$. A given schedule $\mathcal{S}$ determines the set E' of concrete instances of machining and assembly stations as well as maze products (e.g. $\mathsf{Mach0}$, $\mathsf{Maze0}$, $\mathsf{Assy0}$). *Evidence variables* (instantiated from evidence ARVs) are clamped to values representing the actual observations, e.g. $\mathsf{pforceHigh}(\mathsf{Assy0}, \mathsf{T4}) = \mathsf{True}$.

4 Computing Confidence Intervals for Plan Success Probabilities

Now we can use M_{assess} to assess manufacturing plans for individual products. Since we usually cannot have an exact $Pr(\mathcal{G}_i \mid o_{0:t})$, nor hard bounds (Section 2), we propose to compute the confidence interval of $Pr(\mathcal{G}_i \mid o_{0:t})$: "soft" bounds p_u^* and p_l^* according to a predefined probability γ, the *coverage probability*, that $Pr(\mathcal{G}_i \mid o_{0:t})$ will be within these bounds.

Theorem 1. *The bounds p_u^* and p_l^* on $Pr(\mathcal{G}_i \mid o_{0:t})$ are given by the Clopper-Pearson interval [3].*

[3] For technical reasons we modeled it using deterministic fragments (i.e. with prob. values 1.0 and 0.0), the FOL formula however is the more elegant equivalent.

Proof. Let G be a Bernoulli-distributed Boolean random variable (BBRV) with parameter p, which is being sampled in a Bernoulli-process, counting appearances of $G = \text{True}$ and $G = \text{False}$ as a and b, respectively. Let γ be the coverage probability. Then the Clopper-Pearson interval defines bounds $p_l^* = F^{-1}_{a,b,\gamma}(1 - \frac{\alpha}{2})$ and $p_u^* = F^{-1}_{a,b,\gamma}(\frac{\alpha}{2})$, where $\alpha = 1 - \gamma$ and $F^{-1}_{a,b,\gamma} = I^{-1}_{a+1,b+1}$, I being the regularized incomplete beta function; I^{-1} is thus the inverse of the cumulative distribution function (CDF) of the beta distribution. Any manufacturing goal $G_i^{t_{\text{end}}}$ is a BBRV with parameter $p = Pr(\mathcal{G}_i \mid o_{0:t})$. We sample trajectories that correspond to the observations and entail assignments to $G_i^{t_{\text{end}}}$. Therefore, the trajectory sampling can be seen as a sampling of $G_i^{t_{\text{end}}}$. The sampling yields n samples, $n_{G_i^{t_{\text{end}}}}$ goal-achieving and $n - n_{G_i^{t_{\text{end}}}}$ goal-violating. If we now set $G = G_i^{t_{\text{end}}}$, $a = n_{G_i^{t_{\text{end}}}}$, $b = n - n_{G_i^{t_{\text{end}}}}$, the theorem follows.

We quickly recap why this works. We abbreviate $G_i^{t_{\text{end}}}$ to G. The quantities n_G, $n - n_G$ determine a *distribution over* p, which (assuming a uniform distribution over parameters when there are no samples) is given by the beta distribution [11]. The CDF $F_{n_G,n-n_G}(x) = Pr(p \leq x)$ allows to compute the probability that p is at most x. Observe now that the *complement* of the given $\gamma = Pr(p_l^* \leq p \leq p_u^*)$, i.e. the probability that p is *outside* the bounds of the interval, can be written as $Pr(p < p_l^*) + Pr(p > p_u^*) = 1 - \gamma = \alpha$. We can rewrite this equation as $Pr(p \leq p_l^*) + 1 - Pr(p \leq p_u^*) = \alpha$, where all probabilities are represented through the CDF $F_{n_G,n-n_G}(x)$. Now we can use the inverse CDF $F^{-1}_{n_G,n-n_G}(y)$ to compute the bounds p_l^* and p_u^*. Except in extreme cases, we can assume that α is to equal parts composed of $Pr(p \leq p_l^*)$ and 1 - $Pr(p \leq p_u^*)$, i.e. $Pr(p \leq p_l^*) = \frac{\alpha}{2} = 1 - Pr(p \leq p_u^*)$. Resolving for p_l^* yields $p_l^* = F^{-1}_{n_G,n-n_G}(\frac{\alpha}{2})$ and for p_u^* gives $p_u^* = F^{-1}_{n_G,n-n_G}(1 - \frac{\alpha}{2})$.

Of course we would like to have as narrow intervals as possible. Increasing the number of samples gives us narrower intervals. Thus, a practical stop criterion for sampling algorithms is to predefine the size of the interval to be sufficiently small, and then sample until the interval is narrower than this predefined size. Note that an estimate for the success probability itself can be computed in the usual way by dividing the number of goal-achieving samples by the number of all samples, $Pr(\mathcal{G}_i \mid o_{0:t}) \approx \frac{n_G}{n}$.

5 Experimental Results

We inferred the success probability of mazes for three different (hypothetical) scenarios, with corresponding ground models instantiated from $\mathcal{B}_{\text{assess}}$, using three sampling algorithms [12,13,10]. We ran Java implementations of the algorithms on an Intel Core2 Duo with 2.53 Ghz and 4GB of RAM. In all scenarios there were two machining stations (Mach0/1) and one assembly station (Assy0). In the smaller *scenario 1* (310 nodes) Mach0 has become faulty. Its schedule ranges over nine time points for three mazes (Maze0/1/2). Observations have been made up to T4, and a force alarm was triggered at T4 while the assembly station was pushing pins into Maze0. In *scenario 2* (520 nodes) Assy0 is faulty. Here, four mazes (Maze0/1/2/3) are scheduled, covering 12 time points. Observations are available up to T8. The pin assembly is done at T4, T6 and T8 for Maze0, Maze2 and Maze1 respectively. A force alarm is observed at all three time

Table 1. Confidence intervals on success probabilities for the mazes in scenario 1 obtained with likelihood weighting and exact results obtained through variable elimination

coverage rate γ		Number of samples			Exact
		100	2500	10000	
0.95	mazeOK(M2,T7)	$[0.002, 0.054]$	$[0.035, 0.051]$	$[0.035, 0.042]$	0.0387
	mazeOK(M1,T9)	$[0.593, 0.772]$	$[0.591, 0.629]$	$[0.587, 0.606]$	0.6041
	mazeOK(M0,T5)	$[0.000, 0.029]$	$[0.000, 0.001]$	$[0.000, 0.000]$	0.0000
0.999	mazeOK(M2,T7)	$[0.010, 0.162]$	$[0.030, 0.057]$	$[0.035, 0.048]$	0.0387
	mazeOK(M1,T9)	$[0.359, 0.677]$	$[0.573, 0.637]$	$[0.594, 0.626]$	0.6041
	mazeOK(M0,T5)	$[0.000, 0.066]$	$[0.000, 0.003]$	$[0.000, 0.001]$	0.0000
	runtime	0.06	1.61	6.24	58.76
		0.06	1.52	6.00	58.76

points. *Scenario 3* (520 nodes) is similar to the former, with the difference that again Mach0 is faulty. Consequently, at T8 no force alarm is triggered. In all scenarios Maze0 and Maze2 are cut on Mach0, while Maze1 and (in scenarios 2 and 3) Maze3 are cut on Mach1. Further, Maze0/1/2/3 should be finished by T5/9/7/12, respectively. Note that results for different products in the same scenario are obtained simultaneously.

In all scenarios we can infer meaningful bounds on the products' success probabilities and thereby identify jeopardized products. In scenario 2 (Maze0:$[0.000, 0.003]$, Maze1: $[0.000, 0.003]$, Maze2: $[0.000, 0.003]$, Maze3: $[0.002, 0.011]$), the observations strongly indicate a faulty assembly (which is a lot more likely than both Mach0 and Mach1 failing simultaneously), which means that the unfinished Maze3 will certainly fail, too. In scenario 3, in contrast, (Maze0: $[0.000, 0.000]$, Maze1: $[1.000, 1.000]$, Maze2: $[0.000, 0.000]$, Maze3: $[0.908, 0.918]$) Maze3 is very likely to succeed since, given the observations, it is highly likely that Mach0 is faulty. Scenario 1 is less conclusive (see table 1): While Maze2 is clearly certain to fail, there's an uncertain chance that Maze1 will be ok. So all in all the result could lead to these decisions: In scenario 2, stop all manufacturing and in scenario 3, continue to finish Maze3. In uncertain cases such as scenario 1 methods to actively gather information could be triggered [14].

Table 1 illustrates how choosing stricter coverage probabilities γ widens the interval. It also confirms that increasing the number of samples results in better intervals. Good intervals ($\gamma = 0.95$, width less than 0.01) can already be retrieved in under a minute, sometimes even in under a second (scenario 2, SampleSearch), with less than 1000 samples. Choosing a good algorithm seems to depend on the scenarios (see Table 2): for 1 and 2 SampleSearch is best, while for 3, likelihood weighting is the better choice. Notably, the runtime does not strictly depend on the problem size, i.e. no single algorithm can be trusted to be equally quick for all problems. A solution could be to run algorithms simultaneously (taking advantage of current multi-core technology) up to a time limit and then take the result with the narrowest interval, or stop when a predefined interval width is reached.

Note that simplified inference methods such as forward filtering (which one might use for HMMs) are not applicable to the type of problem we considered, because the interactions that we modelled result in several coupled temporal chains, which, in

Table 2. Comparing algorithms on scenarios 1 / 2 / 3 by average number of samples (above) needed to reach a target confidence interval width, and runtime in seconds (below)

	Algorithm								
$\max_I \lvert I\rvert$	likelihood weighting			backward simulation			SampleSearch		
≤ 0.025	5900	320	2020	5820	200	-[4]	6180	200	1380
	3.61	0.90	3.82	1.23	5.44	-	0.84	0.06	13.27
≤ 0.01	36800	1000	11800	37280	300	-	38440	660	5080
	22.42	2.67	21.47	7.00	7.56	-	4.76	0.16	42.08
≤ 0.0025	588020	17420	185820	588860	1200	-	614700	6460	181320
	384.38	50.06	362.08	111.95	31.75	-	79.70	1.40	1551.25

particular, do not have the Markov property. Inference is, therefore, considerably more difficult and presents a challenge as problem sizes increase. Advances in lifted inference [15,16] might soon alleviate this problem.

6 Conclusion and Future Work

We presented a model-based method that samples behavior-trajectories of stations and products in a manufacturing scenario to compute *confidence intervals* for success probabilities of the products, thereby allowing an autonomous manufacturing system to react to failures and other unforeseen events. The method uses a Bayesian logic network (BLN) model of a class of manufacturing systems. Results show that, for multiple scenarios generated from the same abstract BLN, we can indeed identify jeopardized products based on the computed bounds. Future work will concern a direct comparison with our previous work [1] and more experiments in order to assess the method's scalability limits. We are also interested in exploiting the obtained results in order to update the underlying model, for instance, to automatically adapt to parameter drifts of components, or to learn, e.g., failure probabilities in the first place.

References

1. Maier, P., Sachenbacher, M., Rühr, T., Kuhn, L.: Constraint-Based Integration of Plan Tracking and Prognosis for Autonomous Production. In: Mertsching, B., Hund, M., Aziz, Z. (eds.) KI 2009. LNCS, vol. 5803, pp. 403–410. Springer, Heidelberg (2009)
2. Jain, D., Waldherr, S., Beetz, M.: Bayesian Logic Networks. Technical report, Technische Universität München (2009)
3. Clopper, C., Pearson, E.: The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika 26, 404 (1934)
4. Rutten, J., Kwiatkowska, M., Norman, G., Parker, D.: Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems. CRM Monograph Series, vol. 23. American Mathematical Society, Providence (2004)

[4] Backward simulation did not produce any results for scenario 3, because the problem was too ill-conditioned, such that no countable samples could be generated. SampleSearch does not have this problem and will always generate usable samples (given enough time).

5. Althoff, M., Stursberg, O., Buss, M.: Online Verification of Cognitive Car Decisions. In: Proc. IV 2007, pp. 728–733 (2007)
6. Mahtab, T., Sullivan, G., Williams, B.C.: Automated Verification of Model-Based Programs Under Uncertainty. In: Proc. ISDA 2004 (2004)
7. Nau, D.S., Gupta, S.K., Regli, W.C.: Manufacturing-Operation Planning versus AI Planning. In: Integrated Planning Applications: Papers from the 1995 AAAI Spring Symposium, pp. 92–101. AAAI Press, Menlo Park (1995)
8. Kiritsis, D., Neuendorf, K.P., Xirouchakis, P.: Petri Net Techniques for Process Planning Cost Estimation. Advances in Engineering Software 30, 375–387 (1999)
9. Mateescu, R., Dechter, R.: Mixed Deterministic and Probabilistic Networks. Annals of Mathematics and Artificial Intelligence 54, 3–51 (2008)
10. Gogate, V., Dechter, R.: SampleSearch: A Scheme that Searches for Consistent Samples. In: Proc. AISTATS 2007 (2007)
11. Bishop, C., et al.: 2. Probability Distributions. In: Pattern Recognition and Machine Learning, pp. 67–74. Springer, Heidelberg (2006)
12. Fung, R.M., Chang, K.C.: Weighting and integrating evidence for stochastic simulation in bayesian networks. In: Proc. UAI 1989, pp. 209–220. North-Holland Publishing, Amsterdam (1989)
13. Fung, R., Del Favero, B.: Backward Simulation in Bayesian Networks. In: Proc. UAI 1994, p. 227. Morgan Kaufmann, San Francisco (1994)
14. Kuhn, L., Price, B., de Kleer, J., Do, M.B., Zhou, R.: Pervasive Diagnosis: The Integration of Diagnostic Goals into Production Plans. In: Fox, D., Gomes, C.P. (eds.) Proc. AAAI 2008, pp. 1306–1312. AAAI Press, Menlo Park (2008)
15. de Salvo Braz, R., Amir, E., Roth, D.: Lifted First-Order Probabilistic Inference. In: IJCAI, pp. 1319–1325 (2005)
16. Singla, P., Domingos, P.: Lifted First-Order Belief Propagation. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008)

Positions, Regions, and Clusters: Strata of Granularity in Location Modelling

Hedda R. Schmidtke and Michael Beigl

Karlsruhe Institute of Technology, TecO
Vincenz-Priessnitz-Str. 3, 76131 Karlsruhe, Germany
{hedda.schmidtke,michael.beigl}@kit.edu

Abstract. Location models are data structures or knowledge bases used in Ubiquitous Computing for representing and reasoning about spatial relationships between so-called smart objects, i.e. everyday objects, such as cups or buildings, containing computational devices with sensors and wireless communication. The location of an object is in a location model either represented by a *region*, by a coordinate *position*, or by a *cluster* of regions or positions. Qualitative reasoning in location models could advance intelligence of devices, but is impeded by incompatibilities between the representation formats: topological reasoning applies to regions; directional reasoning, to positions; and reasoning about set-membership, to clusters. We present a mathematical structure based on scale spaces giving an integrated semantics to all three types of relations and representations. The structure reflects concepts of granularity and uncertainty relevant for location modelling, and gives semantics to applications of RCC-reasoning and projection-based directional reasoning in location models.

Keywords: Spatial granularity, uncertainty, location model, pervasive computing, ubiquitous computing, scale space, qualitative spatial reasoning.

1 Introduction

Location models are data structures or knowledge bases used in Ubiquitous Computing for representing and reasoning about spatial relationships between so-called *smart objects* or *artefacts*. Smart objects – everyday objects, such as the Mediacups [1] or smart buildings, which contain computational devices equipped with sensors and wireless communication – can react intelligently to their spatial environment and to each other through a location model [2]. Generally, three types of location models are distinguished, with each employing certain characteristic types of representation [9]:

- *Symbolic location models* are qualitative location models in which the relationships between named *regions* are represented.
- *Geometric location models* are quantitative location models in which the *position* is represented by means of measured coordinates.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 272–279, 2010.

– *Hybrid location models* contain symbolic as well as geometric location information. Additionally, compounds of heterogeneous artefacts can be represented by finite sets of regions and/or positions, which we will call *clusters* in the following. A complex large-scale artefact, such as a smart building consisting of many locations and devices, is an example of a cluster.

Qualitative spatial reasoning (QSR, cf. [3] for an overview) about locations would be an ideal reasoning technique to enable interesting new intelligent interaction methods for artefacts, as QSR offers a lightweight reasoning approach that can be implemented well even under resource restrictions, as found in autonomous robots [3] or in the small devices of Ubiquitous Computing. However, application of QSR for location models is currently impeded by seeming incompatibilies between the three formats:

– Regions can be compared well with mereotopological relations [13], but relations such as *overlap* and *proper part* are trivial for positions: two points are either identical or differ. For clusters as sets of positions and/or regions, mereotopological relations can be counterintuitive: are the parts of a cluster its elements or the parts of its elements? Are the two in some sense the same?
– Position representations can be compared qualitatively using directional calculi [10]. But relations such as north-of are considerably more difficult to define for regions [14, 17]: is a hallway surrounding a room north, west, east, or south of the room, or does it have all relations at the same time, or none? The same holds even more so for clusters.
– Clusters and their elements are related through the set-theoretical relation of membership and correpondingly definable relations, such as subset. Like mereotopological relations, set-theoretical notions are trivial for positions, either two positions are identical or different. Counterintuitively however, a cluster of one point and the point itself are not the same location in terms of set-theoretical relations. Set-theoretical notions can be applied to regions in a standard way by interpreting regions as point-sets [4].

The aim of this paper is to show that a set of widely used mereotopological and directional QSR-relations can be interpreted in terms of relations over sets of points in a scale space. The resulting structure gives an integrated semantics to all three types of representations found in location modelling in a way so as to reflect uncertainty and heterogeneity of location measurement in Ubiquitous Computing.

After a brief overview of related works (Sect. 2), we introduce the key notion of σ-points as uncertain coordinates in a scale space in Sect. 3. In Sect. 4, we then show how a set of well-known mereotopological [13] and directional relations [10] can be interpreted with respect to scale spaces. We discuss conclusions and open questions in Sect. 5.

2 Related Works

The mathematical structure we propose is built upon notions from scale-space theory, a theory of multi-scale analysis of signals and images used in computer vision research and signal processing [11]. We apply these notions in a novel way to link qualitative reasoning to uncertain quantitative measurements in Ubiquitous Computing. Furthermore, the work presented in this paper draws upon and generalizes related ideas from research on granularity, spatial databases, and diagrammatic reasoning.

When we use knowledge about spatial phenomena, a main characteristic we have to handle is the *spatial granularity* required for a task. Granularity can be understood as a parameter of the task that determines which objects and relations are applicable for the task. In [16], we proposed a mereological axiomatisation of spatial granularity based on size. We go beyond the concepts introduced in [16] as we also represent limitations of accuracy in location models. Coordinate information from location sensing is always subject to uncertainty and varies considerably with devices [6]. Layered multi-resolution *spatial databases* can store objects having different representations on multiple scales, so that each representation is associated with a certain range of scales [5, 12]. The idea of a scale parameter has also been applied for indexing in spatial databases [7] and to diagrammatic visual inference about interval relations [8].

In contrast to spatio-temporal databases and geographic information systems, which contain only quantitative location information in fixed coordinate systems, location models can contain a mixture of qualitative and quantitative spatial information without a common reference system. Each room in a smart building can have its own reference system. We generalise the idea of a scale-parameter in two respects: on the one hand, we address spaces of arbitrary dimension n, and on the other hand, we use the additional coordinate for representing uncertainty of location measurement. The generalised idea is to represent an n-dimensional measurement coordinate $\boldsymbol{m}$ together with a standard deviation σ describing the accuracy of the measurement device or computation that produced the coordinate value. We obtain an $n+1$-dimensional coordinate, a scale-dependent point, which maps to an n-dimensional range of uncertainty.

3 Scale-Dependent Positions, Regions, and Clusters

In order to account for uncertainty in location measurement devices, we interpret a scale-dependent $n + 1$-dimensional coordinate $(\boldsymbol{m}, \sigma)$, consisting of an n-dimensional coordinate vector $\boldsymbol{m} \in \mathbb{R}^n$ and a scale $\sigma \in \mathbb{R}^+ = \{r \in \mathbb{R} \mid r > 0\}$, by a Gaussian function

$$G(\boldsymbol{x}; \boldsymbol{m}, \sigma) = \frac{1}{\sigma(2\pi)^{\frac{n}{2}}} \exp\left[-\frac{d(\boldsymbol{x}, \boldsymbol{m})^2}{2\sigma^2}\right] \tag{1}$$

that centres around the measured coordinate $\boldsymbol{m}$, with the standard deviation σ obtained from the specification of the measurement device. Here, $d(\boldsymbol{x}, \boldsymbol{y})$ is a distance function. Random observational error in location measurement can be represented in the σ-coordinate: when $\boldsymbol{m}$ has been measured, the actual coordinate is within a range δ of $\boldsymbol{m}$ with probability a, the accuracy of the device:

$$\int \cdots \int_{d(\boldsymbol{x}, \boldsymbol{m}) \leq \delta} G(\boldsymbol{x}; \boldsymbol{m}, \sigma) dx^n = a. \tag{2}$$

Using this relationship between a range δ and a corresponding accuracy a, an application can ensure its accuracy is a by performing its reasoning and calculations on the δ-sphere around $\boldsymbol{m}$ for which (2) holds. For any fixed accuracy a we can then derive a transformation function mapping uncertain coordinate information in $\mathbb{R}^n \times \mathbb{R}^+$ obtained from measurement to corresponding regions of uncertainty in $\mathbb{R}^n$. An application requiring a certain accuracy of, for instance, $a = 95\%$ can look up δ so that (2) holds. In the 1-dimensional case, for instance, we obtain that $\delta = 2\sigma$ for $a = 95\%$ and $\delta = \sigma$ for $a = 68\%$; that is, δ is a multiple $\lambda_a\sigma$ of σ, with λ_a depending on a. The $n+1$ dimensional coordinate $\boldsymbol{e} = (x_1, \ldots, x_n, \sigma)$ of a position can thus be interpreted by the application as the n-sphere of radius $\lambda_a\sigma$ that has its centre at $\boldsymbol{x} = (x_1, \ldots, x_n)$.

In order to avoid confusion between the points of the n-dimensional underlying space $\mathbb{R}^n$ and points, lines, planes, etc. in the $n+1$-dimensional multi-scale space $\mathbb{R}^n \times \mathbb{R}^+$, we continue to call the former *points*, *lines*, *planes*, etc. as before and call the latter *σ-points*, *σ-lines*, *σ-planes*, etc. below. For readability, we use variables $\boldsymbol{e}$, $\boldsymbol{e_1}$, $\boldsymbol{e_2}$ to range over σ-points and variables $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{x_1}$, $\boldsymbol{x_2}$, etc. to refer to points.

Figure 1 illustrates the case of (a) 1-dimensional and (b) 2-dimensional position measurements from devices with different accuracies. For the 2-dimensional case, we obtain a set of discs of $\mathbb{R}^2$, with each disc specified by three coordinates: a centre $c \in \mathbb{R}^2$ and a radius $r \in \mathbb{R}^+$. In order to describe the translation between representations in the two spaces, we define a family of functions $f_a : \mathbb{R}^n \times \mathbb{R}^+ \to 2^{\mathbb{R}^n}$ mapping coordinates, in the $n+1$-dimensional multi-scale space $\mathbb{R}^n \times \mathbb{R}^+$ to n-dimensional spheres in the underlying space $\mathbb{R}^n$ for an accuracy a. We define f_a using a projection function $c : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^n$ yielding the centre of the sphere referred to by a σ-point, and functions $\sigma_a : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^+$ that yield the radius of the sphere for a given accuracy a. Then, the σ_a-sphere around the centre point $c(\boldsymbol{e})$ is the set of all points whose distance to $c(\boldsymbol{e})$ is smaller than $\sigma_a(\boldsymbol{e})$ (3).

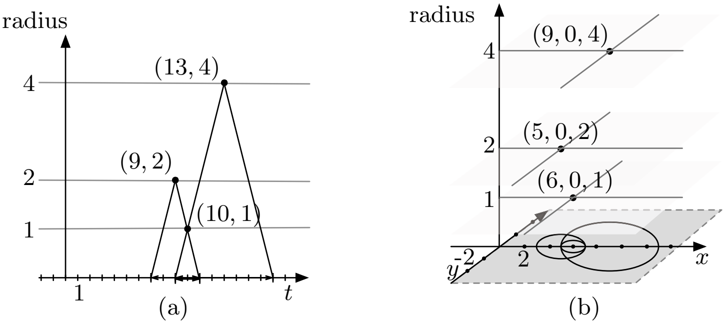


Fig. 1. Scale spaces: a) the interval from $t = 7$ to $t = 11$ with the centre at $t = 9$ and a radius of 2 (i.e., duration of 4) time units is represented as a point at the coordinate $(9, 2)$ in the 2-dimensional scale space of intervals (cf. [8]); b) the open disc $A = \{(x, y) \mid d((x, y), (5, 0)) < 2\}$ with centre at point $(5, 0)$ and radius 2 is represented as a point at the coordinate $(5, 0, 2)$ in a 3-dimensional scale space.

$$f_a(\boldsymbol{e}) = \{\boldsymbol{x} \mid d(\boldsymbol{x}, c(\boldsymbol{e})) \leq \sigma_a(\boldsymbol{e})\} \tag{3}$$

$$f_a(S) = \{\boldsymbol{x} \mid \exists \boldsymbol{e} \in S : d(\boldsymbol{x}, c(\boldsymbol{e})) \leq \sigma_a(\boldsymbol{e})\} \tag{4}$$

We extend the definition of f_a to range over sets S of σ-points (4). We observe:

- The projection function f_a maps any location to an n-dimensional region.
- A σ-point and the singleton set containing it are mapped to the same n-dimensional region in $\mathbb{R}^n$.
- A union of sets S_1, S_2 can map to a connected region $f_a(S_1 \cup S_2)$, without the set $S_1 \cup S_2$ being connected.

The third property is particularly relevant when we want to handle clusters, as clusters are collections of positions and/or regions. It reflects an often desirable property of clusters, to be able to represent space in terms of non-overlapping partitions (cf. [16]).

Taking the three properties together we conclude that positions, regions, and clusters can be interpreted as sets S of σ-points that are distinguished only by internal topological properties: positions are represented by singleton sets, regions by sets that fulfil additional topological criteria such as connectedness [13], and clusters by finite unions of sets.

4 Relations between Sets of $\boldsymbol{\sigma}$-Points

We can now specify the mereotopological and directional relations. For this discussion, we will take the perspective of a single application with a fixed accuracy a. Accordingly, we leave out the application accuracy index a in the following specifications, that is, write f and σ instead of f_a and σ_a and define o, p, EC, etc, instead of o_a, p_a, EC_a, etc.

4.1 Mereotopological Relations

We define functions o and p to express a degree of *overlap* o, *parthood* p, and *similarity* q between two sets of σ-points:

$$o(S_1, S_2) = \inf_{\boldsymbol{e_1} \in S_1, \boldsymbol{e_2} \in S_2} \sigma(\boldsymbol{e_1}) + \sigma(\boldsymbol{e_2}) - d(c(\boldsymbol{e_1}), c(\boldsymbol{e_2})) \tag{5}$$

$$p(S_1, S_2) = \sup_{\boldsymbol{e_1} \in S_1} \inf_{\boldsymbol{e_2} \in S_2} \sigma(\boldsymbol{e_2}) - \sigma(\boldsymbol{e_1}) - d(c(\boldsymbol{e_1}), c(\boldsymbol{e_2})) \tag{6}$$

$$q(S_1, S_2) = \max(p(S_1, S_2), p(S_2, S_1)) \tag{7}$$

$$\lim_{\epsilon \longrightarrow 0} q(S_1, S_2) = d_H(f(S_1), f(S_2)) \tag{8}$$

For sets S_1, S_2 whose σ-points $\boldsymbol{e} \in S_1 \cup S_2$ are point-like, i.e. $\sigma(\boldsymbol{e}) < \epsilon$, the function q converges to the Hausdorff-distance d_H (8).

The RCC-relations [13] of *connection* C, *disconnection* DC, and *part-of* P between sets of σ-points S_1 and S_2 can be interpreted as relations between regions of $\mathbb{R}^n \times \mathbb{R}^+$:

$$\begin{array}{rr} C(S_1,S_2) \text{ iff } o(S_1,S_2) \geq 0 & PO(S_1,S_2) \text{ iff } o(S_1,S_2) > 0 \\ DC(S_1,S_2) \text{ iff } o(S_1,S_2) < 0 & EC(S_1,S_2) \text{ iff } o(S_1,S_2) = 0 \\ P(S_1,S_2) \text{ iff } p(S_1,S_2) \geq 0 & EQ(S_1,S_2) \text{ iff } q(S_1,S_2) = 0 \\ NTP(S_1,S_2) \text{ iff } p(S_1,S_2) > 0 & TP(S_1,S_2) \text{ iff } p(S_1,S_2) = 0 \end{array} \tag{9}$$

Further relations from [13] (PP, TPP, NTPP) can be defined from the above. Note that the σ-point relations based on p can diverge from corresponding relations over the regions projected by f for discontinuous sets S_1, S_2. As discussed above, clusters representing partitions must typically contain such discontinuities. If this is to be avoided in a given application, a scale-dependent hull operation such as the one proposed in [15] has to be employed.

4.2 Directional Relations

For modelling directional relations, we need not only σ-points but also higher-dimensional geometric entities, such as σ-lines and σ-planes. Like lines are defined by two different points, σ-lines can be defined by two σ-points, and σ-planes, by three σ-points. The σ-lines and σ-planes are crucial for interpreting directional relations. The concept of parallelism of directed axes and their quadrants in a plane is the core of reasoning with the directional calculus of Ligozat [10]. Any σ-line can be directed by two of its σ-points so as to give rise to two orderings ($>$ and $<$) between its σ-points and between σ-points on a parallel (10). Three σ-points, in turn, give rise to four different orderings ($<<, ><, >>, <>$) of a σ-plane and, with more than two dimensions, in parallel σ-planes (11):

$$\begin{array}{l} \boldsymbol{e'} >_{\boldsymbol{e_0},\boldsymbol{e_1}} \boldsymbol{e} \text{ iff } \exists\lambda > 0 : \boldsymbol{e'} = \boldsymbol{e} + \lambda(\boldsymbol{e_1} - \boldsymbol{e_0}) \\ \boldsymbol{e'} <_{\boldsymbol{e_0},\boldsymbol{e_1}} \boldsymbol{e} \text{ iff } \exists\lambda < 0 : \boldsymbol{e'} = \boldsymbol{e} + \lambda(\boldsymbol{e_1} - \boldsymbol{e_0}) \end{array} \tag{10}$$

$$\begin{array}{l} \boldsymbol{e'} >>_{\boldsymbol{e_0},\boldsymbol{e_1},\boldsymbol{e_2}} \boldsymbol{e} \text{ iff } \exists\lambda_1 > 0, \lambda_2 > 0 : \boldsymbol{e'} = \boldsymbol{e} + \lambda_1(\boldsymbol{e_1} - \boldsymbol{e_0}) + \lambda_2(\boldsymbol{e_2} - \boldsymbol{e_0}) \\ \boldsymbol{e'} ><_{\boldsymbol{e_0},\boldsymbol{e_1},\boldsymbol{e_2}} \boldsymbol{e} \text{ iff } \exists\lambda_1 > 0, \lambda_2 < 0 : \boldsymbol{e'} = \boldsymbol{e} + \lambda_1(\boldsymbol{e_1} - \boldsymbol{e_0}) + \lambda_2(\boldsymbol{e_2} - \boldsymbol{e_0}) \\ \boldsymbol{e'} <<_{\boldsymbol{e_0},\boldsymbol{e_1},\boldsymbol{e_2}} \boldsymbol{e} \text{ iff } \exists\lambda_1 < 0, \lambda_2 < 0 : \boldsymbol{e'} = \boldsymbol{e} + \lambda_1(\boldsymbol{e_1} - \boldsymbol{e_0}) + \lambda_2(\boldsymbol{e_2} - \boldsymbol{e_0}) \\ \boldsymbol{e'} <>_{\boldsymbol{e_0},\boldsymbol{e_1},\boldsymbol{e_2}} \boldsymbol{e} \text{ iff } \exists\lambda_1 < 0, \lambda_2 > 0 : \boldsymbol{e'} = \boldsymbol{e} + \lambda_1(\boldsymbol{e_1} - \boldsymbol{e_0}) + \lambda_2(\boldsymbol{e_2} - \boldsymbol{e_0}) \end{array} \tag{11}$$

For example, the directional relation system of cardinal directions north, northeast, east, etc. [10], can be described based on a reference system spanned by three σ-points $\boldsymbol{e_0}, \boldsymbol{e_N}, \boldsymbol{e_E}$ – where $\boldsymbol{e_N}$ is to the north of, and $\boldsymbol{e_E}$ to the east of, $\boldsymbol{e_0}$: the linear relations $>_{\boldsymbol{e_0},\boldsymbol{e_N}}$ (to the north of, or N), $<_{\boldsymbol{e_0},\boldsymbol{e_N}}$ (S), $>_{\boldsymbol{e_0},\boldsymbol{e_E}}$ (E), $<_{\boldsymbol{e_0},\boldsymbol{e_E}}$ (W) form the axes of the qualitative reference system, while the four planar relations $>>_{\boldsymbol{e_0},\boldsymbol{e_N},\boldsymbol{e_E}}$ (NE), $<>_{\boldsymbol{e_0},\boldsymbol{e_N},\boldsymbol{e_E}}$ (SE), $<<_{\boldsymbol{e_0},\boldsymbol{e_N},\boldsymbol{e_E}}$ (SW), $><_{\boldsymbol{e_0},\boldsymbol{e_N},\boldsymbol{e_E}}$ (NW) form the quadrants. Note that the three σ-points $\boldsymbol{e_0}, \boldsymbol{e_N}, \boldsymbol{e_E}$ spanning the reference system need to have the same scale σ if we want to obtain a projection-based cardinal direction system, such as shown in Fig. 2.

The directional relations described above require positions $\boldsymbol{e}, \boldsymbol{e'}$. If we want to apply these relations to sets S, S', we need a transformation function that translates extended objects, i.e. regions and clusters, into positions. A transformation from arbitrary sets to positions can be achieved in a number of ways,

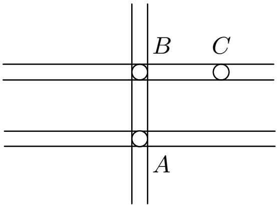


Fig. 2. Projection-based inference about three locations of the same size: C is to the East of B, and B is to the North of A

e.g., by using a σ-point centred on the smallest enclosing sphere, where $e(\boldsymbol{x}, \sigma)$ is the function mapping a centre coordinate $\boldsymbol{x}$ given σ to coordinates $\boldsymbol{e}$ (12):

$$\mathrm{r2p}(S, \sigma) = e(c(\underset{\boldsymbol{e} \in \mathbb{R}^n \times \mathbb{R}^+, P(S, \{\boldsymbol{e}\})}{\arg\min} \sigma(\boldsymbol{e})), \sigma) \tag{12}$$

$$S' >^{\sigma}_{\boldsymbol{e_0}, \boldsymbol{e_1}} S \text{ iff } \mathrm{r2p}(S', \sigma) >_{\boldsymbol{e_0}, \boldsymbol{e_1}} \mathrm{r2p}(S, \sigma) \tag{13}$$

As *r2p* is a partial function, we conclude that directional relations are only applicable for scales σ on which *r2p* is defined. Definitions (10) and (11) can then be extended as exemplified in (13).

5 Summary and Conclusions

We presented an interpretation of mereotopological and directional relations for location models used in Ubiquitous Computing. Our approach respects a range of specific concerns in location modelling, such as uncertainty of location measurement, the mixture of qualitative and quantitative information and the specific types of representations employed in location models, namely, positions, regions, and clusters. We showed how all three representations can be interpreted in a unifying way using a scale space, and how applications can employ this representation to handle uncertainty from sensors and to ensure a required accuracy.

We gave a specification of mereotopological and directional relations in terms of this interpretation so as to allow integration of QSR-mechanisms into location models. Given this interpretation, relations can be computed from the quantitative parts of a location model and can be integrated with the qualitative parts so as to allow qualitative reasoning. The proposed specification is a first step towards an expressive, yet light-weight spatial reasoning mechanism for smart objects.

References

[1] Beigl, M., Gellersen, H.-W., Schmidt, A.: Mediacups: experience with design and use of computer-augmented everyday artefacts. Computer Networks 35(4), 401–409 (2001)

[2] Beigl, M., Zimmer, T., Decker, C.: A location model for communicating and processing of context. Personal and Ubiquitous Computing 6(5/6), 341–357 (2002)

[3] Cohn, A.G., Hazarika, S.M.: Qualitative spatial representation and reasoning: An overview. Fundamenta Informaticae 46(1-2), 1–29 (2001)

[4] Egenhofer, M.J., Franzosa, R.D.: Point set topological relations. International Journal of Geographical Information Systems 5, 161–174 (1991)

[5] Frank, A.U., Timpf, S.: Multiple representations for cartographic objects in a multi-scale tree – an intelligent graphical zoom. Computers and Graphics 18(6), 823–829 (1994)

[6] Hightower, J., Borriello, G.: A survey and taxonomy of location systems for ubiquitous computing. Computer 34(8), 57–66 (2001)

[7] Hörhammer, M., Freeston, M.: Spatial indexing with a scale dimension. In: Güting, R.H., Papadias, D., Lochovsky, F.H. (eds.) SSD 1999. LNCS, vol. 1651, pp. 52–71. Springer, Heidelberg (1999)

[8] Kulpa, Z.: Diagrammatic representation of interval space in proving theorems about interval relations. Reliable Computing 3(3), 209–217 (1997)

[9] Leonhardt, U.: Supporting Location Awareness in Open Distributed Systems. PhD thesis, Imperial College, London, UK (1998)

[10] Ligozat, G.: Reasoning about cardinal directions. Journal of Visual Languages and Computing 9, 23–44 (1998)

[11] Lindeberg, T.: Scale-space theory: A basic tool for analysing structures at different scales. Journal of Applied Statistics 21(2), 225–270 (1994)

[12] Parent, C., Spaccapietra, S., Zimányi, E.: The MurMur project: Modeling and querying multi-representation spatio-temporal databases. Inf. Syst. 31(8), 733–769 (2006)

[13] Randell, D., Cui, Z., Cohn, A.: A spatial logic based on region and connection. In: Knowledge Representation and Reasoning, pp. 165–176. Morgan Kaufmann, San Francisco (1992)

[14] Schmidtke, H.R.: The house is north of the river: Relative localization of extended objects. In: Montello, D.R. (ed.) COSIT 2001. LNCS, vol. 2205, pp. 414–430. Springer, Heidelberg (2001)

[15] Schmidtke, H.R.: Aggregations and constituents: Geometric specification of multigranular objects. Journal of Visual Languages and Computing 16(4), 289–309 (2005), doi:10.1016/j.jvlc.2004.11.007

[16] Schmidtke, H.R., Woo, W.: A size-based qualitative approach to the representation of spatial granularity. In: Veloso, M.M. (ed.) Twentieth International Joint Conference on Artificial Intelligence, pp. 563–568 (2007)

[17] Skiadopoulos, S., Sarkas, N., Sellis, T.K., Koubarakis, M.: A family of directional relation models for extended objects. IEEE Trans. Knowl. Data Eng. 19(8), 1116–1130 (2007)

Soft Evidential Update via Markov Chain Monte Carlo Inference

Dominik Jain and Michael Beetz

Intelligent Autonomous Systems Group
Department of Informatics
Technische Universität München

Abstract. The key task in probabilistic reasoning is to appropriately update one's beliefs as one obtains new information in the form of evidence. In many application settings, however, the evidence we obtain as input to an inference problem may be uncertain (e.g. owing to unreliable mechanisms with which we obtain the evidence) or may correspond to (soft) degrees of belief rather than hard logical facts. So far, methods for updating beliefs in the light of soft evidence have been centred around the iterative proportional fitting procedure and variations thereof. In this work, we propose a Markov chain Monte Carlo method that allows to directly integrate soft evidence into the inference procedure without generating substantial computational overhead. Within the framework of Markov logic networks, we demonstrate the potential benefit of this method over standard approaches in a series of experiments on synthetic and real-world applications.

1 Introduction

A fundamental task in probabilistic reasoning is to compute the probability of a query q given evidence e, i.e. to compute the conditional probability $P(q \mid e)$; and one typically assumes that e is a plain fact that could be represented as a logical conjunction. On many occasions, however, the evidence that we can supply as input to an inference problem does not correspond to firm beliefs. Rather, evidence is often uncertain in the sense that it may represent a belief that either cannot be fully trusted (owing to unreliable mechanisms with which we obtain the evidence) or corresponds directly to a probability distribution that should be believed to hold.

For instance, we may regard the output of a probabilistic classifier as uncertain evidence. Surely it is more appropriate to make full use of the knowledge about the distribution over classes that such a classifier provides than to simply assume that the mode of the distribution is irrevocable fact. Moreover, it may be desirable to incorporate the (soft) beliefs that were computed by external sources of probabilistic information as input to an inference problem rather than to compute everything within a single model – given the fact that inference in all-encompassing probabilistic models is computationally infeasible. It may, therefore, be advisable to find ways of suitably factoring domains into loosely coupled probabilistic fragments and combine them as necessary. Whenever we do combine them, we will usually have to pass on probabilistic information rather than logical facts. The problem of updating beliefs in the light of soft evidence can

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 280–290, 2010.

thus be viewed as a subproblem of probabilistic information interchange. Probabilistic information interchange is particularly relevant within the context of multi-agent systems. Multiple agents that autonomously explore similar environments may want to share their probabilistic findings with other agents, and these agents will have to incorporate the received findings into their own probabilistic knowledge bases and update their beliefs accordingly [1].

In this work, we address precisely the issues pertaining to belief update that are fundamental for applications such as the ones we named above. To put our work in perspective, we first give an overview of semantically different ways in which evidential uncertainty can be interpreted, reviewing the notions of *virtual evidence* and *soft evidence* put forth in the literature (Section 2). Because the former is usually straightforward to handle, we concentrate, for the most part, on the harder problem of *soft evidential update*, i.e. the integration of firm beliefs on probability distributions (i.e. probability constraints). In Section 4, we present two solutions to this problem, one of which is essentially analogous to previous (state-of-the-art) approaches, while the other is fundamentally new in that it extends a Markov chain Monte Carlo (MCMC) method to directly incorporate probability constraints into the inference process, which can be significantly more efficient. We present our methods based on the widely used framework of Markov logic networks (MLNs), a highly general yet simple representation formalism for statistical knowledge [2] (Section 3). We thoroughly evaluate the performance of our newly proposed MCMC method in a series of experiments based on both synthetic and real-world applications (Section 5).

2 On the Semantics of Uncertain Evidence

Whenever evidence on some set of random variables X_E is uncertain, we associate with each assignment $X_E = x_E$ a degree of belief $p_{x_E} \in [0, 1]$. For every assignment $X_E = x_E$, there are (at least) two ways in which we might interpret the evidence and the corresponding degree of belief p_{x_E}:

(1) We have made an observation indicating $X_E = x_E$ and the degree to which this observation is reliable is p_{x_E}.
(2) The degree to which $X_E = x_E$ should be believed is p_{x_E}.

In either case, p_{x_E} is a well-defined degree of belief that should appropriately alter our posterior beliefs on unobserved variables, yet the semantics are fundamentally different. In the former case, we speak of *virtual* (or *intangible*) *evidence* (as the evidence impinges not immediately upon random variables in X but virtual observations thereof), while the latter is referred to as *soft evidence*. In the following, we will explain the differences between these two semantics in more detail.

We consider a full joint distribution over a set of random variables $X = \{X_1, \ldots, X_{|X|}\}$, which is partitioned into observed variables X_E (evidence variables) and unobserved variables X_U, i.e. $X = X_E \uplus X_U$. We denote by $\mathcal{X}$ the set of possible worlds, i.e. the set of possible assignments of values to each of the variables in X. For $x \in \mathcal{X}$, we simply write x as shorthand for $X = x$. Without loss of generality, we assume that all random variables under consideration are Boolean.

2.1 Virtual Evidence

Virtual evidence can be thought of as evidence concerning (undisclosed) observations that are outside of the set of variables X that we consider in our joint probability distribution but that have bearing on a subset X_E of these variables [3]. Technically, virtual evidence on X_E is realized by adding to the model auxiliary evidence variables V_E (corresponding to the observations that were made) and adding the likelihood of an assignment $X_E = x_E$ under these observations as an additional potential, i.e. we add $L(x_E) := P(V_E = v_E \mid X_E = x_E)$ for all $x_E \in \mathcal{X}_E$, where v_E is an arbitrarily chosen value assigned to the auxiliary evidence variables.

Adding virtual evidence on a variable X_i with beliefs $\langle p, q \rangle$ for *true* and *false* respectively can be viewed as adding an observation with a corresponding observation model in which the probability with which we observe $X_i =$ *true* (*false*) if X_i is indeed *true* (*false*) is p (q).[1]

By applying the principle of virtual evidence, we essentially weight each possible world with the likelihood of the evidence variables X_E:

$$\begin{aligned} &P(X \mid v_E) = P(X_U, X_E \mid v_E) = P(X_U \mid X_E) \cdot P(X_E \mid v_E) \\ &\propto P(X_U \mid X_E) \cdot P(v_E \mid X_E) \cdot P(X_E) = P(X_U, X_E) \cdot P(v_E \mid X_E) \\ &= P(X) \cdot L(X_E) \end{aligned} \tag{1}$$

2.2 Soft Evidence

The concept of soft evidence is both simpler and more difficult. It is simpler, because its interpretation is straightforward: Any piece of soft evidence is but a constraint on the probability distribution over assignments to (a subset of) X_E. Soft evidence on a single variable X_i with distribution $\langle p_i, 1-p_i \rangle$ simply requires the probability of $X_i =$ *true* to be p_i. However, as we will see, soft evidence is also the more difficult concept, because its treatment cannot simply be reduced to the addition of auxiliary variables for which potentials can straightforwardly be derived.

Let s_E represent the observations that have led us to believe in the soft evidence that we are given. In its entirety, soft evidence on the set of variables X_E can then be understood as defining the distribution $P(X_E \mid s_E)$. *Jeffrey's rule* [3] provides a way of updating our posterior beliefs on the unobserved variables X_U in the light of our observations:

$$P(X_U \mid s_E) = \sum_{x_E} P(X_U \mid x_E) \cdot P(x_E \mid s_E) \tag{2}$$

We obtain the posterior distribution over $\mathcal{X}$ as

$$\begin{aligned} P(X \mid s_E) &= P(X_U, X_E \mid s_E) = P(X_U \mid X_E) \cdot P(X_E \mid s_E) \\ &= \frac{P(X_U, X_E)}{P(X_E)} \cdot P(X_E \mid s_E) = P(X) \cdot \frac{P(X_E \mid s_E)}{P(X_E)} \end{aligned} \tag{3}$$

[1] Note that the values appearing in a virtual evidence vector (here p and q) need not sum to 1 (only the ratio between any two values matters), but we could equivalently represent them as a distribution (and Section 2.3 further on assumes that we do).

2.3 Discussion

Both virtual and soft evidence can be seen as generalizations of hard evidence. In the special cases where values of 1.0 or 0.0 are specified, both equivalently reduce to hard evidence. For values in between, however, the semantics differ greatly.

From a computational perspective, virtual evidence does not pose any challenges, as it imposes only strictly local dependencies by introducing additional likelihoods, which can be handled by adding appropriate potentials to a probabilistic model [3,4]. Its use is appropriate only if the uncertain information originates from a source that indeed computes mere likelihoods. The effect that a piece of virtual evidence has on the posterior of the respective variable(s) is highly dependent on the probability that is indicated by the model without the respective piece of evidence (and given any further evidence).[2] Soft evidence is concerned precisely with defining that effect, and its treatment therefore needs to take into consideration the fact that several pieces of uncertain evidence cannot usually be treated independently but influence each other.

3 Markov Logic Networks

Markov logic networks (MLNs) [2] combine first-order logic with the semantics of probabilistic graphical models. An MLN L is given by a set of pairs $\langle F_i, w_i \rangle$, where F_i is a formula in first-order logic and w_i is a real-valued weight. For each finite domain of discourse D (set of constants), an MLN L defines a *ground Markov random field* $M_{L,D} = \langle X, G \rangle$ as follows:

1. X is a set of boolean variables. For each possible grounding of each predicate appearing in L, we add to X a boolean variable (ground atom). We denote by $\mathcal{X} := \mathbb{B}^{|X|}$ the set of possible worlds, i.e. the set of possible assignments of truth variables to the variables in X.
2. G is a set of weighted ground formulas, i.e. a set of pairs $\langle \widehat{F}_j, \widehat{w}_j \rangle$, where $\widehat{F}_j$ is a ground formula and $\widehat{w}_j$ a real-valued weight. For each possible grounding $\widehat{F}_j$ of each formula F_i in L, we add to G the pair $\langle \widehat{F}_j, \widehat{w}_j = w_i \rangle$. With each such pair, we associate a feature $\widehat{f}_j : \mathcal{X} \to \{0,1\}$, whose value for $x \in \mathcal{X}$ is 1 if $\widehat{F}_j$ is satisfied in x and 0 otherwise, and whose weight is $\widehat{w}_j$.

$M_{L,D}$ specifies a probability distribution over $\mathcal{X}$ as follows,

$$P(X = x) = \tfrac{1}{Z} \exp \left(\textstyle\sum_j \widehat{w}_j \widehat{f}_j(x) \right) \tag{4}$$

where Z is a normalization constant. The probability of a possible world $x \in \mathcal{X}$ is thus proportional to the exponentiated sum of weights of formulas that are satisfied in x, i.e. $P(x) \propto \exp(\sum_j \widehat{w}_j \widehat{f}_j(x)) =: \omega(x)$. With $s(F) := \sum_{x \in \mathcal{X},\, x \models F} \omega(x)$, we can calculate the probability of any ground formula F_1 given any other formula F_2 as

$$P(F_1 \mid F_2) = \frac{P(F_1, F_2)}{P(F_2)} = \frac{s(F_1 \wedge F_2)}{s(F_2)} \tag{5}$$

[2] For example, if the probability of $X_i = \mathit{true}$ given the other evidence is 0.1, adding virtual evidence on X_i with values $\langle 0.8, 0.2 \rangle$ will cause the probability of $X_i = \mathit{true}$ to become $0.1 \cdot 0.8/(0.1 \cdot 0.8 + 0.9 \cdot 0.2) \approx 0.31$, not 0.8.

4 Soft Evidential Update

In the following, we address the question of how beliefs can efficiently be updated in the light of soft evidence. We will present two methods for the problem of soft evidential update that are considerably more practical than the naive approach involving $2^{|X_E|}$ inference runs that Equation 2 gives rise to, discussing related work along the way.

4.1 Probability Constraints and Iterative Fitting

The problem of soft evidential update is closely related to the concept of imposing probability constraints on a distribution. To make this explicit, we briefly review the *iterative proportional fitting procedure* (IPFP) (cf. [5]). This procedure allows to modify a joint distribution such that it satisfies a set $\mathcal{R} = \{R_k\}$ of probability constraints, a *probability constraint* on a distribution $P(X)$ being a probability distribution $R_k(Y_k)$ over $Y_k \subseteq X$. IPFP guarantees that the distribution $P(X)$ is modified in such a way that the resulting distribution satisfies all of the given probability constraints and is maximally similar to $P(X)$ (with respect to Kullback-Leibler divergence). It achieves this by iteratively adjusting the distribution to fit one of the probability constraints in a round robin fashion. The initial distribution is $Q_0(X) = P(X)$. Then, in the i-th step, IPFP adjusts the (support of the) distribution to satisfy the k-th constraint $R_k(Y_k)$ (where $k = ((i-1) \bmod m) + 1$ if m is the number of probability constraints) using

$$Q_i(X) = Q_{i-1}(X) \cdot \frac{R_k(Y_k)}{Q_{i-1}(Y_k)} \tag{6}$$

Iterative fitting continues until convergence, which can always be achieved provided that the probability constraints are consistent.

Notice that the update rule above is analogous to Equation 3. Hence soft evidence can be handled by applying IPFP with a single constraint $R_1(X_E) := P(X_E \mid s_E)$.

In practice, however, we will rarely be given a joint distribution $P(X_E \mid s_E)$ as evidence, for such distributions are exponential in the size of X_E and therefore impractical to obtain and represent explicitly. Rather, from now on, consider the (perhaps most relevant) case where we are given a number of soft evidences, each concerning a single variable. Let $\{i_1, \ldots, i_{|X_E|}\}$ be the set of indices of variables in X_E. The set of probability constraints $\mathcal{R}$ then contains, for each variable $X_{i_k} \in X_E$, a distribution $R_k(X_{i_k}) = \langle p_k, 1 - p_k \rangle$. Note that this means that the distribution $P(X_E \mid s_E)$ is no longer given – and Jeffrey's rule is inapplicable unless we make additional assumptions. We should, of course, not simply assume independence of the individual pieces of soft evidence, since we do have information about dependencies in the form of the prior $P(X)$. An application of IPFP with the set of constraints $\mathcal{R} = \{R_1, \ldots, R_{|X_E|}\}$ will ensure that this information is appropriately considered.

Unfortunately, a direct application of IPFP is generally infeasible, because it requires us to represent the full-joint distribution over $\mathcal{X}$ explicitly. Therefore, it seems more appropriate to apply iterative fitting not to the full-joint distribution itself but rather to a model that is capable of representing it compactly. Consider a ground Markov random field $M_{L,D} = \langle X, G \rangle$ and the distribution $P(X)$ it represents. For each constraint $R_k(X_{i_k})$, we add to G the ground formula $\widehat{F}_{R_k} := X_{i_k}$ with weight $\widehat{w}_{R_k}$; Initially, we set $\widehat{w}_{R_k} \leftarrow 0$ for all $R_k \in \mathcal{R}$ and thus obtain as the initial distribution $Q_0(X) = P(X)$. We will now show that we can achieve the transition described in Equation 6 by modifying only the respective weight $\widehat{w}_{R_k}$. The probability of X_{i_k} indicated by the current model is $q := s(X_{i_k})/(s(X_{i_k}) + s(\neg X_{i_k}))$ (i), yet R_k requires this probability to become p_k. By adding $\log(\lambda)$ to $\widehat{w}_{R_k}$, we scale the sum of exponentiated sums of weights of possible worlds satisfying X_{i_k}, i.e. $s(X_{i_k})$, by a factor of λ. Therefore, we need to find λ such that $p_k = s(X_{i_k})\lambda/(s(X_{i_k})\lambda + s(\neg X_{i_k}))$ (ii). Combining (i) and (ii) yields $\lambda = p_k \cdot (1-q)/((1-p_k)\cdot q)$ (see also [6]). We can thus achieve the transition from Q_{i-1} to Q_i by applying the update

$$\widehat{w}_{R_k} \leftarrow \widehat{w}_{R_k} + \log\left(\frac{p_k \cdot (1-q)}{(1-p_k)\cdot q}\right) \tag{7}$$

and perform iterative fitting at the level of model parameters. In each step, we need to run an inference algorithm in order to compute q, the current probability of the random variable (ground atom) whose probability we want to fit. Once the procedure has converged, we have the desired model parameters and can update our beliefs based on inference over the model. Henceforth, we refer to the algorithm described above as *IPFP-M* (IPFP at the model level).

Pan et al. have previously proposed a similar method for Bayesian networks, which extends the model with virtual evidence nodes whose conditional distributions they iteratively adjust to fit the probability constraints (Alg. 1 in [7]). Indeed, virtually all related work on the subject of soft evidential update has been centred around variations of IPFP [1,5,7,8]. It has even been suggested to directly apply IPFP but reduce the set of variables for which the joint distribution needs to be explicitly updated from X to X_E (Alg. 2 in [7]) – by first precomputing the prior $P(X_E)$, applying IPFP to $P(X_E)$ and then running an inference algorithm to update the beliefs on unobserved variables using a model extended with the results of IPFP. Thus, the complexity of every IPFP step is exponential in the size of X_E, which renders the algorithm inapplicable as X_E grows larger (it is, however, advantageous if X_E is very small). The same is true for the Big Clique algorithm [1], which takes a similar approach, albeit within a more confined setting (it is formulated strictly as an extension to the junction tree algorithm). IPFP-M, however, scales to larger sets of evidence variables; its performance is largely dependent on the efficiency of the underlying inference method. Most importantly, we can use methods that do not require an explicit representation of a cumbersome full-joint distribution and, in particular, approximate inference methods.

4.2 A Markov Chain Monte Carlo Method

We now introduce a Markov chain Monte Carlo (MCMC) method, which allows to directly integrate soft evidence into the inference procedure – without generating substantial overhead. Specifically, we present an extension of the algorithm MC-SAT [9], which, unlike other MCMC inference algorithms, can soundly handle deterministic and near-deterministic dependencies.

Consider a ground Markov random field $M_{L,D} = \langle X, G\rangle$ in which all formula weights are non-negative. Please note that this is not a restriction, as any pair $\langle \widehat{F}_j, \widehat{w}_j\rangle \in G$ can be equivalently replaced by $\langle \neg\widehat{F}_j, -\widehat{w}_j\rangle$. MC-SAT is a slice sampler that uses one auxiliary variable U_j per ground formula $\langle \widehat{F}_j, \widehat{w}_j \rangle \in G$. The joint distribution over the random variables X and the auxiliary variables U is given as $P(X = x, U = u) = (1/Z) \prod_j I_{[0,\exp(\widehat{f}_j(x)\widehat{w}_j)]}(u_j)$, where I is an indicator function which returns 1 if the argument is within the given interval and 0 otherwise. For a given $x \in \mathcal{X}$, $P(U = u \mid X = x)$ is uniform in the assignments that are within the intervals imposed by x, and the product of the sizes of these intervals is $\omega(x)$ and thus proportional to $P(x)$. $P(X = x \mid U = u)$ is uniform in the slice (subset) of $\mathcal{X}$ where $u_j < \exp(\widehat{f}_j(x)\widehat{w}_j)$ for all j, which implies that if $u_j > 1$, then $\widehat{f}_j(x) = 1$ and $\widehat{F}_j$ must be satisfied.

The Markov chain is formed by alternatingly sampling the auxiliary variables and the actual state as follows. The initial state $x^{(0)} \in \mathcal{X}$ must be one that satisfies all hard constraints in G. Then, in step i, the auxiliary variables are first resampled given the previous state $x^{(i-1)}$. If $\widehat{F}_j$ was satisfied in $x^{(i-1)}$, then with probability $(\exp(\widehat{w}_j) - 1)/\exp(\widehat{w}_j)$, we sample $u_j > 1$ and therefore $\widehat{F}_j$ must also be satisfied in the current step. We thus obtain a set $M = \{\widehat{F}_j \mid u_j > 1\}$ of formulas that must be satisfied and use a SAT sampler to uniformly sample a state $x^{(i)} \in \mathcal{X}$ that satisfies the formulas in M. For this purpose, MC-SAT uses the SampleSAT algorithm [10], which extends the WalkSAT algorithm by injecting randomness via simulated annealing-type moves, allowing near-uniform samples to be obtained. Because SampleSAT requires formulas to be in conjunctive normal form, we apply a conversion beforehand.

We now extend this algorithm to explicitly consider the probability constraints $\mathcal{R}$ corresponding to a set of soft evidences. Intuitively, we would like to achieve the effect of applying MC-SAT to the model we would have obtained by applying IPFP-M to $P(X)$ and $\mathcal{R}$, yet without having to actually compute the weights w_{R_k} that IPFP-M indicates upon convergence. Consider a constraint $R_k(X_{i_k}) = \langle p_k, 1 - p_k \rangle$ and the pair $\langle X_{i_k}, w_{R_k} \rangle$ that IPFP-M would have added to G. Observe that without $\langle X_{i_k}, w_{R_k} \rangle$, there is some relative frequency with which states in which X_{i_k} is *true* would have appeared in MC-SAT's Markov chain. The weight w_{R_k} computed by IPFP-M would achieve that this relative frequency becomes approximately p_k (given that the number of samples is large enough) – simply by specifying the probability with which the Markov chain should remain within the subspace of the support of $P(X)$ where X_{i_k} (or, if w_{R_k} is negative, $\neg X_{i_k}$) is satisfied once it reaches a state where X_{i_k} ($\neg X_{i_k}$) is satisfied. Technically, if X_{i_k} ($\neg X_{i_k}$) was satisfied in the previous step, it is *reselected* for addition to the set M of formulas to satisfy in the current step with probability $1 - e^{-w_{R_k}}$ (respectively $1 - e^{w_{R_k}}$). We can essentially simulate this (without knowing the actual probability with which reselection should occur) by keeping track of the relative frequency with which $X_{i_k} =$ *true* indeed appeared in the Markov chain and, if X_{i_k} was satisfied in the previous step, adding X_{i_k} to M whenever the relative frequency is, thus far, lower than p_k. Inversely, if X_{i_k} was not satisfied in the previous step, we add $\neg X_{i_k}$ to M if the relative frequency of X_{i_k} is greater than p_k. We therefore simulate the effect of w_{R_k} by splitting its impact across the formulas X_{i_k} and $\neg X_{i_k}$.[3] Because the resulting algorithm effectively integrates posterior probability constraints into MC-SAT, we refer to it as *MC-SAT-PC* (see Algorithm 1). We invoke the algorithm with

[3] In general, F with weight $\log(p/(1-p))$ is equivalent to the combination of F with weight $\log(p)$ and $\neg F$ with weight $\log(1-p)$.

Algorithm 1 . MC-SAT-PC(*groundFormulas, probConstraints, numSamples*)

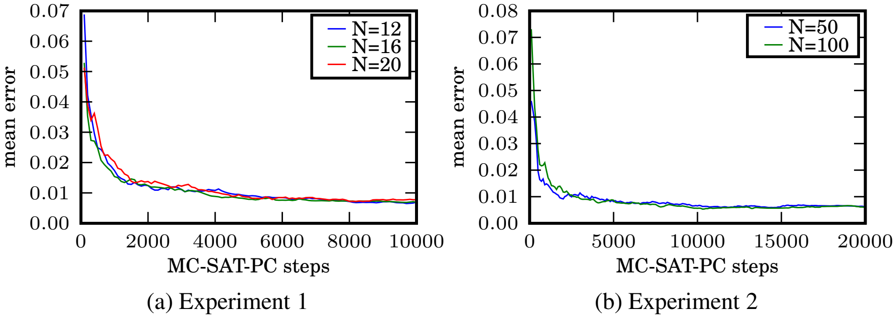
```
$n_k \leftarrow 0$ for all $\langle F_k, p_k \rangle$ in probConstraints
for $i \leftarrow 0$ to $numSamples - 1$ do
    if $i = 0$ then
        $M \leftarrow \{F_k \mid \langle F_k, w_k \rangle \in groundFormulas \wedge w_k$ is hard$\}$
    else
        $M \leftarrow \emptyset$
        for all $\langle F_k, w_k \rangle \in groundFormulas$ do
            if $x^{(i-1)} \models F_k$ then with probability $1 - e^{-w_k}$ add $F_k$ to $M$
        for all $\langle F_k, p_k \rangle \in probConstraints$ do
            if $x^{(i-1)} \models F_k$ then
                if $n_k / i < p_k$ then add $F_k$ to $M$
            else
                if $1 - n_k / i < 1 - p_k$ then add $\neg F_k$ to $M$
    sample $x^{(i)} \sim Uniform(\mathrm{SAT}(M))$
    for all $\langle F_k, p_k \rangle \in probConstraints$ do
        if $x^{(i)} \models F_k$ then $n_k \leftarrow n_k + 1$
```

G as the first argument and $\{\langle X_{i_k}, p_k \rangle \mid R_k \in \mathcal{R}\}$ as the second. Please note that the algorithm is not restricted to constraints on atomic formulas; arbitrary factorizations of the soft evidence could be considered.

5 Evaluation

In the following, we evaluate the performance of MC-SAT-PC by comparing it to IPFP-M, noting that IPFP-M belongs to a class of methods that could presently be considered as the methods of choice whenever the set X_E is not extremely small.

In our first experiment, we compare the results of IPFP-M with exact inference as the underlying inference algorithm (IPFP-M[exact]) to MC-SAT-PC. We randomly generated Markov random fields (MRFs) of three different sizes ($N \in \{12, 16, 20\}$), ten of each size. For an MRF with N boolean variables, we created N random features by generating clauses with a random number of literals (uniformly sampled from $\{1, \dots, 10\}$), drawing weights from $\mathcal{N}(0, (\log(50)/2)^2)$. For each MRF, we randomly chose $N/2$ variables as the set X_E of (soft) evidence variables, sampling beliefs uniformly from $[0, 1]$. We ran IPFP-M[exact] on the inference problems thus obtained, terminating the procedure as soon as the maximum absolute error across all soft evidence variables dropped below $\varepsilon = 0.001$. Typically, IPFP-M[exact] terminated after at most 5 rounds over all constraints (i.e. $5N/2$ iterations and therefore $5N/2$ inference runs). In each iteration, we inferred not only the probability of the constraint to be fitted but the probabilities of all constraints in order to determine convergence, as well as the probabilities of all unobserved variables, $P(X_i \mid s_E)$ for all $X_i \in X_U$, as queries. We ran MC-SAT-PC with the same queries, comparing the results to the ones obtained by IPFP-M[exact] every 100 steps. As shown in Figure 1a, MC-SAT-PC quickly produces results that are, on average, within 0.01 of the true results. After 10000 steps, the maximum error in any query variable across all 30 experiments was 0.035. (The error margins are equivalent to



(a) Experiment 1

(b) Experiment 2

Fig. 1. Mean error in posterior marginals of variables in X_U computed by MC-SAT-PC with respect to results computed by (a) IPFP-M[exact], averaged across all 10 experiments, and (b) IPFP-M[MC-SAT]. N is the number of random variables considered in the respective (series of) experiments.

what one typically obtains with MC-SAT.) These results indicate that, MC-SAT-PC is, in general, a suitable method for the approximation of posterior beliefs in the presence of soft evidence.

Exact inference is infeasible in all but the smallest domains (IPFP-M[exact] took over 80 minutes to converge for $N = 20$), and many approximate inference algorithms fail in the presence of deterministic constraints; MC-SAT is one of very few choices available. In our second experiment, we therefore compare MC-SAT-PC to IPFP-M with MC-SAT as the underlying inference algorithm (IPFP-M[MC-SAT]) in order to evaluate potential time savings. We generated random MRFs of sizes $N = 100$ and $N = 50$. For each experiment, we randomly selected 20 soft evidence variables, and, as before, used the remaining variables as query variables. Owing to the stochasticity of MC-SAT, we had to relax IPFP-M's termination criterion: We stopped the procedure as soon as the mean absolute error across all soft evidence variables dropped below $\varepsilon_1 = 0.01$ and the the maximum error dropped below $\varepsilon_2 = 0.05$. In each iteration of IPFP-M[MC-SAT], we ran MC-SAT for 10000 steps. Figure 1b summarizes the main result: MC-SAT-PC produces roughly the same results; yet it does so in a fraction of the time. While IPFP-M[MC-SAT] ran for 9.5 and 76.6 minutes (3 and 7 rounds) for $N = 50$ and $N = 100$ respectively, MC-SAT-PC drew 20000 samples in less than 35 seconds in each case. The maximum deviation from the results computed by IPFP-M was 0.024 and therefore within the expected range.

We also tested MC-SAT-PC on a complex real-world problem (which in fact provided the initial motivation for the design of

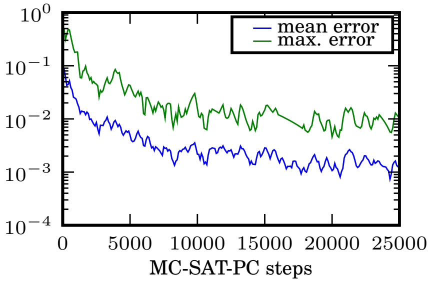


Fig. 2. Experiment 3: Mean and maximum errors across all soft evidence variables for an instance of an MLN for object identity resolution (logarithmic scale); final errors are 0.0014 and 0.0103 respectively.

MC-SAT-PC): temporal object identity resolution, where the goal is to manage associations between observations and (not necessarily uniquely identifiable) objects over time – based on the similarity in appearance between observed entities and models of the objects and based on information on previous observations. We created a Markov logic network for this problem, which is applied iteratively as new observations come in. Previously computed beliefs are provided as soft evidence for a subsequent application of the model (as are beliefs indicating degrees of similarity). An exemplary ground instance of this problem contained 279 random variables and 2688 features; 42 variables were given as soft evidence, 36 as hard evidence. An application of IPFP-M[MC-SAT] to a problem of this size is essentially infeasible, as far too many iterations are required, each of which involves only insignificantly less effort than a full application of MC-SAT-PC. Therefore, because we do not have any results on the unobserved variables to compare against, we plot in Figure 2 the number of steps taken by MC-SAT-PC against the mean and maximum errors we obtain for the soft evidence variables X_E, which (if $|X_E|$ is large enough) are fairly good indicators of convergence. The errors drop fairly quickly; and the results we obtained for X_U were consistent with our common sense expectations.

As with any MCMC method, there can be cases where MC-SAT-PC may take a long time to converge. If, for example, the probability indicated by the model for some X_{i_k} is very low (given the remaining evidence), e.g. 10^{-4}, but R_k requires it to be substantially higher, then a large number of steps may be required for the Markov chain to enter and appropriately explore the subspace where $X_{i_k} = \mathit{true}$. Of course, in such cases, IPFP-M will also have to use a rather large number of steps in the underlying inference algorithm, as we are required to compute an approximately correct non-zero probability for X_{i_k} if Equation 6 is to be applicable and have the desired effect.

6 Conclusion

In this work, we presented MC-SAT-PC, an MCMC algorithm for soft evidential update, which effectively integrates (posterior) probability constraints into the inference procedure.[4] We described its relation to IPFP-based methods and demonstrated that, particularly for hard problems involving a large number of evidence variables, it can improve performance by orders of magnitude without sacrificing accuracy. We presented our methods within the framework of Markov logic networks, yet our results carry over to other formalisms that are subsumed by MLNs (including Bayesian networks and statistical relational models based upon them).

Acknowledgements. This work was partly supported within the DFG cluster of excellence CoTeSys (Cognition for Technical Systems).

References

1. Kim, Y.G., Valtorta, M.: Soft Evidential Update for Communication in Multiagent Systems and the Big Clique Algorithm (2000)

[4] Our implementations of both MC-SAT-PC and IPFP-M are freely available as part of the ProbCog toolbox: http://ias.cs.tum.edu/research/probcog

2. Richardson, M., Domingos, P.: Markov Logic Networks. Mach. Learn. 62, 107–136 (2006)
3. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
4. Li, X.: On the Use of Virtual Evidence in Conditional Random Fields. In: EMNLP 2009: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Morristown, NJ, USA, pp. 1289–1297. Association for Computational Linguistics (2009)
5. Peng, Y., Ding, Z.: Modifying Bayesian Networks by Probability Constraints. In: Proceedings of the 24th Conference on Uncertainty in AI (UAI), pp. 26–29 (2005)
6. Jain, D., Kirchlechner, B., Beetz, M.: Extending Markov Logic to Model Probability Distributions in Relational Domains. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) KI 2007. LNCS (LNAI), vol. 4667, pp. 129–143. Springer, Heidelberg (2007)
7. Pan, R., Peng, Y., Ding, Z.: Belief Update in Bayesian Networks Using Uncertain Evidence. In: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2006), pp. 441–444 (2006)
8. Langevin, S., Valtorta, M.: Performance Evaluation of Algorithms for Soft Evidential Update in Bayesian Networks: First Results. In: Greco, S., Lukasiewicz, T. (eds.) SUM 2008. LNCS (LNAI), vol. 5291, pp. 284–297. Springer, Heidelberg (2008)
9. Poon, H., Domingos, P.: Sound and Efficient Inference with Probabilistic and Deterministic Dependencies. In: AAAI. AAAI Press, Menlo Park (2006)
10. Wei, W., Erenrich, J., Selman, B.: Towards Efficient Sampling: Exploiting Random Walk Strategies. In: AAAI, pp. 670–676 (2004)

Strongly Solving Fox-and-Geese on Multi-core CPU

Stefan Edelkamp and Hartmut Messerschmidt

TZI, Universität Bremen, Germany

Abstract. In this paper, we apply an efficient method of solving two-player combinatorial games by mapping each state to a unique bit in memory. In order to avoid collisions, such perfect hash functions serve as a compressed representation of the search space and support the execution of an exhaustive retrograde analysis on limited space. To overcome time limitations in solving the previously unsolved game *Fox-and-Geese*, we additionally utilize parallel computing power and obtain a linear speed-up in the number of CPU cores.

1 Introduction

Strong computer players for combinatorial games like *Chess* [2] have shown the impact of advanced AI search engines. For many games they play on expert and world championship level, sometimes even better. Some games like *Checkers* [7] have been decided, in the sense that the solvability status of the initial state has been computed.

In this paper we strongly solve *Fox-and-Geese* (*Fuchs-und-Gänse*[1]), a challenging two-player zero-sum game. To the authors knowledge, *Fox-and-Geese* has not been solved yet.

Fox-and-Geese belongs to the set of asymmetric strategy games played on a cross shaped board. The lone fox attempts to capture the geese, while the geese try to block the fox, so that it cannot move.

The first probable reference to an ancestor of the game is that of *Hala-Tafl*, which is mentioned in an Icelandic saga and which is believed to have been written in the 14th century. According to various Internet sources, the chances for 13 geese are assumed to be an advantage for the fox, while for 17 geese the chances are assumed to be roughly equal.

The game requires a strategic plan and tactical skills in certain battle situations[2]. The portions of tactic and strategy are not equal for both players, such that a novice often plays better with the fox than with the geese. A good fox detects weaknesses in the set of geese (unprotected ones, empty vertices, which are central to the area around) and moves actively towards them. Potential decoys, which try to lure the fox out of its burrow have to be captured early enough. The geese have to work together in form of a swarm and find a compromise between risk and safety. In the beginning it is recommended to choose safe moves, while to the end of the game it is recommended to challenge the fox to move out in order to fill blocked vertices.

[1] http://en.wikipedia.org/w/index.php?title=Fox_games&oldid=366500050

[2] An online game can be played at http://www.osv.org/kids_zone/foxgeese/index.html

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 291–298, 2010.

2 Preliminaries

The game Fox-and-Geese is played on a cross-shaped (peg solitaire) board consisting of a 3 × 3 square of intersections in the middle with four 2 × 3 areas adjacent to each face of the central square. One board with the initial layout is shown in Fig. 1. Pieces can move to any empty intersection around them (also diagonally). The fox can additionally jump over a goose to devour it. Geese cannot jump. The geese win if they surround the fox so that it can neither jump nor move. The fox wins if it captures enough geese that the remaining geese cannot block him.

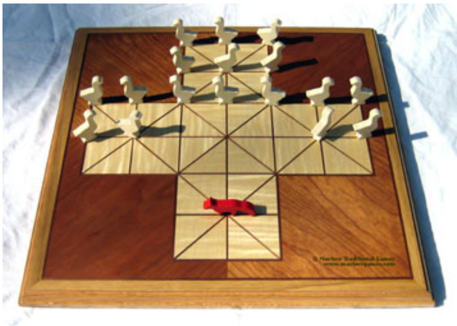


Fig. 1. Initial State of the Two-Player Turn-Taking Game *Fox-and-Geese*

We consider *strongly solving* a game in the sense of creating an optimal player for every possible initial state. This is achieved by computing the game-theoretical value of each state, so that the best possible action can be selected by looking at all possible successor states. For two-player games the value is the best possible reward assuming that both players play optimally.

For analyzing the state space, we utilize a bitvector that covers the solvability information of all possible states. Moreover, we apply symmetries to reduce the time- and space-efficiency of our algorithm. Besides the design of efficient perfect hash functions that apply to a wide selection of games, one important contribution of the paper is to compute successors states on multiple CPU cores.

3 Bitvector State Space Search

Our approach is based on minimal perfect hashing [8]. Perfect hash functions are injective mappings of the set of reachable states to a set of available indices. They are invertible, if the state can be reconstructed given the index. A perfect hash function is minimal if it is bijective.

Ranking maps a state to a unique number, while *unranking* reconstructs a state given its rank. One application of the ranking and unranking functions is to compress and decompress a state.

3.1 Two-Bit Breadth-First Search

Cooperman and Finkelstein [3] showed that, given a perfect and invertible hash function, two bits per state are sufficient to perform a complete breadth-first exploration of the search space. Two-bit breadth-first search has first been used to enumerate so-called *Cayley Graphs* [3]. As a subsequent result the authors proved an upper bound to solve every possible configuration of *Rubik's Cube* [6]. By performing a breadth-first search over subsets of configurations in 63 hours together with the help of 128 processor cores and 7 TBs of disk space it was shown that 26 moves always suffice to unscramble it. More recently, Korf [4] has applied two-bit breadth-first search to generate the state spaces for hard instances of the *Pancake* problem I/O-efficiently, and, together with Breyer, he has extended the idea to construct compressed pattern databases having 1.6 bits [1].

In the two-bit breadth-first search algorithm every state is looked at once per layer. The two bits encode values in $\{0, \dots, 3\}$ with value 3 representing an unvisited state, and values 0, 1, or 2 denoting the current search depth *mod* 3. This suffices to distinguish generated and visited states from ones expanded in the current breadth-first search layer.

3.2 Two-Bit Retrograde Analysis

Retrograde analysis classifies the entire set of positions in backward direction, starting from won and lost terminal ones. Partially completed retrograde analyses have been used in conjunction with forward-chaining game playing programs as endgame databases. Moreover, large endgame databases are usually constructed on disk for an increasing number of pieces. Since captures are non-invertible moves, a state to be classified refers only to successors that have the same number of pieces (and thus are in the same layer), and to ones that have a smaller number of pieces (often only one less).

The retrograde analysis algorithm works for all games, where the game positions can be divided into different layers, and the layers are ordered in such a way that movements are only possible within the same layer or from a higher layer to a lower one. Here each layer is defined by the number of geese on the game board. The rank and unrank functions are different for each layer. In the algorithm rank and unrank stands for the rank, unrank function for the given layer, while the suffix -smaller indicates, that the rank and unrank function belongs to a lower layer. Accordingly the expand-smaller function returns the successors in a lower layer, while expand-equal returns the successors in the same layer.

Retrograde analysis for zero-sum games requires 2 bits per state for executing the analysis on a bitvector representation of the search space: denoting if a state is unsolved, if it is a draw, if it is won for the first, or if it is won for the second player. Additional state information determines the player to move next.

Bit-state retrograde analysis applies backward BFS starting from the states that are already decided. Algorithm 1 shows an implementation of the retrograde analysis in pseudo code. For the sake of simplicity, in the implementation we look at two-player zero-sum games that have no draw. (For including draws, we would have to use the unused value 3, which shows, that two bits per state are still sufficient.) Based on the players' turn, the state space is in fact twice as large as the mere number of possible game positions. The bits for the first player and the second player to move are interleaved, so that it can be distinguished by looking at the *mod 2* value of a state's rank.

Under these conditions it is sufficient to do the lookup in the lower layers only once during the computation of each layer. Thus the algorithm is divided into three parts. First an initialization of the layer (lines 4 to 8), here all positions that are won for one of the players are marked, a 1 stands for a victory of player one and a 2 for one of player two. Second a lookup of the successors in the lower layer (lines 9 - 18) is done, and at last an iteration over the remaining unclassified positions is done in lines 19 - 34. In the third part it is sufficient to consider only successors in the same layer.

In the second part, a position is marked won, if it has a successor that is won for the player to move, here (line 10) $\mathit{even}(i)$ checks who is the active player. If there is no winning successor, the position remains unsolved. Even if all successors in the lower layer are lost, the position remains unsolved. A position is marked lost only in the third part of the algorithm, because not until then it is known how all successors are marked. If there are no successors in the third part, then the position is also marked as lost, because it has either only loosing successors in the lower layer, or no successor at all.

In the following it is shown that the algorithm indeed behaves as expected. The runtime is determined by the number of iterations that dictate work for the passes over the bitvector including ranking, unranking and lookups. A winning strategy means that one player can win from a given position no matter how the other player moves.

Theorem 1. *(Correctness) In Algorithm 1 a state is marked won, if and only if there exists a winning strategy for this state. A state is marked lost, if and only if it is either a winning situation for the opponent, or all successors are marked won for the opponent.*

Proof. The proof is done by induction over the length of the longest possible path, that is the maximal number of moves to a winning situation. As only two-player zero-sum games are considered, a game is lost for one player if it is won for the opponent, and, as the turns of both players alternate, the two statements must be shown together.

The algorithm marks a state with 1 if it assumes it is won for player one and with 2 if it assumes it is won for player two. Initially all positions with a winning situation are marked accordingly, therefore for all paths of length 0 it follows that a position is marked with 1, 2, if and only if it is won for player one, two, respectively. Thus for both directions of the proof the base of the induction holds.

The induction hypothesis for the first direction is as follows: for all non-final states x with a maximal path length of $n-1$ it follows that:

1. If x is marked as 1 and player one is the player to move, then there exists a winning strategy for player one.
2. If x is marked as 2 and player one is the player to move, then all successors of x are won for player two.
3. If x is marked as 2 and player two is the player to move, then there exists a winning strategy for player two.
4. If x is marked as 1 and player two is the player to move, then all successors of x are won for player one.

W.l.o.g. player one is the player to move; the cases for player two are analogously.

Algorithm 1. Two-Bit-Retrograde(m, *lost*, *won*)

```
 1: for all $i := 0, \ldots, m-1$ do
 2:    $Solved[i] := 0$
 3: for all $i := 0, \ldots, m-1$ do
 4:    if $won(unrank(i))$ then
 5:       $Solved[i] := 1$
 6:    if $lost(unrank(i))$ then
 7:       $Solved[i] := 2$
 8:    if $Solved[i] = 0$ then
 9:       $succs\text{-}smaller := expand\text{-}smaller(unrank(i))$
10:       if $\mathrm{even}(i)$ then
11:          for all $s \in succs\text{-}smaller$ do
12:             if $Solved[rank\text{-}smaller(s)] = 2$ then
13:                $Solved[i] := 2$
14:       else
15:          for all $s \in succs\text{-}smaller$ do
16:             if $Solved[rank\text{-}smaller(s)] = 1$ then
17:                $Solved[i] := 1$
18: while ($Solved$ has changed) do
19:    for all $i := 0, \ldots, m-1$ do
20:       if $Solved[i] = 0$ then
21:          $succs\text{-}equal := expand\text{-}equal(unrank(i))$
22:          if $\mathrm{even}(i)$ then
23:             $allone := true$
24:             for all $s \in succs\text{-}equal$ do
25:                if $Solved[rank(s)] = 2$ then
26:                   $Solved[i] := 2$
27:                $allone := allone \,\&\, (Solved[rank(s)] = 1)$
28:             if $allone$ then
29:                $Solved[i] := 1$
30:          else
31:             $alltwo := true$
32:             for all $s \in succs\text{-}equal$ do
33:                if $Solved[rank(s)] = 1$ then
34:                   $Solved[i] := 1$
35:                $alltwo := alltwo \,\&\, (Solved[rank(s)] = 2)$
36:             if $alltwo$ then
37:                $Solved[i] := 2$
```

Assume that x is marked as 1 and the maximal number of moves from position x are n. Then there exists a successor of x, say x', that is also marked as 1. There are two cases how a state can be marked as 1, x' is in a lower layer (lines 17,18) or in the same layer (lines 32,33). In both cases the maximal number of moves from x' is less than n, therefore with the induction hypothesis it follows that all successors of x' are won for player one, therefore there is a winning strategy for player one starting from state x.

Otherwise, if a state x is marked as 2 and the maximal number of moves from position x are n, then there is only one possible way how x was marked by the algorithm (line 34), and it follows that all successors of x are marked with 2, too. Again it follows that there exists a winning strategy for all successors of x, and therefore they are won for player two. Together the assumption follows. The other direction is done quite similar, here from a winning strategy it follows almost immediately that a state is marked. For all paths of length less than n from a state x it follows that:

1. If there exists a winning strategy for player one and player one is the player to move, then x is marked as 1.
2. If all successors of x are won for player two and player one is the player to move, then x is marked as 2.
3. If there exists a winning strategy for player two and player two is the player to move, then x is marked as 2.
4. If all successors of x are won for player one and player two is the player to move, then x is marked as 1.

Assume that the maximal path length from a state x is n. If there exists a winning strategy for player one from x, then this strategy states a successor x' of x such that all successors of x' are won for player one, or x' is a winning situation for player one. In both cases it follows that x' is marked with 1 and therefore x is marked with 1 as well (line 17 or 34).

On the other hand, if all successors of x are won for player two. The successors in the lower layer do not effect the value of x because only winning successors change it (lines 16, 17). In line 31 alltwo is set to true and as long as there are only loosing successors which are marked with 2 by induction hypothesis, it stays true, and therefore x is marked with 2 in line 37, too.

4 Multi-core Parallelization

Modern computers have several cores, and efficient parallelizations reduce the run-time of algorithms by the distribution of the workload to concurrently running threads.

Let S_p be the set of all possible positions in *Fox-and-Geese* with p pieces on a board of size n. The arrangement of geese and the position of the fox together with the player's turn uniquely address a state in the game.

We can partition the state space S_p into disjoint sets $S_{p,0} \cup \ldots \cup S_{p,n-1}$, where the second index is the position of the fox. As $|S_{p,i}| \leq \binom{n-1}{p}$ for all $i \in \{0, \ldots, n-1\}$, an upper bound on the number of reachable states with p pieces is $n \cdot \binom{n-1}{p}$. These states are to be classified in parallel.

In parallel two-bit retrograde (BFS) analysis layers are processed in partition form with increasing number of geese. The fix point iteration to determine the solvability status in one BFS level is the most time consuming computation. Here, we apply a multi-core CPU parallelization. In each layer, n threads (one for each fox location) are created and joined after completion. They share the same hash function.

For improving the space consumption we urge the exploration to flush the sets $S_{p,i}$ whenever possible and to load only the ones needed for the current computation. In the retrograde analysis of *Fox-and-Geese* the access to positions with a smaller number of pieces S_{p-1} is only needed during the initialization phase. As such initialization is a simple scan through a level we only need one set $S_{p,i}$ at a time. To save space for the fix point iteration, we release the memory needed to store the previous layer. As a result, the maximum number of bits needed is $\max_p\{|S_p|, |S_p|/n + |S_{p-1}|)$.

5 Experiments

Table 1 shows the results in strongly solving the game, where we applied parallel retrograde analysis for a increasing number of geese. By using this partitioning the entire analysis stayed in RAM. In fact, we observed that the exploration in the largest problem with 16 geese consumed main memory in the order of about 9.2 GB. The column Time real stands for the time and Time User is the sum of the CPU times.

Table 1. Retrograde Analysis Results for *Fox-and-Geese*

Geese	States	Space	Iterations	Won	Time Real	Time User
1	2,112	264 B	1	0	0.05s	0.08s
2	32,736	3.99 KB	6	0	0.55s	1.16s
3	327,360	39 KB	8	0	0.75s	2.99s
4	2,373,360	289 KB	11	40	6.73s	40.40s
5	13,290,816	1.58 MB	15	1,280	52.20s	6m24s
6	59,808,675	7.12 MB	17	21,380	4m37s	34m40s
7	222,146,996	26 MB	31	918,195	27m43s	208m19s
8	694,207,800	82 MB	32	6,381,436	99m45s	757m0s
9	1,851,200,800	220 MB	31	32,298,253	273m56s	2,083m20s
10	4,257,807,840	507 MB	46	130,237,402	1,006m52s	7,766m19s
11	8,515,615,680	1015 MB	137	633,387,266	5,933m13s	46,759m33s
12	14,902,327,440	1.73 GB	102	6,828,165,879	4,996m36s	36,375m09s
13	22,926,657,600	2.66 GB	89	10,069,015,679	5,400m13s	41,803m44s
14	31,114,749,600	3.62 GB	78	14,843,934,148	5,899m14s	45,426m42s
15	37,337,699,520	4.24 GB	73	18,301,131,418	5,749m6s	44,038m48s
16	39,671,305,740	4.61 GB	64	20,022,660,514	4,903m31s	37,394m1s
17	37,337,699,520	4.24 GB	57	19,475,378,171	3,833m26s	29,101m2s
18	31,114,749,600	3.62 GB	50	16,808,655,989	2,661m51s	20,098m3s
19	22,926,657,600	2.66 GB	45	12,885,372,114	1,621m41s	12,134m4s
20	14,902,327,440	1.73 GB	41	8,693,422,489	858m28s	6,342m50s
21	8,515,615,680	1015 MB	36	5,169,727,685	395m30s	2,889m45s
22	4,257,807,840	507 MB	31	2,695,418,693	158m41s	1,140m33s
23	1,851,200,800	220 MB	26	1,222,085,051	54m57	385m32s
24	694,207,800	82 MB	23	477,731,423	16m29s	112m.35s
25	222,146,996	26 MB	20	159,025,879	4m18s	28m42s
26	59,808,675	7.12 MB	17	44,865,396	55s	5m49s
27	13,290,816	1.58 MB	15	10,426,148	9.81s	56.15s
28	2,373,360	289 KB	12	1,948,134	1.59s	6.98s
29	327,360	39 KB	9	281,800	0.30s	0.55s
30	32,736	3.99 KB	6	28,347	0.02s	0.08s
31	2,112	264 B	5	2001	0.00s	0.06s

The first three levels do not contain any state won for the geese, which matches the fact that four geese are necessary to block the fox (at the middle border cell in each arm of the cross). We observe that after a while, the number of iterations shrinks for a raising number of geese. This matches the experience that with more geese it is easier to block the fox. Recall that all potential draws (that could not been proven won or lost by the geese) are devised to be a win for the fox. The transition, where the fox looses more than 50% of the game is reached at about 15–16 geese. This confirms practical observations that 13 geese are insufficient to win.

The total run-time of about a month for the experiment is considerable. Without multi-core parallelization, we estimated that more than 7 month would have been needed to complete the experiments. Even though we parallelized only the iteration stage of the algorithm, the speed-up on the 8-core machine is larger than 7, showing an almost linear speed-up.

The total space needed for operating our optimal player is about 34 GB, so that in case a goose is captured data is reloaded from disk.

6 Conclusion

In this work we applied linear-time ranking and unranking functions for games in order to execute retrograde analysis on sparse memory. We reflected that such constant-bit state space traversal to solve games is applicable, only if invertible and perfect hash functions are available. The approach features parallel explicit-state traversal of one challenging game on limited space. We studied the application of multiple-core computation and accelerated the analysis. In our experiments, the CPU speed-up is linear in the number of cores. For this we exploited independence in the problem, using an appropriate projection function. The speed-ups compare well with alternative results on parallel search on multiple cores, e.g. [5,9].

References

1. Breyer, T., Korf, R.E.: 1.6-bit pattern databases. In: AAAI (to appear 2010)
2. Campbell, M., Hoane Jr., A.J., Hsu, F.: Deep blue. Artificial Intelligence 134(1-2), 57–83 (2002)
3. Cooperman, G., Finkelstein, L.: New methods for using Cayley graphs in interconnection networks. Discrete Applied Mathematics 37/38, 95–118 (1992)
4. Korf, R.E.: Minimizing disk I/O in two-bit-breath-first search. In: National Conference on Artificial Intelligence (AAAI), pp. 317–324 (2008)
5. Korf, R.E., Schultze, T.: Large-scale parallel breadth-first search. In: National Conference on Artificial Intelligence (AAAI), pp. 1380–1385 (2005)
6. Kunkle, D., Cooperman, G.: Twenty-six moves suffice for Rubik's cube. In: International Symposium on Symbolic and Algebraic Computation (ISSAC), pp. 235–242 (2007)
7. Schaeffer, J., Björnsson, Y., Burch, N., Kishimoto, A., Müller, M.: Solving checkers. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 292–297 (2005)
8. Sulewski, D., Edelkamp, S., Yücel, C.: Perfect hashing for state space search on the GPU. In: International Conference on Automated Planning and Scheduling, ICAPS (2010)
9. Zhou, R., Hansen, E.A.: Parallel structured duplicate detection. In: National Conference on Artificial Intelligence (AAAI), pp. 1217–1222 (2007)

The Importance of Statistical Evidence for Focussed Bayesian Fusion

Jennifer Sander[1], Jonas Krieger[1], and Jürgen Beyerer[1,2]

[1] Lehrstuhl für Interaktive Echtzeitsysteme, Institut für Anthropomatik,
Karlsruher Institut für Technologie, Adenauerring 4, 76131 Karlsruhe
[2] Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB,
Fraunhoferstraße 1, 76131 Karlsruhe
jennifer.sander@kit.edu,
juergen.beyerer@iosb.fraunhofer.de

Abstract. Focussed Bayesian fusion reduces high computational costs caused by Bayesian fusion by restricting the range of the Properties of Interest which specify the structure of the desired information on its most task relevant part. Within this publication, it is concisely explained how Bayesian theory and the theory of statistical evidence can be combined to derive meaningful focussed Bayesian models and to rate the validity of a focussed Bayesian analysis quantitatively. Earlier results with regard to this topic will be further developed and exemplified.

Keywords: information fusion; Bayesian fusion; local Bayesian fusion; statistical evidence; Likelihood ratios; probability of misleading statistical evidence.

1 Introduction

If the range Z of the Properties of Interest is large and if the involved probability distributions are not efficiently representable, the solution of Bayesian fusion tasks causes high computational costs [19]. Local Bayesian fusion approaches [5,18,19,20] reduce these costs by avoiding the complete calculation of the posterior distribution. Local Bayesian fusion is inspired by criminal investigations [5]. With regard to the given fusion task, the information which has to be fused is searched for clues, i.e., for elements of Z which are better supported by the available information than others are. Then, bearing in mind the available resources, Bayesian fusion is concentrated on the identified most task relevant part U of Z.

Ignoring all elements of $Z \setminus U$ delivers a straightforward local Bayesian fusion scheme [18] which we termed focussed Bayesian fusion [6]. Usually, it is not possible to rate the validity of a focussed Bayesian model after the focussing has been done [18]. Because of this, the availableness of reliable construction rules for meaningful focussed Bayesian models is of prime importance at the research on local Bayesian fusion.

Different criteria deliver construction rules which serve this requirement: probabilistic error bounds [18], information theoretic quality indicators [19], and

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 299–308, 2010.

probability interval schemes [20]. All resulting rules deliver the same guidelines for the design of meaningful focussed Bayesian models. Additionally, by each of these rules, the validity of a focussed Bayesian analysis is ratable quantitatively with regard to a certain aspect. This publication addresses probabilistic error bounds which originate from the theory of statical evidence.

While probability in the sense of the Degree of Belief interpretation is an adequate measure for every kind of uncertainty [15,4], Likelihood ratios provide an adequate quantitative statistical evidence measure [11,17]. This has been formalized by Hacking as the Law of Likelihood already in 1965 [14]. In several more recent publications, questions concerning the reliability of observed statistical evidence and (in connection with that) the concept of misleading statistical evidence are discussed. Bounds for the probability of misleading statistical evidence, i.e., for the probability that misleading statistical evidence of a certain strength occurs are given, see for example [8,16,17]. By these bounds, the reliability of statistical evidence becomes quantifiable.

The theory of statistical evidence is not conflicting to Bayesian theory [7], rather it is implicitly an inherent part of it [12]–provided that a Degree of Belief interpretation of probability is adopted. Using the theory of statistical evidence explicitly for the task-specific design of Bayesian models seems to be self-evident. However, by default, Bayesian models are created without respect to the values which the observable quantities adopt in a given task[1].

This paper is organized as follows: Sec. 2 is a short introduction into Bayesian fusion. In Sec. 3, we demonstrate how focussed Bayesian fusion reduces the computational complexity of Bayesian fusion. Probabilistic error bounds for focussed Bayesian fusion derive themselves from the universal bound for the probability misleading statistical evidence which is applicable to every probabilistic model. In Sec. 4, we review this bound after introducing the necessary foundations from the theory of statistical evidence. The application of the concepts from Sec. 4 to focussed Bayesian fusion is done in Sec. 5 and examples for the corresponding proceeding are given in Sec. 6. These examples are much more simpler as the one to which we refer in Sec. 3 with regard to complexity reduction. We chose them intentionally because they confirm the theoretical analysis done in Sec. 5 in an easily comprehensible manner.

2 Bayesian Fusion

Let $z = (z_1, \ldots, z_N) \in Z = Z_1 \times \ldots \times Z_N$, $N \in \mathbb{N}$, denote the Properties of Interest and $d = (d_1, \ldots, d_S) \in D = D_1 \times \ldots \times D_S$, $S \in \mathbb{N}$, denote the information from several information sources. d_s stands for the contribution of information source s, $s \in \{1, \ldots, S\}$. At a given fusion task, d adopts a value according the observed information contributions while the "true" value of the Properties of Interest is unknown. At Bayesian fusion, all available information is transformed into a probabilistic representation in the sense of the Degree of

[1] See for example the discussion in [4] about the generation of a dynamic frame of discourse.

Belief interpretation. For this, all involved quantities are assumed to be random. The Bayesian theorem states how an initial Degree of Belief has to get modified to include additional knowledge in an adequate manner: it holds

$$p(z|d) \propto p(d|z)\,p(z)\ . \tag{1}$$

It is an essential advantage of Bayesian methods that–beside the contributions of the information sources which are included in the inference via the Likelihood $p(d|z)$–the posterior distribution $p(z|d)$ also reflects prior knowledge which is included in the inference via the prior distribution $p(z)$ [6].

If the information contributions are conditionally independent given z, it holds $p(d|z) = \prod_{s=1}^{S} p(d_s|z)$. In this case, it suffices at Bayesian fusion to transform each information contribution d_s individually into a source specific Likelihood $p(d_s|z)$, $s \in \{1, \ldots, S\}$. Storage costs for the saving of $p(d|z)$ get reduced. If $p(d|z)$ has to be approximated using training data, less of them will be necessary to obtain a sufficiently good approximation. The possibility to realize Bayesian fusion via a sequential fusion scheme [5] is another advantageous consequence.

3 Reduction of Computational Complexity by Focussing

The computationally complexity for the necessary operations to obtain the posterior distribution at Bayesian fusion is $\mathrm{O}(|Z|) = \mathrm{O}(\zeta^N)$, $\zeta = \sqrt[N]{\prod_{i=1}^{N} |Z_i|}$, which may be prohibitive in real world tasks. If the actual Bayesian fusion is concentrated on $U \subset Z$ with $|U| << |Z|$, the computational complexity is reduced considerably on $\mathrm{O}(|U|)$.

To clarify this practically, we refer to an example which has been given in [20]: the task of Bayesian fusion is the determination of positions, driving directions and types of cars in a scene. For this, prior knowledge from a street map, IMINT information corresponding to three images of the scene and HUMINT information are fused. The set of possible positions is discretized to approximately 1280 × 960 units. There are four possible driving directions ("south", "north", "west", "east") and five possible car types. At Bayesian fusion, formula (1) has to be evaluated for over $2 \cdot 10^7$ values of the Properties of Interest. In essence, we applied the concept of statistical evidence as described in Sec. 5 to find an adequate subset U of Z for focussed Bayesian fusion. Then, formula (1) has been evaluated only for these values of the Properties of Interest which are included in U. Here, $|U|$ constituted less than 15 percent of $|Z|$. I.e., for the actual Bayesian fusion, the range of the Properties of Interest has been cut substantially.

4 Statistical Evidence

If $p(d|z^*) > p(d|z^{**})$ holds for $z^*, z^{**} \in Z$, the observation d provides statistical evidence in support of z^* vis-a-vis z^{**} because d is more probable under the assumption that the "true" value of the Properties of Interest is z^* than it was z^{**}.

In general, a low value of $p(d|z^*)$ for a certain $z^* \in Z$ does not imply that d represents statistical evidence against z^*: the value of $p(d|z)$ may be low for all $z \in Z$ and d may provide significant statistical evidence in support of z^* compared with each other possible value of the Properties of Interest [3]. A quantitative measure of the statistical evidence which is provided by d can be obtained by pairwise comparisons of Likelihood values: for $z^*, z^{**} \in Z$, the Likelihood ratio $p(d|z^*)/p(d|z^{**})$ measures the strength of the statistical evidence that is provided by d in support of z^* vis-a-vis z^{**} [14,11,17]. Instead of communicating Likelihood ratios with respect to each possible pair of values of the Properties of Interest, the statistical evidence from d can be represented more efficiently by the use of the relative Likelihood function $r(d|z)$ which results if $p(d|z)$ gets scaled to a maximum value of one. $r(d|z)$ communicates the statistical evidence from d in support of each $z \in Z$ vis-a-vis $\arg\max_z p(d|z)$ which is the best supported hypothesis concerning the "true" value of the Properties of Interest [7,17]. An observation d provides misleading statistical evidence of the strength $p(d|z^*)/p(d|z^{**})$ in support of z^* vis-a-vis z^{**} if $p(d|z^*)/p(d|z^{**}) > 1$ holds although the "true" value of the Properties of Interest is z^{**} [17]. Even though it can be misleading, statistical evidence is a valuable concept: useful bounds for the probability of observing misleading statistical evidence of at least a certain strength exist. For $z^*, z^{**} \in Z$, $\epsilon \in (0,1)$, the universal bound delivers[2]

$$\sum\!\!\!\!\!\!\int_{d \in E} p(d|z^{**}) \leq \epsilon \ , \quad E := \{d \in D \,|\, p(d|z^{**})/p(d|z^*) \leq \epsilon\} \ , \tag{2}$$

see for example [8,17]. I.e., under the assumption that the "true" value of the Properties of Interest is z^{**}, the probability for the observation of a value of d which delivers statistical evidence of at least the strength $1/\epsilon$ in support of z^* vis-a-vis z^{**} is bounded by ϵ. The probability of observing misleading statistical evidence is relevant at planning a statistical analysis.

5 Application at Focussed Bayesian Fusion

At focussed Bayesian fusion, Z gets restricted to U on the basis of a pre-evaluation of the observed information prior to the actual Bayesian fusion. Needless to say that the resulting focussed posterior distribution will not represent completely the knowledge about z which is provided by prior knowledge and by d: by the focussing, some information will get lost [19]. Nevertheless, the quality of the focussed posterior distribution can be sufficiently high if the restriction of Z is done in a reasonable manner. The theory of statistical evidence helps to reach this goal: Z gets restricted on the set of these values of the Properties of Interest which are at most consistent with d at a certain level which is specified by ϵ, $\epsilon \in [0, 1)$, [7] if we set

$$U := \{z \in Z \,|\, r(d|z) > \epsilon\} \ . \tag{3}$$

[2] The symbol $\sum\!\!\!\!\!\!\int$ means summation with respect to discrete and integration with respect to continuous components of d.

The choice of ϵ should be conform to the available resources. If U is defined according to (3), (2) saves as error bound for focussed Bayesian fusion: the Degree of Belief that the "true" value of the Properties of Interest is not included in U is bounded by ϵ because it can be identified with the Degree of Belief that the information contributions adopt values which lead to an ignoring of the "true" value of the Properties of Interest at focussed Bayesian fusion.

The statistical evidence which is provided by one single information contribution d_s, $s \in \{1, \ldots, S\}$, is represented by $r(d_s|z)$– a quantity which does not take into account prior information and the other information contributions d_t, $t \neq s$. Hence, an isolated evaluation of the statistical evidence from each d_s is possible. This proceeding may be generally advantageous–but especially for the fusion of heterogenous information sources which have to be evaluated using extremely different kinds of expertise. Here, the analogy between criminal investigations and local Bayesian fusion becomes again obvious: a forensic expert who analyzes a specific kind of data with respect to the delinquent of one or more specific persons has to provide an analysis of the statistical evidence from this data which can subsequently be combined by an investigating detective or a court with the respective prior odds [11] and possibly additional statistical evidence. It is not the job of the forensic expert to analyze also the values of these quantities [2]. At focussed Bayesian fusion, it also makes sense to define U to consist of these values of the Properties of Interest which are at most consistent with at least one of the information contributions at a certain level, i.e.,

$$U := \{z \in Z \,|\, r(d_s|z) > \epsilon \text{ for at least one } s \in \{1, \ldots, S\}\} \tag{4}$$

with an $\epsilon \in [0, 1)$ whose value should be conform to the given resources. Especially in the case of heterogenous information sources, the information contributions may be conditionally independent. As consequence, the error bound for focussed Bayesian fusion gets significantly sharpened to ϵ^S. This fact is easily provable, e.g., by an adaption of the proof of (2) which is given in [8]. If the conditional independence assumption does not hold, the bound ϵ^S is not valid. However, it makes sense to assume that generally in this case, a more optimal bound will be lesser than the poorer–also easily derivable–bound ϵ guarantees [18].

Hence, the concepts from the theory of statistical evidence give basic guidance for the determination of U and the probabilistic error bounds rate the validity of the resulting focussed Bayesian model quantitatively.

As the example in [20] shows, it is usually not necessary to adapt the size of U exactly to the available resources. However, if such an exact adaption should be done, ideally, the precise scheme for the determination of U should be task specific. We will demonstrate this theoreticly in the rest of this section and practically in Sec. 6.

If the size of U is predefined, it is wrong to say that always all S information contributions should be evaluated to determine U in an theoreticly optimal manner. The condition "$r(d_s|z) > \epsilon$ for at least one s" in (4) means that a value z of the Properties of Interest gets ignored if its relative Likelihood with respect to all evaluated information contributions is not exceeding ϵ. If ϵ is fixed, regarding only $T < S$ information contributions for the definition of U will generally lead to a smaller size of U. Because ϵ should be chosen as low as possible according to the given resources, generally, a lower value ϵ_τ should be selected for ϵ if only $T < S$ instead of S information contributions are evaluated for defining U and, as consequence, a lower error bound for focussed Bayesian fusion may result.

However, the use of $T < S$ instead of S information contributions for the definition of U is also not always favorable if the size of U is predefined. E.g., if in the case of the conditional independence of the information contributions, the error bounds are ϵ_τ^T if $T < S$ information contributions are evaluated and ϵ_σ^S if all S information contributions are evaluated with $\epsilon_\tau < \epsilon_\sigma$, it depends on the specific values of ϵ_τ, ϵ_σ, S and T if also $\epsilon_\tau^T < \epsilon_\sigma^S$ holds.

Hence, if the size of U is predefined, the best determination scheme for U– with regard to the error bounds–depends on given fusion task. As discussed in the next two paragraphs, also the costs for the retrieval of the probabilistic representations should be always considered in practice.

In principle, basing the determination of U on the evaluation of only $T < S$ information contributions saves resources: less criteria must be checked to decide if a certain value $z \in Z$ is ignorable. Knowledge corresponding to an information contribution d_s, $s \in \{1, \ldots, S\}$, which is evaluated for the determination of U has to be transformed into a full probabilistic representation in the sense that $r(d_s|z)$ has to be determined for all $z \in Z$. In contrast, for information contributions which are used solely at the subsequent actual focussed Bayesian fusion, the Likelihood has to be determined only for $z \in U$.

On the other hand, if the transformation of all kind of information into a full probabilistic representation consumes too much resources, the information contributions may also get pre-evaluated in a suboptimal manner: also by such a suboptimal pre-evaluation, a probabilistic representation in form of Likelihoods may be obtained such that the theory of statistical evidence is applicable. For example, in [20], a suboptimal pre-evaluation of all information contributions for the determination of U is combined with a subsequent actual focussed Bayesian fusion based on the actual probabilistic information representations. However, we stress that by this proceeding, the resulting error bounds which are based on probabilistic representations of lower quality, will be generally less reliable.

6 Exemplification of Focussed Bayesian Fusion

Here, focussed Bayesian fusion is applied exemplarily at naive Bayesian classification using two data sets from the UCI Machine Learning Repository [1]: Pendigits (16 attributes, 10 classes) and Letter Recognition (16 attributes, 26 classes). The aim of the studies is an easily comprehensible demonstration of the described theoretical results. It is not disclaimed that the number of classes in the data sets may be too low therefor that the application of focussed Bayesian fusion makes sense in reality and, of course, more sophisticated classifiers will outperform our results.

Strictly speaking, the final result of a Bayesian fusion task is the posterior distribution. However, using decision theoretic concepts [4], subsequent decisions can be made on the basis of $p(z|d)$. In a Bayesian classification task, an appropriate decision is choosing the Maximum a Posteriori estimate $\arg\max_z p(z|d)$ which minimizes the expected posterior loss for a zero-one loss function [10]. The naive Bayesian classifier simplifies a Bayesian classification task significantly by assuming that the attributes d_s, $s \in \{1, \ldots, S\}$, are conditionally independent given the class z. It is well known that this classifier often has a good performance although the conditional independence does not hold and although it may not be Bayes-optimal [10]: see e.g. [9,13] and the references given therein.

Here, we implemented a simple form of the naive Bayesian classifier which has been used in [9] for its empirical evaluation: attribute values were discretized in 10 intervals of equal length, zeros in the probabilistic representations were avoided using Laplace corrections. All reported accuracies are averaged over 20 runs. For each run, the data get randomly divided into training and test data. The training data constitute 2/3 of all data. According to the calculated accuracies (Pendigits: ca. 87.81 %, Letter Recognition: ca. 70.77 %), the application of the naive Bayesian classifier is not completely beside the point for the chosen data sets and the stability of the accuracies in terms of sample standard deviations (with respect to 20 runs) is also acceptable (Pendigits: ca. 0.58 %, Letter Recognition: ca. 0.57 %).

In our first study, the decision which classes are ignored is based on the probabilistic representations used for the naive Bayesian classification. The results for Pendigits are depicted in Fig. 1. At rule 1, a class $z \in Z$ gets ignored unless it holds $r(d_s|z) > \epsilon$ for at least one $s \in \{1, \ldots, 16\}$. At rule 2, a class $z \in Z$ gets ignored unless it holds $p(d_s|z) > \epsilon$ for at least one $s \in \{1, \ldots, 16\}$. At rule 3, the ignored classes are selected randomly. At rule 1 and rule 2, different thresholds $\epsilon \in \{i/100 \,|\, i \in \{0, 1, 2, \ldots, 99\}\}$ are applied. Although rule 2 performs better than rule 3, the accuracies are conspicuously below these which are obtained by the application of rule 1 which respects the theory of statistical evidence. Even if ϵ is near to 1, rule 1 does not allow the ignoring of an unreasonable number of classes (for $\epsilon = 0.99$, ca. 33 % of the classes are ignored). The performance of rule 1 is even better than the sharper probabilistic error bound (which assumes the conditional independence of the attributes[3]) leads one to assume.

The results for Letter Recognition are comparable to these which have been obtained for Pendigits. Here, ignoring ca. 19 % of the classes according to rule 1 leads to a extremely low worsening of the accuracy of less then 1 %. For the highest considered threshold $\epsilon = 0.99$, about 55% of the classes are ignored and, by this, the accuracy gets lowered on ca. 65 %.

To demonstrate that the concepts of statistical evidence also apply if other reasonable probabilistic models are used for the determination of U than the one which is used for the actual focussed Bayesian fusion, we conducted a second study for the Pendigits data at which the attributes correspond to

[3] As noted in Sec. 5, it makes sense to assume that a more optimal bound lies between the lower bound ϵ and this bound, here.

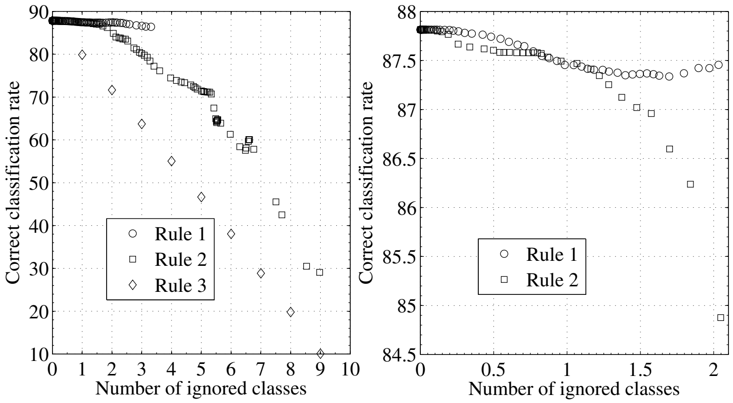


Fig. 1. Change of accuracy of naive Bayesian classification for Pendigits when classes are ignored according to rule 1, rule 2, and rule 3. For rule 1 and rule 2, each marker represents a value which corresponds to a fixed threshold (averaged over 20 runs). I.e., for theses rules, each marker relates (for a fixed threshold ϵ) the average number of classes which are ignored and the average correct classification rate over the 20 runs.

(x, y)-coordinates of hand-written digits. d_s corresponds to a x-coordinate if s is odd and to a y-coordinate if s is even, $s \in \{1, \ldots, 16\}$. By the combination of parts of the information which is delivered by d_1, d_5, and d_7, a new attribute $\tilde{d}_x$ which reports the kind of changes between some of the x-coordinates is created:

$$\tilde{d}_x := \begin{cases} 0, \, d_1 < d_5 < d_7 \\ 1, \, d_1 < d_5 \text{ and } d_7 \leq d_5 \\ 2, \, d_5 \leq d_1 \text{ and } d_5 < d_7 \\ 3, \, d_7 \leq d_5 \leq d_1 \end{cases} . \tag{5}$$

For the determination of U, $\tilde{d}_x$ is evaluated on the basis of the concepts of statistical evidence: at rule 4, a class $z \in Z$ gets ignored unless it holds $r(\tilde{d}_x|z) > \epsilon$, $\epsilon \in \{i/100 \,|\, i \in \{0, 1, 2, \ldots, 99\}\}$. The probabilistic information representation which is used by rule 4 is neither equivalent to the exact Likelihoods nor to its approximations which are used at naive Bayesian classification. However, rule 4 makes sense as Fig. 2 shows. Indeed, for larger values of ϵ, also here an absurdly large number of classes is ignored and, as consequence, the accuracy stays not longer acceptable. However, it is not astonishing that rule 1 (which considers all relative Likelihood functions with respect to the original 16 attributes) outperforms rule 4 (which relates only three of them via one attribute $\tilde{d}_x$) with regard to this aspect. For a reasonable number of ignored classes, the accuracy of rule 4 may be acceptable high. In the cases in that not more than 20 % of the classes are ignored, the accuracy reached with rule 4 even exceeds the accuracy which is reachable using rule 1. Note that the newly created attribute $\tilde{d}_x$ comprises some information with regard to the conditional dependencies between d_1, d_5, and d_7.

Parts of the information which is delivered by d_2, d_6, and d_8 has been used to create a new attribute $\tilde{d}_y$ analogously to $\tilde{d}_x$. When we analyzed the performance

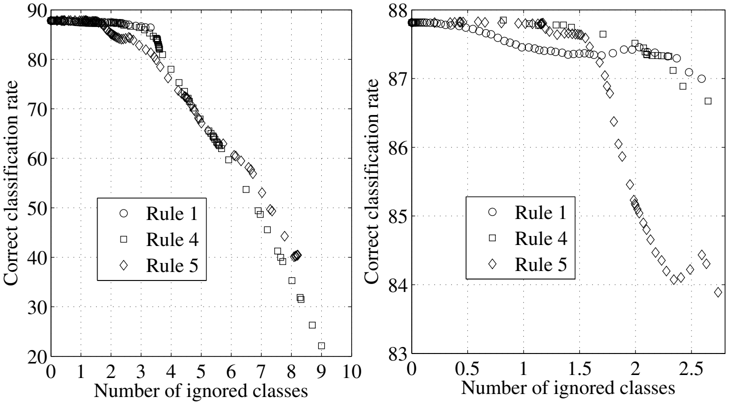


Fig. 2. Change of accuracy of naive Bayesian classification for Pendigits when classes are ignored according to rule 1, rule 4, and rule 5. Each marker represents a value which corresponds to a fixed threshold (averaged over 20 runs).

of both new attributes together, it became obvious that this is a case in that focussed Bayesian fusion based on the analysis of a smaller number of information contributions for the determination of U may be preferable if the size of U is predefined. At rule 5, a class $z \in Z$ gets ignored unless it holds $r(\tilde{d}_x|z) > \epsilon$ or $r(\tilde{d}_y|z) > \epsilon$. It becomes clear from Fig. 2 that rule 4 outperforms rule 5 if the size of U is predefined and not absurdly large. The reason of this has been identified in Sec. 5. The following exemplary numbers clarify it additionally: at rule 4, the lowest value of ϵ for that at least three classes are ignored (on average) is 0.25. Applying rule 5, the lowest value of ϵ for that at least three classes are ignored (on average) is 0.59. Even the sharper probabilistic error bound leads to the conclusion that the use of rule 4 is preferable in this situation.

7 Conclusion

The theory of statistical evidence provides concepts which are extremely valuable at the research on focussed Bayesian fusion. Their application outperforms ad hoc solutions by far. It offers rules for the exact creation of focussed Bayesian models and it provides error bounds by which the validity of a focussed Bayesian fusion is ratable. There exists no unique recipe for the exact creation of focussed Bayesian models if the size of U is predefined. However, the error bounds represent helpful mathematical tools for making this choice.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. Irvine: University of California, School of Information and Computer Science (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`

2. Aitken, C.G.G., Taroni, F.: Statistics and the Evaluation of Evidence for Forensic Scientists. Wiley, Chichester (2004)
3. Bartelborth, T.: Wofür sprechen die Daten? Journal for General Philosophy of Science 35(1), 13–40 (2004)
4. Bernardo, J.M., Smith, A.F.M.: Bayesian Theory. Wiley, Chichester (2004)
5. Beyerer, J., Heizmann, M., Sander, J.: Fuselets – an Agent Based Architecture for Fusion of Heterogeneous Information and Data. In: Dasarathy, B.V. (ed.) Proceedings of SPIE, Multisensor, Multisource Information Fusion: Architectures, Algorithms and Applications 2006, vol. 6242. SPIE, Bellingham (2006)
6. Beyerer, J., Heizmann, M., Sander, J., Gheta, I.: Bayesian Methods for Image Fusion. In: Stathaki, T. (ed.) Image Fusion: Algorithms and Applications. Academic Press, Amsterdam (2008)
7. Blume, J.D.: Likelihood Methods for Measuring Statistical Evidence. Statistics in Medicine 21(17), 2563–2599 (2002)
8. Dempster, A.P.: The Direct Use of Likelihood for Significance Testing. Statistics and Computing 7(4), 247–252 (1997)
9. Domingos, P., Pazzani, M.: On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. Mach. Learn. 29(2-3), 103–130 (1997)
10. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, New York (2001)
11. Edwards, A.W.F.: Likelihood. Cambridge University Press, London (1972)
12. Goodman, S.N.: Toward Evidence-Based Medical Statistics. 2: the Bayes Factor. Ann. Intern. Med. 130(12), 1005–1013 (1999)
13. Ekdahl, M., Koski, T.: Bounds for the Loss in Probability of Correct Classification under Model Based Approximation. J. Mach. Learn. Res. 7, 2449–2480 (2006)
14. Hacking, I.: Logic of Statistical Inference. Cambridge University Press, New York (1965)
15. Lindley, D.V.: The Probability Approach to the Treatment of Uncertainty in Artificial Intelligence and Expert Systems. Statistical Science 2(2), 3–44 (1987)
16. Royall, R.M.: On the Probability of Observing Misleading Statistical Evidence. Journal of the American Statistical Association 95(451), 760–768 (2000)
17. Royall, R.M.: Statistical Evidence: a Likelihood Paradigm. Chapman & Hall, London (1997)
18. Sander, J., Beyerer, J.: Decreased Complexity and Increased Problem Specificity of Bayesian Fusion by Local Approaches. In: Proceedings of the 11th International Conference on Information Fusion, Fusion 2008, pp. 1035–1042. IEEE Press, Los Alamitos (2008)
19. Sander, J., Heizmann, M., Goussev, I., Beyerer, J.: A Local Approach for Focussed Bayesian Fusion. In: Dasarathy, B.V. (ed.) Proceedings of SPIE, Multisensor, Multisource Information Fusion: Architectures, Algorithms and Applications 2009, vol. 7345. SPIE, Bellingham (2009)
20. Sander, J., Heizmann, M., Goussev, I., Beyerer, J.: Global Evaluation of Focussed Bayesian Fusion. In: Braun, J.J. (ed.) Proceedings of SPIE, Multisensor, Multisource Information Fusion: Architectures, Algorithms and Applications 2010, vol. 7710. SPIE, Bellingham (2010)

The Shortest Path Problem Revisited: Optimal Routing for Electric Vehicles

Andreas Artmeier, Julian Haselmayr,
Martin Leucker, and Martin Sachenbacher

Technische Universität München, Department of Informatics
Boltzmannstraße 3, 85748 Garching, Germany
{artmeier,haselmay,sachenba,leucker}@in.tum.de

Abstract. Electric vehicles (EV) powered by batteries will play a significant role in the road traffic of the future. The unique characteristics of such EVs – limited cruising range, long recharge times, and the ability to regain energy during deceleration – require novel routing algorithms, since the task is now to determine the most economical route rather than just the shortest one. This paper proposes extensions to general shortest-path algorithms that address the problem of energy-optimal routing. Specifically, we (i) formalize energy-efficient routing in the presence of rechargeable batteries as a special case of the constrained shortest path problem (CSPP) with hard and soft constraints, and (ii) present an adaption of a general shortest path algorithm (using an energy graph, i.e., a graph with a weight function representing the energy consumption) that respects the given constraints and has a worst case complexity of $O(n^3)$. The presented algorithms have been implemented and evaluated within a prototypic navigation system for energy-efficient routing.

1 Introduction

Dwindling fossil fuel reserves and the severe consequences of climate change are major challenges of the new millennium. Electric vehicles (EVs) offer a potential contribution: they can be powered by regenerative energy sources such as wind and solar power, and they can recover some of their kinetic and/or potential energy during deceleration phases. This so-called recuperation or regenerative braking increases the cruising range of current EVs by about 20 percent in typical urban settings, and often more in hilly areas. However, a more wide-spread use of EVs is still hindered by limited battery capacity, which currently allows cruising ranges of only 150 to 200 kilometers. Thus, accurate prediction of remaining cruising range and energy-optimized driving are important issues for EVs in the foreseeable future.

The unique characteristics of EVs have an impact on search algorithms used in navigation systems and route planners. With the limited cruising range, long recharge times, and energy recuperation ability of battery-powered EVs, the task is now to find energy-efficient routes, rather than just fast or short routes. This modification might appear insignificant at first glance, as it seems enough

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 309–316, 2010.

to simply exchange time and distance values with energy consumption in the underlying routing problem. But on a second glance, several new challenges surface, which require novel algorithms that go beyond existing solutions for route search in street networks.

For example, consider the simple vehicle routing problem shown in Figure 1. The nodes correspond to locations and the edges correspond to road segments. The positive (resp. negative) values at the edges denote consumption (resp. gain) of energy, which we assume is taken from (resp. stored in) a battery with maximum capacity C_{max}. There are two possible paths from source s to destination t: $(s \xrightarrow{2} x \xrightarrow{-1} t)$ and $(s \xrightarrow{-1} y \xrightarrow{2} t)$, both of which are optimal from a classical shortest-path perspective as they have the same overall cost of 1 energy unit. However, if at node s there is only $C_s = 1$ unit of energy left in the battery, the path via y becomes the only feasible path. If instead we start with a battery charged to $C_s = C_{max} \geq 2$ at node s, both paths to t are feasible, but the path via x is now preferable since it has a lower overall energy consumption (the other path via y will require 2 energy units, since recuperation is no longer possible in the segment $s \xrightarrow{-1} y$).

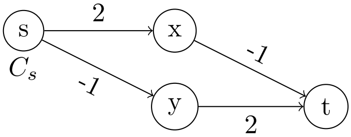


Fig. 1. A simple electric vehicle routing problem

In this work, we address the problem of finding the most energy-efficient path for battery-powered electric cars with recuperation in a graph-theoretical context. This problem is similar to the *shortest path problem* (SP), which consists of finding a path P in a graph from a source vertex s to a destination vertex t such that $c(P) = \min_{Q \in U}(c(Q))$, where U is the set of all paths from s to t, and $c : E \to \mathbb{Z}$ is a weight function on the edges E of the graph. SP is polynomial; the best known algorithm for the case of non-negative edges is Dijkstra [1] with time complexity $O(n^2)$, while in the general case, Bellman-Ford [2], [3] has $O(n^3)$. However, most commonly used SP algorithms like contraction hierarchies [4], highway hierarchies [5] and transit vertex routing [6] can't be applied to our problem because of the negative weights due to recuperation. In addition, SP does not consider the constraints that result from the discharge and recharge characteristics of the EV's battery pack, namely that it neither can be discharged below zero, nor charged above its maximum capacity. An extension of the SP, the *constrained shortest path problem* (CSPP) [7], is to find a shortest path P from s to t among all feasible paths in a graph, where a path P is called feasible if $b(P) \leq T \in \mathbb{N}$ for an additional weight function $b : E \to \mathbb{N}$. Our problem of energy-efficient routing with recuperation can be framed as such a CSPP, but CSPP is known to be NP-complete [8].

In the following, we (i) show how to model energy-efficient routing in the presence of rechargeable batteries as a special case of a CSPP, extending the shortest path problem with problem-specific hard and soft constraints, and (ii) present a novel variant of a general shortest path algorithm that respects these constraints, but still has a polynomial worst case time complexity of $O(n^3)$.

Algorithm 1. ConstrainedGenericShortestPath

```
input: A directed graph $G = (V, E)$, weight function $c : E \rightarrow \mathbb{Z}$, source vertex $s$,
       strategy $S$$\underline{\text{, maximum capacity } C_{max} \text{ and initial } U_s}$
output: A $\underline{\text{prefix bounded}}$ shortest path tree from $s$ $\underline{\text{with respect to absorb}}$
begin
    foreach vertex $v$ in $V$ do
        $d(v) \leftarrow \infty$;
        $p(v) \leftarrow$ null;
    $d(s) \leftarrow 0\underline{+U_s}$;
    $Q \leftarrow \{s\}$;
    while $Q \neq \emptyset$ do
        choose $u$ from $Q$ with strategy $S$;
        $Q \leftarrow Q \setminus \{u\}$;
        foreach successor $v$ of $u$ do
            $d' \leftarrow d(u) + c(u, v)$;
            $\underline{d' \leftarrow \max(d', 0)}$;
            if $d' < d(v)$ $\underline{\text{and } d' \leq C_{max}}$ then
                $d(v) \leftarrow d'$;
                $p(v) \leftarrow u$;
                $Q \leftarrow Q \cup \{v\}$;
end
```

2 Preliminaries

We assume that a road network can be modeled as a directed graph $G = (V, E)$ with $|V| = n$ and $|E| = m$. A *path* P of *length* $|P| = k$ is then a sequence of $k+1$ distinct vertices $(v_1, v_2, \ldots, v_{k+1})$ with $(v_i, v_{i+1}) \in E$ for all $i \in \{1, 2, \ldots, k\}$. Note that a single vertex is a path of length 0. In addition, we assume that a weight function $c : E \rightarrow \mathbb{Z}$ models the amount of energy required or gained when traveling along the edges in the network. The *weight* $c(P)$ of a path P of length k is then defined by $c(P) = \sum_{i=1}^{k} c(v_i, v_{i+1})$. A cycle $C = (v_1, v_2, \ldots, v_k, v_1)$ is called *positive* if $c(C) \geq 0$. For a road network, the absence of negative cycles corresponds to the law of conservation of energy.

Algorithm 1 ([9,10]) without the underlined statements – these show our changes to be explained in the next sections – computes a shortest path tree for a given source vertex and expansion strategy, where the shortest path tree is a directed tree that represents the solution of all shortest path problems with this source vertex, and the strategy determines the order in which vertices are processed. The algorithm extends and improves a tree defined by function p, where $p(v)$ is the predecessor of v in the tree. The distance $d(v)$ is the weight $c(P)$ of the path P from s to v in this tree. The algorithm improves these distance values $d(v)$ for $v \in V$ until the queue Q is empty, and the best path (according to the distance value) to every reachable vertex is found. Note that the algorithm does not terminate if a cycle with negative weight is reachable from the source vertex.

The time complexity of the algorithm with Dijkstra's strategy – choose the vertex with smallest distance value from Q – is known to be $O(m + n \log n) = O(n^2)$ for $m = O(n^2)$ for positive weights but exponential for the general case [11], while using the Bellman-Ford strategy – choose the vertices in a FIFO manner from Q – results in time complexity $O(n^3)$ for arbitrary weights.

3 Prefix-Bounded Shortest Paths

Our goal is to find an energy efficient route for a battery-powered EV in a road network with given energy values, modeled as a directed graph $G = (E, V)$ with weight function $c : E \to \mathbb{Z}$.

Clearly, route segments are only feasible if the required energy does not exceed the charging level of the battery. In addition, due to elevation differences or changes in the cruising speed, on each route there can be road sections where the energy consumption is negative, such that the EV's battery can be recharged. However, we also have to consider that the battery has a certain maximum capacity, beyond which recharging is no longer possible. The two conditions on the charge level of the battery—it cannot be discharged below zero, it cannot store more energy than its maximum capacity—can be viewed as *hard* and *soft constraints*, respectively, on possible routes: a route is infeasible if there is a point where the required energy exceeds the charge level, and a route is less preferred if there is a point where energy could be recuperated but the battery's maximum capacity is exceeded.

For modeling these constraints, some more definitions are needed. Let $P = (v_1, v_2, \ldots, v_k)$ with vertex $v_1 = s$ be a path in G, C_{max} the maximum battery capacity, C_{v_i} the charge level of the battery at vertex v_i, and $U_{v_i} = C_{max} - C_{v_i}$ the remaining storage capacity of the battery at vertex v_i. Then, we define the *absorption function* absorb recursively as

$$\operatorname{absorb}(P^k) = \begin{cases} U_{v_1} & \text{if } k = 1, \\ \max\left(\operatorname{absorb}(P^{k-1}) + c(v_{k-1}, v_k)\right., 0) & \text{if } k > 1 \end{cases}$$

where $P^i = (v_1, v_2, \ldots, v_i)$ is the subpath of P ending in vertex v_i (with $i \le k$). Moreover, we call path P *prefix-bounded by* C_{max}, if $\operatorname{absorb}(P^i) \le C_{max}$ holds for every $i = 1, 2, \ldots, k$.

The problem we want to solve can then be described as follows: given a start point and battery charge level, find a route to a target point that respects the constraints and where the remaining charge is highest. We call this problem the *prefix-bounded shortest path problem*, or for short PBSP. A prefix-bounded path P corresponds exactly to a route which is feasible to use $(0 \le C_{v_i}, U_{v_i} \le C_{max})$ and where $\operatorname{absorb}(P)$ is the remaining storage capacity at the end of the route. Formally, a solution to a PBSP is a path $P \in W$ such that

$$\operatorname{absorb}(P) = \min_{Q \in W}(\operatorname{absorb}(Q))$$

where W is the set of all prefix-bounded paths w from the source vertex s to the the destination vertex t for which the constraint $\mathrm{absorb}(w^i) \leq C_{max} - U_s$ holds for all $i \in \{1, 2, \ldots, (|w| + 1)\}$. Since maximizing the remaining battery charge is equivalent to minimizing the missing battery charge, we can search for a prefix-bounded shortest path tree with respect to function absorb in order to get a solution to the PBSP.

If we only consider graphs with non-negative edges, the PBSP is a special case of a CSPP, where the two weight functions b and c are equivalent. However, algorithms for CSPP are exponential in the worst case, and they do not allow for negative weights. Therefore, we instead propose an algorithm that extends the generic shortest path algorithm by taking the prefix-bound constraints into account. As we will see, the additional prefix-bound constraints don't change the time complexity of the shortest path problem, and thus this algorithm solves the problem in polynomial time ($O(n^3)$ with the Bellman-Ford strategy).

4 Computing Prefix-Bounded Shortest Path Trees

In this section, we modify the generic shortest path algorithm (see Section 2) to compute a prefix-bounded shortest path tree with respect to function absorb, that is, to solve a PBSP. The modifications are shown as underlined statements in Algorithm 1. One modification concerns the initialization of the source vertex with the difference between the maximum capacity of the battery and its current charge (line 5). The other modifications implement the soft constraint that the battery can't be overcharged (line 12) and the hard constraint prohibiting the use of road sections with insufficient energy (line 13). Detailed proofs of correctness for the algorithm and the theorems of this section can be requested from the authors.

Theorem 1. *Algorithm 1 computes for every strategy S a prefix-bounded shortest path tree with respect to function* absorb*.*

In [12], several expansion strategies are presented for solving the general shortest path problem. However, in the context of electromobility, novel ones are needed that take into account the specific energy graph properties, e.g., far fewer negative than positive edge weights. As a first variant, we propose the following expand strategy. For each vertex, a counter is initialized with 0 and incremented each time this vertex is chosen for expansion in line 8; the strategy then chooses a vertex from Q with the smallest counter. This strategy leads to the same time complexity as the FIFO strategy does for the Bellman-Ford algorithm in the general shortest path problem (see [10] for instance).

Theorem 2. *Algorithm 1 using strategy expand has time complexity $O(n^3)$.*

Lemma 1. *Algorithm 1 using strategy dijkstra has time complexity $O(n^2)$ if c is non-negative.*

Now we consider the strategy expand-distance: choose the vertex u from Q which has smallest $d(u)$ among vertices in Q which are expanded least of all. This new strategy, combining both previously presented strategies, guarantees time complexity $O(n^3)$ in any case. In addition, if c is non-negative, we get $O(n^2)$ since at the beginning of the algorithm expand$(u) = 0$ and no vertex is inserted twice into Q. Therefore the following key statement is proven.

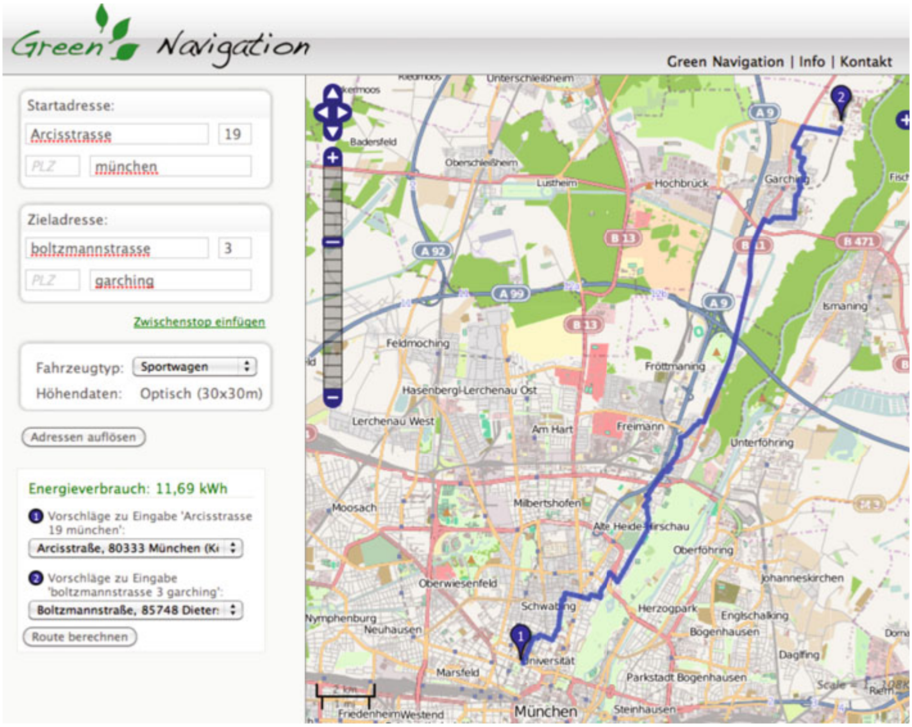
Corollary 1. *Algorithm 1 using strategy expand-distance has time complexity $O(n^3)$ for arbitrary weight function c and $O(n^2)$ if c is non-negative.* □

Since in our graph modeling the energy values there are typically only few edges with negative weight, the time complexity of the expand-distance strategy can be expected to be near $O(n^2)$, and in contrast to the conventional Dijkstra algorithm we are sure to be far away from exponential time. These two advantages are the reason why we suggest Algorithm 1 using strategy expand-distance in the context of routing for electromobility.

5 Prototypic Implementation and Experimental Results

We developed a prototypic software system for energy efficient routing, based on opensource libraries and freely available data. It is possible to access this system online (www.greennav.org). For a given car type, source address and destination address, the system computes a route with minimum energy costs. The data basis consists of the collaborative project OpenStreetMap (OSM), which aims to create and distribute freely available geospatial data, and the altitude map of the NASA Shuttle Radar Topographic Mission (SRTM), which provides digital elevation data with a resolution of about 90m. By combining these two sources, we created a road map with elevation and cruising speed information for every point in this network. The first step was then to derive a graph with weights corresponding to the energy consumption of road sections. For this purpose, we used a simplistic physical model of an EV. The consideration of recuperation induces negative weights for a small percentage of edges.

Figure 2 shows the web interface to our prototype. It is possible to choose source, destination and car type on the left side. The blue path displays the energy efficient shortest path according to the available data and vehicle model. While some of the proposed deviations from a straight (shortest) route are indeed due to energy savings, others originate from an overly simplistic vehicle model and some missing speed tags in the OSM data; future development will address these problems. Within this prototype, we evaluated Algorithm 1 using the four different strategies dijkstra, expand expand-distance, and FIFO. The evaluation was carried out on a section of the OSM map that covers the Allgäu region southwest of Munich, and contains 776,419 vertices and 1,713,900 edges. We set $U_s = 0$ and take $k = 10$ randomly selected source vertices s as input. The individual runtimes x_i were used to calculate the average time $\bar{x} = \frac{1}{k}\sum_{i=1}^{k} x_i$ (first value in columns 2-5 in Table 1) and the variance $\frac{1}{k}\sum_{i=1}^{k}(x_i - \bar{x})^2$ (second value). These calculations were done for different values of C_{max} with steps of 10



Startadresse: Arcisstrasse 19, PLZ, münchen

Zieladresse: boltzmannstrasse 3, PLZ, garching

Fahrzeugtyp: Sportwagen

Höhendaten: Optisch (30x30m)

Energieverbrauch: 11,69 kWh

1 Vorschläge zu Eingabe 'Arcisstrasse 19 münchen': Arcisstraße, 80333 München (K

2 Vorschläge zu Eingabe 'boltzmannstrasse 3 garching': Boltzmannstraße, 85748 Dieter

Fig. 2. Screenshot of our route planner prototype, showing the energy optimal route for an EV from TUM's Munich campus to TUM's Garching campus

Table 1. Runtime in seconds (and variance) for computing energy-optimal paths with algorithm ConstrainedGenericShortestPath, using four different strategies

C_{max} [kWh]	dijkstra	expand	FIFO	expand-distance	average tree size
10	0.28 / 0.00	1.20 / 1.07	0.80 / 0.19	0.29 / 0.00	19695
20	0.33 / 0.01	16.23 / 169.21	23.80 / 963.10	0.36 / 0.01	67211
30	0.44 / 0.02	> 60 / -	> 60 / -	0.54 / 0.03	136884
40	0.64 / 0.07	> 60 / -	> 60 / -	0.85 / 0.15	231475
50	0.86 / 0.13	> 60 / -	> 60 / -	1.22 / 0.35	330229
60	1.00 / 0.18	> 60 / -	> 60 / -	1.54 / 0.55	414227
70	1.13 / 0.19	> 60 / -	> 60 / -	1.84 / 0.68	486764
80	1.22 / 0.19	> 60 / -	> 60 / -	2.11 / 0.78	551194
90	1.32 / 0.16	> 60 / -	> 60 / -	2.37 / 0.68	612294
100	1.41 / 0.11	> 60 / -	> 60 / -	2.59 / 0.59	664905
infinity	1.57 / 0.03	> 60 / -	> 60 / -	3.14 / 0.17	776419

kWh. Additionally, the average size of the corresponding prefix-bounded shortest path tree (number of vertices) is presented. Our experiments were conducted on a Intel Core2 Duo CPU with 2.20 GHz and 2 GB RAM.

In Table 1 it can be seen that the strategies FIFO and expand are far away from practical usability, since we have aborted the individual executions for

$C_{max} > 20$ after one minute without any result. Strategy expand-distance is less than two times slower than dijkstra in our experiments (average of 10 computations), and moreover, the former strategy has the advantage that the worst case time complexity is polynomial.

6 Conclusion

Optimal routing for electrical vehicles with rechargeable batteries will become increasingly important in the future. In this paper, we studied this problem within a graph-theoretic context. We modeled energy-optimal routing as a shortest path problem with constraints, and proposed a family of search algorithms that respect these constraints with a worst case time complexity of $O(n^3)$; in fact, our problem constitutes a tractable variant of the more general constrained shortest path problem. Further research will study the impact of the negative/positive edge ratio in our routing graphs and the development of special tailored heuristics using the law of conservation of energy. In addition, we plan to extend our approach by modeling the energy consumption with stochastic instead of constant values for assessing the risk of running out of energy before arriving at the destination. Finally, it is interesting to extend the framework towards energy-efficient management of a fleet of EVs, for instance in car-sharing scenarios.

References

1. Dijkstra, E.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
2. Bellman, R.: On a routing problem. Quart. of Appl. Math. 16(1), 87–90 (1958)
3. Ford Jr., L.R.: Network flow theory. Technical report, RAND (1956)
4. Geisberger, R., Sanders, P., Schultes, D., Delling, D.: Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In: McGeoch, C.C. (ed.) WEA 2008. LNCS, vol. 5038, pp. 319–333. Springer, Heidelberg (2008)
5. Sanders, P., Schultes, D.: Highway Hierarchies Hasten Exact Shortest Path Queries. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 568–579. Springer, Heidelberg (2005)
6. Bast, H., Funke, S., Sanders, P., Schultes, D.: Fast routing in road networks with transit nodes. Science 316(5824), 566 (2007)
7. Joksch, H.C.: The shortest route problem with constraints. Journal of Mathematical Analysis and Applications 14, 191–197 (1966)
8. Garey, M., Johnson, D.: Computers and Intractibility: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)
9. Gallo, G., Pallottino, S.: Shortest path methods: A unifying approach. Mathematical Programming Studies, vol. 26. Springer, Heidelberg (1986)
10. Cherkassky, B.V., Goldberg, A.V., Radzik, T.: Shortest paths algorithms: Theory and experimental evaluation. Mathematical Programming 73(2) (1993)
11. Johnson, D.B.: A note on Dijkstra's shortest path algorithm. Journal of the ACM 20(3), 385–388 (1973)
12. Zhan, F.B., Noon, C.E.: A comparison between label-setting and label-correcting algorithms for computing one-to-one shortest paths. Journal of Geographic Information and Decision Analysis 4(2), 1–11 (2000)

A Systematic Testing Approach for Autonomous Mobile Robots Using Domain-Specific Languages

Martin Proetzsch[1], Fabian Zimmermann[2], Robert Eschbach[2],
Johannes Kloos[2], and Karsten Berns[1]

[1] Robotics Research Lab, TU Kaiserslautern, 67653 Kaiserslautern, Germany
{proetzsch,berns}@cs.uni-kl.de
[2] Department of Testing and Inspections, Fraunhofer IESE,
67663 Kaiserslautern, Germany
{fabian.zimmermann,robert.eschbach,johannes.kloos}@iese.fraunhofer.de

Abstract. One aspect often neglected during the development of autonomous mobile robots is the systematic validation of their overall behavior. Especially large robots applied to real-world scenarios may cause injuries or even human death and must therefore be classified as safety-critical. In this paper, a generic approach to defining and executing purposeful test runs using domain-specific languages (DSLs) is presented. Test cases can be defined in an appropriate test description language (first DSL). These test cases can be derived automatically using a model-based testing approach, for which a test model has to be created. Hence, a second DSL for the creation of the test model is presented. It is further shown how the generated test cases are automatically executed and evaluated. The paper concludes with the application of the approach to the autonomous off-road robot RAVON.

1 Motivation and Introduction

In the field of autonomous mobile robots, systematic quality assurance is made complicated by the complexity of detection and motion generation systems. Since the failure of a robot's control software can have grave consequences, some basic testing is required before putting it into operation. This testing is often done unsystematically and in an ad-hoc manner. Furthermore, test cases have to be described on a very low and technical level of abstraction. Thus, testing requires complicated programming and adaptation of test cases for new versions.

Some previous research has been conducted to assure the quality of robot control software. There have been efforts to guarantee certain properties of systems using formal verification techniques [1,2]. However, the increasing number of states for complex control systems limits this approach to relatively small subsystems. In [3] fault trees are used for the verification of a mobile robot. Here, only failures identified during fault tree analysis are considered. Further approaches use an architecture with a safety layer to avoid hazards such as collisions [4]. This works for the construction of a new robot, but changing the existing architecture of RAVON would require considerable effort.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 317–324, 2010.

Other contributions deal with performance measures [5,6]. Here, aspects such as usefulness of actions, smoothness and accuracy of paths, time to completion of tasks, and distance traveled are used for evaluating the progress made during system development. However, no information is given on how these techniques can be used to provide a generic approach for generating, executing, and evaluating test cases. A general overview of safety aspects in the field of artificial intelligence is provided in [7]. The paper at hand presents an approach for a high-level description of tests for autonomous mobile robots. It is based on domain-specific languages and leverages model-based testing. The approach is evaluated on the autonomous robot RAVON (Robust Autonomous Vehicle for Off-road Navigation).

2 Test Case Definition

2.1 Test Description Language

Defining test cases for a mobile robot in a generic way requires a test description language capable of expressing the relevant aspects. First, for a mobile robot, a list of *commands* to be executed has to be specified. As the application presented here concerns driving across unknown terrain, commands are expressed as a list of *target points* to be approached. During execution, several *events* can occur, e. g., moving obstacles or deliberatively disturbed sensor data. For a complete test run, a list of commands and events can be defined. For the chosen application, this is a *path* containing target points to reach and events occuring during the traversal.

Besides a definition of the commands, one crucial aspect during test cases is how malfunctions can be detected, i. e., a test oracle has to be defined. Since RAVON uses a behavior-based control system (cf. Section 3), no exact definition of correct system behavior exists. If RAVON should reach a target point, any sequence of actions that brings the robot to the target point within a certain period and without any collision can be seen as correct. Similar problems also occur when testing other autonomous robots. Thus, instead of defining a detailed expected response for each input (here, inputs are the coordinates of the next target point), a set of rules is given. There are two different kinds of rules, called *invariants* and *goals*. Invariants should always hold, while goals should be reached in the end. An example of an invariant is *no collision*, a possible goal is *the robot has reached the last target point*. A further distinction is made as to whether an occurring *violation* causes a test run to *terminate*. The current test case has succeeded if all goals are reached and no invariant has been violated.

These aspects are defined using a domain specific language (first DSL in our approach). This language contains a list of positions and events that are defined and referenced by indexes for defining paths. For the definition of execution and specification checks, predefined types of modules are referenced that are parameterized according to the demands. The presented test case description language is suitable for automatic generation by means of tools.

2.2 Automatic Generation of Test Cases

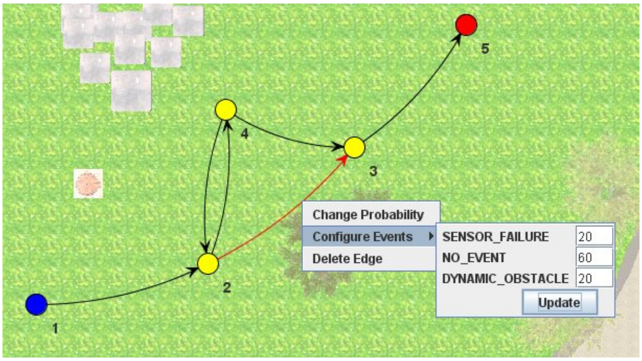
Although the language described in the last section can be used to manually define test cases, further improvement of the testing procedure can be achieved through automatic test case generation. Therefore, this paper deals with model-based statistical testing techniques (MBST) [8] for automatically deriving test cases based on a test model. The MBST approach has been evaluated in [9].

Test model. In MBST, a test model is constructed as a transition system. It contains all possible inputs to the system and usually the corresponding expected outputs. In our case of testing an autonomous robot, a state in the test model, i. e., a state in the transition system, represents a position of the robot on the map. This map is selected according to the expected area of deployment. If the area of deployment is unknown, a map containing the expected type of terrain should be chosen. A stimulus is the command to drive to a target point combined with an event that might occur while driving there. Positions where the robot should start its test drives are marked as start states. All states where a test drive might end are marked as exit states. If a transition from a state A to a state B exists, the command *drive to B* could occur in a test drive when the robot has reached target point A. For random test case generation, each transition is annotated with a probability according to a usage profile, i. e., the test model is a Markov chain.

Language for test model creation. Creating the test model is a complicated process usually done by test experts, e. g., using a technique called sequence-based specification [10,11]. A domain expert is usually not knowledgeable in the construction of test models, but has very detailed information about which configurations of the system will be especially critical. Since autonomous robots are a domain of scientific research, there are often no test experts available at all. Thus, it is recommended providing a domain-specific language that enables the domain expert to create the test model [12].

In our approach, the test model can be easily created by the roboticist as the domain expert. We provide a domain-specific language for the description of test models (the second DSL in our approach). Since the test model is a connected graph depicting the area the robot should drive through during testing, we provide an editor for drawing this graph as a domain-specific language. On the map, points are selected as potential target points by clicking. For each target point, a connection is drawn to all target points that could be reached from this point in the next test step. Target points that can serve as starting points are marked, as are points where a test case can end. The connections between target points can be annotated with a probability value. This probability value is used for the random generation of test cases according to a usage profile.

The annotation is done in the editor by left-clicking on the current transition. We provide probability values between 1 (unlikely) and 10 (very likely) with a default value of 5. The probability values of all outgoing transitions of each state are normalized. Furthermore, the robot expert can select different events



SENSOR_FAILURE 20
NO_EVENT 60
DYNAMIC_OBSTACLE 20

Fig. 1. Example of a test model created with the test model editor. Event probabilities are assigned to the transition from point 2 to point 3.

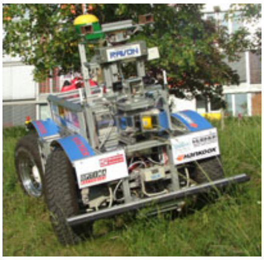


Fig. 2. The autonomous off-road robot RAVON

that can happen using a certain transition in a test step. If more than one event might happen on a drive represented by a certain transition, probability values have to be assigned to all possible events. If no event is explicitly selected and no probability distribution is explicitly defined, no event will occur, i. e., the probability of NO_EVENT is 100 %.

Figure 1 shows the determination of event probabilities for the transition between point 2 and point 3. Here, in 60 % of the times that a drive from point 2 to point 3 occurs as a test step, no event will happen. In 20 %, a dynamic obstacle will occur in front of RAVON; in another 20 %, we inject a sensor failure.

Generation of test cases. As soon as the test model has been created, test cases are automatically generated following different generation strategies. To test each relevant input sequence at least once, coverage criteria are used, e. g., transition coverage. This means that each transition is visited at least once, i. e., each possible movement from a target point to its neighbors with all possible events occurring is executed. To derive more test cases, the strategy of random test case generation based on the probability profile of the transitions is used. The benefit of this strategy is that a larger number of test cases can be derived. These test drives often contain circles. Thus, the robot is tested in many different, sometimes unexpected situations. For the generation of test cases, the tool jumbl [10] is used.

3 Method Application for Ravon

The application platform used here is the off-road robot RAVON, see Fig. 2. The characteristics of RAVON (2.4 m × 1.4 m × 1.8 m, 750 kg) make it indispensable to consider safety aspects. Therefore, dedicated hardware for low-level emergency stops is integrated, i. e., safety bumpers, emergency stop buttons, and a safety chain connecting the involved components. Besides this, RAVON is equipped with several sensor systems for obstacle detection, i. e., fixed laser scanners, a panning laser scanner, and stereo camera systems. RAVON's control system is implemented following the behavior-based control architecture iB2C (integrated Behavior-Based Control) [13]. In iB2C, a standardized interface of behaviors allows adjustment of a behavior's relevance, uniform combination of conflicting commands, and abstract evaluation of a behavior's internal state.

A major aspect in the context of performance analysis is the automation of test runs such that experiments can be executed repetitively. This way, basic safety properties of a control system can be evaluated. Due to the high effort of real-world experiments and missing information about the ground truth, preliminary tests have to be based on a simulation environment providing suitable details of the environment and the robot hardware. Here, SimVis3D [14] is used, which supports the simulation of actors, sensors, and environmental properties. That way, it is possible to check fundamental static and dynamic properties of robotic systems without possible damage to the hardware. As the real system is simulated at the lowest possible level, the sensor processing mechanisms and the control system remain unchanged both for simulated and real-world experiments. A predefined hardware interface allows the seamless exchange of hardware simulation and real system.

3.1 Test Run Automation

Based on the description of test cases presented in Section 2.1, automated test runs have been executed to validate the performance of RAVON's control system with respect to reaching predefined positions without violating the safety margins. The execution of commands and the checks of specifications are realized as an iB2C network (cf. Section 3) that is automatically built up according to the test case description. Based on this modular approach, extensions are easily configurable. Furthermore, the standardized interface of behaviors offers the possibility to observe the current state of the test run system during run-time. Figure 3 shows an example of the proposed structure. The depicted behavior network is connected to the upper interface of the control system and replaces the inputs of an operator.

The *Test run supervisor* behavior deals with coordinating the sequence of runs for a given environment and given specifications. The *Test run execution* behavior generates commands to be executed by the control system, i. e., a sequence of positions to be reached. During system execution, the violation of the defined specifications is evaluated by behaviors. For the given application, invariant violation behaviors observe collisions, elapsed time, and the mobility of the robot. Further criteria for evaluating safety properties can be excessive inclination, excessive velocity, or unstable steering motions. The maximum fusion behavior *(F) Invariant violation* gives an abstract representation of whether any of the properties is violated.

Behaviors evaluating the achievement of goals deal with aspects that have to be valid at least once during a test run. For the given example, the evaluation concerns whether a target position is reached, see Figure 3. Here, the

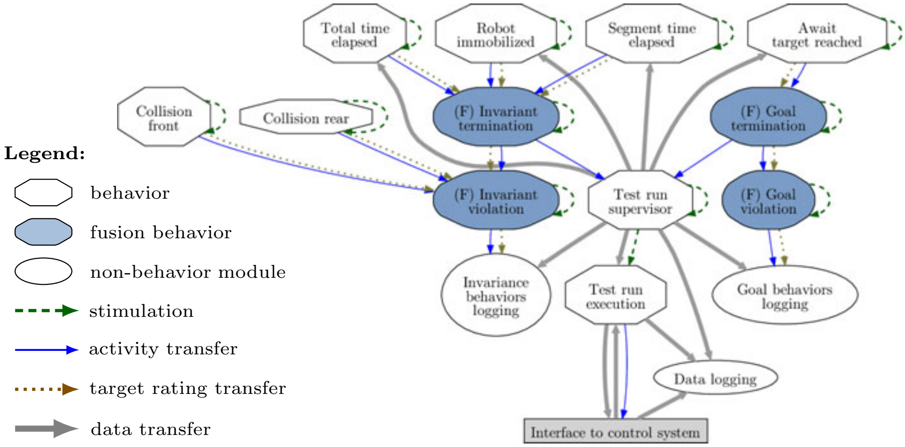


Fig. 3. Behavior network for test run automation

activity of the maximum fusion behavior *(F) Goal violation* therefore is an abstract measure for the accomplishment of all goals. Specifications that are relevant for terminating a test run are also implemented as separate behaviors (the behaviors *Await target reached*, *Robot immobilized*, *Total time elapsed*, and *Segment time elapsed*). This information is transferred to the *Test run supervisor* behavior via fusion behaviors. Their activity output is used to determine when to proceed with the next run. After finishing the last experiment, the program is automatically terminated. For each test run that is finished, the data collected during the experiments are saved and evaluated in terms of specification violations. The proposed approach is applicable both for simplified scenarios to test subsystems and for complex scenarios to validate the cooperation of system components. Although targeted towards simulation environments, the approach can also be used for test runs in real scenarios if the environmental properties can be sufficiently defined. In this case, the specification of invariant properties and goals has to be based on data available on-board.

3.2 Test Run Experiments and Results

The approach presented here has been applied to RAVON to evaluate the performance and safety properties in the given simulation scenario. Several overnight experiments were executed and yielded data for further evaluation. Figure 4 shows the test model and the trace of the robot's position during one of the test runs as an overlay. A selection of the statistical evaluation of the test runs is presented in Fig. 5. Here, for some of the path segments, the number of executions and the number of failed traversals are given. In this context, *failed* corresponds to a violation of any of the given specifications.

The most significant result is the high number of fails for paths 8 to 5. A closer evaluation reveals that the robot collides with the bushes during positioning maneuvers. Here, the vehicle follows a strategy of penetrating possibly traversable

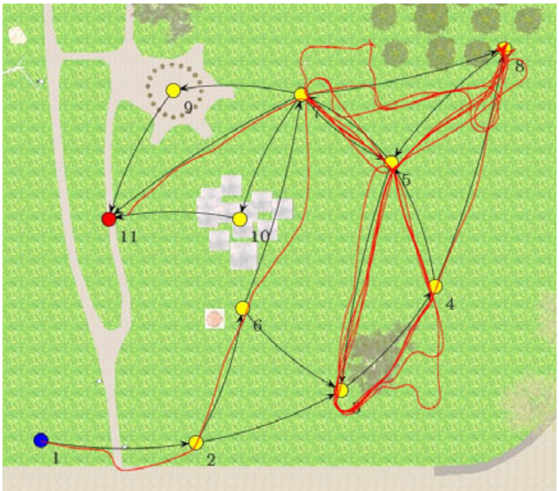


Fig. 4. Test model with overlaid pose trace of an exemplary run from one test case

From	To	Executed	Failed
2	3	50	0
2	6	54	0
3	4	241	3
4	8	106	3
7	10	1	1
7	11	101	3
7	5	30	0
7	8	54	2
7	9	1	0
8	5	160	16
9	11	1	0

Fig. 5. Selection of test run results

structures by slowly driving forward while observing the deflection of a bumper system. This approach has been implemented to allow traversing vegetated areas like grassland. As this happens at a low velocity, it is tolerable with respect to safety. Another interesting aspect is the reachability of points 9 and 10. Due to the low probability, each of them is only approached once. As expected, the result shows that point 10 cannot be reached. The test runs also reveal that the robot succeeds in approaching point 9 although it is blocked by obstacles. This results from the fact that during the approach, the minimal distance between the robot and the target point is less than a given tolerated distance used to mark a point as reached.

The tool support and the facilitated definitions of the test run execution procedure have proven suitable for rapidly adjusting the test run execution to new requirements. That way, several results regarding errors in the control system have been achieved. Furthermore, this approach has been used to validate the simulation environment by executing stress tests on it. In conclusion, contributions to the improvement of many different aspects of complex systems have been achieved.

4 Conclusion and Future Work

The approach presented in this paper provides a possibility for domain experts to easily define a test model that is used to generate an arbitrary number of purposeful test cases. As an automatic evaluation of test run results is provided, this system can be used for automatically checking system properties to find errors introduced during the development process. Future work will include an extension of the set of possible events and available types of violation detection behaviors.

Acknowledgments. This work was partly funded by the German Federal Ministry of Education and Research (BMBF) in the context of the project ViERforES (No.: 01 IM08003).

References

1. Sharygina, N., Browne, J., Xie, F., Kurshan, R., Levin, V.: Lessons learned from model checking a nasa robot controller. Form. Methods Syst. Des. 25(2-3), 241–270 (2004)
2. Kim, M., Kang, K.C.: Formal construction and verification of home service robots: A case study. In: Peled, D.A., Tsay, Y.-K. (eds.) ATVA 2005. LNCS, vol. 3707, pp. 429–443. Springer, Heidelberg (2005)
3. Lankenau, A., Meyer, O.: Formal methods in robotics: Fault tree based verification. In: Proc. Quality Week Europe, Brussels, Belgium (1999)
4. Lankenau, A., Röfer, T.: A safe and versatile mobility assistant. reinventing the wheelchair. IEEE Robotics and Automation Magazine, 29–37 (2001)
5. Rosenblatt, J.: Damn: A distributed architecture for mobile navigation. Journal of Experimental and Theoretical Artificial Intelligence 9, 339–360 (1997)
6. Goldberg, D., Mataric, M.: Design and evaluation of robust behavior-based controllers for distributed multi-robot collection tasks. Robot Teams: From Diversity to Polymorphism (2001)
7. Lüth, C., Krieg-Brückner, B.: Sicherheit in der künstlichen intelligenz. Künstliche Intelligenz 21(1), 51–52 (2007)
8. Prowell, S.: Using markov chain usage models to test complex systems. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, HICSS 2005, p. 318c (January 2005)
9. Hussain, T., Eschbach, R.: Statistical testing of iec 61499 compliant software components. In: Proceedings of INCOM 2009 (2009)
10. Prowell, S., Poore, J.: Foundations of sequence-based software specification 29(5), 417–429 (May 2003)
11. Lin, L., Prowell, S., Poore, J.: The impact of requirements changes on specifications and state machines. Softw. Pract. Exper. 39(6), 573–610 (2009)
12. Kloos, J., Eschbach, R.: Generating system models for a highly configurable train control system using a domain-specific language: A case study. In: Proceedings of A-MOST 2009 (2009)
13. Proetzsch, M., Luksch, T., Berns, K.: Development of complex robotic systems using the behavior-based control architecture iB2C. Robotics and Autonomous Systems 58(1) (2010)
14. Braun, T., Wettach, J., Berns, K.: A customizable, multi-host simulation and visualization framework for robot applications. In: 13th International Conference on Advanced Robotics (ICAR 2007), Jeju, Korea, August 21-24, pp. 1105–1110 (2007)

Collision Free Path Planning for Intrinsic Safety of Multi-fingered SDH-2

Thomas Haase and Heinz Wörn

Institute for Process Control and Robotics*, Karlsruhe Institute of Technology (KIT), D-76131 Karlsruhe, Germany
{thomas.haase,heinz.woern}@kit.edu
http://rob.ipr.kit.edu

Abstract. This paper presents a collision free path planning algorithm for the multi-fingered SDH2. The goal of this algorithm is the autonomous transfer of all fingers into a desired target position. The algorithm is independent of the initial finger position and may be used irrespective of any collision detection algorithm. It will be presented how forbidden finger movements can be discovered. The need for this feature, advantages and important Real-Time characteristics as well as further possibilities are discussed.

Keywords: Multi-fingered SDH2, Path Planning, Intrinsic Safety.

1 Introduction

Nowadays, various multi-fingered robot hands are state of the art in the robotic research. With the introduction of the SCHUNK Dexterous Hand 2.0 the first multi-fingered gripper that may fulfill industrial needs was introduced in 2008. The three fingered SDH2 is equipped with completely integrated electronics. A lot of experience is necessary to control the robot hand. It is desirable to simplify the manageability of the SDH2 for the integration into industrial applications. Therefore, intelligent modules are necessary to simplify the configuration of a grasping process. Fundamental methods of a grasping process should be realized autonomously. An intelligent SDH2 may be able to deliver the operator from tasks like path planning, contact determination or slip detection. Some of these modules may be invisible for the user and some others may be adaptive to special requirements. The main goal of this paper is the development of an algorithm (skill) which realizes the autonomic transfer of the SDH2 into each desired finger position. Even if the SDH2, compared to other robot hands, has only a small number of degrees of freedom, it is not a straightforward problem. Due to different control systems and manipulation tasks, the initial condition as well as the final position can be very complex and unmanageable. Especially due to the distal joints, the autonomous transfer of a finger is complicated. The overlaying corners of the gears and the overlapping finger positions increase

* This project is supported by SCHUNK GmbH & Co. KG, Lauffen.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 325–332, 2010.

the complexity of the problem. There are a lot of different techniques available. The development of collision free path planning algorithms started with the development of computer controlled robot systems. The most commonly used algorithms are variations of the collision-avoiding potential field [2] and the construction of probabilistic roadmaps [3]. The algorithms are developed to handle the high number of degrees of freedom in robot systems. The potential field technique requires a fast distance calculation. An example of how to transfer the potential field algorithm into multi-fingered gripper problems is given in [6]. The method described has the disadvantage that the target position is not definitely reached. The RPM-algorithm executes a required collision detection algorithm several times. The number of executions n_e is up to $n_e > 400$ and depends on the actual position. In that case, the given collision detection for the SDH2 [5] cannot fulfill hard Real-Time constraints. The Rapidly- Exploring Random Tree (RRT) concept is designed for a broad class of planning problems and adaptable to the SDH2. But even the RRT cannot fulfill the Real-Time constraints. Hence, the design of a collision free path planning algorithm to transfer each joint into a target position is the rather goal of this work. The algorithm has to define the motion sequence and should be able to run in Real-Time applications.

1.1 The SDH2

The final version of the SCHUNK Dexterous Hand 2.0 [1] was introduced in 2008 and can be seen in Figure 1. The robot hand consists of three identical fingers mounted on the body of the SDH2 which includes the inner electronics. Each finger possesses two independent degrees of freedom. Combined with an extra pivoting joint the SDH2 provides seven degrees of freedom. The actuators are DC motors coupled with high-ratio gears to achieve maximum torques of up to $M_d = 1.4Nm$ within the distal joints (middle of finger) and $M_p = 2.1Nm$ within the proximal joints (finger root). Due to high torques, there is the possibility that the finger damage each other. In order to prevent major damage, the maximum transferable torque can be reduced with the help of the SDH2 C++ library provided by the manufacturer. This mentioned library is used to control

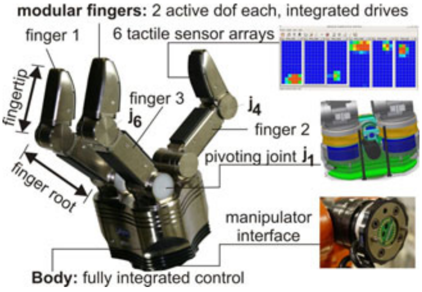


Fig. 1. Multi-fingered SDH2

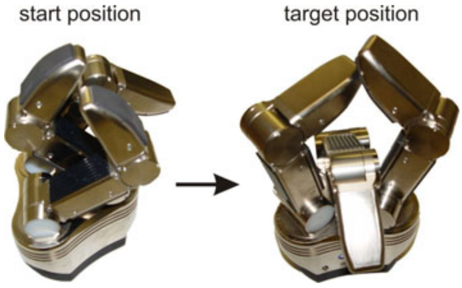


Fig. 2. Transformation Task

the SDH2. Currently, there are two different possibilities to move the fingers of the hand. The first possibility is to simply define angular velocities. A further opportunity is to set target angles for each joint. Once started a motion, the SDH2 calculates and controls the motion sequence on its own. All joints start and finish their motion at the same time. Therewith, it is not clearly defined that the finger will not collide with each other during a movement. This motion sequence has to be taken into account during the development of the path planning algorithm.

2 Functionality of the Collision Free Path Planning

The model of the designed algorithm can be seen in Figure 3. Each finger is described with the help of three points. Therewith and with given joint angles, the position of each finger in relation to the others can be determined. The connecting line between point $p_{l,i}$ and $p_{l,i+1}$ defines the orientation of each phalanx($l \in [1,3], i \in [0,2]$). The developed algorithm requires a special home position H_{SDH2}, Figure 5. It is shown, that all distal joints are set to zero. The home position for the proximal joints H_p is set to $H_p = -40°$:

$$H_{SDH2} = [0, -40, 0, -40, 0, -40, 0] \tag{1}$$

It is not necessary to specify joint zero in more detail. The most important thing is that all direction vectors are away from each other. The fingers of the SDH2 are divided into six phalanges: three fingertips and three root phalanges. The determination of allowed and forbidden movements is the main idea of the following algorithm. The algorithm and some special cases are shown in section 2.1. Section 2.2 deals with the transfer of all joints into the target position.

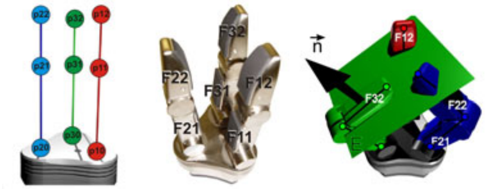


Fig. 3. SDH2 model

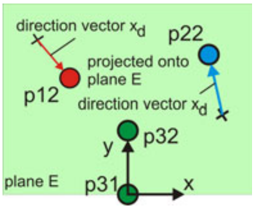


Fig. 4. Intersections

2.1 Identification Of Permitted Movements

The basic idea of this algorithm consists in detecting of movable phalanges. It is checked for each phalanx from Figure 3 whether its direction of motion is blocked by another phalanx or not. The required direction of motion is a decisive question for further calculations. Two points for each phalanx are calculated with the help of the kinematic model and the actual joint angles of the SDH2. Therewith, a plane E is inserted into each upper or lower phalanx. The direction of the normal vector $\bar{n}$ depends on the required motion. It is heading in the same direction as the motion direction. An initial test with plane E checks whether the phalanx of the finger is blocked or not. **A phalanx is released for moving, if there are no intersections with the plane E.** The selected phalanx is moving away from all other phalanges. As shown in Figure 4 finger 1 and finger 3 have intersections with plane E of fingertip 2. In that case, all points which are above that plane (with regard to the normal vector) are projected onto the plane. Now it is checked whether each direction vector x_d is towards the analyzed phalanx. If a direction vector is towards the analyzed phalanx, the phalanx is blocked. **The phalanx is free to move if no direction vector points towards him.** This kind of investigation helps in the decision-making process, whether a phalanx is allowed to move or not. Due to the simple assembling of the model, finger collisions may not be detected in each case. A model and a simple prediction as shown in Figure 4 will not be adequate to prevent finger collisions. Due to the fact, that all finger components are equal to each other, only a small number of special cases exist and it is still possible to realize the path planning algorithm without using collision detection. The SDH2 typical design led to the following additional considerations:

- If gears of other fingers are above a plane, this phalanx is blocked.
- Each phalanx is blocked if it has to move in positive angle direction into its home position. This may be cancelled if no further movements are possible.
- After testing each phalanx, it is feasible, that no one is allowed to move. This happens, if finger one or three prevents movements of finger two. Only if:

  $$j_0 > 85° \qquad \& \qquad j_{1||5} > 50° \qquad \& \qquad j_{2||6} < -60°, \tag{2}$$

  the fingertips are able to block finger two. In that case, the corresponding phalanx (fingertip 1 or 3) is allowed to move to $j_{2||6} = -25°$ in order to free the fingers. The joint angles $j_{2||6} = -25°$ depend on the joint angles of the chosen home position H_{SDH2}. It ensures that finger one and three are not able to collide with each other (due to j_0).
- The gears of finger two and three inhibit their movement into the home position. If finger 2 and 3 are parallel to each other ($j_0 \approx 0$) the gears can collide. In this special case it is necessary to move the fingers in predefined positions which allow passing the gears. These predefined joint angles depend on SDH2 design-related parameters. If:

  $$j_5 > 55° \qquad || \qquad j_5 < -16° \qquad || \qquad j_3 < -19° \tag{3}$$

  finger two and three may pass each other. Whether both phalanges are crossed can be detected with plane E and joint j_0. If $j_0 > 11°$ a crossover is impossible.

- The pivoting joint j_0 is transferred into the home position, if all fingers have finished their movements. Excluded from this are required movements of j_0 if all other phalanges are blocked. This exception has the lowest priority.

The phalanges are tested one after another. If all special cases are checked, there is at least one phalanx left and allowed to move into its home position. This intermediate position n_i is stored and the algorithm repeats the procedure until all joints / phalanges are successively transferred. The result of all calculated intermediate positions is a motion sequence, table 1. This motion sequence must be processed line by line to reach the home position.

The number of intermediate positions n_i in table 1 depends on the inital finger and joint conditions. So far, the number is less than $n_i \leq 7$.

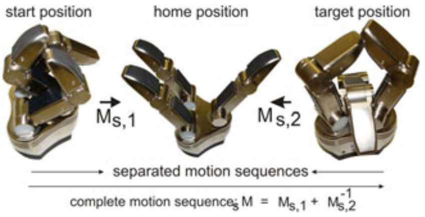

start position
home position
target position
$\vec{M}_{s,1}$
$\overleftarrow{M}_{s,2}$
separated motion sequences
complete motion sequence: M = $M_{s,1}$ + $M_{s,2}^{-1}$

Fig. 5. Algorithm Reversal

Table 1. Motion Sequence

n_i	j_0	j_1	j_2	j_3	j_4	j_5	j_6
0	60	-20	60	-5	90	55	50
1	60	**-40**	**0**	-5	90	55	50
2	60	-40	0	**-40**	**0**	55	50
3	60	-40	0	-40	0	**-40**	**0**
4	**0**	-40	0	-40	0	-40	0

2.2 Algorithm Reversal

With the algorithm from section 2.1 it is possible to free and transfer all SDH2 fingers into a home position, Figure 5. This is achieved by using a geometrical model of the SDH2 and an algorithm which detects forbidden finger movements. As a result of this algorithm a list of intermediate finger positions is established, Table 1. The reversal of the motion sequence leads to the appropriate solution. If the motion sequence allows a collision free transformation of each phalanx into the home position, it also works the other way around. With Table 1 it is possible to transfer each phalanx from the home position into the original start position. The transformation of the SDH2 finger into a target position can be achieved using the same algorithm. The main idea of how to transfer a start into a target configuration is shown in Figure 5: A motion sequence $M_{s,1}$ is calculated to transfer the SDH2 phalanges into the home position. Position 5 in Figure 6 visualizes the predefined home position H_{SDH2}. A second calculation is made to receive the motion sequence $M_{s,2}$ from the target position into the home position. This second motion is performed in the reverse order $M_{s,2} \rightarrow M_{s,2}^{-1}$. Figure 6 and table 2 show a complete transfer of the SDH2 fingers from a defined start position into the target position. Therefore, both motion sequences are performed successively:

$$M_s = M_{s,1} + M_{s,2}^{-1} \quad . \tag{4}$$

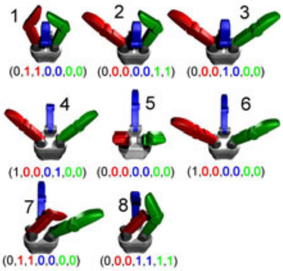


Fig. 6. SDH2 Transformation

Table 2. Complete Motion Sequence with Start and Target Position

n_i	j_0	j_1	j_2	j_3	j_4	j_5	j_6
1	90	-12	45	67	-85	-17	70
2	90	**-40**	**0**	67	-85	-17	70
3	90	-40	0	67	-85	-40	0
4	90	-40	0	-40	-85	-40	0
5	0	-40	0	-40	0	-40	0
6	**65**	-40	0	-40	0	-40	0
7	65	50	60	-40	0	-40	0
8	65	50	60	-5	49	-21	50

3 Path Planning

The path planning bases on the results of table 2. Each joint position $j_{y,x}$, $(x \in [0,6])$ in each row y has to be reached. Therefore, a time based trajectory must be implemented to transfer the finger/joints in each desired intermediate position. The interpolation is realized with the aid of a polynomial $s_{(y,x)}(t)$ of the fifth degree according to equation 5. The distance and time between two intermediate positions are given with $s_e = (j_{(y+1),x} - j_{y,x})$ and t_e.

$$s_{y,x}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{5}$$

$$\dot{s}_{y,x}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \tag{6}$$

$$\ddot{s}_{y,x}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \tag{7}$$

The boundary conditions are defined as: $s(0) = 0$, $s(t_e) = s_e$, $\dot{s}_x(0) = 0$, $\dot{s}_x(t_e) = 0$, $\ddot{s}_x(0) = 0$, $\ddot{s}_x(t_e) = 0$ and lead to:

$$a_0 = 0 \ , \ a_1 = 0 \ , \ a_2 = 0 \ , \ a_3 = \frac{10 s_e}{t_e^3} \ , \ a_3 = \frac{-15 s_e}{t_e^4} \ , \ a_3 = \frac{6 s_e}{t_e^5} . \tag{8}$$

The time based joint trajectory $j_{(y,x)}(t)$ can be calculated with

$$j_{(y,x)}(t) = j_{(y,x)}(0) + s_{(y,x)}(t) \quad t \in [0, t_e] \ . \tag{9}$$

The evaluation of equation 9 delivers the required joint angles, Figure 7.

4 Experimental Results

At IPR, Windows and Linux based Real-Time Systems are used to develop reactive grasping skills. The Real-Time Windows Target as well as the RTAI for Linux executes compiled models with operating frequencies f of up to $f = 1kHz$. Therewith, the calculation time of the algorithm is one of the crucial properties. An output of the developed Simulink model offers the time-based motion sequence. Figure 9 visualizes the output signal of the example in Figure 6 and

Table 2. Currently, the motion sequences are assigned directly to the SDH2. Due to the mentioned control facilities in section 1.1 the interpolation between two intermediate positions is realized internally. In Figure 8, an example of how to integrate the algorithm into the existing application development system can be seen. The movement request realizes the sequence control. The status is set to one if the SDH2 is in motion, otherwise it is zero. Additional sequence control systems may be realized with the evaluation of the actual velocities or the comparison of actual and target positions. Real world applications at the IPR have brought to light that the developed algorithm is rarely used to transfer difficult hand and finger positions. In most cases it is used to free hand positions into the home position or to transfer a secured hand position into a special finger position. For this task, the algorithm is an enrichment. It facilitates the construction of grasping tasks. No further motion sequence algorithms and calculations are required. Therewith, the algorithm fulfills the given requirements.

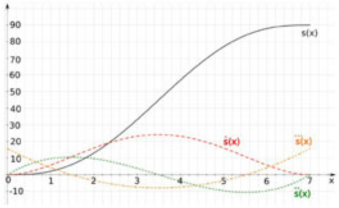


Fig. 7. Trajectory between two points

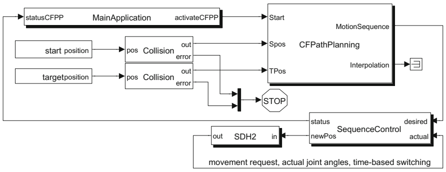


Fig. 8. Simulink Integration Example

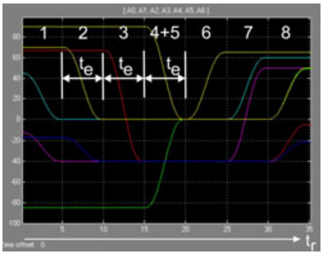


Fig. 9. SDH2 Movements

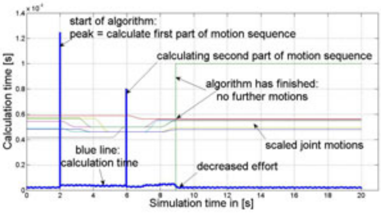


Fig. 10. Calculation Timeline

4.1 Performance and Execution Times

Due to the fact that a complete collision free path planning algorithm consists of two identical but independent algorithms, the calculation of the motion sequence is divided into two calculations. The required calculation times are shown in Figure 10. First, the motion sequence for freeing the fingers from a given starting position is calculated. Therefore an average calculation time of $t_{avg} = 130\mu s$ is needed (xPC Target, Dell Dimension 5100). Once the home position is reached the second motion sequence is calculated to transfer each finger into the target position, Figure 5. While the intermediate positions from section 2.1 are being worked through, the average calculation time is $t_{avg} = 3\mu s$. Due to the fact that no time-consuming collision detection is essential, an offline collision free path planning can be achieved within Real-Time applications. The interpolation of section 3 basically offers the possibility of influencing the time t_e between to int. positions. However, the resulting processing time t_r depends on the unknown number of intermediate positions. On basis of single finger movements and due to the lack of a collision detection algorithm, the transfer of different hand positions is not as quick as it may be. This disadvantage is compensated for by minimum computation times.

5 Conclusions and Future Works

In this paper a collision free path planning algorithm for the multi-fingered SDH2 was introduced. It was shown how to realize a geometrical approach and how to expand the algorithm to transfer the SDH2 into each desired target position. The establishment of a motion sequence and the time based interpolation between two intermediate positions were realized. The principal idea of this algorithm is the transfer of each phalanx into a predefined home position. The reversal of the algorithm is used to achieve each desired finger configuration. The algorithm is successful integrated into the existing Real-Time environment and keeps all Real-Time constraints.

Due to the simplicity the algorithm may be part of the SDH2 firmware. In combination with the existing collision detection [5] a contribution to the further development of the intrinsic safety of the SDH2 can be achieved. Nevertheless, the integration of simplified finger models may be useful to reduce the number of special cases. Sometimes it is not practical to use this algorithm to transfer finger positions. This is the case when all phalanges are allowed to move simultaneously and no collisions can occur. New methods have to be invented for checking both positions with regard to this aspect.

References

1. SCHUNK GmbH & Co. KG, Servo-electric 3-Finger Gripping Hand SDH2 (2010), http://www.schunk.com
2. Barraquand, J., Latombe, J.-C.: Robot Motion Planning: A Distributed Representation Approach. The International Journal of Robotics Research (1991)
3. Kavraki, L.E., Svestka, P., et al.: Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. IEEE Trans. on Robotics and Automation (1996)
4. Lavalle, S.M.: Rapidly-Exploring Random Trees: A New Tool for Path Planning
5. Haase, T., Woern, H.: Real-time collision detection for intrinsic safety of Multi-fingered SDH-2 (2010)
6. Lyons, D.: Tagged potential fields: and approach to specification of complex manipulator configurations. In: IEEE International Conference on Robotics and Automation

Dynamic Bayesian Networks for Learning Interactions between Assistive Robotic Walker and Human Users

Mitesh Patel, Jaime Valls Miro, and Gamini Dissanayake*

ARC Centre of Excellence for Autonomous System, University of Technology Sydney
15 Broadway, Ultimo, Sydney, NSW-2007, Australia
{miteshkumar.n.patel,jaime.vallsmiro,gdissa}@eng.uts.edu.au
http://www.cas.edu.au

Abstract. Detection of individuals intentions and actions from a stream of human behaviour is an open problem. Yet for robotic agents to be truly perceived as human-friendly entities they need to respond naturally to the physical interactions with the surrounding environment, most notably with the user. This paper proposes a generative probabilistic approach in the form of Dynamic Bayesian Networks (DBN) to seamlessly account for users attitudes. A model is presented which can learn to recognize a subset of possible actions by the user of a gait stability support power rollator walker, such as standing up, sitting down or assistive strolling, and adapt the behaviour of the device accordingly. The communication between the user and the device is implicit, without any explicit intention such as a keypad or voice.The end result is a decision making mechanism that best matches the users cognitive attitude towards a set of assistive tasks, effectively incorporating the evolving activity model of the user in the process. The proposed framework is evaluated in real-life condition.

Keywords: Activity Recognition, AI for Robotics, Mobile Robotics, Perception for Human-Robot Interaction, Machine Learning for decision making.

1 Introduction

In order to successfully design a robotic system capable of co-existing with humans it is essential to investigate and understand how the human-robot relation and their interactions can be established [5]. In this ever growing field of human-robot interaction (HRI) the use of modern artificial intelligence(AI) tools is becoming more relevant as they are capable of performing complex tasks that require behaviours normally associated with human intelligence. One of the most

* M. Patel, J. V. Miro and G. Dissanayake are with the Faculty of Engineering and IT, University of Technology Sydney (UTS), NSW 2007, Australia.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 333–340, 2010.

relevant goals of an “intelligent” user interface for a HRI system is to successfully execute the explicit user commands while at the same time account for the implicit cues that are not so easily observed [10].

Intention recognition, as defined in Heinze’s doctoral thesis [2], is the process of becoming aware of the intention of another agent (the user in our case) or as the problem of inferring an agent’s intention through their actions and its effects in the environment. Due to partial observations the communication of intentions between the user and agent becomes an important issue. Intention recognition has found its application in many research areas such as user assistance in aviation monitoring [2],human robot co-operation [10] and many others.

2 Motivation

Assistive technology is increasingly used to offset the impact of impairments resulting from injury, disease, the aging process and related disorders. Typically these technologies have focused on assisting users with mobility impairments. Increasing growth in the number of people with motor disabilities has resulted in considerable research being conducted into developing robotic assistive technologies to address the difficulties that this population faces, and as a result several intelligent systems that use AI and ubiquitous computing techniques have been developed with the older adult population in mind. These include for instance the Nursebot project [9], the PAMM project [1],or the UTS assistive wheelchair project [11]. Researchers in all these projects have explored different avenues to develop robotic assistive technologies using different Probabilistic/Stochastic models.

3 Previous Work

Probabilistic Networks (PN), also known as Belief Networks, has become a popular tool in the AI community as they can represent a complex system using graphical models [4]. Due to the amount of uncertainty involved in designing user assistive systems, researchers have explored the possibility of using various stochastic models like Markov Decision Process Models(MDP)/Partially Observable Markov Decision Process Models (POMDPs) [6], [11],Hidden Markov Models (HMM),Bayesian Network (BN) [3] and Dynamic Bayesian Network (DBN) [12]. The typical temporal dynamics of an assistive system consists of a sensorial system and an engine which, after learning, infers the user behaviour from the newly acquired data.

BNs, HMMs, hybrid HMMs (H-HMMs), auto-regressive HMMs (A-HMM), factorial-HMMs (F-HMMs) can be viewed as variants of DBN [7] as all these models can be represented using graphical models. BN being a static network only works with results from a single slice of time and hence does not work for analyzing an evolving system that changes its modelling behaviours over time.

POMDPs and DBNs are stochastic models for representing the evolution of variables over time. It consists of sequence of time-slices where each time slice contains set of variables representing the state at the current time [12]. In POMDP a control policy computes an action after every observation such that in the long-run the expected utility is maximized [13]. They are advantageous in modelling systems where the expected predictions or results are maximized in the long term. On the other hand, DBNs provides a framework where expected predictions or results are maximized in the immediate time slice and they are computationally more efficient.

4 Proposition

In this paper we propose to use DBN as a probabilistic decision making AI model suitable to be employed in mobility rehabilitative tasks, with the emphasis on the interactions between the machine and the *human in the loop*. The aim of this system is to understand and recognize the intention of a user (among a given set of alternatives) while he/she is using an instrumented powered rollator walker. The intention of a user is generally a hidden state in that it cannot be directly observed. Hence, the robotic agent has the task of deciding what the user really wants to do based on a minimalistic set of unobtrusive indicative inputs from the user.

5 The Dbn Framework

Dynamic Bayesian Networks are a branch of Bayesian Networks for modelling sequential data.They are represented by directed acyclic graphs (DAG) in which nodes are variables and arcs show the conditional independencies among the variables.Each node on the net has a probability table associated containing the conditional probabilities of each of the values that node can take with respect to each of the possible combination of its parent nodes. A typical DBN is depicted in Figure 1. The network only considers the discrete-time stochastic process, and it increases the index time t by one every time a new observation is recorded [7].

$$P(Z_t/Z_{t-1}) = \prod_{i=1}^{N} P(Z_t^i|Pa(Z_t^i)) \tag{1}$$

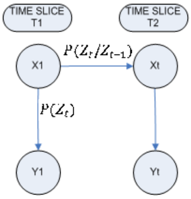


Fig. 1. DBN Structure unrolled for two time slices

A DBN is defined to be a pair, $(B_1, B_{->})$, where B_1 is a Bayesian Network which defines the prior $\mathrm{P}(Z_1)$, and $B_{->}$ is a two slice temporal Bayes net which defines $\mathrm{P}(Z_t/Z_{t-1})$ by means of a directed acyclic graph as given by Equation 1. Except for the first nodes in the first time slice, all the other subsequent nodes have an associated conditional probability distribution (CPD).

To construct a DBN we need to define three clusters of information: the prior distributions which can be termed as the initial probabilities of the state variables, the transition model which is the conditional probability distribution between states, and the sensor model. To specify the transition probability model we must also specify the topology of the connections between successive slices and between the states and evidence variables in each time slice.

6 Experimental Setup

The power walker employed is displayed in Figure 3(a). It is a modified commercial rollator walking frame with four wheels. The base frame has been instrumented with actuators and incremental encoders to the two rear wheels (front wheels are passive), two infra-red (IRs) proximity sensors to detect the presence of the user, four strain gauges (SGs) to detect indicative pressure, two push button switches on each of the walker's handle-bar to recognize the actual intention of the user, a low-level micro-controller and a high-level control computer.

The installed strain gauges are two Micro Measurements 120UR. The differential force between the vertical axis of each handle-bar are used to establish whether the user is holding on the handle-bar and is in readiness to start a task such as sitting down, standing up or ambulation.

The IR subsystem sensor is made up of two Sharp GP2Y0A02YK, which are used to estimate whether the driving user in standing behind the walker and how far they are from it. The motorized actuation subsystem is based around two powerful Matsushita Electric GMX-8MC045A 24VDC reversible gear-head motor with optical encoder and rotary mechanical couplings, driven using a national semiconductor LMD18200 3A, 55V H-Bridge motor driver.

A compact Hokuyo URG-04LX laser range finder is also incorporated in the design for localization and reactive navigation. The URG-04LX is able to report

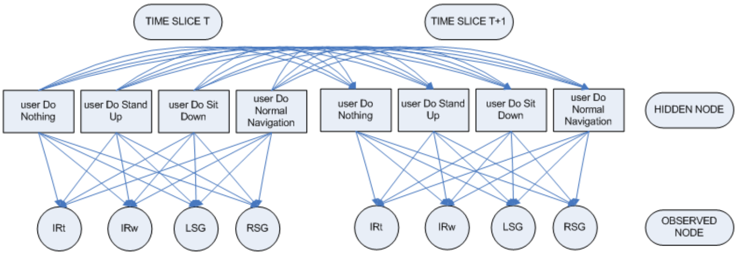


Fig. 2. DBN Model for our application when unrolled for two time slices

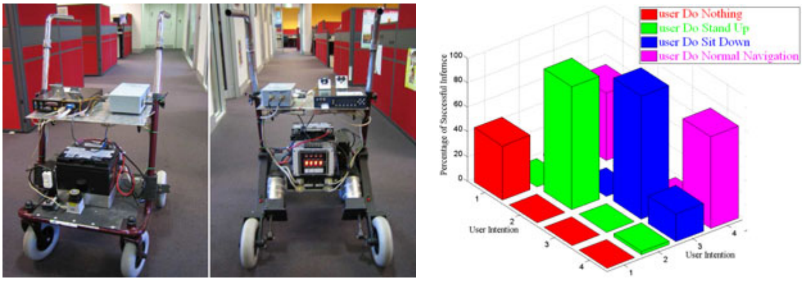
ranges from 0.02 m to 4.0 m (0.0001 m resolution) in a 240^o arc. Its power consumption, 500 MA @ 5V, makes it a natural choice for battery operated vehicles. The localization data obtained using the Hokuyo laser is used to plot the user intention at a particular location as shown in Figure 4.

The two push button switches installed on handles of the walker are used to record the actual intention of the user. This was done to compare the real-time intention of the user with the inference of the DBN model. The users were asked to press the correct switch while executing their intended action.

6.1 DBN as a Decision Making Agent on Walker

The desired DBN model has four observed nodes which represent the data collected from the four sensors on the walker and four hidden nodes which predicts the temporal state of the user. Figure 2 shows the intra- and inter- connections between nodes (in the example presented in this paper), which are then repeated over subsequent time slices.The four temporal states are *Do Nothing*, *Stand Up*, *Sit Down*, *Normal Navigation(Power Assisted Strolling).* The data from the sensor system being noisy, its corresponding nodes in the DBN model are represented as continuous gaussian nodes.The hidden node which predicts the user intention is discrete. The hidden nodes between time slices are interconnected as the future prediction of the user intention depends on the previous intention.

The transition probabilities between time slices were defined using various techniques viz. based on past experiences [6], common laws of operation and performing simulation. Specifically, we collected data from user trial runs while executing the tasks on the walker system, and also reflected what was perceived as accepted behavioural transitions (e.g., users will not sit down immediately after standing up, probability of the user to sit down immediately after standing up will be low...).



(a) Front and Rear View of the Walker (b) Offline inference results of the DBN Model

Fig. 3. (a) The instrumented rollator walker platform showing the laser range finder at the front, the infra-red proximity sensors on top of the black PC controller, servomotors and encoders and force sensors mounted on the handlebars and (b) Percentage accuracy of the DBN Model when tested offline in correctly inferring user intention

7 Experimental Results

The proposed DBN algorithm was evaluated within the domain of our office environment, a typical working space with desks, cubicles, people walking about, open meeting areas, corridors,etc. A 2D bird's eye view of the map can be seen in Figure 4 overlaid with some of the results detailed. The initial testing of the DBN model was done in a simulation environment in Matlab. Simulations in matlab environment also gave us an opportunity to derive the most optimal transition probability matrix.

Off-line testing of the DBN model was done after recording the real-time data from the sensor system while asking the user to perform different tasks. The DBN model was designed to recognize one of the four user intention. Data was recorded from five different users with distinctive physical characteristics and behaviours. Each user was asked to manoeuver the walker such that they perform all the intended tasks during the course so that sensor data for all the user intentions/actions could be recorded. Data sets recorded for each user were merged manually. Each user data set consists of 65 samples for each action, from which 50 samples were used for training the DBN model and 15 (not used in training) for testing. The total number for all users was 250 samples for each action used for training, and correspondingly 75 samples used for testing. The Matlab BNT toolbox [7] was used to design the DBN and perform offline testing. Figures 3(b) graphically depict the accuracy of the inference engine, which indicates successful recognition rates in the order of 81%.It can be seen in Figure 3(b) that the DBN model can correctly predict more distinctive behaviours, such as the action of sitting down and standing up, while at times it gets confused between the *Do Nothing* and *Normal Navigation* actions given the more subtle driving indication variability from user to user as sensed primarily by the strain gauges.

The promising results of the offline testing, encouraged us to test the DBN model in real-time. We used C++ programming environment and Mocapy++

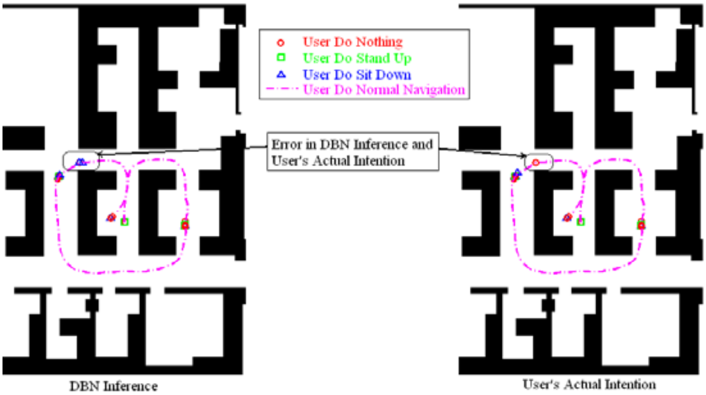


Fig. 4. 2D view of DBN Inference and Users' Actual Intention. The various tasks are depicted in different colours.

Table 1. Inference results obtained from two different users while manoeuvering the walker in a office environment

Users	Total Samples	false Inference By DBN	Accuracy
User 1	179	29	83.79%
User 2	317	12	96.21%

toolbox [8] to perform real-time testing. The data set, used for offline simulation was used to train the DBN model along with the most optimal transition matrix obtained during simulations. The trained DBN model was used to infer human intention based on the realtime data available from the sensors. Results in Figures 4 shows the comparison of the actual user intention (*recorded by pressing the correct combination of the push button switches*) and that inferred by the DBN model while the walker is being manoeuvred around the office by the user. A total of 496 samples of observation were collected from the sensors while the user was performing the task shown in Figure 4. The average recognition rates achieved when using the system for two users (*one user being new to the system and another being known to the system*) was in the order of 91.73%. A detailed break down of the results with each individual user is shown in table 1. The accuracy is higher with *User 2* as he is the designer of the DBN model and the overall walker system, hence he has a better understanding of the functioning of the system. The accuracy of the DBN model does not vary as much when tested with a user new to the system.

8 Conclusion and Future Work

In this paper a generic DBN framework capable of unobtrusively recognizing a set of possible actions by the users of a gait stability support power rollator walker has been presented. Since user intentions do not occur in a fixed sequence hence a stochastic approach has been used to model the relation of intentions and corresponding sensor measurements. The successful implementation of the proposed stochastic framework have shown an intention recognition mechanism that can be readily adapted to allow for more natural human robot interactions. The DBN model being highly flexible, it can be easily extended by adding new intentions and observations to the model. In future we plan to fuse the DBN model along with other static classifiers ("SVM","BN") such that the classifier can be used to provide the low level prediction based on the data available from the sensors, whereas the DBN engine can be used for high level user state inference.

Acknowledgment

This work is supported by the Australian Research Council (ARC) through its Centre of Excellence programme, and by the New South Wales State Government.

The ARC Centre of Excellence for Autonomous Systems (CAS) is a partnership between the University of Technology Sydney, the University of Sydney and the University of New South Wales.

References

1. Dubowsky, S., Génot, F., Godding, S., Kozono, H., Skwersky, A., Yu, H., Yu, L.S.: PAMM - A robotic aid to the elderly for mobility assistance and monitoring: A helping-hand for the elderly. In: IEEE International Conference on Robotics and Automation, pp. 570–576. IEEE, Los Alamitos (2000)
2. Heinze, C.: Modeling Intention Recognition for Intelligent Agent Systems. Ph.D. thesis, The University of Melbourne (2003)
3. Hong, J.H., Song, Y.S., Cho, S.B.: A hierarchical bayesian network for mixed-initiative human-robot interaction. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005), pp. 3808–3813 (April 2005)
4. Jensen, F.V.: An Introduciton to Bayesian Networks. UCL Press (1996)
5. Lund, H.: Modern artificial intelligence for human-robot interaction. Journal of Proceedings of the IEEE 92(11), 1821–1838 (2004)
6. Miro, J.V., Osswald, V., Patel, M., Dissanayake, G.: Robotic assistance with attitude: A mobility agent for motor function rehabilitation and ambulation support. In: IEEE International Conference on Rehabilitation Robotics, ICORR 2009, pp. 529–534 (June 2009)
7. Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning. Ph.D. thesis, University of Califronia, Berkeley (2002)
8. Paluszewski, M., Hamelryck, T.: Mocapy++ - a toolkit for inference and learning in dynamic bayesian networks. BMC Bioinformatics 11(1), 126 (2010)
9. Pineau, J., Montemerlo, M., Pollack, M.E., Roy, N., Thrun, S.: Towards robotic assistants in nursing homes: Challenges and results. Robotics and Autonomous Systems 42(3-4), 271–281 (2003), http://dx.doi.org/10.1016/S0921-88900200381-0
10. Schrempf, O.C., Hanebeck, U.D.: A generic model for estimating user intentions in human-robot cooperation. In: Proceedings of the 2nd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2005), Barcelona, Spain, vol. 3, pp. 251–256 (September 2005)
11. Taha, T., Miro, J.V., Dissanayake, G.: Pomdp-based long-term user intention prediction for wheelchair navigation. In: IEEE International Conference on Robotics and Automation (ICRA 2008), pp. 3920–3925 (May 2008)
12. Tahboub, K.A.: Intelligent human-machine interaction based on dynamic bayesian networks probabilistic intention recognition. Journal of Intelligent and Robotic Systems 45(1), 31–52 (2004/2005)
13. Theocharous, G., Mannor, S., Shah, N., Gandhi, P., Kveton, B., Siddiqi, S., Yu, C.H.: Machine learning for adaptive power management. Intel Technology Journal 10(4) (2006)

From Neurons to Robots: Towards Efficient Biologically Inspired Filtering and SLAM

Niko Sünderhauf and Peter Protzel

Department of Electrical Engineering and Information Technology
Chemnitz University of Technology, 09126 Chemnitz, Germany
{niko.suenderhauf,peter.protzel}@etit.tu-chemnitz.de

Abstract. We discuss recently published models of neural information processing under uncertainty and a SLAM system that was inspired by the neural structures underlying mammalian spatial navigation. We summarize the derivation of a novel filter scheme that captures the important ideas of the biologically inspired SLAM approach, but implements them on a higher level of abstraction. This leads to a new and more efficient approach to biologically inspired filtering which we successfully applied to real world urban SLAM challenge of 66 km length.

1 Introduction

Environmental perception, information processing under uncertainty and navigation in unknown environments are two of the core problems of ongoing research in mobile autonomous systems. We want to direct the readers attention towards a class of relatively new, biologically motivated approaches to filtering, navigation and SLAM and the change of paradigms they express. After reviewing some of this interesting work, we are going summarize our recently published derivation of a novel filter scheme that was inspired by these new approaches and show that the new filter can reproduce the results of the biologically more accurate, but computationally more expensive approaches.

2 Related Work

2.1 A Neural Implementation of the Kalman Filter

Very recently, Wilson and Finkel [17] presented how a Kalman filter can be approximated by a neural attractor network structure. The time-dependent activations $u_{i(t)}$ of the neurons in this network are governed by the internal network dynamics and by the external input. This input $I_{(t)}$ corresponds to the noisy sensor data and is incorporated into the network in an additive way: $u_{(t)} = D(u_{(t-1)}) + I_{(t)}$. Wilson and Finkel were able to show that the function D expressing the network dynamics can be adjusted so that the resulting model equations map directly onto the Kalman filter equations. However, the authors pointed out that this is only the case when the prediction errors are small, i.e. prior and evidence distribution coincide. In case of large prediction errors, "the output of the network diverges from that of the Kalman filter, but in a way that is both interesting and useful." Wilson and Finkel concluded that this behaviour makes the neural-based estimator more robust to changepoints and outliers in the sensor data.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 341–348, 2010.

2.2 Probabilistic Population Codes

Another group of authors developed a model of how uncertainty is manged and processed in the human brain. Their model is called *Probabilistic Population Codes* [6] [2]. Based on experimental results that humans perform "near-optimal Bayesian inference", the authors concluded that neural structures in the human brain are able to represent probability distributions and combine them using a "close approximation of Bayes' law"[6]. According to the proposed model, if there are two population of neurons that respond to the same stimuli and represent a certain piece of information (e.g. the position of a target according to auditory and visual cues), both representations can be fused by *adding* the weighted activities of the two individual neuron populations, i.e. performing a linear combination of the population activity vectors. Under the assumption that the neurons respond to the stimulus in a Poisson-like fashion, the linear combination of the two individual population activities can be shown to be Bayes-optimal [6].

Although not directly stated by [6] and [2], this optimality only holds when the hypotheses represented in the two populations approximately coincide. In later work [8] and [3], the same authors addressed this problem and formulated the "causal inference model" that distinguishes between cases of small and large disagreement between the two populations. Depending on the amount of conflict, the model can either force a fusion of the data or keep the two independent hypotheses.

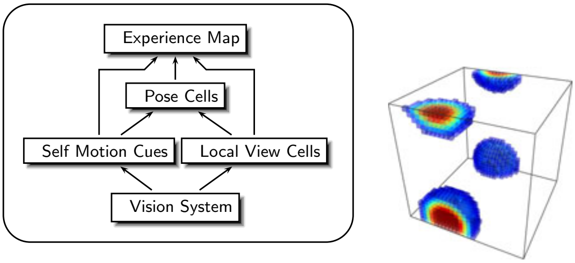
2.3 RatSLAM – A Biologically Inspired SLAM System

RatSLAM [9] is a biologically motivated approach to appearance-based SLAM (*Simultaneous Localization and Mapping*, see [4] and [1] for an introduction and overview) that was successfully applied to a very large (66 km) and demanding urban scenario [10]. This approach differs substantially from the established probabilistic SLAM algorithms [16] in that it makes no use of Bayesian calculus or probabilities. Instead, it borrows important key ideas from biological systems and mimics the behaviour of different types of brain cells that have been found to be involved in spatial navigation tasks in rodents, primates and humans. We shortly reviewed these cells and provided references to the relevant literature in [15].

Fig.1(a) sketches the RatSLAM system. The sole input is provided by the vision system. It provides coarse *self motion cues* by means of visual odometry and is furthermore able to recognize known places the robot already visited. Both the calculation of visual odometry and the place recognition are performed using very simplistic algorithms.

The core of the system is a 3-dimensional *continuous attractor network*, called the *pose cell network* (PCN) that was inspired by the head direction and place cells in rodent brains. Due to the attractor dynamics, mutual excitation and inhibition, self-preserving packets of local activity form in the network of pose cells. These local packets compete, trying to annihilate one another until a stable state is reached. The pose cell network is used to maintain an estimate of the system's current pose in (x, y, θ)-space. Fig. 1(b) illustrates the network and shows how the activity can wrap around the network borders.

Each cell in the PCN can receive additional stimuli from the *local view cells* which inject energy into the pose cell network. These local view cells are driven by the vision system's place recognition. Thus like their biological counterparts, the pose cells



(a) (b)

Fig. 1. (a) The general structure of RatSLAM. The vision system provides the input for odometry information and place recognition. The pose cells are inspired by the rodent head direction and place cells, while the experience map is responsible for loop closing and maintaining a topologically consistent map. **(b)** A screenshot of our C++ implementation of the pose cell network visualized using OpenGL. The network contains a single packet of activity. Notice how the activity wraps around the network borders.

are driven by both self-motion and visual cues. The main task of the PCN is to filter the many spurious false activations that are caused by the erroneous vision based place recognition system that generates a lot of false positive matches between different places. Without proper filtering, many wrong loop closures would be created.

Finally, on the top of the system we find the *experience map* which is responsible for managing a topologically and (to some extend) metrically consistent global map of the environment. It is a graph structure and consists of single *experiences*, each bound to a particular position in the state space and connected to previous and successive other experiences.

3 Deriving a Novel Filter Scheme from RatSLAM

The pose cell network (PCN) in RatSLAM is a three-dimensional attractor network, in general comparable to the one-dimensional network used by Wilson and Finkel in their work [17] mentioned above. While Wilson and Finkel designed their network in a way that it approximates a Kalman filter in case of small prediction errors, the authors of RatSLAM made no such attempt. Despite that, we discussed how the functionality of the pose cell network (PCN) can be compared to and interpreted from a Bayesian viewpoint in [13] and [14]. We pointed out the strong resemblance of the PCN to the histogram filter, a discrete version of the general Bayes filter.

Furthermore, we concluded that both approaches, PCN and histogram filters share the prediction step that predicts $\overline{\mathrm{bel}}(x_t)$ given the prior $\mathrm{bel}(x_{t-1})$ and the state transition model and control inputs. The main difference between both approaches is an *additive update* step performed by the PCN:

$$\mathrm{bel}(x_t) = \eta \left(\alpha p(z_t | x_t) + \overline{\mathrm{bel}}(x_t) \right) \tag{1}$$

where the posterior $\mathrm{bel}(x_t)$ is the weighted sum of the prediction $\overline{\mathrm{bel}}(x_t)$ and the evidence distribution $p(z_t|x_t)$ which is expressed by the local view cell's additive injection of energy into the pose cell network. The equation above contrasts the *multiplicative update* that is performed by Bayes filters, where $\mathrm{bel}(x_t) = \eta \cdot p(z_t|x_t) \cdot \overline{\mathrm{bel}}(x_t)$.

It is apparent that the *additive* incorporation of external data has as well been described in the work concerning probabilistic population codes and the work of Wilson and Finkel we reviewed above. Models of biologically motivated information processing seem to arrive naturally at this additive solution as neurons are generally understood to perform linear combinations, i.e. weighted *summation* of their inputs. Similar to [17], we found this additive solution to perform more optimal in cases of large prediction errors, i.e. when the prediction and evidence distributions largely disagree. Fig. 2(a) compares the additive and the Bayesian multiplicative solutions under a large prediction errror that is not covered by the involved covariances. This situation might occur in the event of a loop closure after the odometry accumulated a lot of under-estimated errors due to a wrong estimation or a degration of the measurement noise (over-confidence). Other possible causes are outliers or an erroneous data association.

Algorithm 3.1. Causal Update Filter (evidences, priors)

```
for all $p_i$ in priors do
    for all $e_j$ in evidences do
        if CONSENT($p_i$, $e_j$) then
            new_posterior ← BAYESIAN_UPDATE($p_i$, $e_j$)
            INCORPORATE(new_posterior, all_posteriors)
            $p_i$.used ← True ; $e_j$.used ← True
        end if
    end for
    if $p_i$.used = False then
        INCORPORATE($p_i$, all_posteriors)
    end if
end for

for all $e_j$ in evidences do
    if $e_j$.used = False then
        INCORPORATE($e_j$, all_posteriors)
    end if
end for

RESCALE_AND_PRUNE( all_posteriors )
best_posterior ← FIND_POSTERIOR_PEAK(all_posteriors)
return best_posterior, all_posteriors
```

Seeking a more efficient implementation of the additive incorporation scheme without using neural attractor networks, we formulated the novel *Causal Update filter* (CUF) in [13] and [14]. Its core idea is to use the multiplicative update where the two distributions (prior and evidence) agree, and the additive update when they disagree. The filter's response therefore is a mixture of the additive and the Bayesian multiplicative update and resembles the causal inference model of [7]: The filter can either fuse two hypotheses or keep both, depending on how much they coincide. Fig. 2(b) illustrates this idea. To achieve this behaviour, and for performance reasons, the prior, evidence and posterior distributions are modelled as multimodal Gaussian distributions. The algorithm (see Algorithm 3.1 for pseudo-code) determines the pairwise consent between the involved Gaussians using a Mahalanobis-like measure: $d_{(N_1,N_2)} = \sqrt{(\mu_1 - \mu_2)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2)}$. A threshold on this consent measure decides whether the Gaussians are incorporated into the resulting posterior in a multiplicative or additive way. Each Gaussian is assigned a weight, that is not changed when the Gaussian is incorporated additively. When two Gaussians are fused because they are in consent, the resulting Gaussian is assigned the sum of the weights of the two parent Gaussians. The weights are used in a pruning and rescale step at the end of the algorithm where posterior Gaussians whose weights are too small are removed from the joint posterior. This way, we achieve a kind of voting scheme where posterior Gaussians that represent consent prior and evidence hypotheses are considered more important.

Using this novel Causal Update filter we were able to completely replace the PCN with a much more efficient approach, while maintaining its desired robustness.

4 Results

In order to prove the CUF to be functionally equal to the pose cell network it was derived from, we adapted our RatSLAM implementation and replaced the PCN by the CUF. Fig. 3 illustrates the general algorithmic layout, details can be found in [14]. We found the SLAM system with the CUF at its heart to be able to perform equally well on the 66 km dataset presented by Milford et al. in [10]. Fig. 2(c) shows the results of the experiment. Compared to the map given in [10] one can identify some flaws (e.g. in the

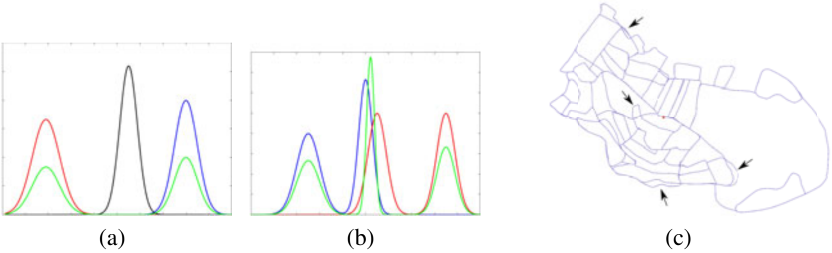

(a) (b) (c)

Fig. 2. **(a)** Bayesian multiplicative and additive update of dissenting Gaussian prior (blue) and evidence (red) distributions along with the respective Bayesian posterior (black) and additive posterior (green). Notice how the additive update bears more intuitive results by splitting the probability mass and keeping two independent hypotheses. **(b)** 1-dimensional example of the CUF. The prior (blue) and evidence (red) distribution are multimodal, each consisting of two peaks. The resulting posterior (green) consists of three Gaussians. Notice how the two hypotheses in consent have been incorporated multiplicatively and the remaining ones have been incorporated additively. **(c)** Final map of St. Lucia (path length 66 km, dimensions 1.8 by 3 km). The Causal Update filter was able to replace the pose cell network and managed to filter the erroneous data from the simplistic scene descriptors. The map has a few flaws (marked by the arrows) where the filter missed a loop closure, but covers the topologic layout and the general metric proportions of the environment.

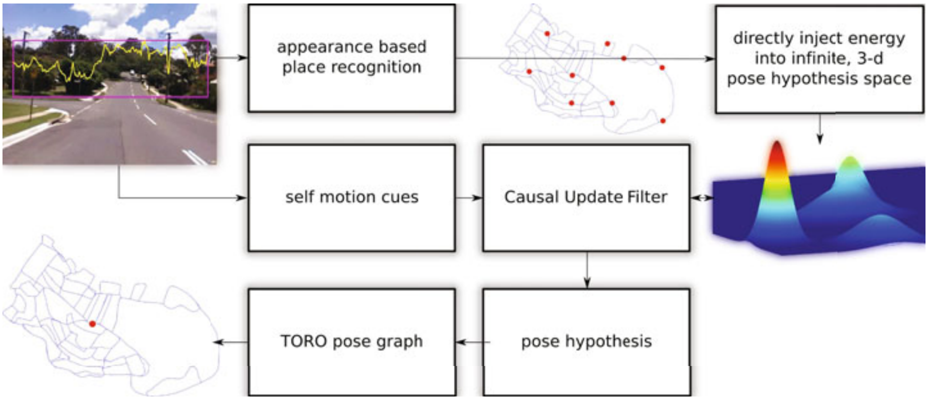

appearance based place recognition
directly inject energy into infinite, 3-d pose hypothesis space
self motion cues
Causal Update Filter
TORO pose graph
pose hypothesis

Fig. 3. The general structure of our modified, biologically inspired SLAM algorithm, using the Causal Update filter. The vision system provides the input for odometry information and place recognition. Notice that the place recognition is highly erroneous and produces a lot of false positives. The Causal Update filter at the center is derived from RatSLAM's pose cell network, but implements its core ideas on a higher level of abstraction. The TORO [5] pose graph is responsible for maintaining a topologically consistent map.

very top left or at the bottom) but in general the road network was successfully captured and the resulting map is topologically correct. Although it is of course not metrically correct, it can be considered semi-metric as the general proportions of the environment are represented adequately. A video showing the working SLAM system is available at our website `www.tu-chemnitz.de/etit/proaut`.

Compared to the original pose cell network, the CUF can be calculated much more efficiently: While incorporating odometry information into the PCN and calculating the network dynamics took 60 ms, the CUF can be calculated in well under 0.1 ms as it involves only a few operations. This massive speed-up comes at no costs or disadvantages, as the CUF is able functionally replace the pose cell network and to reproduce its results.

Furthermore, we reproduced the simple tracking example presented in [17]. In their experiment the authors tracked an object in a one-dimensional environment, using noisy position measurements as sensor input to the neural network structure that was tuned to approximate a Kalman filter. The results of this experiment using the CUF instead of the neural network are shown in Fig. 4. The first 75 timesteps resemble the small prediction error case in the experiment of Wilson and Finkel [17]. In this case, the CUF response is identical to a standard Bayes filter (a Kalman filter in this case, as all involved processes are linear). At time 75 the tracked object is kidnapped and the CUF's response diverges from that of a Kalman filter. We can see how a second, growing hypothesis is introduced at the new position and how the old hypothesis is still maintained, but constantly weakens due to the lack of sensory backup until it finally vanishes around time 120. This behaviour is exactly what was observed by [17] in the event of a so called changepoint. The CUF is able to reproduce these desirable results but can be calculated more efficiently, as no network dynamics are involved.

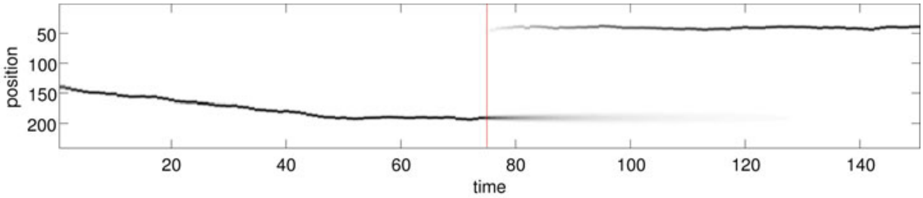


Fig. 4. The CUF in a 1-dimensional tracking experiment similar to those performed by [17]. Gray values indicate the weights of the CUF posteriors following the ground truth. The red line marks the kidnapping of the tracked object. See the text for further explanation.

5 Conclusions and Future Work

As we saw in the review of related work, the additive incorporation of information is common in biologically motivated models of filtering and information processing. Although this additive principle is inherent in neural systems, as neurons perform linear combinations, i.e. weighted summations of their inputs, it contrasts the multiplicative update scheme of Bayesian filter approaches.

Inspired by this fundamental difference between the world of Bayesian calculus and the biologically motivated approaches, we derived the Causal Update filter from the pose cell network of RatSLAM. While the pose cell network captured the additive incorporation principle by sticking closely to the neural nature of its biological archetypes (head direction and pose cells in the mammalian brain), our CUF is a higher abstraction of this principle, leading to an increased efficiency while maintaining the desirable robustness. Despite these abstractions, the CUF performed equally well in the same demanding, 66 km long urban SLAM scenario as the original RatSLAM algorithm [10]. Furthermore, we showed that the CUF is as well able to reproduce the results of a neural network model that was designed to approximate a Kalman filter [17].

In future work we will have to explore the possibilities of the novel CUF filter scheme and apply it to different filter problems where Bayesian techniques are the state of the art today. Seeing if the CUF can be applied to other problems and how well it performs there compared to the established Bayesian filters, remains an open but exciting question.

Equally important will be to establish an analysis and further understanding of the exact nature of the proposed filter. As the additive update can not be directly derived from the laws of Bayesian calculus, the filter seems to be non-Bayesian and closer to alternative formulations of probability that distinguish between *uncertainty* and *ignorance* like for instance Dempster-Shafer [11], the possibility theory of Zadeh [18] or the transferable belief model [12].

Acknowledgements

We would like to thank Michael Milford for providing the St. Lucia video footage that was presented in his paper [10]. Further material on RatSLAM is available at the RatSLAM website (`http://ratslam.itee.uq.edu.au`).

References

1. Bailey, T., Durrant-Whyte, H.: Simultaneous Localisation and Mapping (SLAM): Part II State of the Art. IEEE Robotics and Automation Magazine 13(3), 108–117 (2006)
2. Beck, J.M., Ma, W.J., Latham, P.E., Pouget, A.: Probabilistic population codes and the exponential family of distributions. Progress in Brain Research 165, 509–519 (2007)
3. Beierholm, U., Körding, K.P., Shams, L., Ma, W.J.: Comparing bayesian models for multisensory cue combination without mandatory integration. In: NIPS (2007)
4. Durrant-Whyte, H., Bailey, T.: Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. IEEE Robotics and Automation Magazine 13(2), 99–110 (2006)
5. Grisetti, G., Stachniss, C., Burgard, W.: Non-linear constraint network optimization for efficient map learning. IEEE Transactions on Intelligent Transportation Systems 10(3), 428–439 (2009)
6. Ma, W.J., Beck, J.M., Latham, P.E., Pouget, A.: Bayesian inference with probabilistic population codes. Nature Neuroscience 9, 1432–1438 (2006)
7. Ma, W.J., Beck, J.M., Pouget, A.: Spiking networks for bayesian inference and choice. Current Opinion in Neurobiology 18(2), 217–222 (2008), Cognitive Neuroscience
8. Ma, W.J., Pouget, A.: Linking neurons to behavior in multisensory perception: A computational review. Brain Research 1242, 4–12 (2008)
9. Milford, M.J.: Robot Navigation from Nature. Springer, Heidelberg (March 2008)
10. Milford, M.J., Wyeth, G.F.: Mapping a Suburb with a Single Camera using a Biologically Inspired SLAM System. IEEE Transactions on Robotics 24(5) (October 2008)
11. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
12. Smets, P.: The combination of evidence in the transferable belief model. IEEE Pattern Analysis and Machine Intelligence 12, 447–458 (1990)
13. Sünderhauf, N., Neubert, P., Protzel, P.: The Causal Update Filter – A Novel Biologically Inspired Filter Paradigm for Appearance Based SLAM. In: Proc. of the IEEE International Conference on Intelligent Robots and Systems, IROS (2010)
14. Sünderhauf, N., Protzel, P.: Beyond RatSLAM: Improvements to a Biologically Inspired SLAM System. In: Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, Bilbao (September 2010)
15. Sünderhauf, N., Protzel, P.: Learning from Nature: Biologically Inspired Robot Navigation and SLAM - A Review. In: Künstliche Intelligenz, German Journal on Artificial Intelligence, Special Issue on SLAM. Springer, Heidelberg (2010)
16. Thrun, Burgard, Fox: Probabilistic Robotics. The MIT Press, Cambridge (2005)
17. Wilson, R., Finkel, L.: A Neural Implementation of the Kalman Filter. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C.K.I., Culotta, A. (eds.) Advances in Neural Information Processing Systems, vol. 22, pp. 2062–2070 (2009)
18. Zadeh, L.: Fuzzy sets as the basis for a theory of possibility. Fuzzy Sets and Systems 1, 3–28 (1978)

Haptic Object Exploration Using Attention Cubes

Nicolas Gorges, Peter Fritz, and Heinz Wörn

Institute for Process Control and Robotics
Karlsruhe Institute of Technology
Engler-Bunte-Ring 8, 76131 Karlsruhe, Germany
{gorges,fritz,woern}@ira.uka.de

Abstract. This paper presents a new approach to generate a strategy for haptic object exploration. Each voxel of the exploration area is assigned to an attention value which depends on the surrounding structure, the distance to the hand, the distance to already visited points and the focus of the exploration. The voxel with the highest attention value is taken as the next point of interest. This exploration loop results in a point cloud which is classified using an Iterative-Closest-Point algorithm. The approach is evaluated in a simulation environment which includes in particular a simulation of tactile sensors.

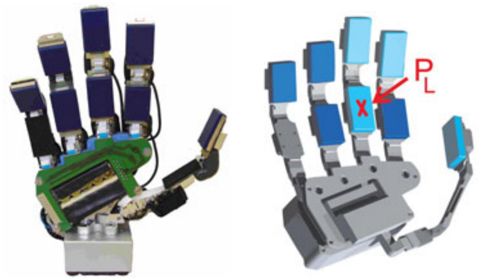
Keywords: Haptic exploration, object recognition, simulation.

1 Introduction

In the recent years there were many ambitions to create robots which help the human with everyday tasks. Handling unknown objects is essential for these kind of tasks. An active palpation sequence with direct interaction with the object allows perceiving object features like shape, texture and weight which are essential for handling an object. The haptic perception of a robotic hand usually includes tactile sensing for surface sensibility and kinesthetic sensor data like the finger positions for depth sensibility. Using the kinematics of the robotic hand, a palpation sequence results in a set of contact points which allows to model [1] and to classify [2],[3] the object. Alternatively, objects can be directly classified in the haptic feature space [4],[5],[6] but the benefit of building a point cloud is a more precise and an exchangeable presentation of the object. Besides the generation of an object model, a strategy for the exploration procedure is needed. Such a strategy should not create a random but target-oriented palpation sequence. Recent haptic exploration strategies so far include an approach using SLAM [7] and a local exploration behavior investigated in [8] using potential fields. The work in [9] describes a set of exploratory procedures which can be used for fine to coarse sensing strategies.

This work presents a novel strategy for haptic object exploration. The first goal is to reduce the number of exploration steps for object recognition and improve the classification results by wisely choosing the next point of interest.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 349–357, 2010.



(a) Real robot hand (b) Simulated hand.

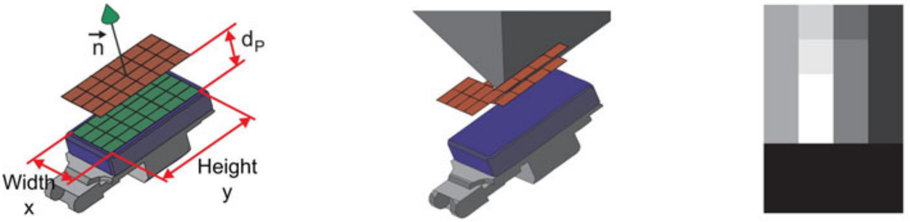
Fig. 1. (a) The real and (b) the simulated robot hand. The point P_L refers to the reference point of the robot hand and is attached to the tactile sensor of the middle finger in this example. The blue patches of the simulated hand indicate tactile sensor matrices with resolution of 7x4 taxels (tactile elements) whereas turquoise patches represent a resolution of 6x4.

The second goal is to provide a mechanism for creating any kind of exploration behavior. The exploration behavior might focus on different local structures like edges, surfaces or corners. Furthermore, the behavior might change between scanning and inspection: extract a homogeneous set of points (scan) and analyze the richest features (inspect). It should also be possible to fluently change between different behaviors. The idea of this work is motivated from visual attention maps [10] where each pixel of an image is assigned to an attention value. Transferring this idea to the haptic level, each voxel of the exploration area is assigned to an attention value which depends on the surrounding structure, the distance to the hand, the distance to already visited points and the focus of the exploration. The voxel with the highest attention value is taken as the next point of interest. Such a procedure involves a complete exploration system which is introduced in this work. This approach is evaluated in a simulation environment. The goal of the exploration procedure is the classification of an object.

After this introduction, the simulation environment used in this work is explained in section 2. The haptic exploration system including the attention space is presented in section 3. Finally, the results are presented in section 4 followed by the conclusions in section 5.

2 Simulation Environment for Haptic Object Exploration

The main components of the used simulation are visualization, collision detection, the kinematics of the robot hand and the tactile sensors. The simulation used in this work is based on Matlab and allows therefore a convenient access to all components. Furthermore, the simulation focuses on tactile sensors which allow a robot hand to detect contact only at certain regions –like the real robot hand. For collision detection we use SWIFT++ [11] which is implemented in C++ and embedded in the simulation.



(a) Sensor definition

(b) Collision with object.

(c) Resulting tactile image.

Fig. 2. A simulated tactile sensor

Of course, several restrictions have to be made for the simulation compared to the real system. Firstly, the object to be explored is stationary floating in space. Currently, the surrounding environment and dynamics are not considered. Secondly, the hand is not linked to a robot arm and can therefore move arbitrarily in space. Both restrictions do not influence the principles of this approach.

2.1 Simulated Robot Hand

The real robot hand [12] has eight degrees of freedom (DOF) and is shown in Fig. 1(a). The thumb, the index finger and the middle finger have 2 DOF each. The ring and the pinkie finger respectively have 1 DOF at the proximal phalanx. The real hand has joint angle sensors and is actuated pneumatically with position control of the fingers. The simulated robot hand is generated from a CAD-model of the real robot hand and is shown in Fig. 1(b).

2.2 Simulated Tactile Sensors

The tactile sensor system of the real robot hand [13] includes 8 tactile sensor matrices located at the inner surfaces of the fingers. The tactile sensor modules are able to pick up a pressure profile using a resistive working principle [14]. The index, the middle and the ring finger have 2 tactile sensor modules each. The thumb and the pinkie finger respectively have only one single sensor module at the distal phalanx. In the simulation, a tactile sensor patch is defined by the number of tactile elements (taxel) in x- and y-direction (height and width), the spatial distance between two taxels and the skin depth. The tactile gray image is generated by adding a thin plate for each taxel in the simulation. Then, each plate is moved along the normal vector of the sensor by the maximum skin depth d_P as shown in Fig. 2(a). The plates represent a taxel before (green) and after (red) translation along the normal vector. While moving, the collision detection is active so that a plate stops as soon as a contact is established (cf. Fig. 2(b)). The taken distance refers to the gray value of the taxel (cf. Fig. 2(c)). This represents the ideal output as a depth profile of the local object structure. To model the real tactile output, a transformation [13] from depth to pressure and

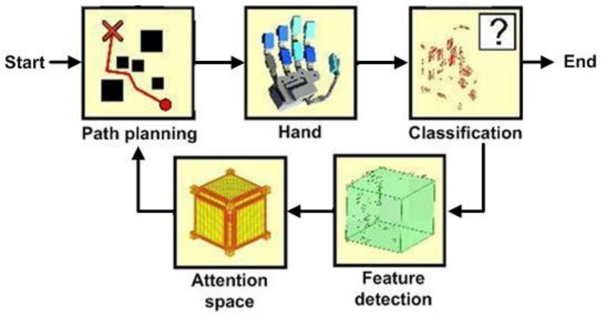


Fig. 3. The work flow of the haptic exploration system and its 5 components

from pressure to voltage has to be applied. Furthermore, each taxel has a different sensitivity and its own characteristic curve. Additionally, the conductive foam of the tactile sensor has the effect of a low-pass filter. For computing contact points, the center of gravity or the local maxima of the tactile image can be taken. Alternatively, all taxels with a response are be taken.

3 Haptic Exploration System

The exploration of an object involves 5 components which are shown in Fig. 3. The *hand* component involves the hand control and signal processing. It includes the processing of the tactile images as well as the approach movement of the robot hand, the adjustment of the hand towards the object surface and the execution of finger movements. The *classification* component serves for the recognition of objects based on a point cloud. If enough points are collected and the object can be clearly classified, the exploration process is finished. The *attention space* is the key component of the system as it chooses the next point of interest. For computing the attention space, *a feature detection* approach based on the point cloud is required. A *path planning* component is needed for reaching the next point of interest (POI) without collision. A reference point P_L on the robot hand is chosen which has to be guided to the POI. The reference point has to be located on a tactile sensor patch. The introduced components are explained in more detail in the following.

3.1 Attention Space

The 3-d attention space is created by representing the explored object as a voxel model and by assigning each voxel an attention value. For storing the 3-d points efficiently, we use an octree representation which recursively divides the 3-d space into octants. Fig. 4 shows a scheme of the components used for computing the attention space. Based on a point cloud, certain features are detected and each feature class is represented as an own feature cube which is a voxel model again. The features cubes are merged by creating a weighted sum.

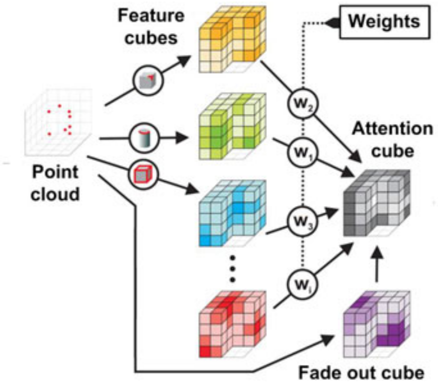


Fig. 4. Computing the attention space

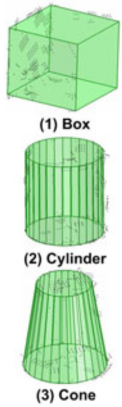


Fig. 5. Primitive fitting

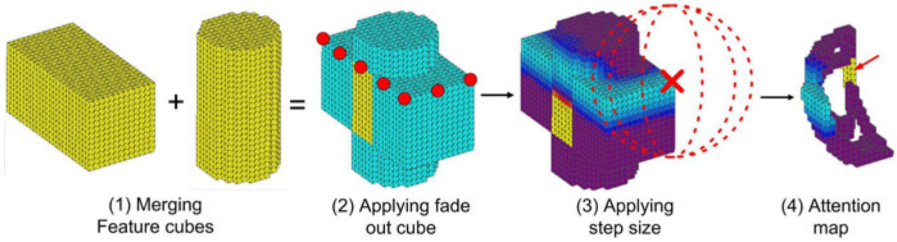


Fig. 6. From features cubes to an attention cube: (1) Superimposing two feature cubes. (2) Applying fade out cube - avoid already visited points (red dots). Red indicates high, turquoise low attention. (3) Applying step size (sphere) - distance between two successive points of interest (cross). (4) Final attention map - take peak (red arrow) as next point of interest.

Local and global features of the object are estimated: the guessed global shape classifies for scanning the object whereas local features are suited for supporting an inspection of the object. A local feature might be e.g. a corner, an edge as well as a planar or curved surface. Additionally, the results of the classification component can be treated as a guess of the global shape and added as feature cubes.

The fade out cube allows to ignore already visited areas and is superposed with the summed up features cubes. The outcome is the attention cube. Its peak is taken as the next point of interest. By adapting the weights, the focus on certain features can be changed. Thus, the exploration strategy is mainly defined by the weights $w_i(t)$ and their temporal behavior. Another factor which is not depicted in Fig. 4 is the step size which is the maximum distance of the hand's reference point to the POI. It allows to reach a compromise between speed and accuracy of the exploration. Fig. 6 shows an example for computing an attention cube.

3.2 Feature Detection

For finding features in a point cloud, we currently use a primitive fitting approach [15]. The simplest features are collinear and coplanar points but also points on a ellipse. Furthermore, the point cloud can be approximated by a set of object primitives. In this work, the primitives to be estimated are currently a box, a cylinder and a cone, as illustrated in Fig. 5. Each object primitive itself can be decomposed into features again, like edges, corners and planar surfaces. The feature detection and primitive fitting approach is based on Ransac [16] (Random Sample Consensus). Ransac takes only a small subset of the measurements into account for estimating a model. To evaluate a model, the number of measurements are counted which lie within a certain error margin. These points are called inliers. The points which are not conform to the estimated model are called outliers. The model with the most supporting inliers is finally chosen. In this work, the measurements are given by 3-dim. points and the models are the primitives.

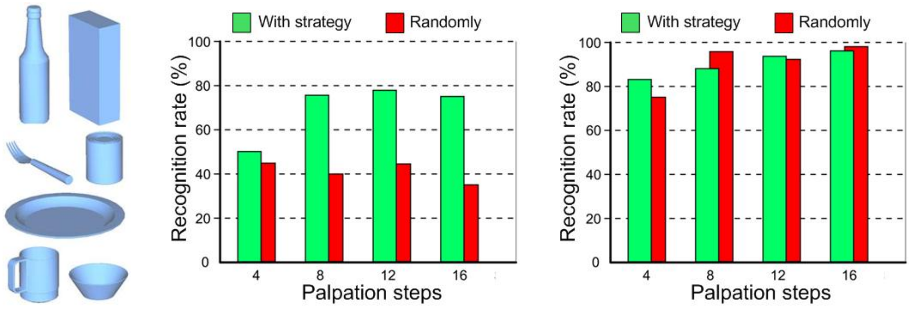
To decompose a point cloud into a set of primitives, the point cloud is searched for every kind of primitive and the primitive with the most inlier is taken. The inliers are removed from the point cloud and the next dominant primitive is sought until a maximum number of search steps has been reached or not enough point are left. Points, which are within a primitive but not close enough to the primitive's surface, cause a negative score.

3.3 Path Planning

For computing the path from one point of interest (POI) to the next, we currently use a very simple approach. To find a definitely collision-free path, we need four way points. The first and last point are given by two successive POIs. The approaching and departing directions are given by the surface normal of the respective POI. The hand and the exploration area are each approximated by a sphere. The sum of both radii plus a security distance provides the distance of the hand to the object center for approaching and departing. The path between the way points used for approaching and departing is given by the accordant circular path around the object center. The orientation of the hand for approaching is chosen according to the reference point of the hand and the accordant normal of the tactile sensor patch.

3.4 Object Classification

For object recognition, we use the Iterative-Closest-Point algorithm [17] (ICP). Given a point cloud of an unknown object and a reference point cloud, the algorithm tries iteratively to find a transformation which minimizes the distance between the point clouds. Superposing both point clouds, for each explored point the closest point of the reference point cloud is found. The outcome is a set of point pairs which are used to compute an Euclidean transformation. The estimated transformation is applied to the object point cloud. The resulting registration error is given by the mean Euclidean distance of all point pairs



(a) Objects. (b) Exploration of a can. (c) Exploration of a bottle.

Fig. 7. Random exploration compared to an exploration with strategy - the recognition rates related to the number of palpations

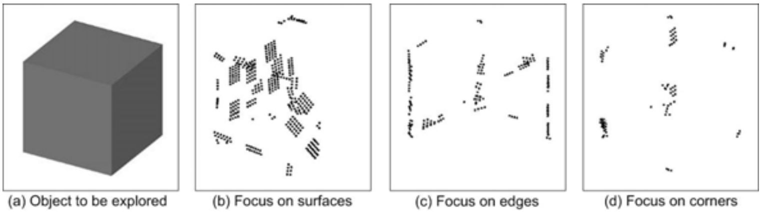


Fig. 8. Different exploration behaviors for a square object

and indicates the quality of the fitting. The process is repeated until the error converges or the maximum number of iterations is reached. Given a set of sample point clouds of each object, the point cloud of an unknown object is assigned to the class of the reference point cloud with the least registration error.

4 Results

The proposed approach is evaluated with everyday objects appearing in a household. The test set includes currently 7 objects, as depicted in Fig. 7(a). Each object is explored in the simulation using the 5-finger hand and the proposed exploration system. Then, the resulting point cloud is assigned to the object class with the most similar reference point cloud. The reference point clouds are generated by transforming a CAD-model of an object into voxel presentation. Then, the voxel model of the object is transformed into a reference point cloud.

The first focus of the evaluation is on the difference between a random exploration and exploration with strategy. Are the classification results improved by an exploration strategy? Therefore, 100 exploration runs were performed with each exploration consisting of up to 16 palpations. Fig. 7(b) shows the results of the exploration of a can. For this object, the recognition rate is up to 80%

with a strategy and only 40% for random exploration. It shows that the strategy has improved the results for the can enormously. The classification results of the bottle are much better: almost 100% with and without a strategy. But there is no significant difference when using a strategy. Looking at the test objects, it obvious that the can might be easily mixed up with the cup or the bowl. Therefore, a goal-directed strategy helps to distinguish between very similar objects. On the other hand, the bottle is easily distinguishable from the rest of the objects with or without strategy. Another focus of the evaluation is on changing the strategy by changing the weights. Fig. 8 shows that the focus of the exploration can easily be set on different structures and therefore generate completely different exploration behaviors.

5 Conclusions

This paper introduced a new approach for modeling a strategy for haptic exploration. Each voxel of the exploration area is assigned to an attention value and the voxel with the highest value is taken as next point of interest. A first version of an exploration system has been introduced which incorporates the proposed attention space. The approach has been successfully evaluated in a simulation environment which in particular includes simulated tactile sensor matrices. The proposed attention space improves the classification of similar objects and allows to generate different exploration behaviors. For future work, the components of the exploration system in general will be enhanced. Especially, the feature extraction and the classification system will be extended. The whole system will be evaluated with a larger test set and be transferred to the real robot system.

Acknowledgments

This work was supported by the German Research Foundation (DFG) within the Collaborative Research Center SFB 588 on "Humanoid robots—learning and cooperating multimodal robots".

References

1. Schmidt, P.A., Maël, E., Würtz, R.P.: A sensor for dynamic tactile information with applications in human-robot interaction and object exploration. Robotics and Autonomous Systems 54(12), 1005–1014 (2006)
2. Allen, P.K., Roberts, K.S.: Haptic object recognition using a multi-fingered dextrous hand. In: Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 342–247 (1989)
3. Magnanini, C., Zanichelli, F.: Haptic object recognition with a dextrous hand based on volumetric shape representations. In: Proc. of the IEEE Int. Conf. on Multisensor Fusion and Integration, Las Vegas, NV, pp. 2–5 (1994)
4. Heidemann, G., Schoepfer, M.: Dynamic tactile sensing for object identification. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, New Orleans, USA, pp. 813–818 (2004)

5. Johnsson, M., Balkenius, C.: Experiments with proprioception in a self-organizing system for haptic perception. In: Towards Autonomous Robotic Systems, Aberystwyth, UK, pp. 239–245 (2007)
6. Gorges, N., Navarro, S.E., Göger, D., Wörn, H.: Haptic Object Recognition Using Passive Joints and Haptic Key Features. Proc. of the IEEE Int. Conf. on Robotics and Automation, Alaska, USA (2010)
7. Schaeffer, M.A., Okamura, A.M.: Methods for intelligent localization and mapping during haptic exploration. In: Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics, vol. 4, pp. 3438–3445 (October 2003)
8. Bierbaum, A., Rambow, M., Asfour, T., Dillmann, R.: A potential field approach to dexterous tactile exploration of unknown objects. In: Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots, pp. 360–366 (December 2008)
9. Allen, P.K., Michelman, P.: Acquisition and interpretation of 3-d sensor data from touch. IEEE Trans. on Robotics and Automation 6(4), 397–404 (1990)
10. Steil, J.J., Heidemann, G., Jockusch, J., Rae, R., Jungclaus, N., Ritter, H.: Guiding attention for grasping tasks by gestural instruction: The gravis-robot architecture. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1570–1577 (2001)
11. Ehmann, S.A., Lin, M.C.: Accurate and fast proximity queries between polyhedra using convex surface decomposition. In: Computer Graphics Forum, pp. 500–510 (2001)
12. Schulz, A., Pylatiuk, C., Kargov, A., Oberle, R., Bretthauer, G.: Progress in the development of anthropomorphic fluidic hands and their applications. In: Mechatronics & Robotics 2004, pp. 936–941 (September 2004)
13. Göger, D., Gorges, N., Wörn, H.: Tactile Sensing for an Anthropomorphic Robotic Hand: Hardware and Signal Processing. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, Kobe, Japan (2009)
14. Weiss, K., Wörn, H.: The working principle of resistive tactile sensor cells. In: Proc. of the IEEE Int. Conf. on Mechatronics and Automation, Canada (2005)
15. Schnabel, R., Wahl, R., Klein, R.: Efficient ransac for point-cloud shape detection. Computer Graphics Forum 26(2), 214–226 (2007)
16. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
17. Besl, P.J., McKay, H.D.: A method for registration of 3-d shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)

Task Planning for an Autonomous Service Robot

Thomas Keller, Patrick Eyerich, and Bernhard Nebel

University of Freiburg
Institut für Informatik
Georges-Köhler-Allee 52
79111 Freiburg
{tkeller,eyerich,nebel}@informatik.uni-freiburg.de

Abstract. In the DESIRE project an autonomous robot capable of performing service tasks in a typical kitchen environment has been developed. The overall system consists of various loosely coupled subcomponents providing particular features like manipulating objects or recognizing and interacting with humans. To bring all these subcomponents together to act as monolithic system, a high-performance planning system has been implemented. In this paper, we present this system's basic architecture and some advanced extensions necessary to cope with the various challenges arising in dynamic and uncertain environments like those a real world service robot is usually faced with.

1 Introduction

The overall aim of the DESIRE[1] project [1] was to develop an autonomous robot capable of performing service tasks in a typical kitchen environment. From the project's beginnings, rather than focus on the accomplishment of predefined scenarios, it was decided to keep the system as unbounded as possible in order to gain maximum flexibility. This led to a system architecture consisting of several loosely coupled subsystems which are able to fulfill basic tasks like manipulating objects, driving autonomously in a dynamic environment or recognizing and interacting with humans. From our point of view, the most important implication of this decision was the need for a *domain-independent* planning system that is able to combine any number of these subsystems in an efficient yet stable manner.

The integration of such a task planner into the system increases the robot's level of intelligence and flexibility by altering the way the robot is controlled, moving from predefined sequences of detailed user instructions to a more sophisticated goal oriented approach. It is not longer required to provide the robot with a fully worked out description of its task (e.g., "Go to the big table, then take the plate, then return and give me the plate!") but rather to state some imperative command (e.g., "Give me a plate!") and leave it to the robot to find a suitable *plan* to achieve the according *goals* on its own, combining the features of the different components in a meaningful way.

[1] DESIRE is a project with several partners from industry and academia and is an abbreviation for "German Service Robotics Initiative".

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 358–365, 2010.

Task planning itself is a thoroughly investigated subfield in artificial intelligence [2]. However, in a robotics context, it has to be dealt with certain aspects complicating its application, including imperfect information about the environment, non deterministic changes, or user interaction. The main contribution of this paper is to show how to overcome these problems and make task planning suitable for everyday use in a robotics context.

The remainder of the paper is structured as follows: In the next section, we present the basics of the used planning system, describe how the current situation of the robot and its surroundings can be described in the *Planning Domain Definition Language* (PDDL) and show how it can be adapted to a dynamically changing and partially unknown environment via continual planning. Section 3 covers how to bridge the gap between several independent planning episodes, while the subsequent section shows how problems containing subproblems of different granularity can be solved efficiently. Before we conclude, we present the whole system architecture of the planning subcomponent in Section 5. Related work is referred to throughout the running text whenever it fits.

2 Planning in Real-World Environments

Given an initial state, a set of actions, and a goal formula, classical planning is about finding sequences of actions turning the initial state into a state satisfying the goal formula. The planning framework we use in this paper is PDDL2.1 Level 3 [11] extended by object fluents as proposed by Geffner [10]. In the following, we will use the term *state variable* to refer to predicates, numeric fluents and object fluents.

The classical planning approach heavily relies on having a complete and certain description of the situation the agent is faced with. Furthermore, actions need to be fully deterministic and the only changes allowed to occur in the environment are due to actions the agent decides to execute. Obviously, these constraints are too restrictive when it comes to modeling a domain depicting the real world: There might be other agents altering the state of the world (including nature), actions might have stochastic or probabilistic effects and the current state of the world might not be fully known.

Typical approaches dealing with such domains are *contingent* and *probabilistic* planning. These approaches try to generate conditional plans and policies (mappings from states to actions), respectively. Unfortunately, both approaches are of much higher complexity [7,8] than classical planning and usually fail to scale in even moderately complex scenarios. Furthermore, it might be impossible to model all potential outcomes of actions in dynamic environments, or concrete probabilities of outcomes are unknown.

Recently, we have proposed an alternative technique to deal with real world environments: *Continual Planning* (CP) [6], in which a continuous loop between *planning*, *plan execution* and *execution monitoring* is performed. In the planning phase, the agent is allowed to postpone the decision of how to fulfill subgoals to a later point in time when more information is available by using *assertions* as part of the plan. Then, the first action of the plan is executed. In the third phase, a monitoring procedure checks whether the remainder of the plan is still executable and still fulfills the goal. If that is not the case or if the plan has to be refined since the next executable action is an assertion, the agent switches to the planning phase. Otherwise, the next action's execution is started and the system continues with monitoring it.

For DESIRE, we have integrated the CP approach within our temporal planning system which is briefly introduced in the next section. Since temporal planning allows for concurrent actions of variable durations, there might be more than one action to start in the execution phase and these actions might have different durations. Furthermore, situations might arise in which an action has already finished, while others are still running. Therefore, we have extended the monitoring component to consider such actions with their remaining durations.

2.1 Base Planning System: TFD

We use *Temporal Fast Downward* (TFD) [4] as the base planning system. TFD is a domain-independent progression search planner built on top of the classical planning system *Fast Downward* [3]. It extends the original system to support durative actions and numeric and object fluents.

TFD solves a planning problem in three phases: First, the PDDL planning task is translated from its binary encoding into a more concise representation using finite-domain variables. This enables the use of heuristics employing hierarchical dependencies between state variables. In the second step, efficient internal data structures for the heuristic and the search component are generated. The most important ones are domain transition graphs for each variable that encode how a state variable's value can be changed, and the causal graph that represents the hierarchical dependencies between different state variables. Finally, a best-first progression search is performed, guided by a numeric temporal variant of the context-enhanced additive heuristic.

2.2 Generation of the Initial State and the Goal Description

The preferential way of assigning jobs to a service robot certainly is to formulate them as an imperative command. To process such a command, a robot needs to be able to recognize that the user's utterance is directed to him and to parse the utterance into an appropriate textual format. In DESIRE, this is done by the *automatic speech understanding* component (ASU). The ASU has a predefined set of grammar based frame structures used to map utterances to keywords and to classify them into categories like 'socialization' or 'instruction'. Additionally, single frames are mapped to classes like 'action' or 'object'. With the help of these keywords, the planner then extracts the information captured in an instruction utterance and transforms the command into a logical formula describing the goal. Basically, this is done by mapping the action frame of the frame structure generated by the ASU to its description in the PDDL domain file. The starting point of the generated goal formula is then the effect of this action. Afterwards, the parameters of the effect are replaced with appropriate fillers extracted out of the utterance (again detected with the help of the frame structure). Parameters for which no filler exists become universally or existentially quantified, depending on the found keywords (e.g., "Bring the salt cup to me!" is transformed to $\exists s(salt\_cup(s) \wedge loc(s) = user)$, while "Bring all salt cups to me!" is transformed to $\forall s(salt\_cup(s) \Rightarrow loc(s) = user)$).

The information about the current world state is distributed among the subsystems, e.g., the information about the positions and orientations of objects is present in the scene-analysis component while the position and internal status of the robot belongs to the self-model. Thus, the planning relevant information has to be gathered and unified in order to generate a coherent initial state. For that purpose, we have developed the *abstract world model* (AWM). The AWM is implemented as a plug-in mechanism and thus does not depend on the currently active components. Each component generates a proxy subcomponent which is responsible for passing the relevant information to the planner in a consistent way. During runtime, the components register their proxies with the planner, allowing the planner to query all relevant information. Based on the information gathered that way the planner generates a globally consistent initial state.

To express information about specific properties like color or shape shared by all instances of an object class, we have developed an ontology framework. The ontologies content is grounded during runtime and added to the initial state. Furthermore, for each object class a place where instances of that type can usually be found is stored in the ontology. If one or more objects obviously required to fulfill a task are missing (e.g., if the goal is to put away all salt cups but none is present), the goal is temporally changed to find these objects and the ontological information can be utilized to fulfill this temporally goal.

3 Global Memory

While Continual Planning serves as a much faster substitution for contingent planning, one problem that has not been addressed yet arises: Typically, planning systems are not designed to consecutively solve planning tasks that possibly depend on each other. Especially the success or failure of an action's execution, which is unknown during the planning episode might cause dependencies that need to be dealt with: While, in the case of an execution failure, the planning system will realize via monitoring that no progress was made in the active plans execution because the system's state did not change in the anticipated way, it might not be able to react on this because the state did not change at all. In this case, the planner will be confronted with the same planning task over and over again, leaving the whole system in an endless loop of generating a plan, failing in its execution, identifying this through monitoring and again generating the same, inexecutable plan.

Our solution to this problem is the *Global Memory* (GM), which keeps additional facts for each action that are added to the initial state *in the next planning episode after that actions execution.* These facts are divided in two sets, one that maintains facts that are applied in the case of successful execution, and the other for execution failure. A fact consists of either atomic or universally quantified state variable assignments, including increase and decrease operators for numeric state variables. Depending on the feedback of the sequencer after an action's execution, the according facts are applied to the GM, and in each planner run the content of the GM is added to the initial state.

An example for the syntax we use to describe GM updates is given in Figure 1. The first line merely states the name of the assigned operator, in this case the grasp action with parameters $?m$ (a movable object), $?g$ (a gripper) and $?l$ (a location). The grasp action has two preconditions that are important with regard to the GM: The scene model of the environment needs to be up-to-date, and the number of times it was unsuccessfully tried to grasp an object $?m$ with gripper $?g$ from location $?l$ in the past may not exceed a certain number of trials (in our case, a maximum number of two trials appeared to be reasonable).

```
GRASP ?m ?g ?l
  SUCC: (not (scene_model_updated))
        (forall ?M ?G ?L ((grasp_unsuccessful ?M ?G ?L) = 0))
  FAIL: (not (scene_model_updated))
        ((grasp_unsuccessful ?m ?g ?l) += 1)
```

Fig. 1. Global Memory description of the grasp action

Both pieces of information cannot be gathered by querying the AWM proxies of the according subarchitectures, as neither scene-model nor manipulation keep track of their history. Due to this reason, the GM is used to pass this information between consecutive planning episodes: Independent of the success of a grasp action's execution, the scene model is considered to be out-of-date, as can be seen in line 2 and 4 – in both cases, `(not (scene model updated))` is added to the initial state. The tracking of unsuccessful grasps on the other hand depends on the success of an action's execution: If the grasp action is executed successfully line 3 is applied and all unsuccessful grasps that ever happened are forgotten, i. e., reset to zero, since by successfully grasping any object the environment changes enough to justify retrying formerly unsuccessful grasps. In case that action's execution fails, the variable that keeps track of unsuccessful grasps is increased by one, as is stated in line 5. This increase allows us to break the endless loop the system possibly enters in case of execution failure: After a maximum number of trials to grasp an object, the according state variable `(grasp unsuccessful ?m ?g ?l)` is bigger than the maximum number of trials given in the grasp action's precondition and the operator is no longer applicable, forcing the planner to find a plan *without* trying to grasp object $?m$ with gripper $?g$ from location $?l$ again. Of course, this kind of modeling the domain leads to situations where the robot is not able to execute its orders anymore because it already tried to grasp an object from all locations with both its grippers. For this case, we added operators to the domain which can only be applied if some operator's execution failed more often than a given threshold allows, and which notify the user that (parts of) the received command cannot be executed.

Note that counting the number of unsuccessful action executions is obviously not the only way to break potential endless loops with the GM. Alternatively we could for instance force the planner to never use an unsuccessfully executed action again in the same state, or, in the case of grasping, the concerned docking position could be altered somewhat, but in the DESIRE scenario the described method worked sufficiently well.

4 Planning with External Modules

Planning in real-world domains requires to solve problems of different granularity: On the one hand, high-level actions like driving to a certain position or grasping a certain object are atomic actions with well-defined symbolic preconditions and effects. On the other hand, how to actually perform such an action might be a difficult subproblem in itself: To reach a certain position it is usually required to invoke a path planning subroutine, and before an object can be grasped a collision-free trajectory needs to be computed.

In previous work we have presented a new approach to solve such types of problems: The use of *semantic attachments* [5]. A semantic attachment is an external procedure called during the planning process to evaluate specific conditions or to directly alter the planning state. By using semantic attachments for subproblems like path planning we combine the advantages of both approaches while circumventing their disadvantages: on the one hand, the high-level planner does not need to care about subproblems since they are dealt with in the semantic attachments, on the other hand, only information actually needed to solve the problem is generated at the time the semantic attachment in invoked.

To deal with the mentioned issue of solving problems of different granularity, we implemented several semantic attachments, in particular for manipulating objects. When planning for *grasping* an object, it quickly falls into place that a purely symbolic representation is insufficient for the task. Having said that, the complete integration of a manipulation planner is far too inefficient, as one call to such a planner usually requires runtimes in the magnitude of seconds and in non-trivial problems hundreds to thousands of such calls are required. Therefore, we used a solution in between by utilizing an approximation procedure as a semantic attachment. This gives us more precise results than purely symbolic planning while staying efficient even in problems of considerable complexity. In dependence of the object's location and the shape of the surface it is located on, the semantic attachment checks whether a given docking position of the robot is appropriate for grasping. For that purpose, it is checked whether the object is within reach of the manipulator in question and whether it is not covered by other objects nearby it. Furthermore, it is ensured that the angle between the robot and the object's position is within some predefined range.

To find an appropriate position on a given surface to *place* an object on, we used a semantic attachment that works as follows: First, the surface is partitioned into grid cells of one square centimeter. Then, the occupied cells are determined on the basis of all other objects on the same surface. Finally, a free area big enough to hold the object and maximizing the remaining free space is chosen as the position to place the object on. Note that all these computations are performed only when they are required.

5 System Architecture

Before we conclude, we give an overview of the whole implemented planning subarchitecture, which is depicted in Figure 2. The heart of it is the TFD/M planner as described in Section 4. Its input is generated by the subcomponents depicted in the box at the bottom left – AWM and GM contribute the systems current state, which corresponds to the planner's initial state, and GoalGen transforms a user's command into a PDDL description of the goal.

As we described in the previous section, symbolic planning is not sufficient in a dynamic real-world environment. This is taken into account by the additional use of semantic attachments (depicted as Modules), which do complex calculations during planning such as checking spatial conditions for grasping or deciding where to place an object.

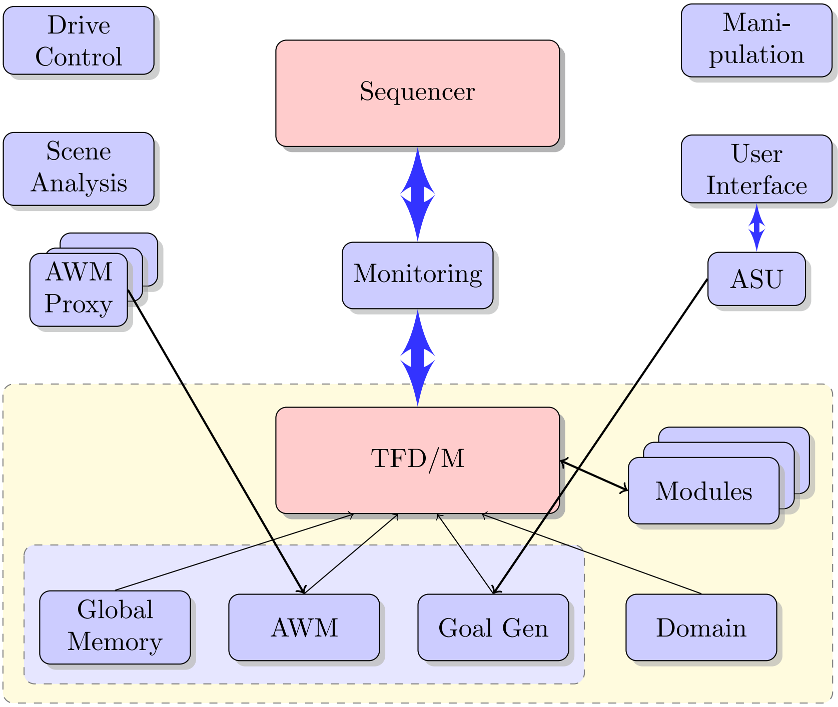


Fig. 2. System Architecture of the Planning subcomponent in DESIRE

The last depicted component that is part of the planning subarchitecture is monitoring, which decides if the planner is triggered at all based on the success of the currently active plan's execution.

6 Conclusion

In this paper we have presented the planning system used in the DESIRE project. We have shown how the current state of the robot is generated from information of distributed subarchitectures and how additional knowledge is passed between consecutive planning episodes to avoid endless loops in continual planning that might arise if some action seems executable on the symbolic level but is indeed not in the real world.

Another difficulty arising when planning in dynamic and uncertain environments is the necessity for calculations that exceed the expressive power of PDDL. With the concept of semantic attachments we have presented a way to overcome this hindrance.

Acknowledgments. This research was supported by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IME01-ALU.

References

1. Plöger, P., Pervölz, K., Mies, C., Eyerich, P., Brenner, M., Nebel, B.: The DESIRE Service Robotics Initiative. Künstliche Intelligenz 8(4), 29–32 (2008)
2. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory and Practise. Morgan Kaufmann Publishers, San Fransisco (2004)
3. Helmert, M.: The Fast Downward Planning System. JAIR 26, 191–246 (2006)
4. Eyerich, P., Mattmüller, R., Röger, G.: Using the Context-enhanced Additive Heuristic for Temporal and Numeric Planning. In: 19th Int. Conf. on Automatic Planning and Scheduling, pp. 130–137. AAAI Press, California (2009)
5. Dornhege, C., Eyerich, P., Keller, T., Trüg, S., Brenner, M., Nebel, B.: Semantic Attachements for Domain-Independent Planning Systems. In: 19th Int. Conf. on Automatic Planning and Scheduling, pp. 114–121. AAAI Press, California (2009)
6. Brenner, M., Nebel, B.: Continual Planning and Acting in Dynamic Multiagent Environments. Journal of Autonomous Agents and Multiagent Systems 19(3), 297–331 (2009)
7. Littman, M.L., Goldsmith, J., Mundhenk, M.: The Computational Complexity of Probabilistic Planning. JAIR 9, 1–36 (1998)
8. Rintanen, J.: Constructing conditional plans by a theorem-prover. JAIR 10, 323–352 (1999)
9. Hoffmann, J., Nebel, B.: The FF Planning System: Fast Plan Generation Through Heuristic Search. JAIR 14, 253–302 (2001)
10. Geffner, H.: Functional STRIPS: a more flexible language for planning and problem solving. In: Logic-Based Artificial Intelligence. Kluwer, Dordrecht (2000)
11. Fox, M., Long, D.: PDDL2.1: An extension to PDDL for expressing temporal planning domains. JAIR 20, 61–124 (2003)

Towards Automatic Manipulation Action Planning for Service Robots*

Steffen W. Ruehl, Zhixing Xue, Thilo Kerscher, and Rüdiger Dillmann

FZI Forschungszentrum Informatik
Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany
{ruehl,xue,kerscher,dillmann}@fzi.de
http://www.fzi.de/ids

Abstract. A service robot should be able to automatically plan manipulation actions to help people in domestic environments. Following the classic sense-plan-act cycle, in this paper we present a planning system based on a symbolic planner, which can plan feasible manipulation actions and execute it on a service robot. The approach consists of five steps. *Scene Mapping* formulates object relations from the current scene for the symbolic planner. *Discretization* generates discretized symbols for *Planning*. The planned manipulation actions are checked by *Verification*, so that it is guaranteed that they can be performed by the robot during *Execution*. Experiments of planned pick-and-place and pour-in tasks on real robot show the feasibility of our method.

1 Introduction

Intelligence is a very important factor for service robots to help people handle daily tasks in household environments. If a service robot is ordered to serve a cup of water, it should be able to autonomously navigate in front of the fridge, open its door, get a bottle of water from it and pour the water into a cup. The robot should be able to perform these actions, understand the effects of them and plan this sequence of actions by itself. Focused on the last point, we present our current work of automatic manipulation action planning for service robots in this paper.

Planning is one of the most investigated problems in area of artificial intelligence. It is normally abstracted to a symbolic level, so that a symbolic planner can handle and solve the problem. Although solutions in the symbolic world can be found, it is still an open problem, how the used symbols can be mapped back to the real world. If a symbolic planner generates an action "grasp the cup", for instance without considering the arm reachability and obstacles around the cup, the plan could not be performed in the real world, although its symbolic formulation is correct. So, detailed knowledge

* The research leading to these results has been supported by the DEXMART Large-scale integrating project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-216239. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 366–373, 2010.

about the scene is required and should also be taken into account by the planner, which leads to the classic sense-plan-act cycle.

To plan executable manipulation actions, we have extended the sense-plan-act cycle with two new steps. In the discretization step, the continuous Cartesian space is discretized to locations which can be handled as symbols. Heuristics for the scene are used to further filter out invalid locations. In the verification step, the resulting symbolic actions are checked with the real robot ability, which ensures the execution of planned actions. Experiments on a real robot performing planned manipulation actions show the feasibility of our method.

The paper is laid out as follows: comparable approaches from literature are reviewed in the next section. The system overview of our proposed solution is presented in Sec. 3. Used planner and formulations are given in Sec. 4, followed by Sec. 5, which contains experiments of planned pick-and-place and pour-in tasks. The paper ends with planned future work and conclusions.

2 State of the Art

Different work has been carried out in the area of action planning for manipulation. Most approaches ignore the problem of grasping and assume an object can be manipulated if there exists a collision free arm configuration, where the Tool Center Point (TCP) is in some fixed relation to the manipulated object.

Different ways of integrating real world motion and grasp planning have been investigated in the literature. The automatic generation of action symbols from a motion planner is shown in [2]. Point pairs are connected to one pick-and-place action, if there is a collision free path between them. The manipulation space is clustered into subsets to avoid dimensional explosion. The opposite approach is taken in [4]. A symbolic planner can connect different locations arbitrarily. The motion command is extended by a “semantic attachment”, which causes a call to a motion planner. It ensures the collision free execution of the robot motion. These approaches however do not consider grasping, but could be extended to do so.

A motion planner is extended to be able to plan regrasp action sequences in [12]. This is done with respect to grasping and collision free trajectories and evaluated in a real world scenario. The planner is still a motion planner and cannot handle arbitrary abstract goals.

The above approaches lack of a way to generate action parametrization for real world actions, considering reachability, grasp-ability and stability of the action’s results. In this paper, we present a system which generates symbolic descriptions of a scene, plans manipulation actions for given goals and executes them in a real world environment.

3 Proposed System Overview

For flexible program adaption, detailed knowledge about the scene is required. Also, symbolic planners are widely used for reasoning about goals. In combination with a executing robot, this leads to the classic sense-plan-act cycle. This simple cycle fails

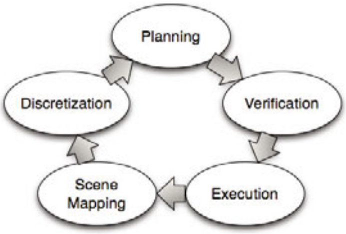


Fig. 1. The processing cycle of the proposed system. A given scene is mapped to symbolic relations. The continuous manipulation space is sampled to a discrete representation. Together with a set of goals, a planner can find a sequence of robot actions to archive the desired goal. The resulting plan is automatically verified and optimized in a simulation environment. Finally, it is executed by the robot, resulting in a new scene. The scene is analyzed again and compared to planned goals and states. If conflicts are detected, the cycle will be executed again to keep the manipulation goal directed.

however, if the scene and robot properties are too complex to be comprehensively modeled for a symbolic planner. We extend the sense-plan-act [10] cycle with a discretization step beforeplanning and a verification step for generated plans. The discretization step maps a scene to an approximation which can be handled by a symbolic planer. The verification step filters out infeasible generated plans, so that it is guaranteed that the planned actions can be executed on the real robot.

The proposed system is sketched in Fig. 1. The scene mapping creates a relational description of an observed scene. The discretization component creates a finite set of discrete symbols to represent relevant aspects of the continuous Cartesian space in which the manipulation task is performed. Those two sets of symbols are the input for the planner which calculates an action sequence describing the manipulation task. The verification component checks this sequence for its feasibility and its consistency with the given goals. Finally, it is executed by the robot.

Scene Mapping. Information from sensors include object positions, their uncertainty and information about obstacle space. The used perception is described in [5,7] . From its data, a relational description of the scene is created. This steps includes associating objects and obstacles to their locations. Relations between objects can be generated. The mechanics of the scene are analyzed, leading to information about the stability of manipulation in a scene. E. g. if one object carries another, it can not be moved, similar to the case, where an object is leaned against another.

Discretization. Discretization in our system is the process of breaking down a continuous Cartesian space of object or TCP poses down to a symbolic description. Object poses from the scene mapping are not sufficient for comprehensive manipulation planning. A task may require a temporary placement of an object to free a location required by another operation. A big set of arbitrary symbols should be avoided for planning, since it would cause the search for a plan to explode [9]. We propose a three step approach using heuristics to generate small sets of symbols with a high probability to be sufficient for a planner to find a solution for a given task.

Empirical subsets are relevant for different manipulation tasks. A temporary place operation will need a location on a flat surface, while the handover of an object is executed in the free space. The selection of this subset is dependent on the set of actions available for the robot. We use the following subsets for different manipulation actions: *"Table plane"* is the area on the (or multiple) table, where most of the manipulation takes place. *"Free space"* for positions where no contact between objects is desired. *"On"* are places onto of other objects, necessary for stacking objects.

Sampling reduces the continuous sets to discrete ones with a fix sampling step or density of symbols in space. Possible sampling methods are random or periodically in a grid base manner. However, this set could still be too big for the planner.

Rating and selection are the last steps. Action and scene dependent rating functions are used to select promising symbols. For instance, for a place position the number of available grasps as a rating indicate how well it is suited to place the object there. Robot arm TCP poses can be rated using the capability map [14].

Planning. From a set of goals, a planing domain and an initial scene, the planner generates a sequence of manipulation actions. The initial scene description is generated from the scene mapping, goals are compiled from the requested actions and the scene model. The planning domain contains knowledge about the robot and general manipulation constraints. Possible grasps for known objects in the scene are precalulated and available from a database [13,8]. The output of the planning component is a sequence of atomic robot actions, linked to their executing robot components.

Verification. In this step, the resulting action sequence is evaluated for its feasibility on the given scene. Arm movement should be guaranteed to be performed without any collision. Minor variations of object position within the accuracy of the observing sensors should have a low probability to interfere with the plan. Methods from the scene mapping can be used to ensure consistency between the simulated and the planed execution. Reached goals are checked for their persistence during further execution. For instance, an object at its final target spot may not be moved again.

Execution. Finally, the planed action sequence is executed on the robot. Methods from verification are used to keep the actions consistent with the manipulation goal.

4 Planner

The used planner uses time lines to represent the course of actions and the state of the world. A time line is a predicate with a set of possible stats for that predicate. For each point in time, exactly one of those states holds. Different states are connected by Allen relations [1]. Allen relation describe temporal relations between two entities with temporal extension like "met-by", "contained-by", "before" or "after".

Time lines are well suited to describe states which are always assigned a value, like the location of an object. On the execution side, they can be used to model robot components, which are able to execute one operation at a time.

4.1 Goal Generation

For different actions, goals for the planer are generated on-line. Pick-and-place operations result in “At(object, location)” goals. Complex manipulation actions may require specific states of the manipulated objects, which have to be provided by the action definition. Some learned actions are bound to offline generated trajectories, which have to be collision free. The planner is able to guarantee this. Therefore, the free space is sampled from arm configurations and projected onto the 2d table. For the intersection of generated positions and the free space, “location free” goals are generated. The planner then can provide actions to move the object to an available free location.

4.2 The Planning Domain

The planning domain is organized in three layers. We define an abstract layer, which mainly deals with objects and their locations. It also provides special object features like “switched on” and the causing action “switch on”. This layer does not model, how these effects are achieved, only their logical dependencies.

Actions in the abstract layer are accompanied by robot-actions in the robot layer. Here, the robot is viewed as a set of abstracted components, for instance hand and arm for grasping.

On the lowest level, robot-actions are mapped to atomic actions for robot components, like a robot arm and the robot commands necessary to operate the robot.

4.3 Implementation

The Europa PSO [3] framework is used for plan generation. Europa is a constraint based planner which supports, to some extend, numerical values and resource planning. Planning domain, goal and initial scene are described in the new domain description language NDDL. Most of the static scene and robot knowledge is provided by manually created NDDL sources. Scene information from observation and goals are compiled into NDDL during runtime.

The planner generates a partially ordered set of actions which achieves the desired goal or times out, if there is not solution found within a given time. The resulting actions are then ordered by their earliest possible execution time with respect to resources required for their execution. The planed sequence is executed in the framework described in [11].

5 Experiments

5.1 Pick-and-Place

We evaluated the proposed system on our bimanual dexterous demonstrator. In the first experiment, the robot is commanded to place a tin in the center of a table, where the targeted spot is blocked by a box. Setup, plan, generated robot program and the simulated execution are shown in Fig. 2. The generated plan is shown in the Gantt diagram on the top of the figure.

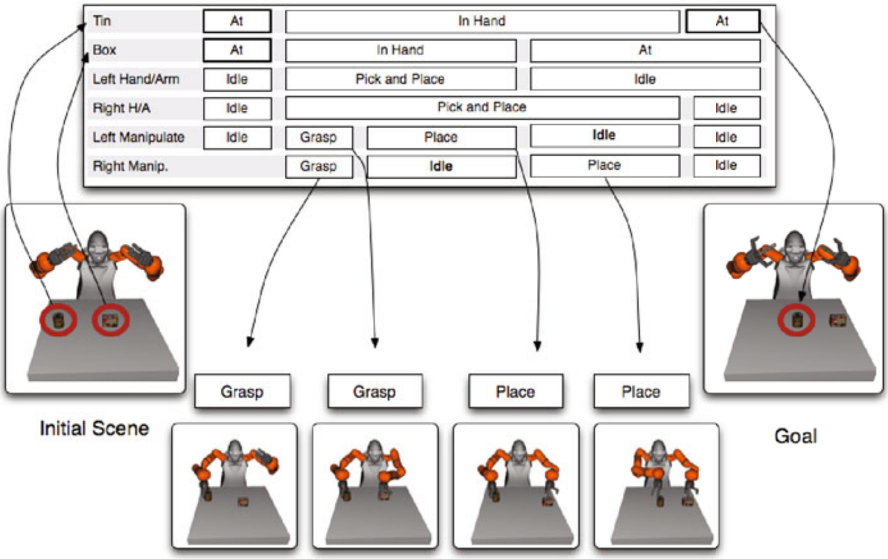


Fig. 2. First evaluation experiment: The targeted spot for the tin is blocked by another object

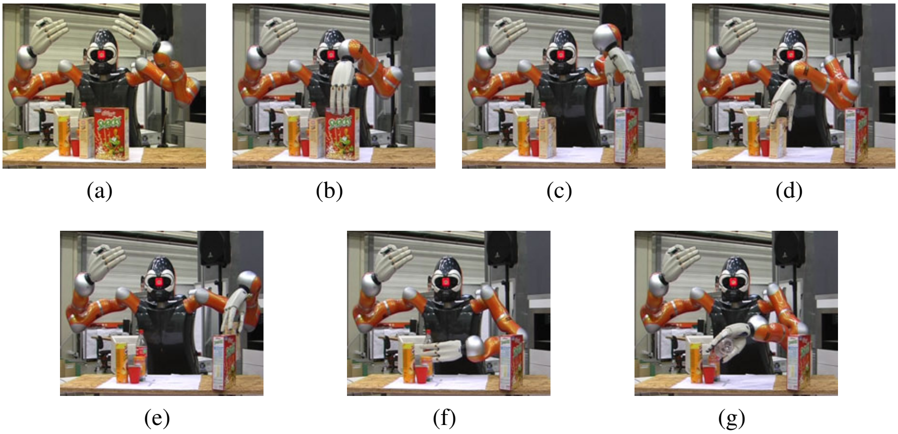


Fig. 3. Evaluation on a real robot. The robot is commanded to execute a "pour-in" operation an a cluttered scene. Therefore, obstacles have to be moved away from the cup.

The top time lines show the most abstract level, the location of the two objects. The location of an object is modeled with the "At" relation. All object positions are known from the scene mapping and are modeled in the planning domain. The only possible action in this domain is "In-Hand", indicating that the object is somehow manipulated by the robots. The details of this manipulation are specified in the next lower level. On this level, only the effects from and on the location are considered. "In-Hand" must meet (precondition) an "At" relation at its start, ending this "At" relation. This models that the object is taken away from its initial position. And, at the end of the manipulation, an new "At" relation starts (effect) for a location that was not occupied by an object before. It is planned, that both objects have to be moved to an new location, where the target location for the tin is taken from the symbol set from the discretization step.

The two lines in the middle of the diagram show the association of the abstract actions to composed robot components and actions. Both object displacements are linked to pick-and-place actions. The "pick-and-place" action must be linked to an "In-Hand" action on the high level with the same start and goal position as the high level action. The planer selects one of a set of possible grasps and an executing manipulator component. This component has to be able to execute the "pick-and-place" action based on its kinematic capabilities, which is modeled as a precondition.

The lowest level shows atomic robotic actions. It is a set of atomic actions for each action on the mid-level and their ordering constraints. Those actions can be mapped directly to the robot. Note, that the grasp actions are planned parallel since they are independent of each other. The place actions are scheduled in sequence, since in the model, the place action provides the location free state, which is a precondition for the place operation in the center.

5.2 Pour-In

The second experiment shows a more complex task. Here, the goal is to execute a learned [6] pour-in operation. The learned trajectory collides with multiple objects in the initial scene. The planning process detects those collisions, generates two alternative positions for the blocking objects and schedules two pick-and-place actions to move the objects out of the way. Finally, the generated pour in manipulation is executed. Pictures of the planned sequence, executed on our demonstrator, can be seen in Fig. 3.

6 Outlook

So far, we have focused on the discretization and planning part of the proposed system. Next, we will improve the heuristics used for location generations and introduce additional knowledge into the planning domains. Especially, knowledge of the initial scene can be exploited for planer configuration. To have a better model, the initial scene can be improved by generating additional relations from the observations, which can be used for finer modeled preconditions. E.g. sensor uncertainty could be used as a precondition in the planner, to prevent grasping object and instead schedule a localization action.

The planning domain will always be a approximation of the real world. Hence, plans which do not lead to the desired goal can be generated. To detect those plans, the verification will be extended to a more precise simulation. Based on its results, generated plans can be modified or discarded in a loop with the planner. Finally the most promising action is selected for execution, according to ratings based on the simulation results.

7 Conclusions

We have shown an approach to utilize an AI planner to generate plans for a service robot in a real world environment. It maps an observed scene to a symbolic description. A discrete set of symbols is generated from the Cartesian manipulation space.

Generated plans are verified in a simulation environment. Using the proposed system, our manipulation demonstrator was able to carry out learned manipulation actions in a previously unknown cluttered environment.

References

1. Allen, J.: Maintaining knowledge about temporal intervals. Communications of the ACM 26(11), 832–843 (1983)
2. Choi, J., Amir, E.: Combining planning and motion planning. In: ICRA 2009: Proceedings of the 2009 IEEE International Conference on Robotics and Automation, pp. 4374–4380. IEEE Press, Piscataway (May 2009)
3. Daley, P., Frank, J., Iatauro, M., McGann, C., Taylor, W.: Planworks: A debugging environment for constraint based planning systems. In: Proceedings of the International Conference on Knowledge Engeneering in Plannig and Scheduling, Monterey, California, USA, pp. 22–26 (June 2005)
4. Dornhege, C., Gissler, M., Teschner, M., Nebel, B.: Integrating symbolic and geometric planning for mobile manipulation. In: IEEE International Workshop on Safety, Security and Rescue Robotics, SSRR (November 2009)
5. Grundmann, T., Eidenberger, R., Zoellner, R.D., Xue, Z., Ruehl, S., Zoellner, J.M., Dillmann, R., Kuehnle, J., Verl, A.: Integration of 6d object localization and obstacle detection for collision free robotic manipulation. In: IEEE International Symposium on System Integration (SII) (December 4, 2008)
6. Jaekel, R., Schmidt-Rohr, S., Xue, Z., Loesch, M., Dillmann, R.: Learning of probabilistic grasping strategies using programming by demonstration. In: IEEE International Conference on Robotics and Automation, Anchorage, USA (May 2010)
7. Kuehnle, J., Xue, Z., Grundmann, T., Verl, A., Ruehl, S., Eidenberger, R., Zoellner, J.M., Zoellner, R.D., Dillman, R.: 6d object localization and obstacle detection for collision-free manipulation with a mobile service robot. In: 14th International Conference on Advanced Robotics (ICAR) (June 22-26, 2009)
8. Miller, A., Allen, P.: Graspit! a versatile simulator for robotic grasping. IEEE Robotics & Automation Magazine 11(4), 110–122 (2004)
9. Nguyen, X., Kambhampati, S., Nigenda, R.S.: Planning graph as the basis for deriving heuristics for plan synthesis by state space and csp search. Artificial Intelligence 135(1-2), 73–123 (2002)
10. Nilsson, N.J.: Shakey the robot. Tech. Rep. 323, SRI International (1984)
11. Ruehl, S.W., Xue, Z., Zoellner, J.M., Dillmann, R.: Integration of a loop based and an event based framework for control of a bimanual dextrous service robot. In: IEEE International Conference on Robotics and Biomimetics, ROBIO (2009)
12. Vahrenkamp, N., Berenson, D., Asfour, T., Kuffner, J., Dillmann, R.: Humanoid motion planning for dual-arm manipulation and re-grasping tasks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA (October 2009)
13. Xue, Z., Kasper, A., Zoellner, J.M., Dillmann, R.: An automatic grasp planning system for service robots. In: 14th International Conference on Advanced Robotics (ICAR) (June 22-26, 2009)
14. Zacharias, F., Borst, C., Hirzinger, G.: Capturing robot workspace structure: representing robot capabilities. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, California, USA, pp. 3229–3236 (October 2007)

Towards Opportunistic Action Selection in Human-Robot Cooperation

Thibault Kruse and Alexandra Kirsch

Intelligent Autonomous Systems Group
Department of Informatics
Technische Universität München*

Abstract. A robot that is to assist humans in everyday activities should not only be efficient, but also choose actions that are understandable for a person. One characteristic of human task achievement is to recognize and exploit opportunities as they appear in dynamically changing environments. In this paper we explore opportunistic behavior for robots in the context of pick and place tasks with human interaction. As a proof of concept we prototypically embed an opportunistic robot control program, showing that the robot exhibits opportunistic behavior using spatial knowledge, and we validated the feasibility of cooperation in a simulator experiment.

1 Introduction

Robots have been identified as a promising approach to enable a more independent life for elderly people. In many cases, elderly people can perform most of their everyday activities on their on, but need assistance in a small number of tasks, that are notwithstanding crucial in the everyday course of life. Our vision is a robot assistant that offers its help to an elderly person to assist in everyday tasks such as preparing meals. In particular, we are interested in planning and plan execution that allow a smooth and natural collaboration.

Human beings are a factor that make the environment of a robot particularly dynamic and difficult to predict. On the other hand, we want to be sure that a human understands what the robot is doing and intending — in other words we want the robot to show not only efficient but also *legible* behavior.

When humans perform tasks, they adapt their behavior to the state of the environment by exploiting opportunities. We think that opportunistic behavior of a robot can benefit its acceptance in two ways: (1) the achievement of goals can be expected to be more efficient, which is probably the underlying reason why humans exploit opportunities and (2) the behavior of the robot will be more human-like and therefore easier to predict for a person. In this paper, we are mostly concerned with the second aspect, although we believe that an efficiency gain can also be expected.

* This work was partially funded by the cluster of excellence CoTeSys and the BFHZ.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 374–381, 2010.

In the context of a robot, we define an opportunity as an alternative or following action to the robot's current action. By this definition, the choice of the next action also falls in the framework of opportunistic behavior. When considering the next action to take, the different choices can be considered as different opportunities.

The concept of opportunistic planning relies on the detection of opportunities. In principal, opportunities could be defined by a variety of models, which might depend on the execution context. In our work, we have considered one specific model for recognizing opportunities for pick-and-place tasks by using the "reachability" of an object as a measure to rate the desirability of an (alternative) action. The criterion of whether an object can easily be reached is determined with respect to the positions of humans in the world. This means that an object, which is spatially near the robot, but which is blocked by a human, is ranked as less reachable than an object to which a free path can be found. This phenomenon is illustrated in Figure 2a. Although the robot ("Auto") is nearer to place A, the object lying at place B are ranked as better reachable, because the "human" blocks the way to the objects at A.

In this paper, we introduce a general concept of opportunistic planning and show its application to a realistic robot application. We attempt to validate its usefulness with respect to the legibility and acceptance of humans.

The next section puts our work in perspective with related work. After that we introduce our general concept of opportunities and details of our specific implementation for a robotic household assistant, which is followed by a first validation of our proof-of-concept implementation. We end with conclusions and future work.

2 Related Work

The conflicting needs of planning versus reactivity also shows in the classification by Hayes-Roth [5], who describes a continuous space of agent control modes. According to that agent design needs to balance early commitment on actions and sequences with reactivity, to avoid the lack of opportunism that comes with early and persistent commitments.

The same argument is made by Schut and Woolridge [11] who describe a framework that allows agents to make commitment choices at runtime rather than at design time, hence they show this approach to outperform agents with fixed policies. They use the well known BDI framework [1,4] for their agent design, to embed reactive planning as intention reconsideration in a given control loop of the BDI framework.

They embed the intention reconsideration into a loop, which in sequence reconsiders and executes. However robotics and HRI require even more opportunism than that, because during the execution of an action opportunities can also occur, such as a human giving way to a previously blocked position. The efficiency gain of opportunism mostly be achieved in this kind of plan-execute-loop, however added legibility is achieved by also recognizing opportunities that occur during the execution of a task. The additional efficiency gains to be expected from that depend on the duration of actions and the frequency of such spontaneous opportunities.

Their computation of the utility of actions is effectively a reactive planner, whose role is to decide which of the currently possible next actions in the plan should currently be pursued. Hayes-Roth has formulated that this way: "At each point in time, many actions may be logically possible, given the current state of the task environment. An intelligent agent must choose among them, either implicitly or explicitly." [6].

This kind of parallel reconsidering of choices has also been researched in [10] using goodness, competence and conflict as categories for action selection criteria. The robot Milou in that work had a controller which would review alternative navigation plans (what could be considered actions in action selection) every 100ms with regard to three fuzzy measures: goodness, competence and conflict. Goodness is an a priori priority ranking of actions, competence the degree by which the actions are currently possible, and conflict the degree by which one action conflicts with parallel actions. Their robot however has no notion of humans in the environment or of human comfort.

To get more information about related action selection algorithms, we recommend the survey in [2].

3 Approach

The basic procedure of our approach is illustrated in Figure 1. The plan, which is executed by the robot is a reactive plan. For the representation of such plans we use the Reactive Plan Language [9], which allows to reason about and change an existing plan at run-time.

The plan execution is constantly monitored by a parallel process that while the current action can be preempted, generates potential opportunities (i.e. alternative actions) and evaluates them according to a given model. If an opportunity is found to be more desirable in the current situation, a second deliberation process ("opportunity filter" in Figure 1) has to check whether the switch between the actions is worth the gain predicted by the model to evaluate opportunities. This step is important to ensure a certain commitment of the robot to the action it has decided upon. By adding costs of switching, the risk of a livelock is reduced. Although in a highly dynamic environment in which humans constantly change the world livelocks might be unlikely, the legibility of the robot's behavior would suffer by constantly re-deciding its course of action.

This mechanism of choosing opportunities is also valid for the choice of the next action if that has not been decided yet. In this case, the plan is empty and there are no switching costs for setting the next action. The mechanisms of choosing the next action is a greedy choice of the action with the highest utility value in the situation.

In the procedure as we have described it, and also in the specific implementation that we will describe next, the whole procedure is completely reactive.

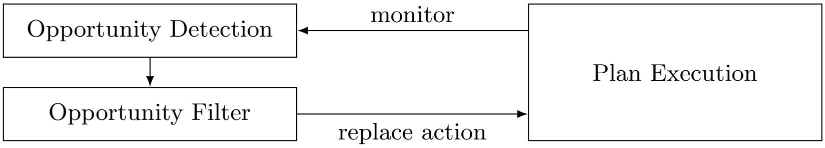


Fig. 1. General procedure for detecting and using opportunities

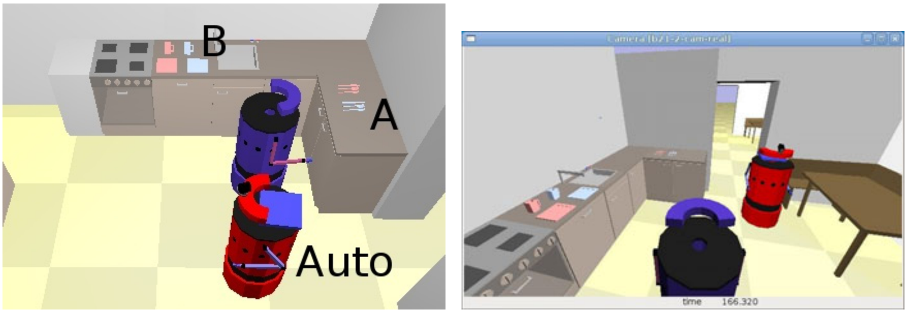
The robot has a set of goals to achieve, which all have the same a priori utility, but may have different dynamic utility. The robot picks goals to achieve greedily from this set of goals and possibly interrupts some action if a better alternative turns up. In this paper, actions could only be interrupted when the robot was in the state of moving towards an item to pick up. However, our general idea of opportunistic planning is that there are some parts of the robot's course of actions that are planned (possibly offering alternatives at execution time as the Reactive Plan Language allows) and other parts can be marked as possible candidates to exploit opportunities.

In our current implementation of the procedure of Figure 1, we want a robot to set the table together with a person, where the goal positions of all objects are predetermined. This scenario allows us to reduce the set of actions to "atomic" pick-and-place tasks, which means that action selection narrows down to the choice of the object to take next. In the general case, the granularity for opportunistic selection is a design choice that affects how much the robot can exploit opportunity as well as how much computation is required at run-time to compare potential actions.

Because our actions are narrowed down to the choice of objects, we defined our model for evaluating an opportunity based on the human-aware reachability of the objects. The idea is to calculate the costs for a path from the robot's current position to a position from which it can grip a certain object. The Human-Aware Navigation Planner (HANP) [12] calculates navigation paths for a robot that take into account human comfort as well as the usual safety criteria to avoid obstacles. We use a variant of HANP [8] that takes into account that the humans in the world are also actively involved in the task and are therefore moving most of the time. We use this navigation planner for the navigation of the robot, but also as a model to predict, which objects are reachable in a human-friendly way, and thus as an evaluation function for opportunities.

Using HANP rather than the Euclidean distance or other simpler heuristics has the benefits of both considering the spatial geometry of the environment (walls and furniture) as well as the comfort of the humans present. Also consider the situation in Figure 2a, where spot A is closer to the robot "Auto", but obstructed by the other agent. Whether at any given time a special opportunity exists for the robot to save overall time can thus be determined as a function of the minimal navigation costs to move to a location of action.

We don't claim that the human-aware reachability of an object is the only good model that can be employed in this context. Other ideas are to consider the reachability of the position where the object has to be put down, the feasibility of subsequent actions (putting an object at the edge of the table can make it more difficult to put other objects into the middle of the table afterwards), as well as preferences and abilities of the human collaborator. But when choosing a model, it is important to ensure that it can be calculated reasonably fast. Our current model is in the order of a second to evaluate all alternatives, which is still acceptable. Another consideration would also be how to filter the possible alternative actions to be evaluated.



(a) (b)

Fig. 2. (a): Robot "Auto" decides between picking up knife at A or plate at B. (b):User-study simulation, user perspective, steered robot at bottom.

For identifying opportunities that are worth to abandon the current action, we apply a small threshold equivalent to the costs of moving roughly 50cm, such that the robot will not oscillate between 2 opportunities due to mechanical constraints. When there is no current action, the threshold is 0.

4 Validation

The validation of opportunistic action planning in human robot interaction requires a setting where the mechanical, spatial and temporal constraints of robots are influencing results. Real robot prototypes have the disadvantage of being unreliable for complex tasks and a safety risk to human subjects.

As a compromise we chose to use a 3D simulator already used in robotics and augment it with a virtual agent to be steered by a human.

In order to show the applicability of our approach, we set up a simulation environment where a human can interact with a robot in a household task of setting the table [7]. In the simulation, two robot-shaped agents are able of picking and placing household items in a 3D kitchen environment, one of them would be steered by a human, the other by an autonomous robot controller.

The human perspective when steering the robot is seen in Figure 2b, it is somewhat behind the robot as suggested to be optimal in [3].

The maximal movement speed for both robots was set to 0.5m/second. Grasping and dropping items in the simulator took about 15 seconds. Those times are constrained by the CPU load for simulation and the controllers of the virtual agents. The autonomous robot would take 25 seconds with most of the time spent on turns on the spot, and slow average velocities while following waypoints. The virtual robot steered by a human would travel similar distances in about 10 seconds.

One of our own trials, shown in Figure 3 illustrates the opportunistic behavior of the robot. The initial situation that was used was one where the robot R would be closer to items at A than items at B. Immediately after launching the robot controller, the Human H was moved towards the Items A. As expected, the Robot R changed its intention when the human presence to A made this paths more costly than a path to B. The independent looping cycles calculating the costs for all the available intentions (10 in this case) too 2 seconds.

We then performed tests with four different human subjects, male and female students of different disciplines, having to set the virtual table with six items (plates, cups, knife). Due to minor failures such as virtual agents dropping items, some trials became invalid and had to be repeated.

Seconds	Log message
587	Executing plan
587	There are 10 open goals to perform next
588	recalculated costs
588	Performing intention: (PICK THE ENTITY (NAME SPOON–BLUE))
590	recalculated costs
591	recalculated costs
593	recalculated costs
595	recalculated costs
597	recalculated costs
597	A different intention has become more attractive: Item THE ENTITY (NAME CUP–BLUE).
597	Performing intention: (PICK THE ENTITY (NAME CUP–BLUE))
599	recalculated costs

Snapshots

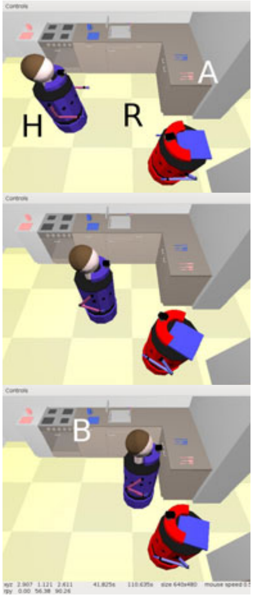


Fig. 3. Logfile and snapshots showing opportunistic change of intention due to spatial circumstance. At time 588 the robot R decided to next pick item SPOON-BLUE shown as location A, and at time 597, the robot instead decided to pick item CUP-BLUE at location B, EVEN though SPOON-BLUE was still closer given Euclidean distance, but has higher path costs due to the human H having moved.

In the experiments the robot calculated the attractivity of open tasks in 1-3 seconds, which was acceptable to the users. When moving to the target, recalculating the attractiveness in parallel took up to 4 seconds, as the motion planner was then also used for actual motion planning, not just for cost estimation. However the robot continuing to its currently committed goal was not adversely impacted by that calculation in parallel.

This shows the approach to use the motion planner for opportunistic action is possible given real-time constraints.

Only in 3 of those 48 moves did the robot change its choice of what item to move next. This is easily explained by the speed of the simulator. In most cases when the robot made an initial choice for an item to pick, the human was already busy in a location of the kitchen and did not move during the robot approach of its target, only in 3 instances did the human move and thus made a different item more attractive.

The subject's rating of the assisting robot was captured on a scale from 1 to 5, 1 meaning not helpful and 5 meaning very helpful, and the average score was 4.2. The subjects were also given the opportunity to give general comments, and the only comments given were about the simulator slowness and of failures due to dropped items, not about the robot behavior.

The experiments could not serve as a basis for further statements about the algorithms, but the mere lack of complaints about the robot behavior may serve as an indication of successful cooperative behavior. The subjects did also not seem to be distracted by the fact that no explicit communication was possible between the autonomous robot controller and the human.

The performance of the plans in our scenario varied a lot due to arbitrary factors, so that even when the robot changed plans, we could not measure the benefit in terms of efficiency.

5 Conclusion and Future Work

We have introduced a general framework for detecting and using opportunities in collaborative human-robot tasks. We have focused on a specific model to identify opportunities — the reachability of places for manipulation considering human comfort in navigation tasks.

We combined opportunistic action selection with a model of human-centric motion planning, using a model of opportunities that allows to suppress actions during their execution in favor of other actions, improving the legibility of the set of robot goals.

We conducted a feasibility study with humans, in a realistic joint table setting scenario in simulation, in which the greedy action selection and opportunistic action switching worked as expected.

One thing we could see from preliminary trials is that when in that table setting scenario an agent had to wait once for the other, from then on the human and robot agent did not compete for the same spots, but visited shared spots of interest alternatingly. This is also a consequence of the rather similar individual pick and place tasks taking the same amount of time. How often such rhythmic patterns occur in real-world cooperation is yet unknown, but could serve as a model for temporal alignment of actions.

We also intend to generalize opportunistic action selection to a finer granularity of tasks, such as combining and interleaving actions, such as picking up one item per robot arm. So far in experiments we only allowed for opportunistic selection of pick and place actions as a whole. A higher impact on robot efficiency (and thus hopefully also on human satisfaction) may be achieved by reactively planning a bit more into the future, such as not just considering the effort to a pickup task, but also the expected effort of placing it in the immediate future. In our example this could help the robot to decide setting the table for a different person, and thus conflicting less at the table.

As research exists on stochastic prediction of human space occupancy for trajectory planning[13], a combination of those works with ours would be promising.

References

1. Bratman, M.: Intention, Plans and Practical Reason. Harvard University Press, Cambridge (1987)
2. Brom, C., Bryson, J.: Action selection for intelligent systems. In: European Network for the Advancement of Artificial Cognitive Systems (2006)
3. Bruemmer, D.J., Few, D.A., Nielsen, C.W.: Spatial reasoning for human-robot teams. In: Hilton, B.N. (ed.) Emerging Spatial Information Systems And Applications, Hershey, PA, USA, pp. 350–372. IGI Publishing (2006)
4. Georgeff, M.P., Pell, B., Pollack, M.E., Tambe, M., Wooldridge, M.: The belief-desire-intention model of agency. In: Rao, A.S., Singh, M.P., Müller, J.P. (eds.) ATAL 1998. LNCS (LNAI), vol. 1555, pp. 1–10. Springer, Heidelberg (1999)
5. Hayes-Roth, B.: Opportunistic control of actions in intelligent agents (1992)
6. Hayes-Roth, B.: Intelligent control. Artificial Intelligence 59, 213–220 (1993)
7. Kirsch, A., Chen, Y.: A testbed for adaptive human-robot collaboration. In: Proceedings of the 33rd Annual German Conference on Artificial Intelligence (2010)
8. Kruse, T., Kirsch, A., Sisbot, E.A., Alami, R.: Dynamic generation and execution of human aware navigation plans. In: Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems, AAMAS (2010)
9. McDermott, D.: A reactive plan language. Technical report, Yale University, Computer Science Dept. (1993)
10. Saffiotti, P.P., Parsons, S., Pettersson, O., Saffiotti, A., Wooldridge, M.: Robots with the best of intentions (1999)
11. Schut, M., Wooldridge, M.: Principles of intention reconsideration. In: AGENTS 2001: Proceedings of the Fifth International Conference on Autonomous Agents (2001)
12. Sisbot, E.A., Marin-Urias, L.F., Alami, R., Simeon, T.: A human aware mobile robot motion planner. IEEE Transactions on Robotics 23, 874–883 (2007)
13. Ziebart, B., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J.A.D., Hebert, M., Dey, A., Srinivasa, S.: Planning-based prediction for pedestrians. In: Proc. IROS 2009 (October 2009)

Trajectory Generation and Control for a High-DOF Articulated Robot with Dynamic Constraints

Marc Spirig, Ralf Kaestner, Dizan Vasquez, and Roland Siegwart

ETH Zurich, Switzerland
mspirig@ethz.ch

Abstract. In this paper, we propose a novel and alternative approach to the task of generating trajectories for an articulated robot with dynamic constraints. We demonstrate that by focusing the effort on the generation process, the design of a trajectory controller becomes a straightforward problem. Our method is efficient and particularly suited for applications involving high-DOF articulated systems such as robotics arms or legs. We claim that our algorithm can be easily implemented by roboticists that do not share a deep background in control theory. Nevertheless, the resulting trajectories ensure a robust state-of-the-art control performance. We show, in simulation and practice, that the approach is well prepared for integration with graph-based planning techniques and yields smooth trajectories.

1 Introduction and Related Work

The problem of trajectory generation for articulated systems is often faced in robotics. Whereas graph-based planning approaches [4,11,7] have become increasingly popular during the last decade, plan execution on platforms with dynamic limits constitutes a challenging task. Given a set of waypoints, no closed-form solution exists to generate trajectories that take into account the velocity and acceleration constraints of a robot. In fact, these constraints are often neglected by many authors. Alternatively, previous work in the field of mobile robotics exclusively considers velocity parameters [5,9] or acceleration limits [6,10] whilst ignoring the other constraint.

Special types of mobile platforms such as soccer-playing robots have introduced popular application scenarios for trajectory generation under dynamic constraints. Plenty of work has been published within that field, and the problem is generally solved using Bezier curves [3]. However, in most safety-concerned domains, Bezier curves yield widely impracticable solutions. This is due to the fact that Bezier curves merely approach the states given by the planner whilst not reaching them accurately.

In the area of manipulator robots, the task of trajectory generation with dynamic constraints has received surprisingly poor attention in the past. Here, the focus commonly lies on trajectory optimization, and most solutions imply a complex controller design [1,8].

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 382–391, 2010.

In this paper, we propose a novel and alternative approach to the problem. We will demonstrate that, by focusing the effort on generating well-suited trajectories for an articulated system, controller design becomes significantly less challenging. We claim that our approach can be easily implemented by roboticists that do not share a deep background in control theory. Nevertheless, the trajectories resulting from the proposed method are computationally inexpensive and ensure a robust state-of-the-art control performance.

The proposed technique has been optimized for applications involving high-DOF articulated systems such as robotics arms or legs. Trajectory generation considers both acceleration and velocity constraints under the condition that all configuration states are accurately reached. We will show that in association with a static, graph-based planning algorithm, our approach may thus be used to generate control-friendly trajectories that fulfill the kinematic as well as the dynamic constraints of a robotics system. We conclude our paper with the design and analysis of a suitable controller which allows for smooth trajectory tracking and rectifies disturbances.

2 Configuration, Path, and Trajectory

We will first introduce a formal description of the state model used throughout this work. To be general, we consider an articulated system with K degrees of freedom (DOF). The k-th degree of freedom is parameterized by the state variable θ_k. With k ranging between 1 and K, we may thus express the *configuration* of the entire system as $\Theta = \{\theta_1, \ldots, \theta_K\}$.

We furthermore consider a static, graph-based motion planner to provide a sequence $\mathcal{P}$ of collision-free configurations to the robot. With N being the number of configurations given by the planner, the i-th configuration in $\mathcal{P}$ shall be denoted Θ_i. Thus, we have $\mathcal{P} = \{\Theta_1, \ldots, \Theta_N\}$. Note that in the context of path planning, the Θ_i are commonly termed *waypoints*, and $\mathcal{P}$ constitutes a *path.*

The plan $\mathcal{P}$ is the starting point for the trajectory generation approach presented in this paper. Therefore, it is important to remark that our static motion planner does not provide any notion of time or intermediate configurations with the Θ_i. However, in order to fully describe a dynamic articulated system those intermediate configurations need to be known explicitly. In other words, we seek to express the state of the k-th degree of freedom through a continuous function $\mathcal{T}_k : t \rightarrow \theta_k$ with t representing time. Note that we will henceforth use the common term *trajectory* in order to refer to $\mathcal{T}_k$.

In the following sections, we will mainly be concerned with deriving a parametric form of $\mathcal{T}_k$. However, solving this problem will involve interpolation methods that would require the time intervals between consecutive configurations to be known. Since time being is free parameter of our method, we instead need to infer $\mathcal{T}_k$ based on dynamic quantities such as the assumed maximum velocity $\dot{\theta}_k^{max}$ or the assumed maximum acceleration $\ddot{\theta}_k^{max}$ of the k-th degree of freedom. For a mobile system, one may for example want $\dot{\theta}_k^{max}$ and $\ddot{\theta}_k^{max}$ to correspond to the safety limits of the motor used to actuate the k-th degree of freedom.

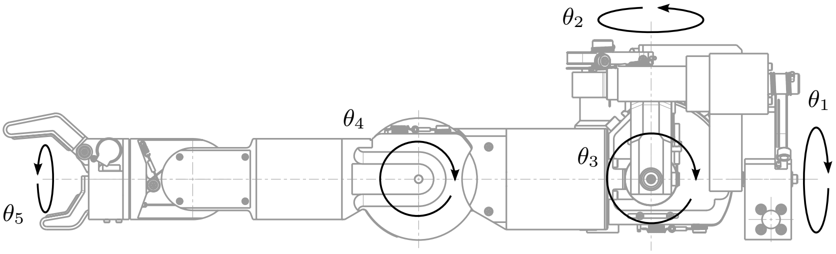


Fig. 1. The embedded robot arm is a 5-DOF articulated system with dynamic constraints. Its configuration may be fully described by the inscribed joint angles θ_1 to θ_5.

To further motivate the problem discussed above, we will introduce an exemplary articulated system with 5 degrees of freedom. Fig. 1 shows the embedded robot arm that has been used in the experimental validation phase of this work. In the application domain of this arm, safety and precision are essential components that may be vastly improved by advanced trajectory generation and tracking methods.

3 Trajectory Generation

In this section, we will present our technique to generate continuous trajectories from discrete paths and a set of dynamic constraints.

As presented above, the input of the trajectory generation algorithm consists of a sequence $\mathcal{P}$ of configurations $\Theta_i = \{\theta_1^i, \ldots, \theta_K^i\}$ provided by the motion planner. In order for the robot to accurately attain the Θ_i, all θ_k^i are required to be reached simultaneously. Moreover, additional information comes in the form of the limit quantities $\dot{\Theta}_{max} = \{\dot{\theta}_1^{max}, \ldots, \dot{\theta}_K^{max}\}$ and $\ddot{\Theta}_{max} = \{\ddot{\theta}_1^{max}, \ldots, \ddot{\theta}_K^{max}\}$. Another natural constraint is given by the velocities at the first and last waypoint, namely $\dot{\Theta}_1$ and $\dot{\Theta}_N$, respectively. Those velocities represent the system's dynamic configuration in the start and the goal state and are set to 0.

A feasible trajectory should thus be defined to comply with the dynamic limits of the system. In contrast, it should not result in an over-conservative motion of the robot.

We suggest to generate feasible trajectories in two stages. First, we estimate the minimum transition times ΔT_i between consecutive waypoints Θ_i and Θ_{i+1} under the assumption of hard dynamic constraints. As we shall see, we thus obtain an intermediate solution which does not satisfy the requirement for a globally optimal motion of the robot. In a second iteration, we therefore apply the estimated time intervals ΔT_i to derive significantly smoother trajectory functions $\mathcal{T}_k$. Although the $\mathcal{T}_k$ are not strictly limited by the hard constraints, we will demonstrate that constraint violations are bound and that the solution provides a feasible input to our trajectory controller.

3.1 Estimating Transition Times

The *Three-cubic Method* introduced by [12] is a procedure that is based on subdividing a profile into three cubic segments, each of which is represented by a third-order polynomial. Their parameters, 12 in total, can be expressed using the static end-point configurations θ_k^i and θ_k^{i+1}, the velocities $\dot{\theta}_k^i$ and $\dot{\theta}_k^{i+1}$ and the time interval Δt_k^i in between. For calculating Δt_k^i, the acceleration and velocity constraints are considered simultaneously. With the obtained estimate for the transition times, it is thus possible to calculate the parameters of the third-order polynomials. The resulting piece-wise function is globally $\mathcal{C}^2$-steady and may therefor directly serve as trajectory solution.

Further information about the Three-cubic Method can be found in the corresponding literature [12]. For K degrees of freedom, the method has to be applied to all N segments and to all K components of the configuration vector. Thus, for the i-th waypoint the algorithm gives a set of K time estimates $\{\Delta t_1^i, \ldots, \Delta t_K^i\}$, one for each degree of freedom. Since we seek to reach all configurations simultaneously without violating any of the dynamic constraints, the largest value in the set of transition times has to be chosen as time interval ΔT_i for generating the i-th trajectory segment.

One aspect that has to be taken into consideration is the fact that the Three-cubic Method expects the velocities in the segment end-points $\dot{\theta}_k^i$ and $\dot{\theta}_k^{i+1}$ to be given. However, following our assumptions these velocities are generally unknown and their estimation seems mostly delicate. We therefore propose to set $\dot{\theta}_k^i$ and $\dot{\theta}_k^{i+1}$ to 0. In turn, this assumption vastly simplifies the Three-cubic algorithm.

Alg. 1 states the application of the Three-cubic Method for the estimation of the i-th transition time ΔT_i for multiple degrees of freedoms. As already proposed by the authors, the datum n defining the division length of a cubic segment into three sub-segments should be set to 16 in order to yield good results.

3.2 Generating Smooth Trajectories

A major problem of directly using the Three-cubic polynomials as segments of a piece-wise trajectory model results from the fact that the start and end-point velocities were chosen to 0. Following such a trajectory would cause our robot to stop and re-accelerate at each of the waypoint configurations Θ_i. Clearly, this will cause a non-smooth navigation behavior of the system whilst consuming additional energy and time.

We therefore suggest to apply the Three-cubic Method only for computing the time intervals ΔT_i. In order to generate globally smooth trajectories $\mathcal{T}_k$, we then use the ΔT_i to perform *cubic spline interpolation* with known intervals.

Our algorithm for the generation of smooth trajectories from a sequence of waypoints is stated in Alg. 2. Here, each cubic spline segment starting with θ_k^i and ending with θ_k^{i+1} is described by the third-order polynomial

$$\mathcal{T}_k^i(t) = a_k^i + b_k^i t + c_k^i t^2 + d_k^i t^3 \tag{1}$$

Algorithm 1. estimateTransitionTime(Θ_i, Θ_{i+1})

```
Input: Consecutive configurations $\Theta_i$ and $\Theta_{i+1}$
Input: First- and second-order dynamic constraints $\dot{\Theta}_{max}$, $\ddot{\Theta}_{max}$
Output: Transition time $\Delta T_i$
foreach $\theta_k^i \in \Theta_i$ do
    if $\theta_k^i \neq \theta_k^{i+1}$ then
        $\Delta t_{k,acc}^i \leftarrow \sqrt{\frac{6 \cdot n}{(n-1) \cdot \ddot{\theta}_k^{max}} \cdot \left|\theta_k^{i+1} - \theta_k^i\right|}$
        $\Delta t_{k,vel}^i \leftarrow \frac{3 \cdot n}{2 \cdot (n-1) \cdot \dot{\theta}_k^{max}} \cdot \left|\theta_k^{i+1} - \theta_k^i\right|$
        $\Delta t_k^i \leftarrow \max\ \left(\Delta t_{k,acc}^i, \Delta t_{k,vel}^i\right)$
    else
        $\Delta t_k^i \leftarrow 0$
    end
    $\Delta T_i \leftarrow \max_k \Delta t_k^i$
end
```

An example trajectory generated by our method from simulated path configurations of a single degree of freedom is depicted in Fig. 2. It becomes evident that the proposed combination of the Three-cubic Method and spline interpolation produces feasible trajectories. As compared to the piece-wise solution obtained by solely applying the Three-cubic Method, the resulting trajectories do not suffer from the lack of smoothness caused by the zero-velocities in the supporting waypoints.

As a matter of fact, the additional smoothness of the trajectories comes at the price of bound constraint violations. This is due to the global nature of the spline function: in some cases, overshoots as those illustrated in Fig. 3 may

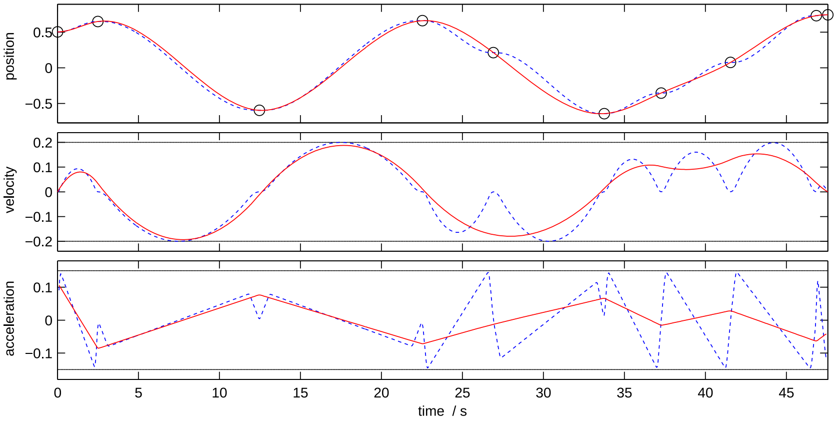


Fig. 2. Illustration of the Three-cubic Method with spline interpolation showing the Three-cubic solution (blue) and the result of the proposed algorithm (red). Waypoints are marked by circles, and horizontal lines represent the dynamic constraints.

Algorithm 2. generateTrajectories($\mathcal{P}$)

```
Input: Sequence of configurations $\mathcal{P} = \{\Theta_1, \ldots, \Theta_N\}$
Output: Set of trajectories $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_K\}$ in spline-parametric form
for $i \leftarrow 1$ to $N - 1$ do
    $\Delta T_i \leftarrow$ esimtateTransitionTime($\Theta_i$, $\Theta_{i+1}$)
end
for $k \leftarrow 1$ to $K$ do
    $\{\dot{\theta}_k^1, \ldots, \dot{\theta}_k^{N-1}\} \leftarrow$ splineInterpolation($\theta_k^1, \ldots, \theta_k^N$, $\Delta T$)
    $\mathcal{T}_k \leftarrow \emptyset$
    for $i \leftarrow 1$ to $N - 1$ do
        $a_k^i \leftarrow \theta_k^i$
        $b_k^i \leftarrow \dot{\theta}_k^i \Delta T_i$
        $c_k^i \leftarrow 3(\theta_k^{i+1} - \theta_k^i) - (\dot{\theta}_k^{i+1} + 2\dot{\theta}_k^i)\Delta T_i$
        $d_k^i \leftarrow (\dot{\theta}_k^{i+1} + \dot{\theta}_k^i)\Delta T_i - 2(\theta_k^{i+1} - \theta_k^i)$
        $\mathcal{T}_k \leftarrow \mathcal{T}_k \cup \{a_k^i, b_k^i, c_k^i, d_k^i\}$
    end
end
```

cause the first-order velocity limits to be exceeded. A closer examination of the situation reveals that the local minimum of the affected spline segment does not correspond to the supporting waypoint as would be the case for the Three-cubic solution. Hence, the maximum gradient of the spline function which is forced to go through the consecutive support point becomes larger than the maximum gradient of the Three-cubic polynomial. We will further analyze the effect of the described constraint violations in Sec. 5.

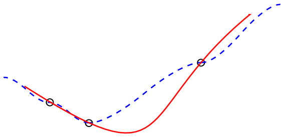


Fig. 3. The additional smoothness of the Three-cubic spline trajectories comes at the price of bound constraint violations. Thus, overshoots of the spline function (red) may cause the gradient limit of the Three-cubic polynomial (blue) to be exceeded.

4 Trajectory Controller

4.1 Control Strategy and Specifications

Given a set of smooth trajectories that widely range within the dynamic constraints of our robot, we are now ready to derive a feasible control strategy.

A first important insight that vastly simplifies our control architecture pertains lies in the fact that, by definition, all trajectories $\mathcal{T}_k$ intersect with the desired waypoint configurations. This safely allows for a decoupled design of K

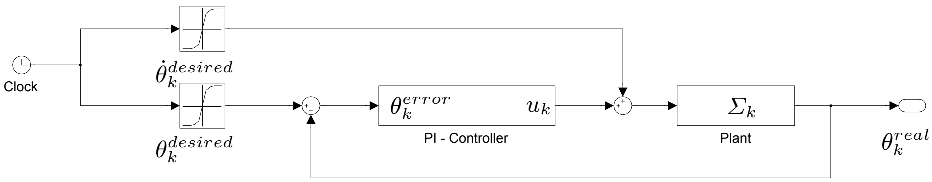


Fig. 4. Illustration of the proposed control strategy for the k-the degree of freedom

independent controllers, each of which is responsible for following the k-th input function. Furthermore, accounting for the dynamic nature of our system we want to control both the position and the velocity of the actuators. The reader may have noticed that these quantities are continuously defined by the steady trajectory functions.

The proposed control strategy can comprehensively be explained by means of Fig. 4. Here, it can be seen that the k-th actuator *plant* is simultaneously fed with the desired velocity profile $\dot{\theta}_k^{desired}$ and the desired position profile $\theta_k^{desired}$. Accounting for disturbances, a feedback control loop is implemented. The measured position θ_k^{real} is compared to the desired position and the position error θ_k^{error} is fed to a *PI* controller. The controller output u_k is then added to $\dot{\theta}_k^{desired}$ and constitutes the effective plant input Σ_k.

4.2 Controller Design

As already mentioned above, we suggest to apply the well-studied PI controller to the trajectory tracking problem. The PI controller is characterized by its ease of implementation and can be formalized as

$$C_{PI}(s) = k_p \cdot \left(1 + \frac{1}{T_i \cdot s}\right), \tag{2}$$

where k_p and T_i denote the control parameters which need to be identified. Note that various standard methods exist for parameter identification, *Åström-Hägglund* or *Ziegler-Nichols* being amongst the most common techniques. For a more detailed discussion of the topic, the interested reader may refer to [2].

In the process of design verification, stability criterions, performance measures, and simulations should be evaluated to give satisfying results. Discretization of the control architecture may then be achieved using *Tustin emulation* with a predefined sampling time.

5 Evaluation

5.1 Simulation

As robustness and stability criterions were fulfilled, performance quality of the trajectory controller was evaluated by comparison between the feedback and the

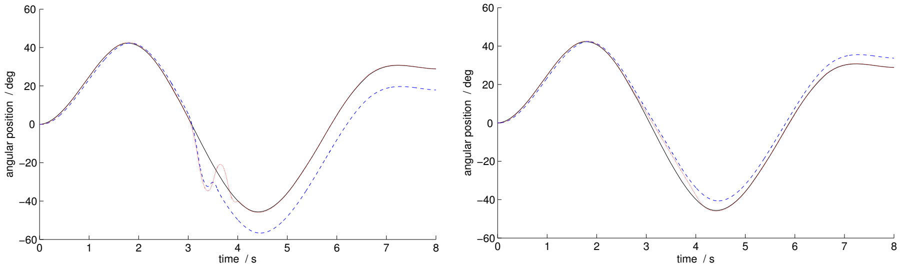


Fig. 5. Comparison of simulated trajectory tracking using feedforward (blue) and feedback control (red). In the left graph, a disturbance was given to the system. The right graph shows a situation where the controllers enter velocity saturation.

feedforward solution. Whereas tracking errors may propagate in an open-loop architecture, disturbances can easily be compensated by closing the loop. This becomes evident in Fig. 5(a).

Since we are expecting first-order constraint violations in the control input, we furthermore validated the system's behavior in cases where the controller enters velocity saturation. The respective results are shown in Fig. 5(b) and exhibit an immediate feedback reaction as soon as the saturation zone is left. In contrast, the error propagates for the feedforward architecture. It shall be emphasized that this is an encouraging property which helps the robot to robustly handle small violations of the trajectory velocity limits.

5.2 Constraint Violations

As already mentioned above, velocity constraint violations may occur as a consequence of the limited overshooting behavior of the trajectory splines. Although spline theory suggests that such overshoots are bound with the size of the spline segments, various simulations were conducted with the aim of examining the relative frequency and the extent of the violations.

The results of this evaluation are best summarized in histogram form as depicted in Fig. 6(a) and Fig. 6(b) for a single and multiple degrees of freedom, respectively. The histograms represent the distribution over normalized trajectory deviations from the predefined velocity limits. In the scope of the analysis, 10, 000 trajectories were generated from random waypoints using the proposed method.

The histogram distributions suggest that we may generally expect a low relative frequency for the occurrence of velocity constraint violations. This is particularly true for high-DOF systems since the estimated transition times comply with the dynamic limits of all degrees of freedom and thus show a clear tendency towards dynamic conservativeness.

5.3 Experimental Results

In addition to the above simulations, the proposed method was evaluated in practice. Here, the presented 5-DOF robotics arm served as experimental

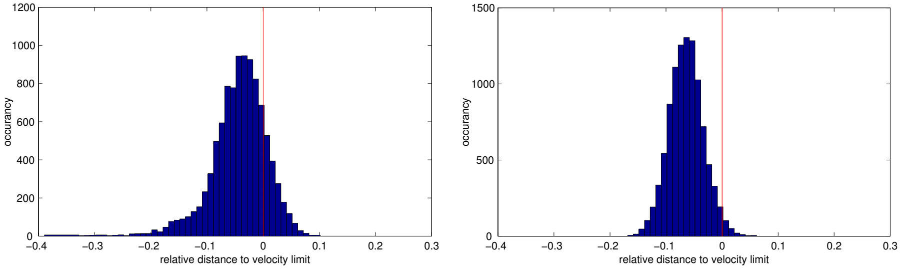


Fig. 6. Histogram distribution over normalized velocity constraint violations for a single (left) and multiple degrees of freedom (right). The red line indicates the velocity limit. Histogram bins to the right of this line correspond to constraint violations.

platform. Configuration space trajectories were generated using an inverse kinematic model and the Three-cubic spline method. The arm's end-effector was instructed to describe several standard profiles involving circular and sinusoidal curves. Communication outages were introduced to simulate the occurrence of tracking errors.

By qualitative analysis and observation, we have found that our approach to generating trajectories in combination with the proposed control architecture yields a remarkably smooth motion of the articulated system.

A quantitative comparison of the control performance between feedforward and feedback architecture is furthermore depicted in Fig. 7. Here, the controller of the arm's shoulder joint experienced a communication outage of 1 second. The feedback results display a robustness which widely resembles the outcome of the above simulations, whilst the feedforward solutions cannot compensate for the propagating position error.

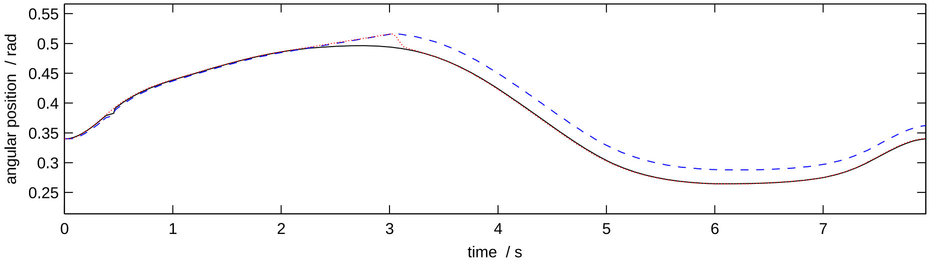


Fig. 7. Experimental results achieved with the embedded robot arm. A communication outage of 1 second was applied to test the control performance of the feedforward (blue) and the feedback architecture (red).

6 Conclusions

We have proposed a novel and alternative approach to the task of generating trajectories for an articulated robot with dynamic constraints. We have furthermore demonstrated that by focusing the effort on the generation process, the design of a trajectory controller becomes a straightforward problem. Our method is efficient, easy to implement, and particularly suited for applications involving high-DOF articulated systems such as robotics arms or legs. We have shown, in simulation and practice, that the Three-cubic spline interpolation is well prepared for integration with a graph-based planner and yields smooth trajectories. Although smoothness comes at the price of bound constraint violations, we have provided evidence that such violations can be expected to be small in magnitud and to have a low frequency of occurrence.

References

1. Fahim, A., Tetreault, M., Necsulescu, D.S.: Robot trajectory optimisation with dynamic constraints. The International Journal of Advanced Manufacturing Technology 3(1), 71–76 (1988)
2. Guzzella, L.: Analysis and Synthesis of Single-Input Single-Output Control Systems, 1st edn. vdf Hochschulverlag ETH Zurich (2007)
3. Jolly, K.G., Kumar, R.S., Vijayakumar, R.: A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits. International Journal of Robotics and Autonomous Systems 57(1), 23–33 (2009)
4. Lavalle, S.M.: Rapidly-exploring random trees: A new tool for path planning. Technical Report, Computer Science Dept., Iowa State University (1998)
5. Lee, C., Xu, Y.: Trajectory fitting with smoothing splines using velocity information. International Conference on Robotics and Automation 3, 2796–2801 (2000)
6. Lepetič, M., Klančar, G., Škrjanc, I., Matko, D., Potočnik, B.: Time optimal path planning considering acceleration limits. International Journal of Robotics and Autonomous Systems 45, 199–210 (2003)
7. Likhachev, M., Ferguson, D., Gordon, G., Stentz, A., Thrun, S.: Anytime dynamic A*: An anytime, replanning algorithm. In: Proc. of The International Conference on Automated Planning and Scheduling (ICAPS), pp. 262–271 (2005)
8. Luca, A.D., Lanari, L., Oriolo, G.: A sensitivity approach to optimal spline robot trajectories. Automatica 27(3), 535–539 (1991)
9. Prado, M., Simón, A., Carabias, E., Perez, A., Ezquerro, F.: Optimal velocity planning of wheeled mobile robots on specific paths in static and dynamic environments. Journal of Robotic Systems 20(12), 737–754 (2003)
10. Shiller, Z., Gwo, Y.-R.: Dynamic motion planning of autonomous vehicles. IEEE Transactions on Robotics and Automation 7(2), 241–249 (1991)
11. Stentz, A.: Optimal and efficient path planning for partially-known environments. Kluwer International Series in Engineering and Computer Science, pp. 203–220 (1997)
12. Tondu, B., Bazaz, S.A.: The three-cubic method: An optimal online robot joint trajectory generator under velocity, acceleration and wandering constraints. The International Journal of Robotics Research 18(9), 893–901 (1999)

Adaptive Motion Control: Dynamic Kick for a Humanoid Robot

Yuan Xu and Heinrich Mellmann*

Institut für Informatik, LFG Künstliche Intelligenz
Humboldt-Universität zu Berlin, Germany
{xu,mellmann}@informatik.hu-berlin.de
http://www.naoteamhumboldt.de

Abstract. Automatic, full body motion generation for humanoid robots presents a formidable computational challenge. The kicking motion is one of the most important motions in a soccer game. However, at the current state the most common approaches of implementing this motion are based on key frame technique. Such solutions are inflexible and cost a lot of time to adjust robot's position. In this paper we present an approach for adaptive control of the motions. We implemented our approach in order to solve the task of kicking the ball on a humanoid robot Nao. The approach was tested both in simulation and on a real robot.

1 Introduction

Automatic, full body motion generation for humanoid robots presents a formidable computational challenge due to

- the high number of degrees of freedom;
- complex kinematic and dynamic models;
- balance constraints;
- collision free;
- switching target dynamically;
- cope with unexpected external forces.

However, the humanoid robots of today still do not satisfy the aforementioned demands and their level of dynamically stable mobility is insufficient in the context of the real and uncertain environment. During the past 30 years, many studies have been conducted on motion of humanoid robots, especially on biped locomotion control, and many methods have been proposed [1,2,3,4,5]. Search techniques, such as Rapidly-exploring Random Trees [6], are applied in the full body motion planning of humanoid robot, but only in a static environment. The kicking motion is one of the most important motions in a soccer game. At the current state the most common approaches of implementing this motion are key frame based techniques [7,8]. However, such solutions are inflexible, i.e., in order

* Both authors contributed equally to this work. The research team is grateful for the comments of the reviewers that help improve the manuscript.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 392–399, 2010.

to adjust the aimed direction of the kick the robot has to walk around the ball. Such adjustments cost a lot of time especially when some precise adjustments have to be done, e.g., for a penalty shoot. Re-planning method [9] based on an off-line computation is proposed to adapt the kicking motion, but a feasible sub-set of the motion parameters is considered only. In this paper we present an approach for adaptive control of the motions. As an application we implement the adaptive kick on the robot Nao, a humanoid robot used in the RoboCup *Standard Platform League* (SPL).

The paper is organized as follows: in section 2, we describe our approach to the adaptive motion control for a humanoid robot; the experimental results are given in section 3, followed by conclusion and discussion in section 4.

2 Adaptive Kick Motion Control

In order to enable the robot to perform an adaptive motion we have to consider the following three aspects: reachable space, motion planning and stabilization.

In our approach we model the motion in Cartesian space, the joint trajectories are generated by inverse kinematic.

We have to ensure the adaptivity and to satisfy the conditions like stability at the same time. Thus, it is an obvious idea to describe the motion itself as an optimization problem, e.g., minimize the angle between the foot and the target directory during the kick preparation or maximize the speed of the foot to get a strong kick. Solving such complex optimization problem may be a difficult job. In order to ensure the adaptivity in real time, the motion trajectories are not calculated explicitly. Rather, we calculate the next position of the foot in each cycle by local optimization. Since the conditions change according to the sensory input, e.g., seen position of the ball, the resulting motion trajectory changes continuously. Of course, it requires the conditions to be defined in a way not to run into an unexpected local minimum or maximum (depending on formulation). In the following we present the basic structure of the kicking motion.

The kick is divided into four phases: preparation, retraction, execution and wrap-up phase. In the preparation phase the robot moves the body to one foot and lift the other. In the second phase the robot retract the foot according to the visual input. After retraction is finished, the robot execute the kick. In the last phase the robot put the lifted foot back to the ground and goes to the initial position. If the situation changes and kick is impossible anymore, e.g., the ball is too far away, the robot can break up the kick and change directly to the wrap-up phase at any time. The adaptation to the visual input is done in the retraction phase. The stabilization is necessary in all four phases.

Now we formulate the kicking task geometrically as follows: The input of the algorithm, i.e., the *kick request* is given by a pair $(\mathbf{p}_b, \mathbf{v}_b) \in \mathbb{R}^3 \times \{\mathbf{v} \in \mathbb{R}^3 : \|\mathbf{v}_b\| = 1\}$, where $\mathbf{p}_b$ is the point which should be moved (e.g., the center of mass of the ball) and $\mathbf{v}_b$ denotes the desired movement direction of the ball (e.g., direction to the goal). Because the front of Nao's foot is round, the collision

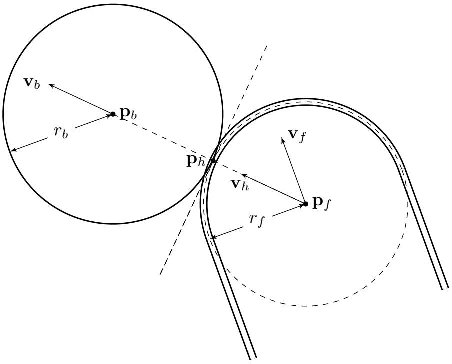


Fig. 1. The *kick request* is defined by $(\mathbf{p}_b, \mathbf{v}_b)$, the target of the foot motion is denoted by $(\mathbf{p}_f, \mathbf{v}_f)$, and $\mathbf{p}_h$ is the *hitting spot.* r_b and r_f are the radius of ball and half width of foot respectively. The involved directional vectors (v_b, v_h and v_f) are unit vectors. We can calculate $\mathbf{p}_h = \mathbf{p}_b - \mathbf{v}_b \cdot r_b$, $\mathbf{p}_f = \mathbf{p}_h - \mathbf{v}_b \cdot r_f$, $\mathbf{v}_h = \mathbf{v}_b$. Direction $\mathbf{v}_f$ results from the motion planning.

between ball and foot can be simplified as collision between two balls, as shown in Fig.1. Thus, the motion direction after the collision is defined by the *hitting spot* $\mathbf{p}_h$, i.e. the collision point between ball and foot. Consequently, the actual task is to move the foot in a trajectory which crosses the point $\mathbf{p}_f$, where the foot collides with the ball. Note that motion direction $\mathbf{v}_f$ results from the motion trajectory and has only influence of the strength of the kick.

2.1 Reachable Space

The *reachable space* of a robot is defined as the set of points that can be reached by its end effector, with respect to a reference frame of the robot. HRP-2 uses numerical methods to generate reachable space for arm manipulation[10]. By *reachable space* in the kicking task of a humanoid robot, we mean the space which can be reached by one foot, while the robot stands stably on the another. The *reachable space* is defined by some basic constraints that a humanoid robot has to satisfy during the kick, including the kinematic constraint (e.g., the limits of joint angles; and the collision constraint.) and the balance constraint (the robot should stands stably with one foot).

The *reachable space* of an end effector that rotates and translates in $\mathbb{R}^3$ is a six-dimensional manifold. In order to reduce the number of variables in the space, we decided to represent the space of the reachability by a three dimensional grid. Thereby, we don't consider the rotation of the end effector (e.g., foot), i.e., some of the points in the grid might be reachable only with a special rotation. The main reason for this decision is the simplicity and representational power, the resulting *reachable space* is a subset of $\mathbb{R}^3$. We generated the *reachable space* according to the physical limitations in an experiment on the real robot. For that, we let the robot move the end effector to every of the reachable points which generated by kinematic constraint, and record the reached points at the same time. In this experiment, all the constraints listed above are considered. Fig. 2 (left) illustrates the *reachable space* of the kicking foot generated by our experiments.

2.2 Motion Planning

As already discussed, we can divide the kick into four phases: preparation, retraction, execution and wrap-up phase. Considering this approach there are two questions arising: how to calculate the retracting point and how to calculate the fastest possible kicking trajectory.

In order to analyze these problems we simplify it to a two dimensional case. For that, we assume that the height of the kicking foot constantly equals to the radius of the ball.

Retraction Point. To answer the first question from above we consider the requirements for the retraction of the foot. First of all the robot should retract the foot as far as possible from the hitting point $\mathbf{p}_h$ to get the maximal load. This is, of course, a very naive assumption, as the maximal impulse is given by maximal joint velocities and the posture of the robot. Additionally, the retraction point should be chosen in the way that the retracted foot points as much as possible in the requested kicking direction $\mathbf{v}_b$. Of course, the retraction is limited by the reachable space of the foot and also by stabilizing ability of the robot. The problem of stabilization will be discussed later in the section 2.3, for simplicity reasons we assume at this point that the robot can stabilize our motion and focus on the reachability constraints.

In order to express the requirements mentioned above we define the following function

$$g : \mathbb{R}^2 \to \mathbb{R}, \quad \mathbf{p} \mapsto g(\mathbf{p}) := \frac{(\mathbf{p}_h - \mathbf{p})^t \cdot \mathbf{v}_h}{\|\mathbf{p}_h - \mathbf{p}\|}. \tag{1}$$

Obviously, it holds $g(\mathbf{p}) = 1$ if, and only if $(\mathbf{p}_h - \mathbf{p}) = \lambda \mathbf{v}_h$ for a certain $\lambda \in \mathbb{R}$. We can use this function to satisfy the direction requirement. Now, for a given $\delta \in [0, 1]$ we define the function

$$f_\delta : \mathbb{R}^2 \to \mathbb{R}, \quad \mathbf{p} \mapsto f_\delta(\mathbf{p}) := (1 - \delta) \cdot \|\mathbf{p} - \mathbf{p}_h\| + \delta \cdot (1 + g(\mathbf{p})) \tag{2}$$

This function combines the conditions for the distance and angle. We can determine the optimal retraction point $\mathbf{p}_r$ as a maximum of f_δ over the reachability grid Ω, i.e.,

$$\mathbf{p}_r = \mathrm{argmax}_{\mathbf{p} \in \Omega}(f_\delta(\mathbf{p})) \tag{3}$$

The parameter δ describes the importance of the angle requirement compared to the distance requirement, i.e., if $\delta = 0$ only the distance is maximized without taking the direction of the kick into account. Note, that δ strongly depends on the size of the reachable space. An optimal δ can be found by experiments, we used the value $\delta = 1 - 10^{-3}$ in our tests. Fig. 2 illustrates an example of a calculated retraction point inside the reachability grid.

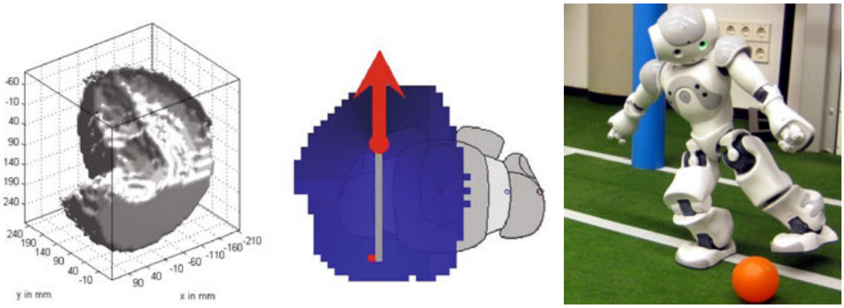


Fig. 2. The *reachable space* of the kicking foot is approximated by a three dimensional grid is shown on the left. The retracting point is calculated according to the reachability grid. Middle figure illustrates the grid of all reachable positions in the xy-plane (blue) whereas the height is fixed to the radius of the ball. Additionally the requested kick point $\mathbf{p}_f$ and direction $\mathbf{v}_f$ are marked by red arrow. The calculated point of retraction $\mathbf{p}_r$ is marked by a red square (the cell in the reachability grid which maximizes the function f_δ). The figure on the right shows the executed preparing motion according to the calculations shown in the left image.

After finding a point $\mathbf{p}_r$ we can interpret the value $g(\mathbf{p}_r)$ as a measure for the precision of the kick, i.e., in the case $g(\mathbf{p}_r) = 1$ the direction of the foot movement $\mathbf{v}_f$ corresponds to the desired direction of the kick $\mathbf{v}_h$. This value can be passed to the behavior as a prediction of the kick result. Based on it the behavior could decide whether to finish the kick or to break it up, if it is not precise enough.

This approach can be easily extended to the three dimensional case.

Trajectory of the Kick. After the preparation is done, i.e., the foot reached the retraction point $\mathbf{p}_r$ the robot has to move the foot towards the hitting spot $\mathbf{p}_h$ to kick the ball. Usually we want to do it as fast as possible. However, moving the foot along the fastest path may cause problems, e.g., the foot may collide with the ground. In our current implementation we move the foot along the shortest path in the reachability grid, which allows to prevent such collisions. It can be improved by using of the shortest path in the joint space. However, in this case we have to adjust this path according to the reachability grid in order to avoid collisions.

2.3 Stabilization

Keeping balance in the single-support phase is one of the major problems. During this phase, the robot is supported only on one foot, so it is more difficult for it to cope with disturbances. Some disturbances, like adjusting kicking foot according to ball, can make the robot lose stability. We introduce a feedback control to modify the reference trajectory according to the sensor information.

Since the poses of legs and position of the hip is already determined by motion trajectory, the stabilizer can adjust the body inclination to satisfy the static

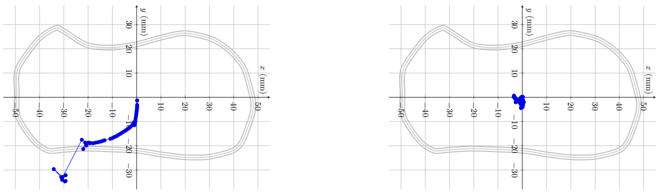


Fig. 3. Robot changes from standing by double feet to the standing on one leg, i.e., prepare for the kick. The left figure shows that center of mass jumps out of support polygon without stabilizer; the center of mass is kept in the center of support polygon with stabilizer in the right figure.

stable criterion, i.e., the center of mass should be in the support polygon. The *Body Inclination Control* is implemented as follows: in the first step the center of mass and support polygon are calculated from sensor data, in the second step the body inclination is adjusted to minimize the difference between center of mass and the center of support polygon. The P control rule is applied as the first trial. The Fig.3 illustrates the center of mass is kept in the center of foot while the robot stabilizes itself during the kicking.

3 Experimental Results

The approach proposed in this paper was implemented and tested on a real Nao robot and in simulation. In this section we present the results of experiments we performed to evaluate the kick motion. A video showing the experiments performed on the real robot can be found here `http://www.informatik.hu-berlin.de/~naoth/media/video/dynamic_kick.mp4`.

In order to test the adaptivity performance of the proposed kick motion we performed two different experiments: in the first experiment we let the robot kick

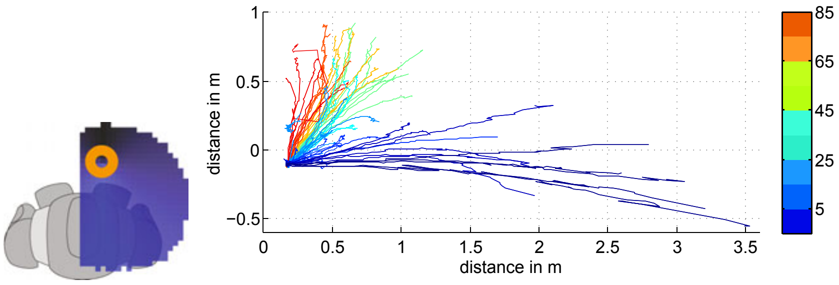


Fig. 4. The kick is performed with the same starting position of the ball but in different directions: (left) position of the ball and the reachability grid; (right) observed trajectories of the ball for each kick direction (respective colors)

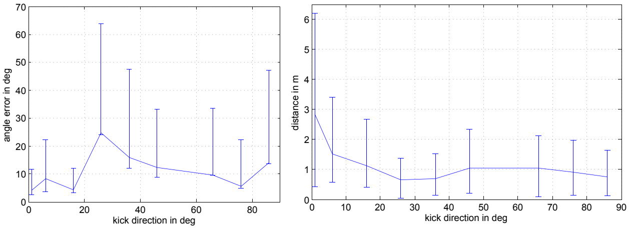


Fig. 5. (left) the angle error of the resulting position of the ball; (right) the maximal kick distance of the ball for each kick direction;

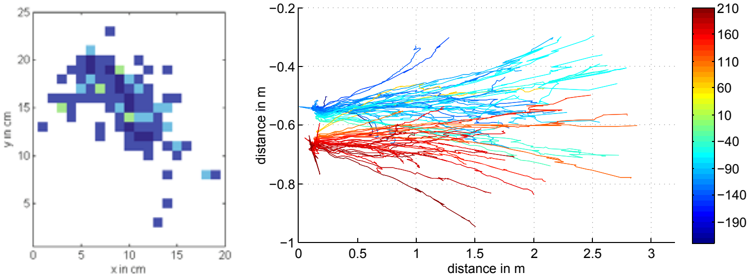


Fig. 6. The kick is performed with the same direction but from random positions: (left) positions of the ball. The color indicates the number of kicks from the according position (the greener, the more); (right) observed trajectories of the ball for each kick position (respective colors)

from the same position but to different directions between forward (0°) to the left (90°). The direction is changed in 10° steps. For each direction we executed 5 kicks. In the second experiment we let the robot kick straight forward, thereby the position of the ball is changed randomly. In this experiment we executed 100 kicks.

The Fig. 4 and Fig. 5 illustrate the results of the first experiment. Here it can be observed, that the resulting direction of the kick has a general negative offset. Another observation is the "hole" for the kick directions around 20°. Both can be explained with the insufficiency of the analytical assumptions for the kick geometry made, e.g., the foot has not a precise form of a circle and the kick is not fully elastic. The Fig. 5 shows the results of the second experiment. Here we can see that the direction of the kick doesn't vary a lot for changing position of the foot.

It can be said that the results of the experiments are very promising. But, they also show that there is a need of more sophisticated algorithms (e.g., neural networks) for the adaptation of the foot to ensure precise kicks to any direction.

4 Conclusion and Future Work

In this paper, we have presented the adaptive motion control for a humanoid robot. It enables the robot to kick the ball in different positions and to different intended directions. All calculations are done on line, therefore the kick can be adapted with vision feedback in real time. In our experiments we could show that the presented approach is able to accomplish adaptive kick.

In the future, we are interested in learning good strategy and parameters by observation from vision. To achieve this fast, the learning algorithm will be investigated in simulation firstly, and applied to real robot late. Our next step is to include avoiding objects, i.e., avoiding the ball and leg of opponent before kicking, and the dynamic stabilization during kicking execution is also very important.

References

1. Vukobratovic, M., Borovac, B.: Zero-moment point- thirty five years of its life. International Journal of Humanoid Robotics 1(1), 157–173 (2004)
2. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by a simple three-dimensional inverted pendulum model. Advanced Robotics 17(2), 131–147 (2003)
3. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by using preview control of zero-moment point. In: ICRA, pp. 1620–1626. IEEE, Los Alamitos (2003)
4. Collins, S.H., Wisse, M., Ruina, A.: A three-dimensional passive-dynamic walking robot with two legs and knees. I. J. Robotic Res. 20(7), 607–615 (2001)
5. Grillner, S.: Neurobiological bases of rhythmic motor acts in vertebrates. Science 228, 143–149 (1985)
6. Kuffner Jr., J.J., Kagami, S., Nishiwaki, K., Inaba, M., Inoue, H.: Dynamically-stable motion planning for humanoid robots. Auton. Robots 12(1), 105–118 (2002)
7. Burkhard, H.D., Holzhauer, F., Krause, T., Mellmann, H., Ritter, C.N., Welter, O., Xu, Y.: Nao-team humboldt 2009. Technical report, Humboldt Universität zu Berlin (2009)
8. Röfer, T., Laue, T., Müller, J., Bösche, O., Burchardt, A., Damrose, E., Gillmann, K., Graf, C., de Haas, T.J., Härtl, A., Rieskamp, A., Schreck, A., Sieverdingbeck, I., Worch, J.H.: B-human team report and code release (2009), `http://www.b-human.de/download.php?file=coderelease09_doc`
9. Lengagne, S., Ramdani, N., Fraisse, P.: Planning and fast re-planning of safe motions for humanoid robots: Application to a kicking motion. In: The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, USA (October 2009)
10. Guan, Y., Yokoi, K., Zhang, X.: Numerical methods for reachable space generation of humanoid robots. The International Journal of Robotics Research 27(8), 935–950 (2008)

An Extensible Modular Recognition Concept That Makes Activity Recognition Practical

Martin Berchtold[1], Matthias Budde[2], Hedda R. Schmidtke[2], and Michael Beigl[2]

[1] Institute of Operating Systems and Computer Networks (IBR), TU Braunschweig
[2] Institute of Telematics, Pervasive Computing Chair, Karlsruhe Institute of Technology (KIT)

Abstract. In mobile and ubiquitous computing, there is a strong need for supporting different users with different interests, needs, and demands. Activity recognition systems for context aware computing applications usually employ highly optimized off-line learning methods. In such systems, a new classifier can only be added if the whole recognition system is redesigned. For many applications that is not a practical approach. To be open for new users and applications, we propose an extensible recognition system with a modular structure. We will show that such an approach can produce almost the same accuracy compared to a system that has been generally trained (only 2 percentage points lower). Our modular classifier system allows the addition of new classifier modules. These modules use Recurrent Fuzzy Inference Systems (RFIS) as mapping functions, that not only deliver a classification, but also an uncertainty value describing the reliability of the classification. Based on the uncertainty value we are able to boost recognition rates. A genetic algorithm search enables the modular combination.

1 Introduction

While many systems dealing with activity recognition in mobile and ubiquitous computing have emerged over the years, there are still challenges to which no solution has been presented so far. In this work, we try to tackle one of the biggest issues in mobile computing: modular combination and extension of activity recognition systems. Conventionally, activity recognition is mostly done with a single monolithic classifier. Such classifiers are large and their usage is inflexible. Having large complex classifiers is problematic, since in mobile computing the processing power of devices is limited and the battery service life is an important factor. A comparison between monolithic and modular classifiers has been conducted in [3]. Classification systems that require large computing resources are not practical, as often real-time reaction is needed. E.g., our prototype activity recognition system using a mobile phone should be able to react on a detected activity in real-time, without disturbing the phone's main functionality.

In activity recognition, inflexibility of classifiers is problematic. A general classifier would need to recognize common daily activities (e.g. 'walking'), but also activities specific for a certain group of persons (e.g. 'dancing') or activities that are rarely carried out (e.g. 'climbing'). Overall, a general recognizer would need to be very complex, covering hundreds of activities while only a few of them are actually required for one person. It is well known that large complex recognizers tend to have lower performance than small, focused classification systems. All this brings us to the need for a dynamic,

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 400–409, 2010.

modular classification system, in which it is possible to add or remove classifier modules on demand. In this paper we propose a modular classifier approach based on Recurrent Fuzzy Inference Systems (RFIS). We give a detailed case study showing how the system reacts if a new classifier set is added to a pre-existing one. We present that, using a bit masking that is identified through a genetic algorithm, we can improve the progression of old and new modules in a dynamic queue.

In other application fields, the combination of classifiers has been investigated for many years now. In general, two approaches are differentiated: combination on the abstract level and on the measurement level. In abstract level combination, classifiers are computed separately and then logically combined, based on class labels or rankings of classes [14]. In [16], classifiers are combined on the measurement level to recognize handwritten Chinese characters. In their proposal, classifiers that map feature vectors onto measurement vectors are combined. [2] proposes a similar attempt, also for handwritten character recognition. Other publications on the topic of dynamic classifier combination with similar approaches are [11] and [12]. All the aforementioned approaches have in common, that either classifiers or other superimposed selection and activation methods are needed to combine all classifiers. This requires that the whole set of classes is known at design time, while in our system, selection and activation occur inherently on the module level. This means that new classes – and therefore classifier modules – can be added to a pre-existing classifier set and only one classifier is active at each point in time. This considerably reduces resource consumption.

There has been a lot of research in activity recognition over the years. An exemplary work is [13], in which a triaxial accelerometer sensor was used to detect eight activity classes. Several algorithms in various combinations are investigated in four different settings on their classification accuracy. Since the employed sensor is mounted on a fixed position in the test persons pelvic region and the algorithms that are used do not qualify for modular and dynamic usage, the recognition rates are not really comparable. [6] and [10] both do activity recognition for mobile phones, where [6] only uses sensors and resources native to the phone for sensor acquisition and classification. Both approaches do not qualify for modular classification, nor do they recognize a significant amount of activities with high accuracy.

The remainder of this paper is structured as follows: First, the modular classifier is explained, with its RFIS mapping function, the machine learning algorithm to identify the RFIS, and the modularity. Second, we describe how modularity can be enabled via a bit vector masking, which is found through a genetic algorithm. The results are presented in the evaluation section and the paper is concluded in the last section.

2 Modular Recurrent Fuzzy Classification

The classification process starts with (1) the feature extraction on the data delivered by the sensors. Then (2) the resulting feature vector is reduced to a one-dimensional value through a mapping function. This value is (3) assigned to a class, based on a set of fuzzy numbers. Steps (2) and (3) are carried out by the modules. As mapping function (step 2) we use a Recurrent Fuzzy Inference System (RFIS) [5]. Due to the used RFIS and the fuzzy numbers-based classification, our modular classifier system provides an uncertainty value, that is sensitive to the respective class that is detected. This solves another problem of activity recognition, namely that patterns are only separable to a certain degree, since the position of the mobile device (e.g. commodity phone) is not always fixed. If an uncertainty measure is present for each classification, a filtering upon this can improve reliability by far. Furthermore, with FISs a bit masking is possible, which enables each dimension of every rule to be 'activated' or 'deactivated' separately. Thereby, the classifier modules can be optimized to react to each other in a dynamic queue. This is possible without losing the original capabilities of the RFIS.

In this section we first specify the RFIS, followed by an explanation of the machine learning algorithm which identifies the mapping function on training data. Lastly, the overall classifier system is explained and how its modularity works.

2.1 Recurrent Fuzzy Inference System

Takagi, Sugeno and Kang [15] (TSK-) FISs are fuzzy rule-based structures, which are especially suited for automated construction. In TSK-FISs, the consequence of the implication is not a functional membership to a fuzzy set, but a linear function, which is defined according to the rule as follows:

$$\text{IF } \mu_j(\overrightarrow{v}_t) \text{ THEN } f_j(\overrightarrow{v}_t), \text{ with } f_j(\overrightarrow{v}_t) := a_{1j}v_1 + .. + a_{nj}v_n + a_{(n+1)j} \quad (1)$$

Since we deal with highly correlated features, we employ covariant Gaussian membership functions, which are defined with the overall output **S** of the TSK-FIS as follows:

$$\mu_j(\overrightarrow{v}_t) := e^{-\frac{1}{2}(\overrightarrow{v}_t - \overrightarrow{\mathbf{m}}_j)\Sigma_j^{-1}(\overrightarrow{v}_t - \overrightarrow{\mathbf{m}}_j)^T} \quad \text{and} \quad \mathbf{S}(\overrightarrow{v}_t) := \frac{\sum_{j=1}^{m} \mu_j(\overrightarrow{v}_t) f_j(\overrightarrow{v}_t)}{\sum_{j=1}^{m} \mu_j(\overrightarrow{v}_t)} \quad (2)$$

The outcome of the mapping at time t is fed back as input dimension n for the TSK-FIS mapping at $t+1$. The recurrence not only delivers the desired uncertainty level, but also stabilizes and improves the mapping accuracy. Instead of 'Recurrent TSK-FIS' we use the simpler term RFIS in the remainder of this paper. More details on the RFIS can be found in [5].

2.2 RFIS Identification Algorithm

The RFIS is completely described by the parameters a_{ij} of the linear functional consequence f_j and the mean vector $\overrightarrow{\mathbf{m}}_j$ and the covariance matrix Σ_j of the membership functions μ_j. These values are identified upon an annotated training feature set via a five step algorithm, is described in the following an in more detail in [9].

1. Data Annotation and Separation: The training data $\mathcal{V}^{tr}$ is separated according to the class c_j the data pairs belong to. Clustering on each subset delivers rules that can be assigned to each class. **2. Clustering:** A combination of Subtractive [7] and Gath-Geva [8] clustering is used, where the Subtractive clustering gives the initial cluster centers and the upper bound for the amount of clusters for the Gath-Geva clustering. The selection of initial cluster centers for the Gath-Geva clustering is done via a genetic algorithm according to the Mean Squared Error (MSE) of the resulting RFIS. The output of the clustering algorithms combination is the amount of rules m, the mean vector

$\overrightarrow{\mathbf{m}}_j$ and covariance matrix Σ_j of the membership functions μ_j. **3. Least Squares:** Linear regression identifies the parameters a_{ij} of the linear consequence function f_j of the rules $j = 1, .., m$. Minimizing the quadratic error – the quadratic distance between the desired output and the actual output – leads to an overdetermined linear equation to be solved. **4. Recurrent Data Set:** The output of the TSK-FIS **S** is now calculated over the training data $\mathcal{V}^{tr}$. This output is shifted by one, with a leading zero, and then added to the training data set $\mathcal{V}^{tr}$ as additional dimension. All data pairs for time $t > 1$ have the output of the FIS mapping of $t - 1$ in the recurrent dimension n. For this data set the steps 1 to 3 are repeated. **5. Stop Criterion:** There are two values qualifying for a stop criterion: the mean quadratic error or the classification accuracy. While the mean quadratic error mostly improves the expressiveness of the reliability value and less the accuracy, optimizing on the classification accuracy only improves the percentage of correct classifications. For our case study, we decided on a fixed number of iterations.

2.3 Modular Classification

Since the abilities of monolithic classifiers are limited, we use a modular approach to cope with this problem. Instead of using one classifier to classify on all classes $\mathcal{C}$, we use several classifiers $\mathbf{M}_i : \mathcal{V} \rightarrow \mathcal{C}_i$ (with $i = 1, .., N$) each classifying on a small subset $\mathcal{C}_i \subseteq \mathcal{C}$ of classes. The subsets $\mathcal{C}_i$ are chosen according to the classes $c_{ij} \in \mathcal{C}_i$ semantics, therefore each subset $\mathcal{C}_i$ has its own meta semantic. We call this meta semantic ‘conditional context’. To not only recognize the respective classes $c_{ij} \in \mathcal{C}_i$, but also the transition between classifiers $\mathbf{M}_i$, each module yields a complementary class $\overline{c}_i$ as well. All classifiers are chained in a dynamic queue, where the last classifier classifying on a class different from the complementary class is put first in queue. The idea behind this re-organization of the classifier queue is, that modules are successive in the queue, which are successive in recognition of input features. Therefore, when the currently ‘active’ module recognizes the complementary class $\overline{c}_i$, preferably not the whole queue needs to be tested for capabilities of classifying this feature vector $\overrightarrow{v}_t$, but only the next module in the queue.

To train this kind of queued classifiers, we need to train them on the respective classes $c_{ij} \in \mathcal{C}_i$ and on the complementary class $\overline{c}_i$. The training $\mathcal{V}_i^{tr}$ and check data sets $\mathcal{V}_i^{ck}$ for a classifier module $\mathbf{M}_i$ are unified with a selection of input data pairs of all other classifiers $\mathcal{V}_i^{\overline{c}} \subset \bigcup_{k \neq i} \mathcal{V}_k^{tr}$. This selection is labeled zero – which indicates the complementary class $\overline{c}_i$ – and added to the normal training and check data sets of this classifier. The actual training and check data is therefore $\mathcal{V}_i^{tr \cup \overline{c}}$ and $\mathcal{V}_i^{ck \cup \overline{c}}$, which are called $\mathcal{V}_i^{tr}$ and $\mathcal{V}_i^{ck}$ throughout the rest of this paper for reasons of simplicity.

The output of the RFIS $\mathbf{S}(\overrightarrow{v}_t)$ at time t is the normalized weighted sum of the functions $f_j(\overrightarrow{v}_t)$ of the rules j. The returned values numerically encode the classes. The assignment of the RFIS mapping result $\mathbf{S}_i(\overrightarrow{v}_t)$ to a class is done fuzzily, so the result is not only a class identifier, but also a membership, representing the reliability of the classification process. Each class c_{ij} is interpreted by a set of a triangularly shaped fuzzy numbers [1] (eqn. 3):

$$\mu_{c_{ij}}(x) = \begin{cases} \max(0, 1 - \frac{c_{ij} - x}{\alpha}) \text{ , when } x \leq c_{ij} \\ \max(0, 1 - \frac{x - c_{ij}}{\alpha}) \text{ , when } x > c_{ij} \end{cases} \tag{3}$$

The mean of the fuzzy number is the identifier c_{ij} itself. The crisp decision – i.e. which identifier is the mapping outcome – is carried out based on the highest degree of membership to one of the fuzzy numbers in the class $\mathbf{K}_i$ (eqn. 4).

$$\mathbf{K}_i(x) = \begin{cases} (c_{i1}, \mu_{c_{i1}}(x)) & \text{, when } \mu_{c_{i1}}(x) = m_i(x) \\ \vdots & \vdots \\ (\overline{c}_i, \mu_{\overline{c}_i}(x)) & \text{, when } \mu_{\overline{c}_i}(x) = m_i(x) \end{cases} \text{, with } m_i(x) = \max(\max_j(\mu_{c_{ij}}(x)), \mu_{\overline{c}_i}(x)) \quad (4)$$

The overall output of the classifier module $\mathbf{M}_i$ (eqn. 5) is a tuple $(c_{ij}, \mu_{c_{ij}})$ of a class identifier and the membership to it, where $c_{ij} \in \mathcal{C}_i \cup \overline{c}_i$ and $\mu_{c_{ij}} \in [0, 1]$.

$$\mathbf{M}_i(\overrightarrow{v}_t) := \mathbf{K}_i(\mathbf{S}_i(\overrightarrow{v}_t)) \quad (5)$$

3 Enabling Modularity

In this paper we focus on the modular combination of many fuzzy classifiers, classifying on subsets $\mathcal{C}_i$ of the overall recognized set $\mathcal{C}$ of classes. Especially, we are analyzing the capabilities of adding new classifiers modules and therefore new classes to a pre-existing classifier module set. This is not possible without an adaption of the pre-existing set of modules, since the modules are aware of each other, so they can work together in a dynamic queue. This awareness is residing in the recognition of the previously described complementary class, which also needs to be trained on.

3.1 Generalization on New Modules

When adding a new classifier module to an existing classification system we need to modify the state transition in the dynamic queue. This transition is done when the active module $\mathbf{M}_i$ recognizes the complementary class $\overline{c}_i$ ($\widehat{=}$ 0). Since adding new modules presumes that no data for identifying the complementary classes of the old modules is available, the old modules are not able to detect a transition onto the new modules. This means that the new modules rarely get activated (detection errors or mapping far outside the model) and therefore have no chance to be part of the classification.

3.2 Bit Vector Adaption Mechanism

If new classifier modules are added to the existing queue, the old ones need to be adapted, so that they recognize transitions to the new classifiers. Through an adaption technique, in which the dimensions of each rule for each pre-existing classifier module are 'activated' or 'deactivated', transition capabilities improve without changing the classification accuracy. The original classifier module is preserved and can be restored at any time. The optimal combinations of 'active/inactive' rule dimensions is searched for with a genetic algorithm, since the full search has an exponential runtime. The data for optimum search is the training data that the new modules were identified on, combined with the original training data.

The adaption is done via a bit vector, that specifies the 'active' and 'inactive' dimensions of each rule for one module $\mathbf{M}_i$. Therefore the bit vector $bit_{\mathbf{M}_i}$ for module $\mathbf{M}_i$, which has n input dimensions and m_i rules, has a length of $b_i := n \cdot m_i$. To use the bit vector, an interpretation function $I(\mathbf{M}_i(\overrightarrow{v}_t), bit_{\mathbf{M}_i})$ is defined, that 'switches' the rules dimensions temporarily, without changing the module $\mathbf{M}_i$ permanently. The interpretation function I is defined as a function mapping a module $\mathbf{M} \in \mathbb{F}(\mathbb{R}^n, \mathbb{R})$ together with a bit vector $bit_{\mathbf{M}} \in \{0,1\}^*$ of appropriate length to a module $\mathbf{M}'$. More details on this bit vector approach can be found in [4].

3.3 Genetic Algorithm Search Space

To determine the respective classifier module's bit vector, a genetic algorithm is used. The space that has to be searched is 2^{b_i} for module $\mathbf{M}_i$. A complete search would therefore have a runtime of $O(2^{b_i})$, which makes it impossible to calculate in a reasonable amount of time. In our experience, the genetic algorithm can find a suboptimal, but appropriate solution in a time span that is acceptable for our application. Nevertheless, we are currently investigating methods to limit the search space.

4 Evaluation

For evaluation we use a typical mobile computing device, the commodity phone. Nowadays commodity phones come with a variety of sensors, such a microphone, proximity sensor, GPS and accelerometers. We focus on accelerometer sensors, since with these sensors we can recognize most types of activity events. The phone is required to be with the user, e.g. in his pocket or held in his hand. With our modular classifier approach, we not only can recognize activity classes, but also the place the phone is positioned on. We call this 'conditional context' and get this directly through the respective activated classifier module. We need this conditional context to reach high recognition rates, as recognized acceleration patterns from the same activity differ heavily with it.

4.1 Application

The device chosen by us is the 'OpenMoko Freerunner' phone which comes with two 3-D accelerometer sensors. The sensors have a sampling rate of about 100Hz. With a window size of 8 samples we end up with 12.5 feature vectors per second. The features are mean and variance, since they are efficiently calculable and enable good recognition results. With this feature extraction, the resulting feature vector has 12 dimensions. Adding the recursive dimension makes a total of 13 dimensions. This feature vector is mapped onto the respective activity class via the currently 'active' module. If the active module classifies onto the complementary class, the next module is activated for this feature vector. This procedure is repeated for this feature vector until one of the modules classifies onto a class different from the complementary one.[1] This classifier module is then put first in queue and the next feature vector is processed. To increase reliability – and thereby accuracy – filtering upon the uncertainty is done. The filtering reduces the amount of output classes, but for most applications a few classes per second

[1] in case all classifiers including the last one classify onto $\overline{c_i}$, the overall output is the complementary class. This means the feature vector cannot be classified correctly by the given queue.

Table 1. Conditional contexts, classes and classifier modules for the acceleration sensor

Conditional Context	Context Class	Class No.	Classifier Module
Phone in users trouser pocket: *no movement*	user is sitting	1	$\mathbf{M}_1$
	user is standing	2	
	user is lying	3	
Phone in users trouser pocket: *movement*	user is walking	4	$\mathbf{M}_2$
	user is climbing stairs	5	
	user is cycling	6	
Phone on table:	no movement	7	$\mathbf{M}_3$
Phone in users hand:	just holding	8	$\mathbf{M}_4$
	talking on phone	9	
	typing text message	10	
Phone in users trouser pocket:	user is sitting in bus	11	$\mathbf{M}_5$
	user is standing in bus	12	
Phone in users trouser pocket: *dancing*	user is dancing (style 1)	13	$\mathbf{M}_6$
	user is dancing (style 2)	14	
	user is dancing (style 3)	15	

are more than enough, where the reliability is most important. Here we assume that most activities are not changing faster than in seconds.

The original classifier module queue classifies onto ten classes and recognizes four conditional contexts. To evaluate the addition of new classifiers, two more modules are put in queue. These recognize five classes and two conditional contexts. All activity classes, conditional contexts and respective classifier modules are shown in table 1.

The training data $\mathcal{V}_i^{tr}$ for classifier module $\mathbf{M}_i$ consisted of 400 data pairs and the check data $\mathcal{V}_i^{ck}$ of 200 pairs for each class c_{ij}. The data for training the complementary class $\overline{c}_i$ consisted of 800 and for the check data of 400 pairs, which was randomly selected out of the training and check data for the other classifiers. Both sets $\mathcal{V}_i^{tr}$ and $\mathcal{V}_i^{ck}$ were randomized in slices of 30 pairs, so the recurrence could be trained and tested. Each classifier modules' $\mathbf{M}_i$ RFIS mapping function $\mathbf{S}_i$ was trained for 100 epochs, after which the RFIS achieving the best classification accuracy for combination of training and check data was chosen.

The evaluation data $\mathcal{V}^{ev}$ had about 2000 data pairs per class, so 160 seconds per activity. This evaluation set was also randomized according to slices of 20 data pairs ($1, 67$ seconds), which in our experience is the quickest transition occurring between activities. Since most of the false classifications are occurring when changing from one activity class to another (due to recurrence), this is a stress test for the classifier modules. In a real application we estimate even better results than presented in the following, because activities change less often.

4.2 Evaluation Results

First we have a closer look into the results of classification, when all classifier modules get trained on a collective data set. When no filtering upon the uncertainty value is done, the overall recognition rate lies at only 58%. This is too low for practical use, but with our recurrent fuzzy classifier approach, we can improve the results significantly. As mentioned before, activity data from sources without fixed position is hardly separable. For this a filtering is needed, where the classifications with low reliability are separated from the recognitions with high reliability.

With the filtering upon the uncertainty value with a threshold of $\tau = 0.96$, we can boost the overall recognition rate by 27.4 *pp* (*percentage points*) up to 85,4%. The corresponding confusion matrix is shown in table 2. This is a recognition rate for a system which can be used in real applications. But the filtering reduces the amount

Table 2. Theoretical optimum without knowledge of all training data. Filter threshold: $\tau = 0.96$. Overall recognition rate: 85,4%.

	91%										**75%**				
	$\mathbf{M}_1$			$\mathbf{M}_2$			$\mathbf{M}_3$	$\mathbf{M}_4$			$\mathbf{M}_5$		$\mathbf{M}_6$		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	99.7	0.0	0.4	0.1	1.5	2.2	0.4	0.0	0.0	1.1	0.4	5.1	0.0	1.5	5.2
2	0.0	90.9	0.1	0.0	0.0	2.1	0.0	0.0	0.0	0.3	0.3	0.0	0.4	0.7	0.7
3	0.0	0.0	98.9	0.0	0.0	5.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	3.0
4	0.0	0.1	0.0	92.6	1.5	0.6	0.0	0.0	0.0	0.3	0.0	0.2	6.1	2.2	0.7
5	0.0	0.1	0.0	1.0	49.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	17.0	0.7	0.0
6	0.0	0.0	0.1	0.0	0.0	88.1	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	6.7
7	0.0	0.0	0.0	0.1	1.5	0.0	98.2	0.0	1.3	3.4	0.0	0.0	0.0	2.2	7.5
8	0.2	0.2	0.0	0.0	0.5	0.3	0.0	98.3	0.1	0.0	0.0	0.0	0.4	2.9	3.7
9	0.0	0.0	0.0	0.1	0.0	0.1	0.2	0.6	98.3	0.3	0.0	0.0	0.0	2.2	0.0
10	0.0	0.0	0.0	0.0	0.5	0.0	0.5	0.0	0.0	91.3	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	1.0	0.1	0.0	0.0	0.1	0.0	99.1	0.0	0.4	0.7	0.7
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	86.9	0.0	5.8	9.7
13	0.1	8.8	0.4	5.9	43.9	0.4	0.0	0.0	0.1	1.3	0.1	0.2	75.1	0.7	0.0
14	0.0	0.0	0.0	0.1	0.0	0.6	0.6	1.0	0.0	2.1	0.1	1.7	0.4	67.2	14.9
15	0.0	0.0	0.0	0.1	0.0	0.1	0.1	0.0	0.0	0.0	0.0	5.9	0.0	13.1	47.0
	69.3	60.3	73.7	15.1	6.2	26.5	56.8	42.1	50.3	27.7	40.8	25.0	9.2	4.4	4.6

Table 3. Modular approach without the need for knowledge of all training data in all training steps that is not masked with a bit vector. Threshold: $\tau = 0.96$. Overall recognition rate: 68,6%.

	95%										**17%**				
	$\mathbf{M}_1$			$\mathbf{M}_2$			$\mathbf{M}_3$	$\mathbf{M}_4$			$\mathbf{M}_5$		$\mathbf{M}_6$		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	99.3	0.2	1.7	0.4	0.8	2.8	0.5	0.2	0.2	4.5	8.6	0.0	0.8	7.7	2.3
2	0.0	98.9	0.9	0.4	0.1	0.6	0.0	0.1	0.2	0.7	73.8	1.2	0.3	16.9	6.8
3	0.0	0.0	92.9	0.0	0.0	0.2	0.0	0.1	0.0	0.0	0.0	26.0	0.0	26.2	34.1
4	0.0	0.0	0.0	86.4	2.1	0.0	0.1	0.0	0.1	0.0	0.0	0.0	4.1	1.5	0.0
5	0.0	0.3	1.9	12.9	95.4	0.2	0.2	0.1	0.3	1.5	0.0	0.6	92.2	20.0	18.2
6	0.0	0.0	1.7	0.0	0.6	95.4	0.0	0.0	0.0	0.0	0.8	0.0	0.2	6.2	6.8
7	0.2	0.0	0.0	0.0	0.0	0.0	99.0	0.0	0.0	11.2	0.4	0.0	0.0	0.0	0.0
8	0.3	0.2	0.6	0.0	0.4	0.7	0.1	99.4	0.0	2.2	0.0	15.0	0.9	13.8	20.5
9	0.0	0.0	0.2	0.0	0.0	0.0	0.2	0.1	99.2	0.0	0.0	1.2	0.3	0.0	2.3
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	79.9	0.0	0.0	0.2	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	16.4	0.0	0.0	3.1	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	55.5	0.0	1.5	2.3
13	0.0	0.4	0.0	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.0	3.1	0.0
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.8
	51.0	75.7	14.3	29.4	26.9	21.4	59.1	79.3	67.5	9.8	7.5	3.8	22.0	2.1	1.5

of classifications the recognition system has as output. The percentage of remaining classifications for each class after filtering is displayed in each table in the last row of the confusion matrices. Since the classifier modules originally produce 12.5 classifications per second, a percentage of less than 8% remaining classifications could result in an effective output of less than one class per second. Usually, a person's activities do not change that fast, but it is still possible that this could result in missing an activity event.

Taking a closer look, we can see that most of the classes (9 out of 15) are recognized with an accuracy of over 90%, as required for activity recognition applications. The classes with lower recognition percentages could have overlapping membership

Table 4. Modular combination without the need for knowledge of all training data in all training steps, masked with a bit vector. Threshold: $\tau = 0.96$. Overall recog. rate: 84,0% (optimal 85%).

	89%										**73%**				
	$\mathbf{M}'_1$			$\mathbf{M}'_2$			$\mathbf{M}'_3$	$\mathbf{M}'_4$			$\mathbf{M}_5$		$\mathbf{M}_6$		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	99.6	0.1	0.3	0.2	1.4	3.1	0.3	0.0	0.0	0.8	0.3	5.1	0.0	1.5	5.8
2	0.0	92.1	0.1	0.0	0.0	2.7	0.0	0.0	0.0	0.0	0.2	0.1	0.4	0.8	2.2
3	0.0	0.0	96.1	0.0	0.0	5.9	0.0	0.0	0.2	0.0	0.0	0.0	0.4	0.0	2.9
4	0.0	0.2	0.2	87.2	7.7	0.3	0.0	0.1	0.0	0.3	0.1	0.3	12.4	3.8	1.4
5	0.0	0.1	0.1	1.6	46.4	0.2	0.0	0.0	0.0	0.0	0.0	0.0	16.3	0.0	0.7
6	0.2	0.0	2.7	0.0	0.0	84.9	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.0	5.8
7	0.0	0.1	0.0	0.2	1.9	0.0	98.1	0.0	2.1	1.4	0.0	0.0	0.0	2.3	7.2
8	0.1	0.2	0.0	0.0	0.5	0.3	0.0	98.4	0.1	0.0	0.0	0.0	0.4	3.1	3.6
9	0.0	0.0	0.0	0.2	0.0	0.2	0.2	0.6	97.3	0.5	0.0	0.0	0.0	2.3	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.0	0.0	94.8	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	1.0	0.2	0.0	0.0	0.1	0.0	99.1	0.0	0.0	0.8	0.7
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	86.4	0.0	6.9	8.7
13	0.1	7.4	0.5	10.0	41.1	0.6	0.0	0.0	0.1	0.8	0.2	0.2	69.0	0.0	0.0
14	0.0	0.0	0.0	0.2	0.0	1.0	0.8	0.9	0.0	1.4	0.1	1.5	0.4	64.1	13.8
15	0.0	0.0	0.0	0.2	0.0	0.6	0.1	0.0	0.0	0.0	0.0	6.5	0.0	14.5	47.1
	69.0	59.9	65.7	7.9	6.6	24.6	57.9	43.2	47.5	26.8	42.4	24.6	8.6	4.2	4.8

functions in the RFIS mapping, which is indicated through mutually misclassified classes. A good example for this circumstance is given by classes no. 5 and no. 13, where $43,9\%$ of data for classifier no. 5 is misclassified on no. 13 and 17% from no. 13 on no. 5. These two provide a good example for classes whose patterns are hardly separable. The overall recognition rates in our evaluation are significantly lowered due to the fact that we have classes with overlapping membership functions in our system.

Next, we examine the classification accuracies when a new set of classifiers is added to a pre-existing one. The original classifier set consisted of modules $\mathbf{M}_1$ to $\mathbf{M}_4$. To this, the set containing modules $\mathbf{M}_5$ and $\mathbf{M}_6$ is added. The results of the union without bit masking the classifiers are shown in table 3. Here, the recognition rate for the original modules is 95% as required. The added modules $\mathbf{M}_5$ and $\mathbf{M}_6$ are rarely activated, because the only case where they could be activated is when misclassifications occur and all original modules recognize the complementary class. After the genetic algorithm has found a bit masking for the original classifiers, the recognition rates increase significantly (table 4): the overall rate for all classes is just 2 *pp* lower compared to the upper limit we achieved when training all modules together. The still high recognition rates for the ten original classes indicate, that the original classification capabilities of these modules have not changed much. A recognition rate of 6 *pp* less compared to the non bit masked classifier modules and an increase to 15 recognized classes is a really good result compared to other activity recognition systems.

5 Conclusions

We have presented an approach for modular classification in activity recognition feasible for mobile devices, where Recurrent Fuzzy Inference Systems (RFIS) are utilized. This approach solves two main issues in practical activity recognition. One is the addition of new classifier modules to a pre-existing set of modules. We solve this problem with a bit vector masking of the classifier modules, so that the new set of modules are recognized in the dynamic queue of classifiers. The other issue is, that activity classes are often hardly separable, especially when the sensor data sources have no fixed position. This problem is solved through the used RFIS in combination with a filtering. The filtering separates reliable from unreliable classifications according to the uncertainty value provided by the RFIS mapping in the classifier modules.

We have shown in a case study, that the modular combination of new and pre-existing classifiers has nearly the same recognition rates (only 2 *pp* less accurate) as a classification system where all modules get trained together. Also the filtering on the uncertainty value achieves a boost of recognition up to 95%, which in the end makes activity recognition practical.

Acknowledgements

This work has been (partially) supported by the NTH School for IT Ecosystems.

References

1. AbuAarqob, O.A., Shawagfeh, N.T., AbuGhneim, O.A.: Functions defined on fuzzy real numbers according to zadehs extension. International Mathematical Forum 3(16) (2008)
2. Aksela, M., Laaksonen, J.: Adaptive combination of adaptive classifiers for handwritten character recognition. Pattern Recogn. Lett. 28(1), 136–143 (2007)
3. Berchtold, M., Riedel, T., Beigl, M., Decker, C.: Awarepen - classfication probability and fuzziness in a context aware application. Ubiquitous Intelligence and Computing (2008)
4. Berchtold, M., Riedel, T., van Laerhoven, K., Decker, C.: Gath-geva specification and genetic generalization of takagi-sugeno-kang fuzzy models. In: Proceedings of the SMC 2008 (2008)
5. Berchtold, M., Beigl, M.: Increased robustness in context detection and reasoning using uncertainty measures: Concept and application. In: Proceedings of the AmI 2009 (2009)
6. Brezmes, T., Gorricho, J.L., Cotrina, J.: Activity recognition from accelerometer data on a mobile phone. In: Proceedings of the IWANN 2009, pp. 796–799 (2009)
7. Chiu, S.: Method and software for extracting fuzzy classification rules by subtractive clustering. IEEE Control Systems Magazine, 461–465 (1996)
8. Gath, I., Geva, A.B.: Unsupervised optimal fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(7), 773–781 (1989)
9. Günther, H., Simrany, F.E., Berchtold, M., Beigl, M.: A tool chain for a lightweight, robust and uncertainty-based context. In: ARCS – CoSDEO Workshop (2010)
10. Györbíró, N., Fábián, A., Hományi, G.: An activity recognition system for mobile phones. Mobile Networks and Applications 14(1), 82–91 (2009)
11. Ianakiev, K.G., Govindaraju, V.: Architecture for classifier combination using entropy measures. In: Proceedings of the MCS 2000 (2000)
12. Kirchhoff, K., Bilmes, J.A.: Dynamic classifier combination in hybrid speech recognition systems using utterance-level confidence values. In: ICASSP 1999, pp. 693–696 (1999)
13. Ravi, N., Nikhil, D., Mysore, P., Littman, M.L.: Activity recognition from accelerometer data. In: Proceedings of the 17th IAAI, pp. 1541–1546 (2005)
14. Singh, S., Singh, M.: A dynamic classifier selection and combination approach to image region labelling. Signal Processing: Image Communication 20(3) (2005)
15. Tagaki, T., Sugeno, M.: Fuzzy identification of systems and its application to modelling and control. Systems, Man and Cybernetics (1985)
16. Xiao, B., Wang, C., Dai, R.: Adaptive combination of classifiers and its application to handwritten chinese character recognition. In: Int. Conference on Pattern Recognition (2000)

Online Workload Recognition from EEG Data during Cognitive Tests and Human-Machine Interaction

Dominic Heger, Felix Putze, and Tanja Schultz

Cognitive Systems Lab (CSL)
Karlsruhe Institute of Technology (KIT), Germany
{dominic.heger,felix.putze,tanja.schultz}@kit.edu

Abstract. This paper presents a system for live recognition of mental workload using spectral features from EEG data classified by Support Vector Machines. Recognition rates of more than 90% could be reached for five subjects performing two different cognitive tasks according to the flanker and the switching paradigms. Furthermore, we show results of the system in application on realistic data of computer work, indicating that the system can provide valuable information for the adaptation of a variety of intelligent systems in human-machine interaction.

1 Introduction

Machines play an important role in our everyday lives as matters of communication, work, and entertainment. However, in the interaction with humans, machines widely neglect different internal states of their users with the consequence of unnatural interaction, inadequate actions and inefficient adaptation to the user. Research in Human Machine Interaction and Affective Computing approaches this problem by developing systems that sense the current situation of the user and adapt to it. In this paper we contribute a system for automatic assessment of mental workload on short segments of EEG data for real-time adaptation of intelligent systems.

Terms, such as mental workload, task demand, engagement, vigilance, and others are often imprecisely used in literature to describe a human internal state of mental effort. Throughout this paper we use the term 'workload' as the amount of mental resources that are used to execute a current activity.

Numerous types of applications can strongly benefit from the ability to adapt themselves according to a detected workload level of their users. For example, in industrial production an optimal productivity of the operators could be maintained by monitoring their level of workload. Intelligent assistants, such as interactive driver assistance systems in future cars, could attempt to delay communication in difficult traffic situations to more suitable points of time in order to shift and balance the user's workload and improve safety while driving. In the near future, humanoid robots will be commercially available as household robots or to assist in elderly care. Therefore, social skills of robots in the interaction with humans will become essential. The robot could adapt the strategies of

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 410–417, 2010.

its spoken dialog system, for example by using shorter utterances in situations where its owner has a high workload. Thus, workload recognition can be a first step towards a social and empathic behavior.

The assessment of a user's workload must have a high temporal resolution, i.e. a workload prediction made from a short segment of data, so that a system can instantaneously react to it by adaptation. In contrast to other biosignals, such as heart rate variability, or skin conductance, EEG is a direct measure of the electric activity of the brain. Therefore, EEG can directly reflect mental processes and is the most suitable measure for workload estimations on the basis of time slices as short as one second length. In many situations, especially in professional working environments, sensors can be integrated into professional clothing (e.g. helmet), which attenuates concerns on wearing EEG devices.

2 Related Work

The analysis of workload from EEG data has a tradition in the psychological community. However, there is still no consensus of workload effects on the EEG. This section outlines systems that have a focus on the automatic computational assessment of workload based on EEG data.

Gevins and Smith [1] evaluated data from subjects performing different tasks of computer interaction and sequential memorization of stimuli (n-back tests). They used spectral features of the theta and alpha frequency bands from chunks of four second length and applied subject-specific multivariate functions and neural networks for discrimination of three different workload levels.

Berka et al [2] developed the B-Alert system for monitoring alertness and cognitive workload in different operational environments using a wireless EEG sensor headset. They applied the system to several tasks, such as sleep deprivation studies, military motivated monitoring (Warship Commander Task), reaction and digit sequence identifaction, as well as image memorization tasks. They predicted four states of alertness using a linear discriminant function on spectral features out of the range 3-40Hz derived from one second long epoches of data.

Kohlmorgen, et al. [3] measured workload of car drivers. They applied a highly parameterized feature extraction optimized for each subject and classification by Liner Discriminant Analysis. Parameters include data segment length (10-30 seconds), channel selection, spatial filters, and frequency bands (within 3-15Hz). Workload was induced by a secondary auditory reaction task and a tertiary task, which consisted of mental calculation or following one speaker in a recording of simultaneous voices. The authors could show that mitigation of high workload situations for the driver based on online detection of workload leads to improvements in reaction time for most subjects.

Honal and Schultz [4] analyzed task demand from EEG data recorded in lecture and meeting scenarios. They used Support Vector Machines (SVMs) and Artificial Neural Networks for classification and regression of features from short time Fourier transform (2 second epoches). To make brain activity measurements less cumbersome, they also evaluated a comfortable headband in addition to a standard EEG cap.

Putze, Jarvis, and Schultz [5] proposed a multimodal recognizer for different levels of cognitive workload. They recorded EEG data in addition to skin conductance, pulse, and respiration and classified it on one minute windows in a person independent way using SVMs. For their evaluation they used data recorded from subjects performing a lane change task in a driving simulator, while solving visual and cognitive secondary tasks.

The current paper contributes an online workload classification system based on one second long segments of EEG data and presents post-processing by voting and temporal smoothing of the prediction results. Results for training on one cognitive test paradigm and application of the system on the data of another are presented, which shows robustness in task variation of the proposed classifier. Finally, we show the abilities of the system to estimate a person's workload during realistic data of computer work. This indicates that a system trained on standardized data, such as cognitive tests and resting periods, can successfully be applied for detection of workload in realistic scenarios.

3 System Setup

In this section we describe the system setup of the proposed system. Figure 1 shows a block diagram of the processing stages involved.

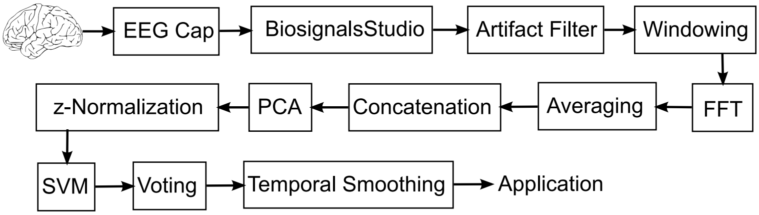


Fig. 1. Processing stages of the workload recognition system

3.1 Recording Setup

An active EEG-cap (Brain Products actiCAP) has been used for recording of EEG data. 16 active electrodes placed at positions FP1, FP2, F3, F4, F7, F8, T3, T4, C3, Cz, C4, P3, P4, Pz, O1, and O2 according to the international 10-20 system [6] have been recorded with reference to FCz. The impedance of each electrode was kept below 20 kΩ during each session. Amplification and A/D-conversion has been done using a 16 channel VarioPort biosignals recording system by Becker Meditec using a sampling rate of 256 Hz.

For data recording and stimulus presentation for the cognitive tests we used BiosignalsStudio (BSS) [7], which is a flexible framework for multimodal biosignal recording that has recently been developed at the Cognitive Systems Lab in Karlsruhe. BSS is also used as input layer for real-time EEG data acquisition during the online application of the proposed workload detection system.

3.2 Pre-processing and Feature Extraction

Contamination by artifacts is a severe problem for a reliable estimation of workload from EEG signals. Predominantly eye movement artifacts and muscular artifacts can be found in EEG signals recorded under non-laboratory conditions. Therefore, we applied a heuristic approach for artifact detection using thresholds on the signal power and its slope to identify artifacts in each data segment. Contaminated data segments are dropped and not used for classification or training.

The incoming stream of raw data from each electrode is cut into segments of one second length overlapping by 0.5 seconds. Each segment is multiplied by a Hamming window function to reduce spectral leakage. Afterwards, the windowed data segments are transformed to frequency domain using FFT. Three adjacent frequency bins are combined by averaging, which reduces noise in the data and lowers the dimensionality of the feature space. Thus, each coefficient in the resulting vector represents a frequency range of 3 Hz. The spectral features of each channel are concatenated to a final feature vector. As EEG data from neighboring channels and frequencies are highly correlated, we apply Principal Component Analysis (PCA) to further reduce the dimensionality of the feature space. A reduction from 224 dimensions (4-45Hz, 16 channels) to 100 dimensions explains 79.5-86.3% (mean 83.3%) of variance in the training data. The final vector for classification consists of z-scores of the PCA coefficients, i.e. normalization of each coefficient by subtracting the mean of each feature and dividing by its standard deviation determined on the training data.

3.3 Recognition and Post-Processing

For recognition we apply Support Vector Machines (SVMs) with linear kernels (LibSVM implementation [8]) to discriminate between low and high workload conditions. The penalty constant to control misclassification is set to $C = 1$ for each subject (parameter optimization by cross-validation did not improve the results). For each time step we calculate a majority vote using k previous predictions. This way the stability of recognition results can be increased, by the cost of temporal resolution. For the experiments presented in this paper we used $k = 3$, which gives a still high temporal sensitivity as the SVM delivers predictions every 500ms, while eliminating outliers and noise in the estimations.

Adaptation of an intelligent interaction system using binary workload estimations is usually unsuitable, as rapid changing of system behavior would appear unnatural to the user. Therefore, workload estimations of longer duration are required in addition to predictions gathered only from small data segments. To provide such information of workload trends with increased stability, we calculate the following workload index, which is a simple linear temporal smoothing of the prediction results:

$$load\_index(x) = \sum_{t=x-(l-1)}^{x} \frac{pred(t)}{l},$$

where x is a particular point of time, l is the smoothing length, and $pred(t)$ is the voted binary prediction by the SVM at time t. The temporal integration results in a workload index value in the range between 0 and 1. For the evaluations on computer work data we chose $l = 30$, i.e. linear smoothing on a time period of 15 seconds.

4 Evaluation

4.1 Cognitive Test Data

For training and evaluation of the system, workload and resting conditions from five subjects have been recorded. All of them are male students or employees of the Karlsruhe Institute of Technology (KIT).

We used the flanker paradigm and the switching paradigm for induction of workload. In both of these cognitive tasks subjects repeatedly react to stimuli presented on a display by pressing one of two keys on a keyboard using their left and right index finger.

During the flanker test, different horizontal arrays of five arrows are displayed (e.g. <<><<). Subjects respond as quickly as possible to the orientation of the middle arrow by pressing the corresponding left or right key. During the switching task, digits are presented on the screen surrounded by a dashed or solid square. A dashed square requires the subjects to indicate whether the stimulus is greater or lower than 5, while subjects need to decide whether the digit is odd or even, when a digit is surrounded by a solid square. Both tests require concentration and alertness and are especially demanding on visual-perceptual information processing. Workload is enhanced by the executive control required to overcome the interference of the presented stimuli in the flanker task and task switching decisions due to multitasking in the switching task.

We decided for these tasks for workload induction, because they allow a widely standardized evaluation of the system as the same stimuli are presented to each person in a controlled fashion. They only require little physical activity that could lead to artifacts caused by musclular activation (EMG) or eye movements. Moreover, they have some behavioral patterns similar to usual computer work (e.g. multitasking) and induce a constant level of workload, which is required to derive features from short time slices of data.

In contrast to the workload data, two resting conditions have been recorded, where the subjects were asked to relax without any activity keeping their eyes open. The workload recognition system is trained on the two classes of data: workload, i.e. subjects performing the flanker or switching task, and resting, i.e. subjects performing no particular task.

Due to temporal effects, such as high correlation of neighboring feature vectors calculated from small segments of EEG data, we did not use cross-validation for our evaluations. Instead, we used one cognitive task and one resting period as training data and evaluated the system on the other cognitive task and the second resting period recorded for each person. These evaluations also show the task robustness of the proposed system. Therefore, two systems were trained and evaluated for each subject by switching training and evaluation data.

Approximately six minutes of resting and workload data have been used for training of the system, with a balanced number of samples for each condition.

Table 1 shows the recognition results for different frequency bands that are usually used in EEG analyses [9], including theta, alpha, beta, and gamma band, as well as a full frequency range (4-45 Hz). Lower and higher frequencies are left out because they are more sensitive to artifacts.

Table 1. Recognition results on different frequency bands

Frequency Band / Subject ID	θ 4-7 Hz	α 8-13 Hz	β 14-38 Hz	γ 38-45 Hz	$\theta - \gamma$ 4-45 Hz
1	69.6%	74.4%	88.4%	96.4%	95.6%
2	81.6%	89.1%	84.5%	85.9%	86.0%
3	57.4%	83.9%	99.8%	99.4%	99.8%
4	50.5%	48.8%	74.4%	77.3%	75.2%
5	63.6%	82.3%	93.2%	94.7%	98.3%
Mean Recognition Rate	64.5%	75.7%	88.1%	90.7%	91.0%
Standard Deviation	11.9	15.9	9.5	9.0	10.3

The results indicate, that especially high frequencies are relevant for the classification task. This hypothesis is supported for example by [10], however we cannot confirm high recognition performance using theta and alpha activity (e.g. as proposed in [1]).

In addition to these person dependent results we used a leave-one-out cross-validation scheme for person independent evaluation of the system, i.e. both constellations for training and evaluation of each person are left out once from the training data for evaluation. Spectral features from frequency range 4-45 Hz again gave the best recognition results. The system delivered a recognition rates between 67.4% and 87.4% (mean 72.2% sd=9.0) using the data of the flanker and switching experiments.

4.2 Computer Work Data

In addition to the cognitive tests, we qualitatively evaluated the system on more realistic data in which subject 3 performs different tasks of common computer work. The classifier is trained on the same data as above, i.e. using data from a cognitive task and a resting period. The recording session consists of the following tasks of computer work:

It starts with a resting period of three minutes, i.e. no interaction with the computer. Next, a one minute long period of mail reading shows moderate workload classification results. Then, six minutes of writing an email have been recorded which show quite high workload. Next, starting approximately at minute ten, the subject has done undemanding internet surfing until minute 13, where the subject has started computer programming until approximately minute 16. Finally, another resting period of approximately one minute length has been recorded.

Figure 2 shows the time course of the load index during the session with the manually marked boundaries between the different periods. The blue curve gives the workload index as predicted from the workload recognizer. The results indicate that boundaries between workload periods can be identified with reasonable accuracy. Smoothing of the binary classification results, reveals not only relaxed and high workload conditions, but also can show moderate workload as in the mail reading condition, where a non extremal worload level is maintained over a period of several minutes. This can be explained by the uncertainty of the classifier in situations of medium workload or situations that repeatedly require high workload for short amounts of time.

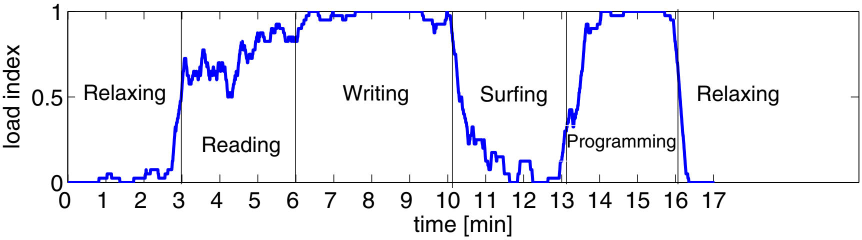


Fig. 2. Smoothed recognition results during different tasks of computer work

We also successfully applied the proposed workload recognizer to data of solving an test exam in computer science - showing high workload. In addition, the system has been applied to EEG recordings in a driving simulator. The results indicate that driving on a highway results in a rather relaxed user state, while demanding traffic situations are indicated by high workload.

5 Conclusion

In this paper we have proposed a system for assessment of workload from EEG data that can easily run in real-time on standard desktop computer or laptop. Excellent recognition results could be shown for person dependent classification for training and evaluation on two standardized cognitive tasks paradigms. Furthermore, we showed in a qualitative evaluation that a system can be trained using cognitive tests and applied to realistic data, such as computer work. The proposed system has been successfully demonstrated live several times, with different subjects and different scenarios, which indicates that it can provide valuable information for the adaptation of a variety of intelligent systems in human-machine interaction.

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) within Collaborative Research Center 588 “Humanoid Robots - Learning and Cooperating Multimodal Robots” [11].

6 Outlook and Future Work

In this paper we have shown the applicability of a system to detect workload in realistic scenarios. More detailed analyses and systematic evaluations are needed to get more insights of the capabilities and limitations of the system and the proposed workload index.

For a real application in human-machine interaction a major drawback of the proposed system is the fact that the acceptance of users to wear an EEG-cap outside experimental scenarios is rather low. Therefore, we experimented with less invasive wearable devices, which are more fashionable and, for example, integrated in clothes such as a hat or a headband. Furthermore, new electrode technologies, which do not require the use of conductive gel could strongly improve the usability and acceptance of the system.

Additional work needs to address the discrimination of different cortical activation patterns to identify used and available resources of a person. Such information could be helpful for an intelligent system to find a suitable adaptation scheme. For example, when intensive visual but low auditive processing is recognized, a system might provide the same information through an acoustic signal or speech synthesis, instead of using the visual communication channel.

References

1. Gevins, A., Smith, M.: Neurophysiological measures of cognitive workload during human-computer interaction. Theoretical Issues in Ergonomics Science 4(1), 113–131 (2003)
2. Berka, C., Levendowski, D., Cvetinovic, M., Petrovic, M., Davis, G., Lumicao, M., Zivkovic, V., Popovic, M., Olmstead, R.: Real-time analysis of EEG indexes of alertness, cognition and memory acquired with a wireless EEG headset. International Journal of Human-Computer Interaction 17(2), 151–170 (2004)
3. Kohlmorgen, J., Dornhege, G., Braun, M., Blankertz, B., Müller, K., Curio, G., Hagemann, K., Bruns, A., Schrauf, M., Kincses, W.: Improving human performance in a real operating environment through real-time mental workload detection. In: Toward Brain-Computer Interfacing, pp. 409–422
4. Honal, M., Schultz, T.: Determine task demand from brain activity. In: International Conference on Bio-inspired Systems and Signal Processing (2008)
5. Putze, F., Jarvis, J., Schultz, T.: Multimodal Recognition of Cognitive Workload for Multitasking in the Car. In: Accepted for International Conference on Pattern Recognition 2010 (2010)
6. Jasper, H.: The 10-20 electrode system of the International Federation. Electroencephalography and Clinical Neurophysiology 10, 371–375 (1958)
7. Heger, D., Putze, F., Amma, C., Wand, M., Plotkin, I., Wielatt, T., Schultz, T.: BiosignalsStudio: A flexible Framework for Biosignal Capturing and Processing. In: Dillmann, R., et al. (eds.) KI 2010. LNCS (LNAI), vol. 6359, pp. 33–39. Springer, Heidelberg (2010)
8. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001)
9. Zschocke, S.: Klinische Elektroenzephalographie. Springer, Heidelberg (2002)
10. Koles, Z., Flor-Henry, P.: Mental activity and the EEG: task and workload related effects. Medical and Biological Engineering and Computing 19(2), 185–194 (1981)
11. CRC588: Collaborative research center 588 humanoid robots - learning and cooperating multimodal robots, http://www.sfb588.uni-karlsruhe.de/

Situation-Specific Intention Recognition for Human-Robot Cooperation

Peter Krauthausen* and Uwe D. Hanebeck

Intelligent Sensor-Actuator-Systems Laboratory (ISAS),
Institute for Anthropomatics,
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
Peter.Krauthausen@kit.edu, Uwe.Hanebeck@ieee.org

Abstract. Recognizing human intentions is part of the decision process in many technical devices. In order to achieve natural interaction, the required estimation quality and the used computation time need to be balanced. This becomes challenging, if the number of sensors is high and measurement systems are complex. In this paper, a model predictive approach to this problem based on online switching of small, situation-specific Dynamic Bayesian Networks is proposed. The contributions are an efficient modeling and inference of situations and a greedy model predictive switching algorithm maximizing the mutual information of predicted situations. The achievable accuracy and computational savings are demonstrated for a household scenario by using an extended range telepresence system.

1 Introduction

Recognizing a human's intentions, plans, and actions is crucial to the facilitation of nonverbal human-robot cooperation. Intention recognition [1] is the process of estimating the force driving human actions based on noisy observations of the human's interactions with his environment. For example, a robot embedded in a household can assist the human at its best, when estimating whether the human wants to *cook, wash, etc.*, based on observations such as his location, grasping activity, and object interactions [2]. In general, approaches to intention recognition may be categorized into symbolic approaches [3], probabilistic approaches, [4], [5] and blends thereof [6]. The key difference between symbolic and probabilistic approaches is that in the former the possibility of an intention is deduced, while in the latter the probability of an intention is inferred. In this paper, intention recognition is considered a discrete-time state estimation problem formalized in a *Dynamic Bayesian Networks* (DBN) [7], with a state containing the set of intentions to be estimated [2], [8], [9]. In order to allow for an intuitive interactive cooperation, efficient online inference in these models must be achieved. The main challenge to be addressed is the large size of the measurement systems caused by the large number of features sensed.

The problem of inference with large measurement models is at the intersection of three research areas: *structured Bayesian Networks* (BN) [10],[11], *switching models*,

* This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Center SFB 588 on "*Humanoid robots – Learning and Cooperating Multimodal Robots*".

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 418–425, 2010.

and *sensor selection*. A hierarchical fusion of measurements is modeled by a measurement system in form of a BN. Existing approaches to modeling and solving large-scale *structured BNs* include *Object-Oriented BNs* (OOBN) [12] and *Situation-Specific BNs* [13]. Although facilitating the handling of large BN greatly by the introduction of reusable objects and class structures, it is hardly possible to construct a BN for a specific query at each time step and still obtain real-time performance. In contrast, *Multi-nets* [14] are focused on dynamically *switching* the structure of a single network based on the state estimate. Extending this approach to considering only a subset of the measurement model for one BN may be understood as a *sensor selection* problem [15] over time. The specific difficulty for the intention recognition lies in the complexity of the measurement system, i.e., layered information fusion. In [16], an approach to selecting subsets of sensors for such a model, maximizing the mutual information **I** between the state and sensor subsets, was proposed. For the used sensor synergy graph, **I** has to be computed for all pairs of sensors.

In this paper, inference in large models with many features is addressed, where number of features prohibits the exhaustive calculations in [16]. The specific properties of the intention recognition problem are exploited to circumvent the exhaustive calculation, by using the natural decomposition of the problem into situations. A model predictive switching between reduced models is proposed. The reduced models correspond to sets of intentions summarized in situations and include the respective subsets of the measurement system, i.e., BN with less nodes and smaller state spaces. Performing inference with the reduced models only, computational savings can be obtained at a modest loss in accuracy.

2 Intention Recognition

Given a specific situation, a human has a set of coarse intentions, which manifest in fine-grained interactions with the world—the actions. These definitions are abstract and need to be instantiated for each application at hand, i.e., the set of possible actions, levels of abstractions, and the intentions need to be found for each restricted task. Given these limitations, a temporal causal forward model (CFM) for specific tasks and situations is derivable, as depicted in Fig. 1. Nodes reflect quantities, e.g., intentions or actions, and edges correspond to relations among them. Intention recognition is the process of inferring the human state by fusing the various online incoming observations (e.g., video streams or tracking results) according to this CFM, which encodes the human's rationale. The inset in Fig. 1 shows a detailed view of a complicated measurement system. This causal forward model may be simply converted into a BN, i.e., a probabilistic graphical model, by interpreting each quantity, e.g., the intention, as a random variable and the relations among random variables as conditional densities, e.g., $f(\underline{a}_t|\underline{a}_{t-1},\underline{i}_t)$ These conditional densities f quantify how likely value combinations are. For our purpose, it is necessary to model relations between hybrid sets of variables, i.e., sets of continuous and discrete valued variables, as well as nonlinear dependencies. For this reason, hybrid BN with mixture conditional densities are employed [17], as they allow for a unified treatment. Fig. 2 shows a block diagram representation of the BN for intention recognition. Here, $\underline{d}_t$ denotes domain knowledge, $\underline{s}_t$ the situations, $\underline{i}_t$ the

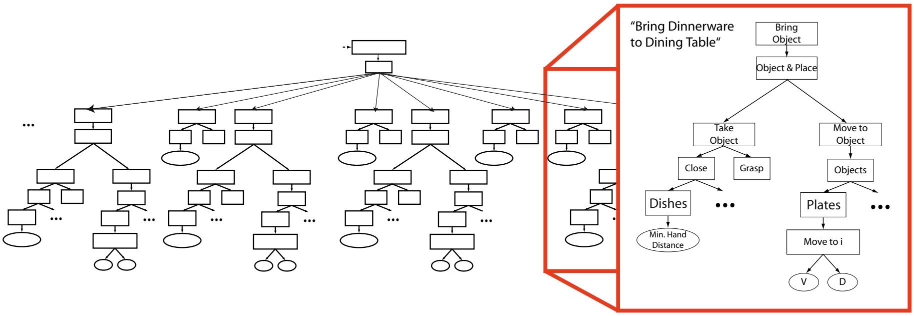


Fig. 1. Temporal causal forward model for one time step and detailed action recognition fragment, visualizing a part of the large measurement systems

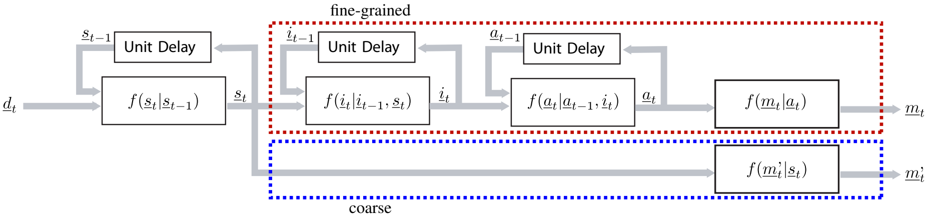

fine-grained
coarse
$\underline{s}_{t-1}$
Unit Delay
$\underline{i}_{t-1}$
Unit Delay
$\underline{a}_{t-1}$
Unit Delay
$\underline{d}_t$
$f(\underline{s}_t|\underline{s}_{t-1})$
$\underline{s}_t$
$f(\underline{i}_t|\underline{i}_{t-1},\underline{s}_t)$
$\underline{i}_t$
$f(\underline{a}_t|\underline{a}_{t-1},\underline{i}_t)$
$\underline{a}_t$
$f(\underline{m}_t|\underline{a}_t)$
$\underline{m}_t$
$f(\underline{m}'_t|\underline{s}_t)$
$\underline{m}'_t$

Fig. 2. Block diagram of the BN for the intention recognition. Note that the conditional density $f(\underline{m}_t|\underline{a}_t)$ subsumes the entire BN measurement system, e.g., corresponding to the inset in Fig. 1.

intentions, $\underline{a}_t$ the actions, $\underline{m}_t$ and $\underline{m}'_t$ the measurement variables for time step t, which will be explained in the next two sections. The corresponding conditional densities also are given, showing the temporal dependency of consecutive time steps. For the rest of this paper, it is assumed that $\underline{d}_t$ and $\underline{s}_t$ remain constant and only the model defined by $\mathscr{M}_t := \{\underline{i}_t, \underline{a}_t, \underline{m}_t\}$ will be considered. In order to infer the intention, information about the domain $\underline{d}_t$ and preceding time steps will be propagated forward, whereas measurements will be processed by a Bayesian backward inference step [17].

3 Online Model Switching

For non-trivial applications, the models constructed in Sec. 2 entail hierarchical and sequential models, relating atomic actions to multilevel action sequences. The resulting model is too large to allow for an interactive human-robot-cooperation. In this section, an online switching between reduced models is proposed to alleviate this problem.

Modeling Situations

A situation is a set of necessary conditions for human behavior, i.e., a set of conditions restricting the set of possible intentions. The constraints for situation s_i may be user given or determined automatically, e.g., spatio-temporal constraints may be obtained from an analysis of the human's movement profiles. The relations are assumed to be given in conjunctive normal form, similar to

$$\text{Cooking} : \text{'Near stove'} \wedge (\text{'12 AM'} \vee \text{'Food present'}) \equiv s_i : (\wedge \vee \underline{m}'_{i,t}) \,. \tag{1}$$

The $m'_{i,t}$ correspond to measurable quantities. Note that $\underline{m}'_t = [m'_{d,t} \dots m'_{e,t}]^{\mathrm{T}}$ is not identical with $\underline{m}_t$. Typically, $m'_{i,t}$ are coarse features used exclusively for situation assessment. Incorporating these constraints into the overall BN corresponds to appending a variable $\underline{s}_t = [s_{1,t} \cdots s_{n,t}]^{\mathrm{T}}$, where each entry corresponds to a specific situation. For each s_i, variables for the terms The $m'_{i,t}$ are introduced and connected via *inverse-OR* nodes. The resulting clauses are then connected to *inverse-AND* nodes to arrive at a BN corresponding to (1). Performing this operation for all $s_{i,t}$ yields a measurement system for the situation model. In Fig. 2, this measurement system corresponds to the bottom system. Yet, as the situation is causal to the intentions $\underline{i}_t$ the situation BN can be merged with the intention recognition BN by introducing the dependency $f(\underline{i}_t|\underline{s}_t)$. One obtains a larger BN, where the intention recognition works as a fine-grained measurement model and the newly constructed situation BN functions as a coarse model. The dynamical development of the situation is accounted for by introducing $f(\underline{s}_{t+1}|\underline{s}_t)$.

Reduced Models

The complete BN for the intention recognition $\mathscr{M}$ resembles the causal forward model in Fig. 1 and is substituted by a set of reduced models $\mathscr{M}^k$

$$\mathscr{M}_t := \{\underline{i}_t, \underline{a}_t, \underline{m}_t\} \equiv \left\{\mathscr{M}_t^1 := \left\{\underline{i}_t^1, \underline{a}_t^1, \underline{m}_t^1\right\}, \mathscr{M}_t^2 := \left\{\underline{i}_t^2, \underline{a}_t^2, \underline{m}_t^2\right\}, \dots\right\} . \tag{2}$$

The model $\mathscr{M}_t$ is defined by all intentions $\underline{i}_t = [i_{1,t} \cdots i_{o,t}]^{\mathrm{T}}, i_{j,t} \in \mathscr{A}_t$, all action subsystems $\underline{a}_t$ and the respective measured variables $\underline{m}_t$ as well as the respective probabilistic models, e.g., $f(\underline{i}_t|\underline{s}_t)$. The reduced models $\mathscr{M}_t^k$ are defined by only some intentions $\underline{i}_t^k = [i_{1,t} \cdots i_{h,t}]^{\mathrm{T}}, i_{j,t} \in \mathscr{A}_t^k \subseteq \mathscr{A}_t$ and the respective action subsystems with corresponding measured variables defined accordingly. The index t emphasizes the dependency on the time step. Note that the conditional densities need to be adapted for switching between different state spaces, e.g., over time $f(\underline{i}_{t+1}^k|\underline{i}_t^k)$ and $f(\underline{i}_t^k|\underline{s}_t)$. The coarse static measurement system of $\underline{m}'_t$ for the situation recognition is contained in all $\mathscr{M}_t^k$, to allow for fast situation recognition. This measurement system is not subject to switching.

Switching Algorithm

Based on the above situation model and the $\mathscr{M}_t^k$, an online model switching algorithm is proposed for choosing the model $\mathscr{M}_{t+1}^k$, which maximizes the reduction of uncertainty, measured by the mutual information $\mathbf{I}$, over the future situation $\underline{s}_{t+1}$ given $\underline{i}_{t+1}^k$. Additionally, a term $P(\mathscr{M}_t, \mathscr{M}_{t+1})$ penalizing a frequent model switching is added to the objective function

$$\mathscr{M}_{t+1}^* = \arg\max_{\mathscr{M}_{t+1}^k} \underbrace{\mathbf{I}(\underline{s}_{t+1}; \underline{i}_{t+1}^k) + \lambda\, P(\mathscr{M}_t, \mathscr{M}_{t+1}^k)}_{=:V(\mathscr{M}_{t+1}^k)} . \tag{3}$$

Maximizing (3), the model $\mathscr{M}_{t+1}^{*}$ is selected, which is most consistent with the expected situation at $t+1$, which is predicted using $f(\underline{s}_{t+1}|\underline{s}_t)$. The parameter λ is a weight, representing our belief in the need for penalization. Furthermore, is assumed, that the $\mathscr{M}_t^k$ are discrete-valued only.

Alg. 1 summarizes the online model selection. It should be noted that this algorithm is greedy and only selects the best model with respect to (3) for a one-step horizon. Additionally, in contrast to [16], not $\mathbf{I}(\underline{s}_{t+1};\underline{m}_{t+1})$ is calculated, but an approximation in form of the intention is calculated—allowing an online application of the approach. Additionally, the calculation of $\mathbf{I}(\underline{s}_{t+1};\underline{i}_{t+1}^k)$ is approximated by neglecting the dependency $f(\underline{i}_{t+1}|\underline{i}_t)$. Besides the dynamics of $\underline{s}_{t+1}$ and $\underline{i}_{t+1}$, the measurement systems might contain dynamic dependencies too, which need to be considered. Regarding scalability, the algorithm is based on the decomposability of the human's intentions according to situations. The performance regarding estimation quality and computational savings is governed by the specific decomposition chosen. The experiments show, that given such a decomposition, the approach allows for efficient intention recognition.

Algorithm 1. Situation and Intention with Online Model selection

```
Input: Models $\mathscr{M}_t^k$, initial model $\mathscr{M}_0$, $\lambda$, penalty function $P(.,.)$
  for all Time-steps $t$ do
    Perform inference with $\mathscr{M}_t^k$, e.g., message passing [10], [17] $\rightarrow f(\underline{s}_t)$ // Inference
    for all $\mathscr{M}_{t+1}^k$ do
      Calculate $G_t^k := V(\mathscr{M}_{t+1}^k)$ // Model evaluation
    end for
    ${\mathscr{M}_{t+1}^k}^* = \arg\max_{\mathscr{M}_t^k} G_t^k$ // Model selection
  end for
```

4 Experiments

The proposed approach will be implemented for close cooperation with a humanoid robot in a household setting. To validate the approach, an extended range telepresent virtual household scenario is employed. In the rest of this section, this testbed and the results are discussed.

Telepresent Virtual Household Setting. For the experiments, the *extended range telepresence system* from [18] was used. It allows the user to move in a virtual 1:1-scale kitchen model of the original household. Head and hand positions are tracked by an acoustic system in his real environment and are mapped to the virtual world. Thus, the human test person moves naturally in his environment and the virtual environment. The noise characteristics are similar to the expected capability of the vision system of the real robot. Additionally, the human grasping activity is measured by means of a bluetooth cyber-glove device. During the actual experiments, a test person was instructed to carry out a typical action sequence in a kitchen: *lay table*, *prepare a meal* followed by *clean dishes*. For this sequence, the test person has to cross a room, pick up some dinner ware, and bring it to the table on the opposite side of the kitchen. After

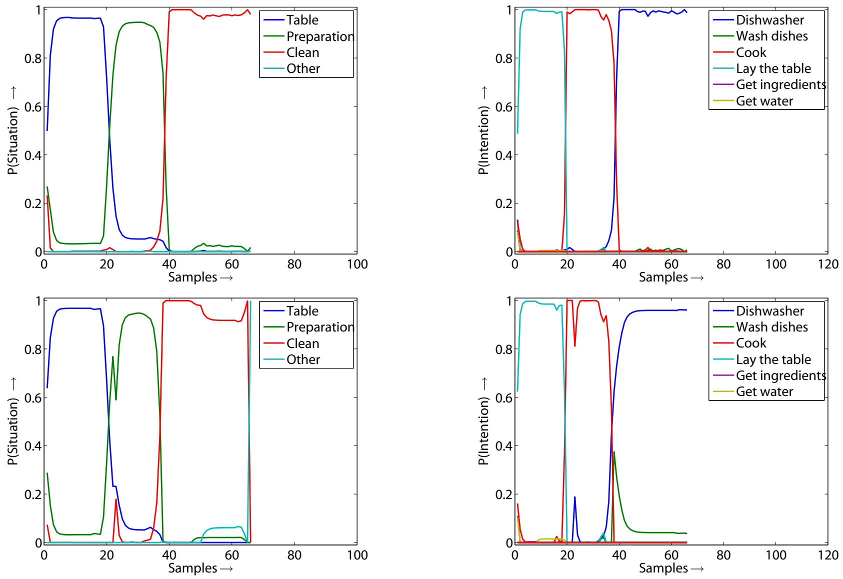


Fig. 3. Situation (left) and intention estimates (right) using the full (top) and reduced (bottom) models with the proposed online switching approach over time

fetching some ingredients, these are put into cooking pot. The pot is put onto the stove. Later, the initially used dinnerware is picked up and put into the dishwasher. For this test setup, only a subset of the actually available situations and intentions is used.

Results. In this section, the achieved accuracy is compared to the computational savings of using the reduced models only for two model sizes. In Ex. 1 ten intentions were estimated with a BN of 319 nodes. Ex. 2 comprises 15 intentions and 625 nodes. Each model was decomposed into four reduced models with about 200-300 nodes per reduced model. For the small model, the situation and intention estimates are given in Fig. 3. The top images in Fig. 3 show the estimation results over time for the unswitched approach and the lower images show the estimation results for the switched approach. The estimates of the two approaches align nicely. Yet, some abrupt changes in the estimates can be observed. These occur when the model switches, as can be seen in Fig. 4, due to the different parametrization and state spaces of the models. In Fig. 4, the error, as the absolute difference between the estimates using the full and the reduced model, are given. Tab. 1 gives a statistic of the error in the situation and intention estimates for both models. In these tables the maximum and average mean absolute difference over all situations/intentions is given with its variance. The errors are averaged over time. Throughout the experiment the average error is modest, except where the model is switched. This leads to the conclusion that the approach works well, but a smooth model transition needs to be further investigated. The approach is relatively robust to misclassifications as can be seen in Fig. 4 for Ex. 2 where a situation is misclassified, leading to a drastic increase in error but after an additional situation change the approach aligns with the

Table 1. Maximum and average difference between the situation estimates $\underline{s}_t$ (intention estimates $\underline{i}_t$) using the full and the reduced model and the respective standard deviation for both experiments. On the right, the used average number of nodes and computation time per step are given.

	$\underline{s}_t$	Absolute Error	$\underline{i}_t$	Absolute Error		Full	Switched
Ex. 1:	Max.	0.050 ±0.0099	Max.	0.060 ±0.0112	Avg. #Nodes	319	224.7
	Avg.	**0.025 ± 0.0024**	Avg.	**0.012 ± 0.0004**	Avg. Time in s	0.032	0.025

	$\underline{s}_t$	Absolute Error	$\underline{i}_t$	Absolute Error		Full	Switched
Ex. 2:	Max.	0.274 ±0.1701	Max.	0.325 ±0.1584	Avg. #Nodes	625	233.1
	Avg.	0.137 ±0.0428	Avg.	0.065 ±0.0063	Avg. Time in s	**0.234**	**0.026**

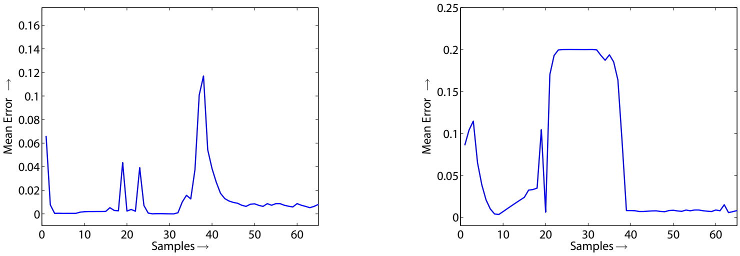


Fig. 4. Mean absolute error of $\underline{i}_t$ over time for Ex. 1 (left) and Ex. 2 (right)

true situation. Regarding the computational savings, we compare the average number of nodes used in the BN given in Tab. 1 for Exp. 1 and Exp. 2. In both experiments, the employed BN are structured like Fig. 2. For both experiments the average number of nodes used is significantly reduced and the computation time is decreased by up to one order of magnitude. Note, even though Exp. 2 contains larger models, only a modest increase in the average number of nodes can be observed. In Exp. 2 larger parts of the model may be ignored, showing that the result will scale favorably for real scenarios.

5 Conclusion

In this paper, an approach to efficient intention recognition based on situation-specific model switching is proposed. To this end, a way of modeling situations and its integration with a BN, encoding the causal forward model of the human's rationale, was explicated. Based on a measurement system for the coarse situation assessment, which checks constraints for objects and places, a model predictive approach to online switching of reduced Dynamic Bayesian Networks for the intention recognition was proposed. At each time step, the best reduced model is selected based on the maximization of mutual information of the predicted situation and intention estimates, avoiding the expensive mutual information calculations for all sensors. Using an extended range telepresence system, the proposed approach was shown to deliver computational savings of up to one order of magnitude at an acceptable error level. In future work, it should be investigated if longer prediction horizons allow for improved switching performance and the neglected dynamic dependencies may be considered.

References

1. Schmidt, C.F., Sridharan, N.S., Goodson, J.L.: The Plan Recognition Problem: An Intersection of Psychology and Artificial Intelligence. Artificial Intelligence 11(1-2), 45–83 (1978)
2. Schrempf, O.C., Hanebeck, U.D., Schmid., A.J., Wörn, H.: A Novel Approach to Proactive Human-Robot Cooperation. In: Proceedings of the 2005 IEEE International Workshop on Robot and Human Interactive Communication (ROMAN 2005), Nashville, Tennessee, pp. 555–560 (2005)
3. Kautz, H.A.: A Formal Theory of Plan Recognition and its Implementation. In: Reasoning About Plans, pp. 69–125. Morgan Kaufmann Publishers, San Mateo (1991)
4. Pynadath, D.V., Wellman, M.P.: Probabilistic State-Dependent Grammars for Plan Recognition. In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI 2000), pp. 507–514. Morgan Kaufmann Publishers, Inc., San Francisco (2000)
5. Bui, H.H.: A General Model for Online Probabilistic Plan Recognition. In: Proc. of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 1309–1315 (2003)
6. Geib, C.W., Goldman, R.P.: Partial Observability and Probabilistic Plan/Goal Recognition-ijcai 2005 workshop on modeling others from observations (2005)
7. Murphy, K.: Dynamic Bayesian Network: Representation, Inference and Learning. PhD thesis, UC Berkeley (2002)
8. Tahboub, K.A.: Intelligent Human-Machine Interaction Based on Dynamic Bayesian Networks Probabilistic Intention Recognition. Journal of Intelligent and Robotic Systems 45(1), 31–52 (2006)
9. Krauthausen, P., Hanebeck, U.D.: Intention Recognition for Partial-Order Plans Using Dynamic Bayesian Networks. In: Proceedings of the 12th International Conference on Information Fusion (Fusion 2009), Seattle, Washington (2009)
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan-Kaufmann Publishers, Inc, San Francisco (1988)
11. Koller, D.: Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge (2009)
12. Koller, D., Pfeffer, A.: Object-Oriented Bayesian Networks. In: Proceedings of the 13th Annual Conference on Uncertainty in AI (UAI 1997), Providence, Rhode Island, pp. 302–313 (1997)
13. Laskey, K., Mahoney, S.: Network Fragments: Representing Knowledge for Constructing Probabilistic Models. In: Proceedings of the 13th Annual Conference on Uncertainty in AI (UAI 1997), Providence, Rhode Island, pp. 334–341 (1997)
14. Bilmes, J.: Dynamic Bayesian Multinets. In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 38–45. Morgan Kaufmann Publishers, Inc., San Francisco (2000)
15. Williams, J.L.: Information Theoretic Sensor Management. PhD thesis, Massachusetts Institute of Technology (2007)
16. Zhang, Y., Ji, Q.: Efficient Sensor Selection for Active Information Fusion. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 99 (2009), ISSN: 1083–4419
17. Schrempf, O., Hanebeck, U.D.: Evaluation of Hybrid Bayesian Networks using Analytical Density Representations. In: Proc. of the 16th IFAC World Congress (IFAC 2005), Czech Republic (2005)
18. Rößler, P., Beutler, F., Hanebeck, U.D., Nitzsche, N.: Motion Compression Applied to Guidance of a Mobile Teleoperator. In: Proceedings of the 2005 IEEE International Conference on Intelligent Robots and Systems (IROS 2005), pp. 2495–2500 (2005)

Towards High-Level Human Activity Recognition through Computer Vision and Temporal Logic

Joris Ijsselmuiden[1] and Rainer Stiefelhagen[1,2]

[1] Fraunhofer IOSB Karlsruhe
[2] Institute for Anthropomatics, Karlsruhe Institute of Technology
joris.ijsselmuiden@iosb.fraunhofer.de, rainer.stiefelhagen@kit.edu

Abstract. Most approaches to the visual perception of humans do not include high-level activity recognitition. This paper presents a system that fuses and interprets the outputs of several computer vision components as well as speech recognition to obtain a high-level understanding of the perceived scene. Our laboratory for investigating new ways of human-machine interaction and teamwork support, is equipped with an assemblage of cameras, some close-talking microphones, and a videowall as main interaction device. Here, we develop state of the art real-time computer vision systems to track and identify users, and estimate their visual focus of attention and gesture activity. We also monitor the users' speech activity in real time. This paper explains our approach to high-level activity recognition based on these perceptual components and a temporal logic engine.

Keywords: activity recognition, computer vision, temporal logic.

1 Introduction

There has been much progress lately in the visual perception of humans and their interaction with other people as well as with machines [1]. Most studies in this area however, focus on sub-problems like tracking or gesture recognition. What is often missing, is a second layer that spans a variety of perceptual components and fuses and interprets their outputs.

Our goal is to develop a framework for high-level human activity recognition. To achieve this, we fuse the available outputs from perception, cast them into a temporal framework, and use a logic engine containing context knowledge to deduce high-level facts. Regardless of the sensor setup or application domain, this framework should provide an abstract understanding of the given scene. For example, we can detect a group meeting or two people working together at a display. This can be used to adapt user interfaces accordingly, to automatically generate reports and visualizations, and to provide perceptual components with top-down knowledge. Application domains are: ambient intelligence and smart environments [2], but also robotics, surveillance and videosearch.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 426–435, 2010.

This work is part of a larger project where computer vision and a wide array of other techniques are used to develop alternatives to the traditional mouse and keyboard controlled GUI. We are working towards real-world applications in control rooms for fire brigade, police, medical services, technical relief, military, and private security firms [3]. Besides for applications in human-machine interaction, we use computer vision, speech recognition, and high-level activity recognition, to automatically generate reports and visualizations.

The paper is organized as follows. In Section 2, we discuss some related work in activity recognition. Section 3 briefly explains the computer vision and speech recognition systems we use. Then, Section 4 describes our approach to multimodal fusion and activity recognition on top of these perceptual components. First experiments are presented in Section 5. And Section 6 provides the conclusion and some thoughts on future work.

2 Related Work

As the performance of computer vision systems increases, the interest in multimodal fusion and high-level activity recognition increases with it. A survey on existing approaches to activity recognition can be found in [4]. And [5] describes the state of the art in multimodal fusion for human-computer interaction.

Our methods for multimodal fusion and activity recognition were inspired by [6], where a temporal framework is used to define composite interactions between people in terms of atomic actions. Typical interactions they want to detect are: fighting, greeting, assault, and pursuit. Much like ourselves, they take advantage from the fact that perception is only concerned with learning and detecting the atomic actions. Complex compositions are provided by the layer on top. They also propose a probabilistic reasoning component that solves for missing and superfluous atomic actions.

In [7], instead of cameras and microphones, they use wearable motion and RFID sensors for recognizing activities of daily living. Nonetheless, we were able to draw inspiration from their work, since we strive for a general purpose framework, independent of sensor setup and application domain. They use an emerging patterns approach and sliding time windows to classify sequential, interleaved, and concurrent activities. A more classical approach is [8], where the perceptual outputs are fed to Hidden Markov Models after some preprocessing. Here, speech detection, ambient sound detection, tracking, and posture estimation are used to classify social activities in an ambient intelligence setting.

In [9], Fuzzy Metric-Temporal Horn Logic is used to generate natural language descriptions from vehicle trajectories. A similar approach, applied to human movement patterns, is presented in [10]. The work presented in [12] is concerned with storyline extraction from sports videos using and-or graph representations. This approach overcomes some of the limitations of Hidden Markov Models and Dynamic Bayesian Networks, because not only the model parameters are learned, but the model structures too. In [11], and-or graphs are used to generate text descriptions for a large dataset containing many different types of videos and images. Note that these last four studies are not just concerned with classifying activities. They also generate corresponding reports in natural language.

The novelty and contribution of our work lies in the fact that we fuse and interpret a large variety of real-time perceptual components in a single model. Also, we use a high-level temporal logic based approach as opposed to low-level statistical methods. More novelty is provided by our application domain: human-machine interaction and teamwork support in crisis response control rooms.

3 Perception

Our experiments take place in a laboratory of six by nine meters, equiped with eleven cameras. Four are located in the room's upper corners and one fish-eye camera is mounted at the ceiling's center. Another four look down from the ceiling onto the area in front of our videowall. And two active pan-tilt-zoom cameras are mounted on the walls at head height. Each computer vision component uses a specific subset of these. For speech recognition, we use four close-talking microphones. And on the interaction side, a videowall serves as the prime user interface. All components described in this paper run on eight off-the-shelf PCs, connected through a dedicated LAN middleware. We currently use four perceptual components: tracks and identities, visual focus of attention, gestures, and speech (see Figure 1). They provide the following information: who are present in the room, where are they located, what are they looking at, what gestures are they performing, and what are they saying. To obtain a complete scene understanding, the perceptual information is supplemented by information that is not grounded in perception: what objects are in the room, where are they located, and what is currently being displayed on the room's user interfaces.

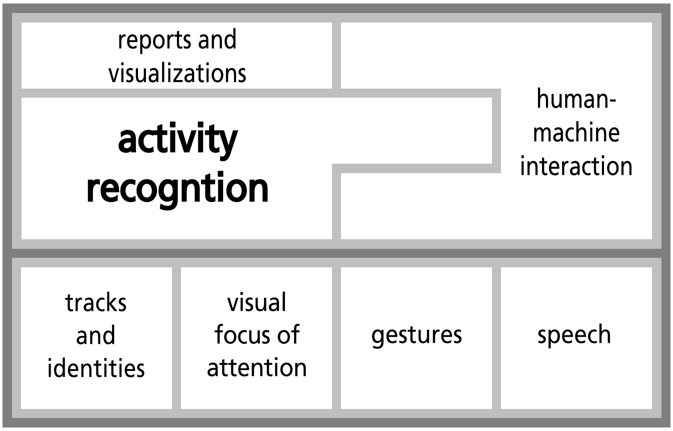

reports and visualizations
activity recogntion
human-machine interaction
tracks and identities
visual focus of attention
gestures
speech

Fig. 1. Schematic representation of the system's components. The bottom layer provides the top layer with perceptual information about the people in the room.

Tracks and Identities. Users' locations and identities are arguably the most fundamental aspects of the observed scene. Tracking is performed using a multi-level particle filter approach on the four corner cameras and the central fish-eye camera [13]. For face identification, we use the two pan-tilt-zoom cameras on the walls and a DCT-based local appearance model [14].

Visual Focus of Attention. The person or object being looked at is another important factor in the perception of human interaction. Our laboratory's unconstrained camera placement yields low resolution images that put strong limitations on the perception of peoples' eyegaze, so we make approximations using head pose angles. The four corner cameras are used to generate head appearance hypotheses (i.e. head positions, sizes, and angles), and a combined particle filter framework rates the hypotheses with local shape descriptors and artificial neural networks. The successive deduction of a person's visual focus of attention (person or object) is then obtained by exploiting a linear relationship between measured head pose and actual gaze angles [15].

Gestures. Teamwork at large videowalls plays an important role in control room design [3]. Touch-sensitive surfaces seem to offer a natural way of interaction here. However, touch alone is not sufficient, because it forces users to walk along the videowall, and objects at the top can simply be out of reach. We overcome this limitation by adding the possibility of pointing at the videowall from a distance. The four cameras around the videowall are used for 3D body reconstruction through a visual hull approach implemented on the GPU [16]. The whole process is based on video cameras alone and it does not require a special surface.

Speech. Close-talking microphones and state of the art speech recognition software [17] provide us with an extra modality for activity recognition and human-machine interaction. We can currently use speech recognition combined with the pointing gestures described above to add tactical symbols to a digital map. And we can monitor who is speaking and who is silent for up to four users.

4 Multimodal Fusion and Activity Recognition

We use the perceptual information described above, knowledge from the domain of human interaction, and a temporal logic engine, to fuse the different sources and deduce high-level facts in real time. In other words, tracks, identities, visual focus of attention, gestures, and speech, but also information about furniture, user interfaces, and other objects, are fused and analysed to build an abstract model of the situation in the room. While keeping in mind our application domain [3], we strive for general purpose solutions to high-level activity recognition. Deduced high-level facts can be used to adapt user interfaces accordingly, to automatically generate reports and visualizations, and to provide the perceptual layer with top-down knowledge. Typical situations we want to detect are: two people having a conversation, two people working together at the videowall, a group meeting, other subgroup-constellations, and users' roles in control room settings.

The perception layer also controls human-machine interaction directly, without a component for multimodal fusion and activity recognition in between. For example, we use tracks and identities to display user specific information close to the corresponding user. And a mixture of hand gestures and speech commands can manipulate objects on the videowall. In such cases, the human-machine interaction component performs its own limited fusion and interpretation. For handling complex situations however, a dedicated component for multimodal fusion and activity recognition is essential.

4.1 Model Taxonomy

The first step is to use the LAN middleware to subscribe to all the output streams generated by the perception layer. Each message received in this manner triggers the proper events as follows. During initialization, a scenario object is constructed containing a situation object for time $t = 0$, representing an empty room. Then, events are triggered that correspond to objects being put into the room, resulting in a situation object for $t = 1$. After initialization, a timestep and corresponding situation object is added to the scenario for each message batch received. Each situation is a copy of the last, augmented through events that are triggered by the incoming messages (see Figure 2).

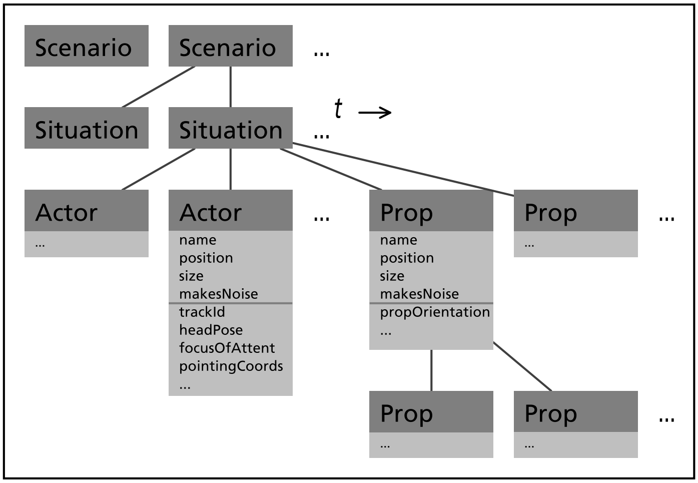


Fig. 2. Schematic representation of the model taxonomy. Actors represent the people in the room, and props can be user interfaces, furniture, or other objects. Situations are ordered in time and transitions between them are triggered by events in the perception layer. This provides the basis for temporal reasoning.

Each situation contains two types of entities: actors and props. The actor objects correspond to the people in the room. Currently, they contain identity, position, size, and knowledge about head pose, visual focus of attention, pointing gestures, and speech activity. All information needs to be assigned to the correct actor object, which requires context knowledge and reasoning. The props contained in each situation represent the objects in the room: furniture and their components, user interfaces and the elements displayed on them, and any other objects that might be there. Props' components are themselves props to allow for arbitrary complexity.

These descriptions of actors and props are used as facts by a logic engine composed of domain knowledge rules. It deduces high-level facts, both within each situation and over sequences of situations, which amounts to temporal reasoning. The logic engine is implemented using the Castor Logic Library, which allows for seamless multiparadigm programming. The power and flexibility of C++ and its imperative paradigm can be mixed at will with the reasoning abilities and convenience of the Castor Logic Library and its declarative, PROLOG-like paradigm [18]. Castor is a small and elegant header-only template library that is open source and without dependancies. Truth conditions for atomic predicates can be of arbitrary complexity, using the full C++ language. And the predicates formed like this are immediately available for spatial, temporal, and logical composition.

4.2 Temporal Logic

We adopt the temporal interval relations before, meets, overlaps, starts, during, finishes, and equals, as defined in [19]. Temporal composition allows us to classify situations that involve changing behavior over time, and it filters out observations that are too short-lived to be of interest. The following example represents our current definition of coordinated interaction, where two people are interacting with a videowall and a third person is seated at a table, giving them instructions (Figure 3, bottom row, second column). For the experiments described in Section 5, we implemented five such rules.

$$\begin{array}{l}\forall\ x, y, z\ \ (TalksTo(x, y, t, u)\ \vee\ TalksTo(x, z, t, u))\ \wedge\\ CloseTo(x, table, t', u')\ \wedge\ CloseTo(y, videowall, t', u')\ \wedge\\ CloseTo(z, videowall, t', u')\ \ \wedge\ \ y \neq z\ \ \wedge\ \ t\ \text{ DURING }\ t'\\ \rightarrow\ CoordinatedInteraction(x, y, z, t)\end{array}$$

$$\begin{array}{lcl}CloseTo(x, y, d, t, u) & \iff & CloseTo(x, y, d) \text{ in at least } u \text{ timesteps of interval } t\\ CloseTo(x, y, d) & \iff & \text{Distance between } x \text{ and } y \text{ is smaller than } d\\ TalksTo(x, y, t, u) & \iff & Talks(x, t, u),\ LooksAt(x, y, t', u'), \text{ and } t \text{ DURING } t'\\ Talks(x, t, u) & \iff & Talks(x) \text{ in at least } u \text{ timesteps of interval } t\\ LooksAt(x, y, t, u) & \iff & LooksAt(x, y) \text{ in at least } u \text{ timesteps of interval } t\\ t\ \text{ DURING }\ t' & \iff & t_{begin} > t'_{begin}\ \text{ and }\ t_{end} < t'_{end}\end{array}$$

5 First Experiments

We performed five group activities that also occur in crisis response control rooms: individual work, table meeting, presentation, coordinated interaction (see Section 4.2), and standing meeting. We recorded each activity seperately for three minutes, and we recorded two ten-minute sequences, containing all five activities (see Figure 3). There were three actors and four props involved: one director, two other staff members, a videowall, a table, and two posters representing individual workstations.

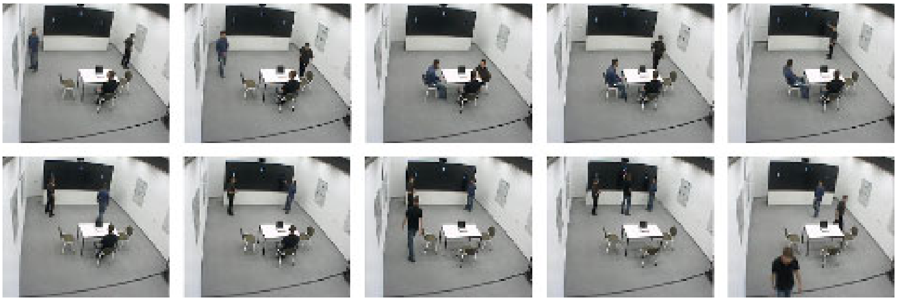


Fig. 3. Consecutively: individual work, transit 1, table meeting, transit 2, presentation, transit 3, coordinated interaction, transit 4, standing meeting, and transit 5

The actors were speaking and moving around freely and they pretended to interact with the videowall. Pointing activity however, was not used for the results presented in this paper. Also, we have only measured the head pose of the director for this experiment. Below are the five sentences generated upon detection, along with their truth conditions in simplified notation. Spatial, temporal, and logical composition for these activities is analogous to the example in Section 4.2. These rules would become more complex if we integrate gestures, head pose for all actors, and other perceptual components. Abbreviations are: dr = director, $s1$ = staff member 1, $s2$ = staff member 2, tb = table, vw = videowall, $p1$ = poster 1, and $p2$ = poster 2. Curly brackets signify disjunction.

- Director, S1, and S2 are doing individual work: $IndividualWork(dr, s1, s2)$ if: $CloseTo(dr, tb), CloseTo(s1, p1), CloseTo(s2, p2)$
- Director, S1, and S2 are in a table meeting: $TableMeeting(dr, s1, s2)$ if: $CloseTo(dr, tb), CloseTo(s1, tb), CloseTo(s2, tb), Talks(\{dr, s1, s2\})$
- S1 holds a presentation for Director and S2: $Presentation(s1, dr, s2)$ if: $CloseTo(s1, vw), Cl.To(dr, tb), Cl.To(s2, tb), Talks(s1), LooksAt(dr, s1)$
- S1 and S2 are interacting under Director's supervision: $Coord.Int.(s1, s2, dr)$ if: $CloseTo(s1, vw), CloseTo(s2, vw), CloseTo(dr, tb), TalksTo(dr, \{s1, s2\})$
- Director, S1, and S2 are in a standing meeting: $StandingMeeting(dr, s1, s2)$ if: $CloseTo(dr, vw), CloseTo(s1, vw), CloseTo(s2, vw), Talks(\{dr, s1, s2\})$

Classifying these five activities is not a hard problem. In fact, one can make the correct classifications using only tracking information. But to show the potential of the presented system, the conditions for each activity were purposefully made harder to fulfill. For example, we set the constraint that three people sitting at a table do not form a meeting yet if nobody is speaking. Under the listed truth conditions, and with empirically chosen parameter values for d, t, and u, the system achieves reasonable classification results. A perfect classification of the five isolated three-minute recordings would be achieved if *only* the correct activity is detected throughout the entire corresponding recording. Over all five three-minute recordings, corresponding to the five activities listed above, we achieve an average precision score of 0.744, and an average recall score of 0.738. Performance on the two mixed recordings is similar.

False positives and false negatives can have several causes. First, the recordings were not annotated. We had to assume as ground truth that each activity was performed non-stop, throughout the corresponding recording. Second, the output from the perceptual layer is not always flawless. And third, the threshold d for $CloseTo(x, y, d)$, was set to $2.5m$ to make the tracker less powerful as a classifier. This had the effect that actors were often close to two props simultaneously, making two activities true at the same time, and thus decreasing the precision score without a notable increase in recall. Also note that t and u were given high values so that atomic predicates had to be true for a considerable amount of time before their temporal counterparts become true. This had the desirable effect of an increase in precision without a notable decrease in recall. However, if t and u were set too high, search spaces became too large for real-time operation.

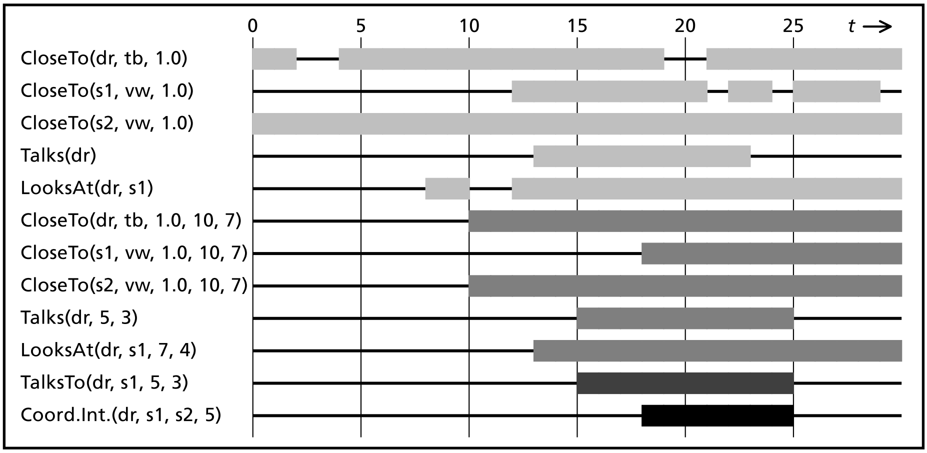


Fig. 4. A detailed view of the system's output. The 30 timesteps on the time axis correspond to three seconds of recorded perceptual information (see Figure 3 and Section 4.2). Abbreviations are: dr = director, $s1$ = staff member 1, $s2$ = staff member 2, tb = table, vw = videowall, $p1$ = poster 1, and $p2$ = poster 2

Figure 4 provides a detailed view of the system's output. The 30 timesteps that are displayed here, correspond to a three second interval from the coordinated interaction recording. The graph illustrates how the spatial, temporal, and logical composition from Section 4.2 reacts to a sequence of real perceptual information. Although we used an interval from the middle part of the recording, the system was started at $t = 0$, having no knowledge about what was perceived before. This is why none of the temporal predicates are true during the first ten timesteps in Figure 4. For the sake of clarity, we chose lower values for d, t, and u during the generation of this graph: $d = 1.0m$, $t = 10$ timesteps, and $u = 7$ timesteps for $CloseTo(x, y, d, t, u)$ and even smaller values for t and u in other predicates. For practical applications, behaviors that spread over longer time intervals are of course more interesting. Also note that $t = 10$ is a short way of saying that t is an interval containing ten timesteps.

6 Conclusion and Future Work

In this paper, we presented our progress towards a framework for high-level human activity recognition. Regardless of the sensor setup or application domain, the presented system fuses the outputs of several perceptual components, casts them into a temporal framework, and uses a logic engine containing context knowledge to deduce high-level facts, thus providing an abstract understanding of the given scene. Application domains include ambient intelligence, smart environments, robotics, surveillance, and videosearch. The novelty and contribution of our work lies in the fusion and interpretation of a large variety of real-time perceptual components into a single model. And in the use of a high-level temporal logic based approach as opposed to low-level statistical methods. More novelty comes from our current application domain: new tools for human-machine interaction and teamwork support in crisis response control rooms.

Future versions of the presented system will deal with more complex and detailed scenarios. Our focus in the future will lie on thorough empirical evaluations using videos with ground truth annotations and systematic evaluation metrics. We will benefit from the fact that the underlying perceptual components are constantly being improved and extended. For example, articulated body models and detailed knowledge of display activity will be available in the near future. We are also looking into the integration of more large screens, traditional workstations, table-displays, tablets, handhelds, and speakers. In parallel to these developments, the amount of possible facts and rules will increase. Also, we are envestigating the integration of alternative methods to obtain more subtle classifications, such as n-valued or fuzzy logic, possible worlds models, default logic, parameter evolution, and statistical learning. Finally, we will pursue the generation of natural language reports and 3D scene visualizations, as well as alternative application domains.

Acknowledgments. This work is supported by the FhG Internal Programs under Grant No. 692 026.

References

1. Waibel, A., Stiefelhagen, R. (eds.): Computers in the Human Interaction Loop. Springer, London (2010)
2. Nakashima, H., Aghajan, H., Augusto, J.C. (eds.): Handbook of Ambient Intelligence and Smart Environments. Springer, New York (2010)
3. Ivergard, T., Hunt, B.: Handbook of Control Room Design and Ergonomics: A Perspective for the Future, 2nd edn. CRC Press, London (2008)
4. Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine Recognition of Human Activities: A Survey. Circ. Syst. Vid. Techn. 18(11), 1473–1488 (2008)
5. Thiran, J.-P., Marques, F., Bourlard, H. (eds.): Multimodal Signal Processing, Theory and Applications for Human-Computer Interaction. Academic P., Oxford (2010)
6. Ryoo, M.S., Aggarwal, J.K.: Semantic Representation and Recognition of Continued and Recursive Human Activities. Int. Jour. of Computer Vision 82, 1–24 (2009)
7. Gu, T., Wu, Z., Tao, X., Pung, H.K., Lu, J.: epSICAR: An Emerging Patterns based Approach to Sequential, Interleaved and Concurrent Activity Recognition. In: 7th Conf. on Pervasive Computing and Communications. IEEE P., New York (2009)
8. Brdiczka, O., Langet, M., Maisonnasse, J., Crowley, J.L.: Detecting Human Behavior Models from Multimodal Observation in a Smart Home. IEEE T. Automation Science and Engineering 6(4), 588–597 (2009)
9. Gerber, R., Nagel, H.-H.: Representation of Occurrences for Road Vehicle Traffic. Artificial Intelligence 172, 351–391 (2008)
10. Gonzalez, J., Rowe, D., Varona, J., Xavier Roca, F.: Understanding dynamic scenes based on human sequence evaluation. Im. Vis. Comput. 27(10), 1433–1444 (2009)
11. Yao, B.Z., Yang, X., Lin, L., Lee, M.W., Zhu, S.-C.: I2T: Image Parsing to Text Description. Proceedings of the IEEE 99, 1–24 (2010)
12. Gupta, A., Srinivasan, P., Shi, J., Davis, L.S.: Understanding Videos, Constructing Plots; Learning a Visually Grounded Storyline Model from Annotated Videos. In: Conf. on Computer Vision and Pattern Recog., pp. 2004–2011. IEEE P., New York (2009)
13. Bernardin, K., Gehrig, T., Stiefelhagen, R.: Multi-Level Particle Filter Fusion of Features and Cues for Audio-Visual Person Tracking. In: Stiefelhagen, R., Bowers, R., Fiscus, J.G. (eds.) RT 2007 and CLEAR 2007. LNCS, vol. 4625, pp. 70–81. Springer, Heidelberg (2008)
14. Ekenel, H.K., Jin, Q., Fischer, M., Stiefelhagen, R.: ISL Person Identification Systems in the CLEAR 2007 Evaluations. In: Stiefelhagen, R., Bowers, R., Fiscus, J.G. (eds.) RT 2007 and CLEAR 2007. LNCS, vol. 4625, pp. 256–265. Springer, Heidelberg (2008)
15. Voit, M., Stiefelhagen, R.: Deducing the Visual Focus of Attention from Head Pose Estimation in Dynamic Multi-view Meeting Scenarios. In: 10th International Conference on Multimodal Interfaces, pp. 173–180. ACM Press, New York (2008)
16. Schick, A., van de Camp, F., Ijsselmuiden, J., Stiefelhagen, R.: Extending Touch: Towards Interaction with Large-Scale Surfaces. In: Interactive Tabletops and Surfaces 2009, pp. 127–134. ACM Press, New York (2009)
17. Soltau, H., Metze, F., Fugen, C., Waibel, A.: A One-pass Decoder Based on Polymorphic Linguistic Context Assignment. In: 2001 Automatic Speech Recognition and Understanding Workshop, pp. 214–217. IEEE Press, New York (2001)
18. Naik, R.: Blending the Logic Paradigm into C++ (2008), http://mpprogramming.com
19. Allen, J.F., Ferguson, G.: Actions and Events in Interval Temporal Logic. Journal of Logic and Computation 4(5), 531–579 (1994)

Towards Semantic Segmentation of Human Motion Sequences

Dirk Gehrig[1], Thorsten Stein[2], Andreas Fischer[2],
Hermann Schwameder[2], and Tanja Schultz[1]

[1] Institute for Anthropomatics,
[2] Institute for Sport and Sport Science,
Karlsruhe Institute of Technology, Germany
dirk.gehrig@kit.edu

Abstract. In robotics research is an increasing need for knowledge about human motions. However humans tend to perceive motion in terms of discrete motion primitives. Most systems use data-driven motion segmentation to retrieve motion primitives. Besides that the actual intention and context of the motion is not taken into account. In our work we propose a procedure for segmenting motions according to their functional goals, which allows a structuring and modeling of functional motion primitives[1]. The manual procedure is the first step towards an automatic functional motion representation. This procedure is useful for applications such as imitation learning and human motion recognition. We applied the proposed procedure on several motion sequences and built a motion recognition system based on manually segmented motion capture data. We got a motion primitive error rate of 0.9 % for the marker-based recognition. Consequently the proposed procedure yields motion primitives that are suitable for human motion recognition.

1 Introduction

In the field of robotics exists an increasing need for knowledge about human motions, as a humanoid robot has to be empowered with knowledge about motion sequences [1]. Given the continuous nature of motion, there is an unlimited number of motion sequences that can be performed. Therefore, it is impossible to enumerate a complete set of motion primitives. Boundaries of motion primitives are often arbitrarily defined, making it difficult to automate the motion segmentation process [2]. However, humans tend to perceive motion in terms of discrete motion primitives [3,4] and thus motion segmentation is still considered useful for some applications, including imitation learning [5] and human motion recognition [6].

Various different approaches can be found in the literature as to what should be seen as a motion primitive and to how these motion primitives can be modeled

[1] This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) within Collaborative Research Center 588 “Humanoid Robots - Learning and Cooperating Multimodal Robots”

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 436–443, 2010.

[5]. Various types of motion primitives are used ranging from low-level motions, e. g. moving the hand forward, up to complex motions such as setting the table. Currently, most approaches are data driven [8,9] and exhibit a gap between kinematic and functional motion representations. In many scenarios it is not sufficient to know the kinematic or dynamic parameters of a motion, since the goal of the motion might not be reached although the execution of the motion is correct. For example, if a robot wants to grasp a glas, it has to make sure that the glas is properly grasped. It is not sufficient, if the robot only performs a motion trajectory based on kinematic and dynamic parameters, which does not result in grasping the glas.

In this paper we start to bridge this gap by looking at the problem from top-down. We think that a system for decomposing motions into motion primitives should take the goals of a motion into account. It is not suitable to decompose a motion in an arbitrary way, since the goals of the motions have to be fulfilled to perform the motion properly. To the best of our knowledge there is no system for decomposing arbitrary daily-life motions into motion primitives based on functional information. In our work we propose a system which allows us to retrieve a motion decomposition into motion primitives based on functional knowledge. Relatively few papers have so far dealt with higher abstraction levels of human motions which touch the border of semantics. Some papers try to segment the data based on object relationships [10,11]. We do not use these appoaches since we also want to segment communicative gestures which are not object related. Another step in the direction of functional motion representation has been done by Guerra-Filho and Aloimonos [12]. They started to close the semantic gap between a WordNet and sensorimotor information by grounding a set of primitive words. Similar ideas have been presented by Ivanov et al. [13], whereas they assume a natural decomposition of motions into low-level primitives and higher-level semantic information. In our work we propose a systematic approach for a manual segmentation of motions into motion primitives based on the motion goals. The manual segmentation is a first step towards an automatic functional motion segmentation and will act as a baseline for the automatic segmentation.

2 A Functional Procedure to Identify Motion Primitives

In this section a heuristic procedure is introduced that enables a decomposition of voluntary motions into motion primitives. Before analyzing the motion sequence in detail, the motion context should be considered (e. g. which objects are in the environment and where are the objects located). The motion context needs to be defined since this information is necessary to constitute the solution space. After the solution space is issued, an analysis of the motion sequence itself has to be carried out. The individual elements of the motion are primitives, which carry a specific function according to the overall goal of the motion sequence. These primitives therefore are called *functional primitives*. We distinguish between *main functional primitives* and *supporting functional primitives*. Main functional primitives appear at least once during the motion and determine the goal of the motion. In contrast, supporting functional primitives are not directly related to the goal of the motion and functionally dependent on other functional primitives. Besides this functional relation there is a temporal relation (Fig. 1). Thereby, *preparatory supporting functional primitives* improve the situation for subsequent functional primitives. In contrast, *assistant supporting function primitives* improve the execution of concurrent functional primitives. Finally, *transitional supporting functional primitives* transform the present motion situation into a new situation [14]. The two axes of the diagram in Fig. 1 represent the functional and temporal relationships of the motion primitive. For a more precise representation of the dependencies lines and arrows are used. An arrow hereby specifies a functional relationship between two motion primitives while a line specifies only a temporal relationship.

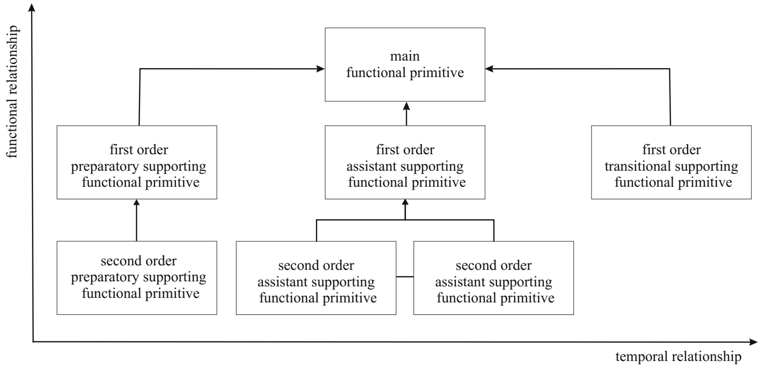

functional relationship
main functional primitive
first order preparatory supporting functional primitive
first order assistant supporting functional primitive
first order transitional supporting functional primitive
second order preparatory supporting functional primitive
second order assistant supporting functional primitive
second order assistant supporting functional primitive
temporal relationship

Fig. 1. Types and relationships of different functional primitives

There are three possibilities for the decomposition of a motion sequence into functional primitives. In the case of the *inductive functional structuring* perceivable motions of the performer are the starting point for the decomposition. These motions are performed because they fulfill certain functions in the context of the motion goal and therefore these motions lead to functional primitives. The origin of the *deductive functional structuring* are not motions but the motion goal and the motion context. Thereby the motion goal has to be decomposed into sub-goals and according actions of the performer have to be defined. Due to the phenomenon of motor equivalence [15] in biological motor control different motions can be defined that fulfill the same goal. A third possibility is a *combined functional structuring* which corresponds to a synthesis of the inductive and deductive structuring [14]. For each of the identified functional primitives *temporal* and *positional* constraints have to be examined. This has to be done at the beginning, during and at the end of each functional primitive, e. g. where the hands of the person must be, at the beginning, during and at the end of each functional primitive. It also has to be specified, whether the motion primitive has to be performed in a certain period of time. Finally, the segmentation of the motion into different functional primitives can be applied. The procedure does not guarantee that the decomposition always results in the same motion primitives and the same structure. The procedure is mainly a possibility to retrieve functionally plausible motion primitives, whereas the plausibility may depend on the desired application. The decomposition of a motion sequence results in different possible structures of motion primitives. Also the actual performed motion might be different for the same motion primitive. In other words the object positions and the used limb (e. g. left arm versus right arm) have to be taken into account for the performed motion. If for example an object is already at the desired position it can happen that no motion has to be performed for a motion primitive.

3 Application of the Procedure

In this section we applied the above introduced procedure to a daily-life motion in a kitchen, cutting an apple (see Fig. 2). The motion sequence is part of the scenario of the Collaborative Research Center (CRC) 588 - Humanoid Robots. The goal of the CRC 588 is the construction of humanoid service robots that share their activity space with human partners. For the application of our procedure we assume that a person is facing a table. The result of the decomposition strongly depends on the environmental conditions, e. g. present object. The objects involved in the task "cutting an apple" are an apple, a knife and a cutting board. At the beginning and the end of the motion sequence the objects are placed on the table.

We used a *deductive functional structuring* to decompose the motion sequence into motion primitives. Fig. 3 shows two structures of the task "cutting an apple". For simplicity reasons we assume that only one motion primitive is performed at a time. Multiple motion primitives performed at the same time will be addressed in future work. The dotted line represents an arbitrary number of repetitions of the motion primitive "cut apple". In our case the motion primitives have no temporal constraints besides the temporal structure shown in Fig. 3. At the

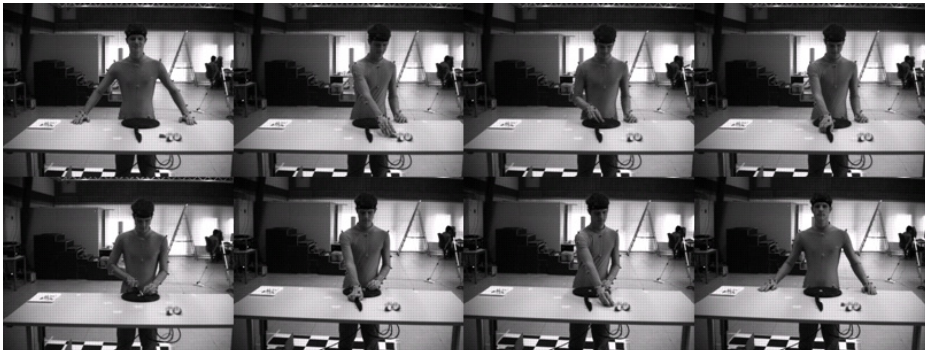


Fig. 2. Complex human motion sequence in a kitchen scenario: cutting an apple

beginning of the primitive we have no constraints induced by the motion context. The positional constraints during the motion primitives are: FP4: cut at apple, other hand at apple. Constraints at the end of motion primitives are: FP1: apple on top of cutting board, FP2: knife on top of apple, FP3: hand at apple, FP6: knife at original position, FP7: apple leftover at original position.

4 Evaluation of the Procedure

For the evaluation we applied the procedure to five tasks in total: *cutting an apple, pouring water, grating an apple, stiring, mashing.* The decomposition resulted in 24 motion primitives including the ones described in Sec. 3. For training purposes of a recognizer, a subject performed each task 20 times in a single session. For data acquisition each motion primitive was always done with the same hand. We manually segmented the 100 motion sequences based on the retrieved motion structure and build a motion recognition system. We evaluated, what recognition performance can be reached when using the extracted motion primitives. We tested the recognition system without using a motion grammar to guide the recognition process, when using a statistical bigram model and when using a motion grammar deduced from the motion structure. For the segmentation and the grammar of the task cutting an apple we used the upper motion structure in Fig. 3.

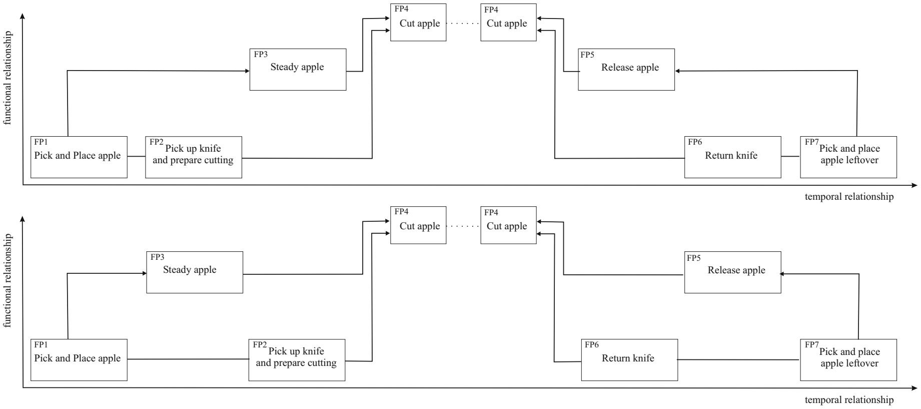


Fig. 3. Two functional and temporal structures of a complex human motion sequence in a kitchen scenario: cutting an apple

Due to the specific task, we only collected data from the persons upper body. The motion sequences were simultaneously recorded with a Vicon motion capture system and the camera system of a humanoid robot head (see Fig. 2). Only the marker-based data was used to build a recognition system and to evaluate our procedure. For the marker based motion capture 10 Vicon cameras were used to capture the motion of the subject with 20 fps. To capture the human motions 35 reflecting markers were attached to the subjects upper body, head and arms. The Vicon system output 3-dimensional positions and labels of the markers. Based on these marker information the related joint angle trajectories were calculated using a kinematic model [16]. As a result, the kinematic model outputs per time step one feature vector consisting of the 24 joint angles.

Our human motion recognition system features the one pass IBIS decoder [17], which is part of the Janus Recognition Toolkit JRTk [18]. We used this toolkit to recognize human motions based on joint angle velocities. The following paragraphs describe the components of our system, i. e. the input features, the model topology, the model initialization, training, and optimization, as well as the decoding strategy.

Feature vectors: The marker-based recognition system uses a 24-dimensional feature vector as input, consisting of 24 joint angle velocities from the upper body, which were calculated based on the joint angles resulting from the kinematic model. The input feature vectors are normalized by mean subtraction and normalizing the standard deviation to 1.

HMM models: Each motion primitive is statistically modeled with a left-to-right Hidden Markov Model (HMM). The number of states was optimized in cross-validation experiments as described below. Each state of the left-to-right HMM has two equally likely transitions, one to the current state, and one to the next state. The emission probabilities of the HMM states are modeled by Gaussian mixtures. The number of Gaussians per mixture was also optimized in the cross-validation experiments. A motion sequence was modeled as a sequential concatenation of these motion primitive models. In total, we discriminated 5 types of human motion sequences as mentioned above, consisting of the 24 different motion primitives.

Model initialization: To initialize the HMM models of the motion primitives, we manually segmented the data into the motion primitives. The manually segmented data were equally divided into one section per state, and a Neural Gas algorithm was applied to initialize the HMM-state emission probabilities.

Model Training: For HMM model training and development we used 10-fold cross-validation on the 100 motion sequences. For the experiments we varied the number of Gaussians between 1 and 64 and the number of states between 1 and 12. HMM training was performed featuring the Viterbi EM algorithm based on forced alignment on the unsegmented motion sequences.

Decoding: Decoding of the systems was carried out as a time-synchronous beam search. Large beams were applied to avoid pruning errors. We did three different types of decodings. First we did not use a motion grammar, whereat all transitions from one motion primitive to another are equally likely (1/24). In a second experiment we used a statistical bigram language model, where the probability of a motion primitive depends on the primitive before. As a third experiment we used a motion grammar deduced from the motion structure. Recognition performance is reported in terms of motion primitive error rate.

Results: When using 9 states for each motion primitive and 4 Gaussians for each state, we got a motion primitive error rate of 4.2 % without using a motion grammar. When using a simple automatically generated statistical bigram model we got an error rate of 2.3 % whereas when using the deduced grammar, we only got an error rate of 0.9 %. Figure 4 shows that the increase in recognition rate can be found in every of the five kitchen tasks.

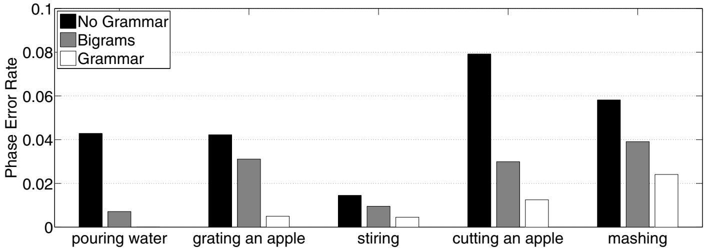


Fig. 4. Recognition error rates for 5 motion sequences

5 Conclusion and Future Work

We showed a way how to approach the task of functional motion decomposition. The proposed procedure is not a step by step instructions manual for segmenting human motions into motion primitives since there is no general way how to do that. It is still necessary to approach the segmentation of human motions systematically to achieve motion primitives, which are essential in the sense that the goals of the motions are represented in the motion primitives. If these motion primitives are used for motion generation or motion recognition, by fulfilling the goals of each motion primitive the achievement of the overall goal of the motion sequence is guaranteed to be reached. We applied the procedure on our motion capture data and performed human motion recognition based on the motion primitives and their temporal and functional structure. We got a motion primitive error rate of 0.9 % for marker-based recognition when using the motion grammar deduced from the motion structure. This shows, that the proposed procedure yields promising motion primitives and grammars. Nevertheless, this approach should be combined with automatic human motion segmentation approaches to automatically learn new motion primitives. The first step in the direction of automization will be the automatic deduction of motion grammars based on the proposed motion structuring. Besides, the effort to build such a motion structure will be reduced step by step through automization of the retrieval of the motion structure. In addition to the automization the possibility of multiple motion primitives performed at the same time will be addressed.

References

1. Schaal, S.: The New Robotics – towards human-centered machines. HFSP J. 1(2), 115–126 (2007)
2. Kahol, K.: Gesture Segmentation in Complex Motion Sequences. In: Proceedings IEEE International Conference on Image Processing, pp. 105–108 (2003)
3. Giese, M., Poggio, T.: Neural Mechanisms for the Recognition of Biological Movements. Nature Reviews 4, 179–192 (2003)
4. Rizzolatti, G., Fogassi, L., Gallese, V.: Neurophysiological Mechanisms Underlying the Understanding and Imitation of Action. Nature Reviews 2, 661–670 (2001)
5. Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: ICRA 2009: Proceedings of the 2009 IEEE International Conference on Robotics and Automation, pp. 1293–1298 (2009)
6. Gehrig, D., Kuehne, H., Woerner, A., Schultz, T.: HMM-based Human Motion Recognition with Optical Flow Data. In: 9th IEEE-RAS International Conference on Humanoid Robots, Humanoids (2009)
7. Krueger, N., Piater, J., Woergoetter, F., Geib, C., Petrick, R., Steedman, M., Ude, A., Asfour, T., Kraft, D., Omrcen, D., Hommel, B., Agostino, A., Kragic, D., Eklundh, J., Kruger, V., Dillmann, R.: A Formal Definition of Object Action Complexes and Examples at different Levels of the Process Hierarchy (2009)
8. Barbic, J., Safonova, A., Pan, J.-Y., Faloutsos, C., Hodgins, J.K., Pollard, N.S.: Segmenting Motion Capture Data into Distinct Behaviors. In: Graphics Interface, pp. 185–194 (2004)
9. Reng, L., Moeslund, T.B., Granum, E.: Finding Motion Primitives in Human Body Gestures. In: Gibet, S., Courty, N., Kamp, J.-F. (eds.) GW 2005. LNCS (LNAI), vol. 3881, pp. 133–144. Springer, Heidelberg (2006)
10. Aksoy, E.E., Abramov, A., Woergoetter, F., Dellen, B.: Categorizing Object-Action Relations from Semantic Scene Graphs. In: IEEE International Conference on Robotics and Automation, pp. 398–405 (2010)
11. Sridhar, M., Cohn, G.A., Hogg, D.: Learning functional object categories from a relational spatio-temporal representation. In: 18th European Conference on Artificial Intelligence (2008)
12. Guerra-filho, G., Aloimonos, Y.: Towards a sensorimotor WordNet SM: Closing the semantic gap. In: Proc. of the International WordNet Conference, GWC (2006)
13. Ivanov, Y.A., Bobick, A.F.: Recognition of Visual Activities and Interactions by Stochastic Parsing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 852–872 (2000)
14. Goehner, U.: Einfuehrung in die Bewegungslehre des Sports, Teil 1: Die sportlichen Bewegungen (Introduction to human movement science, part 1: sports movements). Hofmann, Schorndorf (1992)
15. Kelso, J.A.S., Fuchs, A., Lancaster, R., Holroyd, T., Cheyne, D., Weinberg, H.: Dynamic cortical activity in the human brain reveals motor equivalence. Nature 392, 814–818 (1998)
16. Simonidis, C., Seemann, W.: MkdTools - human models with Matlab. In: Wassink, R. (ed.) The 10th International Symposium on 3D Analysis of Human Movement - Fusion Works (2008)
17. Soltau, H., Metze, F., Fügen, C., Waibel, A.: A one-pass decoder based on polymorphic linguistic context assignment. In: ASRU, pp. 214–217 (2001)
18. Finke, M., Geutner, P., Hild, H., Kemp, T., Ries, K., Westphal, M.: The Karlsruhe-Verbmobil speech recognition engine. ICASSP 1, 83–86 (1997)

Author Index