Karlsruher Schriften
zur Anthropomatik

Band 32

David Münch

Begriffliche Situationsanalyse aus Videodaten bei unvollständiger und fehlerhafter Information

KIT Scientific Publishing

David Münch

Begriffliche Situationsanalyse aus Videodaten bei unvollständiger und fehlerhafter Information

Karlsruher Schriften zur Anthropomatik
Band 32
Herausgeber: Prof. Dr.-Ing. Jürgen Beyerer

Eine Übersicht aller bisher in dieser Schriftenreihe erschienenen Bände finden Sie am Ende des Buchs.

Begriffliche Situationsanalyse aus Videodaten bei unvollständiger und fehlerhafter Information

von
David Münch

KIT Scientific Publishing

Dissertation, Karlsruher Institut für Technologie
KIT-Fakultät für Informatik, 2017

Impressum

Scientific Publishing

Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark of Karlsruhe Institute of Technology. Reprint using the book cover is not allowed.

www.ksp.kit.edu

This document – excluding the cover, pictures and graphs – is licensed under the Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0): https://creativecommons.org/licenses/by-sa/4.0/deed.en

The cover page is licensed under the Creative Commons Attribution-No Derivatives 4.0 International License (CC BY-ND 4.0): https://creativecommons.org/licenses/by-nd/4.0/deed.en

Print on Demand 2017 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 1863-6489
ISBN 978-3-7315-0644-7
DOI 10.5445/KSP/1000066975

Begriffliche Situationsanalyse aus Videodaten bei unvollständiger und fehlerhafter Information

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

David Münch

Aus Freudenstadt

Tag der mündlichen Prüfung:	31. Januar 2017
Erster Gutachter:	Prof. Dr.-Ing. Rainer Stiefelhagen
Zweiter Gutachter:	Prof. Bernd Neumann, Ph.D.

Kurzfassung

In den letzten Jahren hat durch die zunehmende Präsenz visueller Sensoren die Menge an Bild- und Videodaten enorm zugenommen. Damit geht auch ein steigender Bedarf an einer automatischen semantischen Auswertung von Bildfolgen einher. Als Beispiele einer solchen semantischen Auswertung sei die schritthaltende Erkennung komplexer Situationen in Bildfolgen oder die nachträgliche Suche nach bestimmten Situationen in Bildfolgen im Sinne der Videoforensik genannt.

Das automatische Erkennen von Situationen in Videodaten ist ein in vielen Teilen noch nicht gelöstes Problem. Einerseits gibt es direkte, meist lernende, Verfahren, die mit massiv vielen Trainingsdaten versuchen, einen Bezug zwischen den Eingabedaten, z.B. bestimmten Bildmerkmalen, und den zu erkennenden Situationen herzustellen. Eine andere Möglichkeit besteht darin, die Situationserkennung hierarchisch und modellierend anzugehen. Vertreter dieser Verfahren, die zeitliche, räumliche und logische Eigenschaften oft in einer geeigneten höheren Logik repräsentieren, sind die in dieser Arbeit verwendeten Schlussfolgerungsmethoden basierend auf der unscharfen metrisch-temporalen Logik eingeschränkt auf das Hornfragment (FMTHL) und Situationsgraphenbäume (SGTs). Diese Methoden erlauben eine generische Modellierung auch sehr komplexer Situationen und die Wiederverwendung des repräsentierten Wissens. Allerdings weisen sie nur geringe Robustheit gegen Variationen des Auftretens von Situationen auf und reagieren meist auch empfindlich auf fehlerhafte und unsichere Informationen.

In dieser Arbeit werden FMTHL und SGTs auf die Erkennung komplexer Situationen in Bildfolgen im Videoüberwachungskontext angewendet. Es wird die Frage beantwortet, wie die Robustheit der Erkennung komplexer Situationen in natürlichen Umgebungen aufrechterhalten werden kann, in denen fehlerbehaftete, unvollständige und verrauschte Daten verarbeitet werden.

Der Formalismus der begrifflichen, auf FMTHL und SGTs basierenden Situationserkennung, wurde um die Behandlung von Unschärfe und Ableitbarkeit aller zutreffenden Hypothesen [Münch et al., 2011a] erweitert. Diese konzeptionelle Erweiterung ermöglicht es, die Eingabedaten entsprechend der Realität abzubilden und durch den gesamten Inferenzprozess die Unsicherheit zu propagieren. Des Weiteren haben die untersuchten Diskursbereiche gezeigt, dass sich ein beobachteter Agent gleichzeitig in mehr als einer Situation befinden kann. Daher erlaubt die Ableitung aller zu diesem Zeitpunkt zutreffenden Situationen die erschöpfende Beschreibung der beobachteten Szene.

Im Vergleich zu perfekten (künstlich erzeugten) Daten gibt es bei realen Daten Effekte, welche die Robustheit der nachfolgenden Verarbeitungsschritte erheblich schwächen. Auf der Ebene der Eingabedaten kann Rauschen und zum Teil eine gewisse Unvollständigkeit durch Vorverarbeitungsfilter [Münch et al., 2012a] reduziert werden. Wohingegen auf semantischer Ebene Lücken von Beobachtungen dazu führen können, dass einzelne Situationen nicht ausgeprägt und die zeitlich nachfolgenden Situationen somit niemals erreicht werden können. Das Konzept der kontrollierten Halluzination [Münch et al., 2012a] wurde entwickelt, um auf semantischer Ebene diese Unvollständigkeiten zu kompensieren.

Der Diskursbereich Videoüberwachung umfasst Situationen mit vielen Agenten und deren Beziehungen untereinander. Die bisherige Modellierung von Relationen stößt dabei an ihre Leistungsgrenze. Um mit dieser Komplexität umgehen zu können, wurde der Schlussfolgerungsprozess um eine gruppenzentrierte Inferenz [Münch et al., 2012b, IJsselmuiden

et al., 2014] erweitert. Modellierbare Abhängigkeiten sind jetzt nicht mehr binär, sondern können beliebig auf Mengen definiert werden. Semantisch ähnliche Eingabeinformationen können zusammengefasst werden, um die Handhabbarkeit in realen Systemen zu erhöhen.

Die in dieser Arbeit untersuchten Szenarien gehören dem Diskursbereich der Videoüberwachung im Innen- und Außenbereich an. Neben der Nutzung öffentlich verfügbarer Datensätze wurde auch ein eigener Datensatz geschaffen, um die einzelnen methodischen Erweiterungen systematisch mit künstlichen und realen Daten auszuwerten. Mit jeder einzelnen Erweiterung wird eine Steigerung der Erkennungsleistung von Situationen erlangt. Mit einem aktiven Kamerasystem [Münch et al., 2013b] kann die Situationserkennung Einfluss auf die Eingabedaten nehmen. Die Auswertung umfasst das in SGTs neu modellierte Hintergrundwissen der erwarteten Situationen und die in FMTHL zugrunde liegende Basistaxonomie.

Die generische Anwendbarkeit der auf FMTHL und SGTs basierenden Situationserkennung wird gezeigt, indem dieser Ansatz erstmalig auf die automatische Erkennung von Aktivitäten des täglichen Lebens in intelligenten Umgebungen eingesetzt wurde. In diesem Fall existieren keine bildbasierten Eingabedaten, sondern lediglich binäre Sensoren, die Zustände der Umgebung, wie z.B. Licht an oder Licht aus, erfassen. Eine Übertragung von Wissen aus dem Videoüberwachungskontext ist in Teilen möglich; die Schnittstelle zu den Sensoren muss angepasst werden und neue bisher nicht vorhandene Situationen müssen neu modelliert werden. Die Übertragbarkeit von Wissen innerhalb des Diskursbereichs ist bis auf Schnittstellenanpassungen möglich. Lernende Verfahren werden hier eingesetzt, um Verhalten personenspezifisch zu identifizieren und um schließlich die Modellierung von Hintergrundwissen zu unterstützen.

Insgesamt wurde gezeigt wie die Robustheit der Situationserkennung bei natürlichen Szenarien trotz der dort real auftretenden Komplexität und der fehlerbehafteten, unvollständigen und verrauschten Informationen aufrechterhalten werden kann.

Abstract

In recent years the rising number of surveillance cameras has caused a vast amount of video data. As a consequence, the demand for tools to automatically analyze video data on a semantic level is increasing. An example for semantically analyzing video data is the recognition of complex situations or searching preprocessed videos for certain situations.

Recognizing situations in video data is still an unsolved problem. On the one hand, there are direct approaches – mostly learning-based methods – that correlate the input data with the help of massive training data directly to recognized situations. On the other hand, there are hierarchical approaches: They consist of several layers where spatio-temporal and logical relations are encoded on distinct abstraction levels. These approaches are often built on higher-order logics. In this thesis situation recognition is based on fuzzy metric temporal horn logic (FMTHL) and situation graph trees (SGTs). Using these methods, it is possible to model complex situations generically and reuse existing knowledge. However, these methods react sensitively when dealing with variations of situations and real noisy input data.

In this thesis FMTHL and SGTs are applied for complex situation recognition in video surveillance applications. The following question is answered: How to maintain the robustness of recognizing complex situations in natural environments dealing with erroneous, missing, and noisy data?

The formalism of FMTHL and SGTs for recognizing situations conceptually was extended to deal with noise and the instantiation of all possible hypotheses [Münch et al., 2011a]. This extension allows the propagation

of the uncertainty throughout the inference process. Additionally, the different domains show that an agent can be the subject of several different situations at the same time. Thus, an extended situation recognition algorithm allows an exhaustive situation recognition of the current scene.

While using real data – in contrast to perfect (artificially generated) data – there occur effects which weaken subsequent processing steps substantially. On the level of input data noisy and incomplete data can be handled with preprocessing filters [Münch et al., 2012a]. On the semantic level missing evidence prevents situations from instantiation and subsequent situations cannot be reached. Controlled hallucination [Münch et al., 2012a] is introduced to deal with missing evidence on the semantic level.

In the domain of video surveillance there occur situations with many agents which are related to each other. Former modeling techniques are limited to binary relations. In this work knowledge representation is extended by set-based relations [Münch et al., 2012b, IJsselmuiden et al., 2014]. Semantically similar input data can be clustered to reduce complexity, too.

The scenarios addressed in this work stem from video surveillance in- and outdoors. Besides using public available datasets, an own dataset was created to evaluate all the methodical improvements on artificial and real data.

The generical applicability is shown while applying FMTHL and SGTs to activities of daily living. In that domain there does not exist visual input data, instead binary sensors such as a light switch. Portability of knowledge from the video surveillance domain is partly possible. Within the domain – unless interface adaptions – portability of knowledge is easily possible. Learning based methods are applied to model person-centric situations and support the generation of background knowledge.

In total it is shown that situation recognition is maintained robust, even in natural environments with their complexity and erroneous, missing, and noisy data.

Danksagung

Diese Arbeit ist während meiner Tätigkeit in der Abteilung Objekterkennung am Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung in Ettlingen entstanden. Ich war in der glücklichen Lage von vielen herausragenden Personen Unterstützung in vielfältiger Art und Weise zu erfahren.

An erster Stelle gilt mein Dank Dr. Michael Arens. Er hat mich in vielen langen Diskussionen immer wieder neu für mein Thema begeistert, die richtigen Fragen gestellt und Rahmenbedingungen geschaffen, die Freiheiten und zielgerichtetes Arbeiten möglich machen. Mein Teamleiter Dr. Wolfgang Hübner hat zur richtigen Zeit Feedback und Freiräume zur Erstellung dieser Arbeit gegeben.

Mein Dank gilt Prof. Dr.-Ing. Rainer Stiefelhagen für die investierte Zeit, die detaillierte und konstruktive Kritik, die Betreuung als Externer und das erste Gutachten. Herrn Prof. Dr. Bernd Neumann danke ich für die Diskussionen, Anregungen und die Übernahme des Korreferats.

Danken möchte ich auch Prof. Dr.-Ing. Jürgen Beyerer, Prof. Dr. Walter Tichy und Prof. Dr.-Ing. Tamim Asfour für das Interesse an und dem Feedback zu dieser Arbeit.

Mit Dr.-Ing. Joris IJsselmuiden wurde immer ein reger Austausch von Ideen, Werkzeugen und Inhalten gepflegt. Viele Kolleginnen und Kollegen haben mir bei der Ausarbeitung dieser Arbeit immer wieder Korrekturen und Verbesserungsvorschläge unterbreitet. Besonderer Dank an Dr.-Ing. Bastian Ibach und Dr.-Ing. Eckart Michaelsen.

Viele Kolleginnen und Kollegen haben mir die Zeit am Institut zu einer guten Zeit gemacht. Seien es Diskussionen, Arbeiten oder Aktivitäten abseits der Arbeit. Im Besonderen sind das meine Zimmerkolleginnen und -kollegen Ann-Kristin Grosselfinger, Stefan Becker, Hilke Kieritz und Ronny Hug.

Ebenso danke ich meinen ehemaligen Studentinnen und Studenten Ann-Kristin Grosselfinger, Tobias Zepf, Sebastian Bauer, Camilo Ramirez, Andrew Katumba, Alexander Meier und Martin Metzger.

Ettlingen, im Januar 2017 *David Münch*

Inhaltsverzeichnis

Akronyme

ADL	Aktivitäten des täglichen Lebens
BRISK	Binary Robust Invariant Scalable Keypoints
CNN	Convolutional Neural Network
CVS	Cognitive Vision System
DBN	Dynamisches Bayes'sches Netz
FHL	unscharfe Hornlogik
FL1	unscharfe Logik erster Ordnung
FMTHL	unscharfe metrisch-temporale Logik eingeschränkt auf das Hornfragment
LSTM	Long-Short-Term-Memory-Network
MLN	Markov-Logic-Network
MTHL	metrisch-temporale Hornlogik
MTL	metrisch-temporale Logik
OWL	Web-Ontology-Language
PL1	Prädikatenlogik erster Ordnung
SGT	Situationsgraphenbaum
SURF	Speeded Up Robust Features
SWRL	Semantic-Web-Rule-Language

Symbolverzeichnis

$\widetilde{\forall}\mathcal{F}$	Allabschluss $\widetilde{\forall}\mathcal{F} \equiv \forall v_1 \ldots \forall v_n \mathcal{F}$ mit $\{v_i \vert 1 \leq i \leq n\}$ Menge aller freien Variablen in $\mathcal{F}$.
$\widetilde{\exists}\mathcal{F}$	Existenzabschluss $\widetilde{\exists}\mathcal{F} \equiv \exists v_1 \ldots \exists v_n \mathcal{F}$ mit $\{v_i \vert 1 \leq i \leq n\}$ Menge aller freien Variablen in $\mathcal{F}$.
$\mathcal{F}_\Sigma(\mathcal{V})$	Die Menge aller Formeln über der Signatur Σ zur Menge $\mathcal{V}$ von Variablen.
$\Box$	temporaler Alloperator.
μ_A	Zugehörigkeitsfunktion der unscharfen Menge $\widetilde{A}$.
$\widetilde{\mathcal{P}}(\mathcal{U})$	Unscharfe Potenzmenge von $\mathcal{U}$: Menge aller unscharfen Mengen über $\mathcal{U}$.
$\widetilde{R}$	unscharfe Relation.

1 Einleitung

Das inhaltliche Verstehen von Bildfolgen kann durch Verknüpfen von Methoden der Bildverarbeitung mit logischer Inferenz geleistet werden. In dieser Arbeit wird eine kognitive Architektur zur Erkennung von Situationen in Überwachungsszenarien ausgearbeitet. Diese Architektur basiert auf einem kognitiven System und wurde konkret realisiert.

Mögliche Anwendungsbereiche der videobasierten Situationsanalyse werden in Abschnitt 1.1 vorgestellt. Im darauf folgenden Abschnitt 1.2 wird die Zielsetzung dieser Arbeit dargestellt. In Abschnitt 1.3 wird das kognitive System vorgestellt und dessen Anwendbarkeit motiviert, in Abschnitt 1.4 auf die eigenen Beiträge dieser Arbeit eingegangen und anschließend die Gliederung der vorliegenden Arbeit vorgestellt.

1.1 Anwendungsbereiche

Im Folgenden wird auf die Anwendungsbereiche für die Situationsanlyse im Einzelnen eingegangen.

Videoüberwachung im Innen- und Außenbereich ist der im Wesentlichen dieser Arbeit zugrundeliegende Diskursbereich. Die Anzahl an Videokameras in öffentlichen Gebäuden und im Außenraum steigt rapide an; begründet und legitimiert durch stetig wachsende Terrorangst in der westlichen Welt[1,2]. Beispielsweise schätzt man die Zahl an Videoüber-

[1] http://www.sueddeutsche.de/muenchen/videoueberwachung-in-muenchen-stadt-der-augen-1.2316618 (19.03.2017)

[2] Die Neufassung von §15a PolG NRW.

wachungskameras im Vereinigten Königreich auf 4 - 5.9 Millionen Stück[3]. Man kann davon ausgehen, dass diese Kameras nicht nur zur Abschreckung existieren und dass somit ein großer Bedarf an automatischer Situationsanalyse in Videodaten besteht, da menschliches Personal zur manuellen Überwachung der Videokameras die Situationsanalyse in diesem Umfang nicht leisten kann. Ein System zur automatischen Situationsanalyse würde nicht nur die beobachteten bedrohlichen Verhalten erkennen, sondern diese im Idealfall auch prädizieren und dem Überwachungspersonal melden. In gleicher Weise kann die Situationsanalyse auch immer dann eingesetzt werden, wenn strukturierte Verhalten analysiert werden sollen. Dies ist im öffentlichen Straßenverkehr meist der Fall. So kann die Situationsanalyse zur Überwachung von Straßenkreuzungen [Arens et al., 2008] eingesetzt werden, um dann aus der gesteuerten Kreuzung eine geregelte zu machen, um z.B. den Verkehrsfluss entsprechend dem Verkehrsaufkommen zu optimieren. Auch ist ein Einsatz bei Verhalten möglich, die durch ein Protokoll definiert sind, wie zum Beispiel bei der Fahrzeugmontage [Voulodimos et al., 2011] oder Sicherheits- und Wartungstätigkeiten. In [Münch et al., 2016] wird ein System vorgestellt, das die Wartung und Abweichungen davon in einem Serverraum sowie bei der Sicherheitspatrouille in einem Fußballstadion analysiert und bei Bedarf Alarm schlägt.

Technisch ist es möglich, Wohnungen oder spezielle Räume wie z.B. Kontrollräume mit vielen Sensoren auszustatten. Allein dadurch wird aber noch lange keine **Intelligente Umgebung** geschaffen. Dazu bedarf es der Situationsanalyse, welche die mit den Sensoren aufgezeichneten Daten auswertet und in Zusammenhang setzt. Ein Ziel ist dabei, die Situationen, in denen sich der Agent – hier die Umgebung – befindet, zu erkennen und dann Aktionen auszulösen, die den persönlichen Komfort der Bewohner erhöhen. Eine weitere Anwendungsmöglichkeit ist die Überwachung älterer Personen, um im Notfall automatisch Unterstützung anzufordern.

[3] http://www.cctv.co.uk/how-many-cctv-cameras-are-there-in-the-uk/ (19.03.2017)

Andere Möglichkeiten ergeben eine detaillierte textuelle Protokollierung der tatsächlich durchgeführten Handlungen, wie beispielsweise in einem Kontrollraum [IJsselmuiden, 2014].

Bei der **Suche in Bildfolgen im Sinne der Videoforensik** spielt die Situationsanalyse eine große Rolle. Unter der Annahme, dass die bildverarbeitenden Verfahren immer besser und leistungsfähiger werden, können beliebige Videosequenzen prozessiert und das darin Beobachtete zur Situationsanalyse verwendet werden, um dann die Ergebnisse in komprimierter – beispielsweise textueller oder ausgezeichneter Sprache – abzuspeichern. Dies würde es dann ermöglichen, im Nachhinein nach bestimmten Situationen in Videos effizient zu suchen. Die Anwendung beschränkt sich natürlich nicht nur auf Videodaten, sondern kann grundsätzlich auch auf anderen Daten arbeiten, sofern diese geeignet aufbereitet sind. Eine Möglichkeit, Information mit Bedeutung zu versehen, demonstriert beispielsweise das Semantic Web.

In der Entwicklung von Kraftfahrzeugen spielen drei Aspekte eine entscheidende Rolle: Die Steigerung der Sicherheit, des Komforts und der Ökonomie. Dazu kommen vermehrt **Fahrerassistenzsysteme** (ADAS) zum Einsatz. Im Gegensatz zu passiven Systemen unterstützen ADAS den Fahrer aktiv. Das kann durch aktives Eingreifen in die Steuerung – wie eine Verminderung des Bremsdrucks beim Antiblockiersystem – oder nur durch einfache Warnhinweise wie bei der Einparkhilfe geschehen. Komplexere ADAS wie ein aktiver Spurhalteassistent erfordern die Erfassung der Umwelt [Geiger, 2013], um dann darin aktiv und selbständig zu agieren. Mit einer umfassenden Situationsanalyse der Umwelt sind weitere und robustere Vorhersagen der Verhalten anderer Verkehrsteilnehmer möglich.

Im Sport wird z.B. beim Fußball und Basketball das Verhalten der einzelnen Spieler und der gesamten Mannschaft analysiert. Ohne technische Hilfsmittel ist dies eine herausfordernde Aufgabe. Eine erste Erleichterung bei der **Automatischen Spielanalyse** ist die Unterstützung von Kameras und bildverarbeitenden Verfahren, die die Position und Identität der einzel-

nen Objekte über der Zeit festhalten, sowie ein intelligenter Spielball. Weitere Intelligenz kann durch eine umfassende Situationsanalyse hinzugefügt werden. Damit wäre es sogar denkbar, einen automatischen Kommentator zu erschaffen, um die Moderatoren zu ersetzen.

Service- und Industrieroboter unterscheiden sich zu den bisher genannten Anwendungsbereichen dadurch, dass ihr Fokus auf der Interaktion mit der Umwelt liegt. Ihre Hauptanwendung ist z.B. das Montieren von Teilen, das Reinigen des Fußbodens oder die Durchführung von Tätigkeiten in der Küche. Eine Situationsanalyse kann ihnen helfen, ihre eigentlichen Aufgaben zuverlässiger durchzuführen, indem die Umwelt umfassender wahrgenommen wird und auf erkannte Störeinflüsse reagiert werden kann.

1.2 Zielsetzung

Nach [Nagel, 1988] wird ein Verhalten durch eine Schablone der „generisch beschriebenen Situation“ dargestellt. Konkrete Ausprägungen davon sind **Situationen**. „Ein Agent befindet sich demnach in einer bestimmten Situation, wenn er selbst und seine Umwelt in einem bestimmten Zustand sind und der Agent bestimmte Handlungsoptionen besitzt“ [Arens, 2004]. Die **Situationsanalyse** ist das Suchen der detailliertesten Situationen für einen Agenten zu allen Zeitpunkten.

Das automatische Erkennen von Situationen in Videodaten ist ein in vielen Teilen noch nicht gelöstes Problem. Einerseits gibt es direkte, meist lernende, Verfahren, die mit massiv vielen Trainingsdaten versuchen, einen Bezug zwischen den Eingabedaten, z.B. bestimmten Bildmerkmalen, und den zu erkennenden Situationen herzustellen. Probleme treten auf, wenn nicht ausreichend Trainingsdaten zur Verfügung stehen. Eine andere Möglichkeit besteht darin, die Situationserkennung hierarchisch und modellierend anzugehen. Ein Vertreter dieser Verfahren, die zeitliche, räumliche und logische Eigenschaften oft in einer geeigneten höheren Logik

repräsentieren, ist eine technische Realisierung des in Abschnitt 1.3 vorgestellten Cognitive Vision System (CVS) unter anderem durch Schlussfolgerungsmethoden basierend auf unscharfe metrisch-temporale Logik eingeschränkt auf das Hornfragment (FMTHL) und SGTs. Diese Methoden erlauben eine generische Modellierung auch sehr komplexer Situationen und die Wiederverwendung des repräsentierten Wissens. Allerdings weisen sie nur geringe Robustheit gegen Variationen im Auftreten von Situationen auf und reagieren meist auch empfindlich auf fehlerhafte und unsichere Informationen. In dieser Arbeit werden FMTHL und SGTs auf die Erkennung komplexer Situationen in Bildfolgen im Videoüberwachungskontext angewandt.

Dabei wird die Frage beantwortet, wie die Robustheit der Erkennung von komplexen Situationen in natürlichen Umgebungen aufrechterhalten werden kann, in denen fehlerbehaftete, unvollständige und verrauschte Daten verarbeitet werden.

1.3 Ein kognitives System

Ein kognitives[4] System kann seine Umgebung wahrnehmen und durch Denken, Erinnern und Lernen verstehen und dadurch Information umgestalten. Dies kann sich auf die begriffliche Beschreibung der Umgebung beziehen, aber auch ein aktives Eingreifen in die Szene bedeuten. Die visuelle Wahrnehmung ist ein ausgezeichneter Sinn beim Menschen. Ein kognitives Sichtsystem trägt diesem Umstand Rechnung und stellt das visuelle System in den Vordergrund – auch wenn andere Wahrnehmungsmodalitäten zusätzlich möglich sind. Die konkrete technische Umsetzung eines kognitiven Systems erfordert die Abbildung der einzelnen kognitiven Fähigkeiten. Bei der Umgestaltung der Information der Bilddaten einer

[4] lat. cognoscere: erkennen, erfahren, kennenlernen. „The mental action or process of acquiring knowledge and understanding through thought, experience, and the senses." (Oxford Dictionary)

beobachteten Szene zu Information in Form einer begrifflichen Beschreibung werden die Eingaben auf das Wesentliche reduziert. D.h. im Verlauf der Umgestaltung entspricht die Information einem immer höheren Abstraktionsgrad. Bedingt durch die unterschiedlichen Fähigkeiten, die unterschiedlich komplex sind, entsteht ein kognitives System als ein hierarchisches Modell bestehend aus Schichten entsprechend ihrem Abstraktionsgrad. Auch beim menschliche Gehirn geht man davon aus, dass es funktional hierarchisch organisiert ist.

Eine mögliche konzeptionelle Umsetzung ist das Cognitive Vision System (CVS) aus [Nagel, 2000, 2004]. Es wurde initial für das Erkennen von Verhalten bei der videobasierten Verkehrsüberwachung eingesetzt. Bereits in diesen Arbeiten ist der Informationsfluss bidirektional: Es wird nicht nur die Information einer beobachteten Szene in die Beschreibung der dort stattfindenden Geschehen transformiert, sondern aus den Geschehen wandert Information zurück in das Bewegungsmodell der Objekte der beobachteten Szene. Weitere Funktionalität wurde in [Arens et al., 2008, Nagel, 2010] durch eine mächtige Wissensrepräsentation und eine natürlichsprachliche Komponente hinzugefügt. Abbildung 1.1 gibt einen Überblick über das CVS. Die Architektur ist in drei große Schichten untergliedert.

Das *interaktive Teilsystem* (IS) ist die Schnittstelle zur Welt. Es beinhaltet die *Sensor-Aktuator-Ebene* (SAL). Auf dieser Ebene liegt Information als Bild oder Sprache vor und wird durch Sensoren wie beispielsweise Kameras erfasst. Ein Display zur Ausgabe visueller Information ist genauso ein Aktuator, wie eine aktiv steuerbare Kamera. Der Speicher auf dieser Ebene ist das sensorische Gedächtnis.

Im *quantitativen Teilsystem* (QL) können unterschiedliche anwendungsspezifische Teilsysteme eingefügt werden, wie z.B. das *visuelle Teilsystem* (VS), das visuelle Information umgestalten kann, das *natürlichsprachliche Teilsystem* (NS) oder das *informationsbasierte Teilsystem* (IS). Das visuelle Teilsystem gestaltet Information um, und zwar von den bildgebenden Sensoren bis zu einer dreidimensionalen Szene. In die andere Richtung soll

begriffliche Szeneninformation auf z.B. Displays visualisiert werden. Beim natürlichsprachlichen Teilsystem werden Audioinformationen über eine textuelle Form in Sprache und Begriffe umgestaltet. In die andere Richtung sollten begriffliche Szeneninformationen durch Text oder Sprache ausgegeben werden. Das informationsbasierte Teilsystem kümmert sich um die Umgestaltung von Information aus oder nach beispielsweise digitalen Datenbanken oder Archiven. Die Wahrnehmung digitaler Information ist dem Menschen direkt nicht möglich, wohl aber einem technischen System.

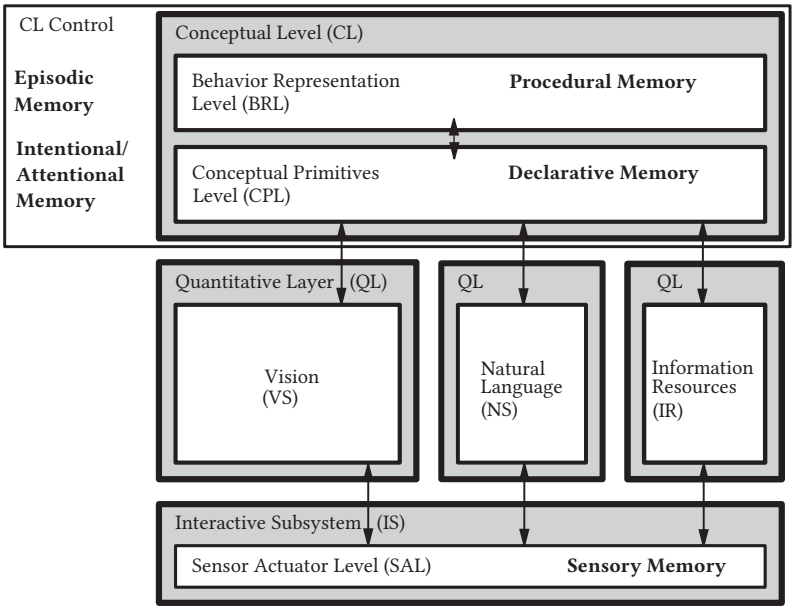


Abbildung 1.1: Überblick über das Cognitive Vision System basierend auf [Nagel, 2000, 2004]. Siehe Abschnitt 1.3 für eine detaillierte Beschreibung.

Im *begrifflichen Teilsystem* (CL) wird die bisher quantitativ vorliegende Information auf und durch Begriffe abgebildet. Beide Repräsentationen, die quantitative aus dem QL und die Begriffe aus dem CL, sind in ihrer formalen Natur unterschiedlich. Somit ist eine Abbildung immer mit einem

bedeutungsbezogenen Unterschied behaftet: der Semantischen Lücke [Smeulders et al., 2000]. Um die semantische Lücke zu schließen ist Expertenwissen notwendig um in der *Begrifflich-Primitiven-Ebene* (CPL) die Abbildung von quantitativer Information in Begriffe zu definieren. Im CPL wird Information in Form von Fakten gespeichert und bearbeitet. Im Speicher, dem deklarativen Gedächtnis, liegen grundlegende Regeln, die das Basiswissen einer Welt beschreiben. Mit Fakten und dem Basiswissen kann Inferenz betrieben werden mit dem Ziel, neue Fakten abzuleiten. Liegt Hintergrundwissen über den betrachteten Diskursbereich vor, können Fakten zu komplexeren Fakten aggregiert werden. Der Zeithorizont, in dem die Information des CPL valide sind, ist nicht groß. Zeitlich ausgedehntere Zusammenhänge werden in der *Verhaltens-Repräsentations-Ebene* (BRL) repräsentiert. Im dort ansässigen prozeduralen Gedächtnis liegt das Hintergrundwissen, um die begrifflich repräsentierten Teilzusammenhänge aus dem CPL in einen zeitlichen Kontext einzubetten. Die Ursache für den aktuell vorliegenden Teilzusammenhang liegt i.d.R. in der Handlung eines beobachteten Agenten. Zusammenhängende Handlungen werden als Verhalten bezeichnet. Die *Steuerung* (CL Control) ist ebenfalls Bestandteil des CL. Das intentionale Gedächtnis verkörpert die Ziele der Inferenz und im attentionalen Gedächtnis wird die konkrete Art der Informationserfassung festgelegt. Die von der Inferenz identifizierten konkreten Verhalten und Teile davon werden im episodischen Gedächtnis abgelegt. In Tabelle 1.1 sind die identifizierten Gedächtnisse zusammengefasst [Anderson et al., 2004, Tulving and Craik, 2005].

Gedächtnis	sensorisch	deklarativ	prozedural	episodisch	intentional	attentional
Funktion	Sensor-daten	Kenntnisse, Regeln	Fakten, Semantisches Wissen	erkannte Verhalten	Wert-zuweisung, Ziele	Wert-zuweisung, allgemein

Tabelle 1.1: Die unterschiedlichen Gedächtnisse im Cognitive Vision System.

Unter einem anderen Blickwinkel kann das CVS auch so betrachtet werden, dass es in einer Richtung eine Verkettung von Abstraktionen und in entgegengesetzter Richtung eine Verkettung von Konkretisierungen realisiert. Bereits Arens [Arens, 2004] stellt fest, dass das CVS eine mögliche Modellierung eines kognitiven Systems sei, und dass es fraglich sei, ob sich jede technische Realisierung immer an die theoretisch erarbeiteten scharfen Trennungen der Schichten halten würde. Es bleibt aber das Ziel.

1.4 Schwerpunkte und Beiträge der Arbeit

Dieser Arbeit liegen die folgenden Hauptbeiträge zugrunde:

Beitrag #1: Behandlung von Unschärfe. Der Formalismus der begrifflichen auf FMTHL und SGTs basierenden Situationserkennung wurde um die Behandlung von Unschärfe und Ableitbarkeit aller zutreffenden Hypothesen erweitert. Diese konzeptionelle Erweiterung ermöglicht es, die Eingabedaten entsprechend der Realität abzubilden und durch den gesamten Inferenzprozess die Unsicherheit zu propagieren. Des Weiteren haben die untersuchten Diskursbereiche gezeigt, dass sich ein beobachteter Agent gleichzeitig in mehr als einer Situation befinden kann. Daher erlaubt die Ableitung aller zu diesem Zeitpunkt zutreffenden Situationen die erschöpfende Beschreibung der beobachteten Szene.

Beitrag #2: Fehlende Information. Im Vergleich zu perfekten (künstlich erzeugten) Daten gibt es bei realen Daten Effekte, welche die Robustheit der nachfolgenden Verarbeitungsschritte erheblich schwächen. Auf der Ebene der Eingabedaten kann Rauschen und zum Teil eine gewisse Unvollständigkeit durch Vorverarbeitungsfilter reduziert werden. Wohingegen auf semantischer Ebene Lücken von Beobachtungen dazu führen können, dass einzelne Situationen nicht ausgeprägt werden können und die konsekutiven Situationen somit niemals erreicht werden können. Das

Konzept der kontrollierten Halluzination wurde entwickelt, um auf semantischer Ebene diese Unvollständigkeiten zu kompensieren.

Beitrag #3: Komplexitätsreduktion. Besonders der Diskursbereich Videoüberwachung umfasst Situationen mit vielen Agenten und deren Beziehungen untereinander. Die bisherige Modellierung von Relationen stößt dabei an ihre Leistungsgrenze. Um mit dieser Komplexität umgehen zu können, wurde der Schlussfolgerungsprozess um eine gruppenzentrierte Inferenz erweitert. Modellierbare Abhängigkeiten sind jetzt nicht mehr binär, sondern können beliebig auf Mengen definiert werden. Semantisch ähnliche Eingabeinformationen können zusammengefasst werden um die Handhabbarkeit in realen Systemen zu erhöhen.

Beitrag #4: Generische Anwendbarkeit. Die generische Anwendbarkeit der auf FMTHL und SGT basierenden Situationserkennung wird gezeigt, indem dieser Ansatz erstmalig auf die automatische Erkennung von Aktivitäten des täglichen Lebens in intelligenten Umgebungen eingesetzt wurde. In diesem Fall existieren keine bildbasierten Eingabedaten, sondern lediglich binäre Sensoren, die Zustände der Umgebung, wie z.B. Licht an oder Licht aus, erfassen. Eine Übertragung von Wissen aus dem Videoüberwachungskontext ist in Teilen möglich; die Schnittstelle zu den Sensoren muss angepasst werden und neue, bisher nicht vorhandene Situationen müssen neu modelliert werden. Die Übertragbarkeit von Wissen innerhalb des Diskursbereiches ist bis auf Schnittstellenanpassungen möglich. Lernende Verfahren werden hier eingesetzt um Verhalten personenspezifisch zu identifizieren und um schließlich die Modellierung von Hintergrundwissen zu unterstützen.

Beitrag #5: Technische Umsetzung und Evaluierung. Die in dieser Arbeit untersuchten Szenarien gehören dem Diskursbereich der Videoüberwachung im Innen- und Außenbereich an. Neben der Nutzung

öffentlich verfügbarer Datensätze wurde auch ein eigener geschaffen, um die einzelnen methodischen Erweiterungen systematisch mit künstlichen und realen Daten auszuwerten. Mit jeder einzelnen Erweiterung wird eine Steigerung der Erkennungsleistung von Situationen erlangt. Die Auswertung umfasst sowohl das in SGTs modellierte Hintergrundwissen der erwarteten Situationen, als auch die in FMTHL zugrunde liegende Basistaxonomie.

1.5 Aufbau der Arbeit

Kapitel 2 gibt einen Überblick über den Stand der Forschung bei der Situationsanalyse. Es werden die verschiedenen Familien der Ansätze vorgestellt und deren Vor- und Nachteile diskutiert. Schließlich wird die eigene Arbeit im Hinblick auf den Stand der Forschung eingeordnet.

Kapitel 3 beschreibt die technischen Grundlagen, mit denen das CVS realisiert wurde. Hintergrundwissen wird als SGTs modelliert und diese können mit dem neuen SGTyEditor erstellt werden. Der Prozess der Situationsanalyse – die Situationsgraphenbaumtraversierung – wird ebenfalls vorgestellt.

Kapitel 4 erweitert die Situationsanalyse um die Behandlung von Unschärfe und die Multihypothesenfähigkeit.

Kapitel 5 geht auf das Fehlen von Information ein. Fehlt Information auf der Ebene der Eingabedaten, dann kann diese oft geeignet interpoliert werden. Fehlt sie auf einer höheren begrifflichen Ebene, dann können fehlende Fakten kontrolliert halluziniert werden.

Kapitel 6 präsentiert zwei unterschiedliche Herangehensweisen zur Reduktion der Komplexität bei der Situationsanalyse. Zum Einen werden

Methoden des maschinellen Lernens verwendet, um die Eingabedaten semantisch vorzufiltern, und zum Andern wird die Modellierbarkeit auf Mengen eingeführt.

Kapitel 7 stellt eine geeignete Systemarchitektur zur technischen Umsetzung des CVS vor. Ebenso werden konkret verwendete Bildverarbeitungsmodule wie Objektdetektor, Objektverfolgung und Aktionserkennung beleuchtet. Anschließend zeigen verschiedene Auswertungen die Performance der oben eingeführten Erweiterungen.

Kapitel 8 beantwortet die Frage, inwieweit die auf FMTHL und SGTs basierende Situationserkennung auf andere Diskursbereiche übertragen werden kann. Weitere Untersuchungen werden bei der Übertragbarkeit innerhalb des Diskursbereichs angestellt und lernende Verfahren zur Identifikation von Merkmalen eingesetzt.

Kapitel 9 schließt diese Arbeit mit einer Zusammenfassung ab und in einem Ausblick werden vielversprechende Ideen zur Weiterentwicklung und Weiterführung dieser Arbeit vorgestellt.

2 Stand der Forschung und Einordnung der Arbeit

Die Situationsanalyse in Bildfolgen erfordert gleichermaßen Teilkomponenten aus den Bereichen der Künstlichen Intelligenz, des Maschinellen Lernens und der Bildverarbeitung. Erst eine Integration dieser unterschiedlichen Komponenten ermöglicht eine geschlossene Betrachtungsweise auf die Situationsanalyse bei unvollständiger und fehlerhafter Information, wie sie bei realen Daten natürlicherweise vorliegt.

Die Übersichtsartikel [Turaga et al., 2008, Lavee et al., 2009, Aggarwal and Ryoo, 2011, Vishwakarma and Agrawal, 2013, Ye et al., 2012] und die Bücher [Gottfried, 2009, Gong and Xiang, 2011] bieten neben der Betrachtung von Verfahren für einfache Aktionen auch einen Überblick über das aktuelle Feld der Situationsanalyse. Motiviert durch Anwendungen wie die semantische Beschreibung von Videos werden in der Arbeit [Turaga et al., 2008] einfache Aktionen und Situationen (dort: „activities“) verwendet, vorhandene Arbeiten vorgestellt und diese diskutiert. Als zweistufiger Abstraktionsprozess von Eingabesignalen zu Zwischenmerkmalen zu Situationen wird die Situationsanalyse in [Lavee et al., 2009] aufgefasst. Die Anwendung, in deren Kontext die Verfahren diskutiert werden, ist hauptsächlich Videoüberwachung im Innen- und Außenbereich. Verfahren für sowohl einfache Aktionen als auch Interaktionen und Gruppenaktivitäten werden in [Aggarwal and Ryoo, 2011] als auch in [Vishwakarma and Agrawal, 2013] sytematisch vorgestellt und verglichen. Im Gegensatz zu den bisher angerissenen Arbeiten diskutiert [Ye et al., 2012] Verfahren zur Situationserkennung, die nicht auf Videodaten als Primärquelle arbeiten,

sondern im Kontext von Pervasive Computing unter Einsatz vieler alternativer Sensoren zur Datenerfassung eingesetzt werden.

Die Abgrenzung zwischen Verfahren zur Aktionserkennung und Verfahren zur Situationsanalyse ist fließend und zu einem gewissen Teil auch überschneidend. In den folgenden Abschnitten sind die verschiedenen Ansätze zur Situationsanalyse nach der von [Aggarwal and Ryoo, 2011] vorgeschlagenen methodenbasierten Taxonomie dargestellt und vergleichend betrachtet. Ein erstes Kriterium, um verschiedene Ansätze zu unterscheiden, bietet der die Architektur betreffende interne Aufbau. Einerseits kann die Situationsanalyse hierarchisch angegangen werden, indem das Problem in immer kleinere Teilprobleme aufgeteilt wird, andererseits kann diese Hierarchie auch sehr flach sein, nämlich nur eine Schicht beinhalten, dann spricht man von direkten Verfahren.

2.1 Direkte Verfahren

Direkte Verfahren sehen von außen so aus, dass sie die Eingabedaten direkt in Ausgabedaten umwandeln. Man nennt diese Verfahren auch Black-Box-Verfahren. Dabei kann das „Innenleben“ der Black-Box durchaus sehr komplex sein; oft ist es schwierig, die Entscheidungen, die in dieser Black-Box getroffen werden, zu verstehen oder zu interpretieren. Nach der Taxonomie von [Aggarwal and Ryoo, 2011] können direkte Verfahren noch weiter in Raum-Zeit- und sequentielle Verfahren unterteilt werden. Direkte Raum-Zeit-Verfahren können robust einfache und periodische Aktionen und Gesten erkennen, auch unter schwierigen Bedingungen wie Rauschen und schlechter Beleuchtung. Für die Situationsanalyse sind sie nicht geeignet, weil sie die dort erforderliche Komplexität im Allgemeinen nicht abbilden können. Wesentlich besser nutzen die direkten sequentiellen Verfahren die zeitlichen Veränderungen der verwendeten Merkmale. Aber auch hier geht die abbildbare Komplexität bisher nicht weiter als zu einfachen Interaktionen zwischen zwei Personen, wie ein effizientes auf Coupled-

Hidden-Semi-Markov-Modellen basierendes Verfahren [Natarajan and Nevatia, 2007].

Vor- und Nachteile Direkter Verfahren

Auf anderen Anwendungsgebieten als der Situationserkennung mit weniger zeitlich und räumlich verschränkter Information wie beispielsweise bei Objektdetektion und Aktionserkennung funktionieren direkte Verfahren sehr gut und sehr effizient. Ein Grund dafür sind die Menge an dafür vorhandenen Trainingsdaten, aus denen Raumbezug, Zeitbezug und Wissen abgeleitet werden kann. Damit sind die Einsatzgebiete immer sehr konkret abgesteckt und durch die Trainingsdaten vorgegeben. Die direkten Verfahren lassen sich nicht ohne Weiteres auf andere Szenen übertragen und schon gar nicht in anderen Diskursbereichen anwenden. Andererseits sind diese Verfahren abgrenzbare, in bestimmten Umgebungen gut funktionierende, austauschbare Module. Für die Situationsanalyse, die über Interaktionen zweier Personen hinausgeht, sind sie wenig geeignet.

2.2 Hierarchische Verfahren

Die Alternative zu direkten Verfahren sind hierarchische Verfahren. Die Grundidee ist, dass nicht vom Bildmerkmal auf die Situation geschlossen wird, sondern eine oder mehrere Zwischenschichten existieren. Situationen bauen dann auf Zwischenergebnissen (im Folgenden: atomaren Aktionen) von niederen Verfahren wie Objekttracking, Aktionserkennung oder Personenidentifikation auf. Diese niederen Verfahren können dann unabhängig von höheren Verfahren zuerst angewandt werden. Die Trennung in verschiedenen Schichten macht das Problem nicht nur von der Komplexität und Laufzeit handhabbarer, sondern erlaubt auch gewissermaßen die Wiederverwendung von Ergebnissen der atomaren Aktionen aus niederen Schichten. Damit befindet sich die Wissensmodellierung auf einer abs-

trakteren Ebene und kann somit generischere Situationen modellieren. Ein entscheidender Vorteil gegenüber direkten Verfahren ist die Möglichkeit, mit komplexeren Strukturen umzugehen, wodurch hierarchische Methoden auch für die Situationsanalyse einsetzbar sind. Im Folgenden werden hierarchische Verfahren entsprechend der eingesetzten Schlussfolgerungsmethoden vorgestellt und diskutiert.

2.2.1 Statistische Verfahren

Statistische Verfahren verwenden fast immer eine Art von Probabilistischen Graphischen Modellen. Dabei handelt es sich strukturell um gerichtete Graphen. Die Abhängigkeiten zwischen den einzelnen Knoten werden durch gerichtete Kanten ausgedrückt. Eine gute Übersicht über traditionelle Methoden liefert [Berger, 1985], jüngere Entwicklungen sind in [Koller and Friedman, 2009] dargestellt.

Eine gängige Herangehensweise ist ein Layered-Hidden-Markov-Modell mit zwei Schichten [Oliver, 2002]. In der ersten Schicht werden die atomaren Aktionen erkannt, dann damit die komplexeren. Anwendungsszenarien sind u.a. Interaktionen in Besprechungsräumen. In [Wojek et al., 2006] werden Aktivitäten mit mehreren Personen in Büroumgebungen erkannt – wie z.B. „Meeting“, „Discussion“ oder „Phone Call“. Sowohl für Audio- als auch für Videoeingabedaten erkennen Hidden-Markov-Modelle atomare Aktionen, die in einem weiteren Hidden-Markov-Modell in der nächst höheren Ebene zur erkannten Aktivität fusioniert werden. Die Aufteilung der Audiodaten, Videodaten und der Fusion in verschiedene Hidden-Markov-Modelle reduziert den Trainingsaufwand und erleichtert die Modellierbarkeit. Um die Robustheit zu erhöhen, wurde ein raumweises Personentracking integriert, damit die Identitäten von Personen aufrecht erhalten werden können. In [Rosario et al., 1999] werden Coupled-Hidden-Markov-Modelle für die Erkennung von Verhalten mit zwei beteiligten Agenten eingesetzt, siehe dazu Abbildung 2.1. Dynamische Bayes'sche Netze (DBNs)

zur Erkennung von Objektinteraktionen im Anwendungsbereich Perimeterschutz werden in [Fischer and Beyerer, 2012] vorgestellt. Dort werden die Parameter für das DBN initial automatisch bestimmt. Die Diskursbereichsübertragbarkeit zu maritimen Szenarien wird in [Fischer, 2012] gezeigt.

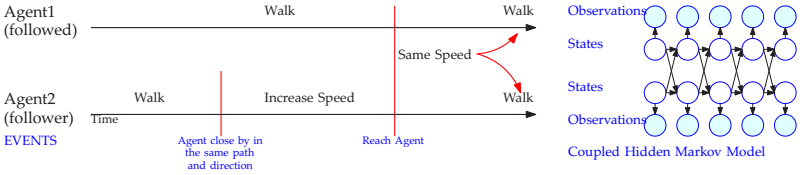


Abbildung 2.1: Coupled-Hidden-Markov-Modelle werden in [Rosario et al., 1999] eingesetzt, um das kooperative Verhalten zweier Agenten zu identifizieren. In den verborgenen Zuständen wird beispielsweise das Verhalten zweier Agenten, von denen einer den anderen verfolgt, modelliert.

Die folgenden Arbeiten gehen über Interaktionen zwischen Agenten hinaus und behandeln Gruppensituationen. In [Dai et al., 2008] wird ein ereignisgesteuertes, mehrschichtiges Dynamisches Bayes'sches Netz für ein dynamisches Kontextmodell eingesetzt, um Interaktionen in Gruppen unter Beachtung des Kontexts zu analysieren. Zweischichtige Hidden-Markov-Modelle verwendet [Zhang et al., 2006] für die Erkennung von Gruppensituationen in Besprechungsszenarien. Dynamic Probabilistic Networks werden in [Gong and Xiang, 2003] zur Detektion von Gruppensituationen eingesetzt.

Arbeitsabläufe im industriellen Kontext werden in [Kosmopoulos, 2012] durch Hidden-Markov-Modelle und Bayes'sche Filter erkannt. Hierarchische Hidden-Markov-Modelle zur Erkennung unterschiedlicher Essensabläufe werden in [Nguyen et al., 2005] eingesetzt.

Vor- und Nachteile Statistischer Verfahren

Statistische Verfahren stellen ein wohldefiniertes statistisches Rahmenwerk dar, das auch – wie die Literatur oben zeigt – gut mit den Problemen realer Daten wie Rauschen oder fehlende Information umgehen kann. Zum Teil müssen zwar initiale Modelle von Hand geschaffen werden, doch dann können hier sehr effektiv Verfahren des Maschinellen Lernens eingesetzt werden. Aufgrund der durch die Anwendungen meist vorgegebenen Szenarien sind die Trainingsdaten oft beschränkt, was zu Effekten wie Überanpassung oder zu wenig Information führen kann. Wenn die Situationen komplexer werden, ist oft die initiale Schaffung eines Modells eine Herausforderung. Im Vergleich zu beschreibungsbasierten Verfahren tragen die einzelnen Elemente eine eher schwierig zu interpretierende oder gar keine begriffliche Semantik.

Statistische Verfahren stehen und fallen mit den ihnen zur Verfügung stehenden Trainingsdaten. Ist der Anwendungsbereich kompakt und bilden die Trainingsdaten die vorkommenden Situationen gut und ausreichend ab, dann werden in eben diesem Anwendungsbereich die Situationen gut erkannt. Das erlernte Wissen stützt sich auf erlernte Korrelationen und nicht – wie bei vielen beschreibungsbasierten Verfahren – auf die Beziehung zwischen Ursache und Wirkung. Die Übertragbarkeit von Wissen wird zwar angestrebt, ist aber ohne größere Anstrengungen kaum möglich. Zeitliche Zusammenhänge können – sofern sie sequenziell ablaufen – sehr gut modelliert werden, und zwar so lange, bis Teilzusammenhänge zeitlich parallel modelliert werden sollen.

2.2.2 Syntaktische Verfahren

Syntaktische Verfahren repräsentieren Verhalten als symbolische Zeichenketten. Jedes Zeichen steht dabei für eine atomare Aktion. Wie bei den statistischen Verfahren ist es auch hier erforderlich, dass die atomaren Aktionen von anderen Verfahren bereits erkannt worden sind. Die Menge aller

Verhalten kann man dabei als formale Sprache über den atomaren Aktionen auffassen. Die Situationserkennung entspricht dann dem Wortproblem der Sprache.

Stochastisch kontextfreie Grammatiken werden zur Erkennung von Zeigegesten in [Ivanov and Bobick, 2000] eingesetzt. In dieser manuell erstellten formalen Grammatik ist die Menge aller validen Zeigegesten zur Darstellung eines Quadrats abgebildet. In der Arbeit [Kitani et al., 2008] werden stochastisch kontextfreie Grammatiken unter der Behandlung von Unsicherheit weiterentwickelt. Schließlich wird mit stochastisch kontextfreien Grammatiken die Human Activity Language geschaffen, siehe [Guerra-Filho and Aloimonos, 2007, Aloimonos et al., 2009]. In diesen Arbeiten werden Situationen inkrementell beschrieben: Eine Situation, ein Wort der Human Activity Language, besteht aus mehreren Teilaktionen (Morpheme), die wiederum aus mehreren Basissymbolen (Phoneme) bestehen. Siehe dazu auch Abbildung 2.2. Eine andere Art von Grammatik wird in [Geib, 2009] in Form von Combinatory Categorial Grammars verwendet. Diese erlauben späte Entscheidungen für effektivere Planungsstrategien.

Attributierte Grammatiken, eine Erweiterung von stochastisch kontextfreien Grammatiken, werden in der Arbeit [Joo, 2006] in einem Parkplatzszenario eingesetzt. Hier werden nicht nur Situtionen erkannt, sondern es wird auch explizit auf mögliche Abweichungen des modellierten Normalverhaltens eingegangen. Die der Grammatik hinzugefügten Attribute werden verwendet, um Bedingungen auf Ebene der Eingabemerkmale und zeitliche Abhängigkeiten zu modellieren.

In der Arbeit [Barrett et al., 2016] wird eine Sprache definiert, um Situationen im Einzelbildern zu beschreiben. Der Fokus liegt dabei auf der Extraktion des handelnden Agenten, des Objekts einer Aktion und der Aktion selbst.

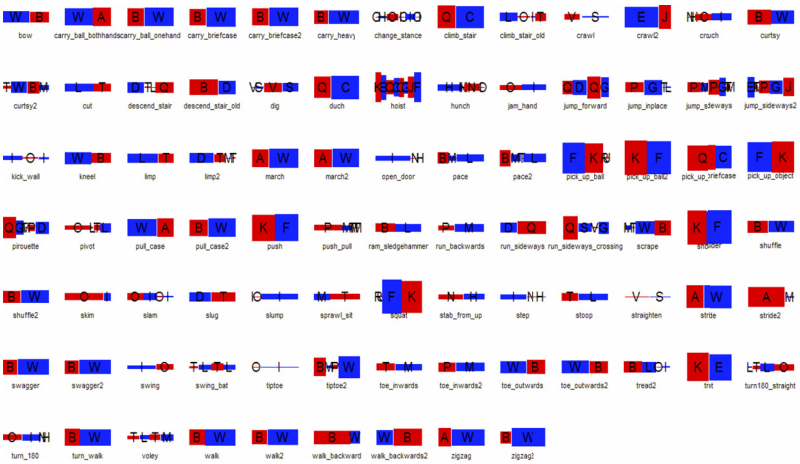


Abbildung 2.2: Ein Wort der Human Activity Language besteht aus mehreren Teilaktionen (Morphemen), die wiederum aus mehreren Basissymbolen (Phonemen) bestehen. Visualisierung der Basissymbole am Beispiel der Bewegung der rechten Hüfte. Die Symbole und Farben stehen für unterschiedliche kinematische Gelenkbewegungen. Abbildung aus [Guerra-Filho, 2007].

Vor- und Nachteile Syntaktischer Verfahren

Die Vor- und Nachteile syntaktischer Verfahren verhalten sich analog zu denen statistischer Verfahren, bis auf einen entscheidenden Unterschied: Durch die Natur der Grammatiken sind syntaktische Verfahren per se hierarchisch, dafür allerdings wenig für verrauschte oder fehlerhafte Daten geeignet und sie scheitern an Lücken. Die zeitlichen Beschränkungen von parallelen Handlungen sind auch hier erfüllt, im Wesentlichen ist man auf sequenzielle Situationen beschränkt, weil eine Grammatik keine Modellierung von parallel stattfindenen Handlungen vorsieht. Das Wissen ist in der Grammatik in Form von Produktionsregeln abgelegt. Diese zu erstellen ist aufwendig, wird bei zunehmender Situationskomplexität immer umfangreicher und unübersichtlicher und bietet nur eine eingeschränkte Flexibi-

lität. Man braucht aber keine oder nur wenige Lerndaten und entsprechend ist die Auswertung nicht durch Trainingsdaten begrenzt, sondern durch die Fantasie des Schöpfers.

2.2.3 Beschreibungsbasierte Verfahren

Die bisher betrachteten Verfahren kommen an die Grenzen ihrer Leistungsfähigkeit, wenn zeitlich ausgedehnte Situationen v.a. mit mehreren Agenten beschrieben werden sollen. Kann ein Experte dem Verfahren Wissen in Form von sprachlich gefassten zeitlichen und örtlichen Zusammenhängen und Begrifflichkeiten über erwartete Situationen auf einer semantisch interpretierbaren Ebene mitteilen, dann müsste dieses Wissen nicht gelernt werden, sondern könnte direkt verwendet werden. Dies realisieren beschreibungsbasierte Ansätze, die meist durch eine nicht-triviale Logik oder logikähnliche Struktur realisiert sind.

Sie sind in ihrer Struktur per se hierarchisch aufgebaut. Ähnlich zu den oben genannten hierarchischen Verfahren bilden auch kleine konkrete Teilzusammenhänge größere abstraktere Zusammenhänge, mit dem Unterschied, dass beschreibungsbasierte Verfahren tiefere Hierarchien aufbauen können.

Maßgeblich limitiert werden die oben genannten Verfahren durch Einschränkungen bei der Modellierung gleichzeitiger Abhängigkeiten. Der Allen-Kalkül [Allen, 1983, Allen and Ferguson, 1994] hebt diese Einschränkungen auf, indem er es ermöglicht, Relationen über gleichzeitige bzw. sich überschneidende Zusammenhänge auszudrücken. Diese grundlegende Errungenschaft ist die Basis vieler Verfahren [Pinhanez and Bobick, 1998, Siskind, 2001, Vu et al., 2003, Gupta et al., 2009].

Algorithmisch anwendbar wird der Allen-Kalkül, wenn er z.B. in ein Past-Now-Future-Netzwerk konvertiert wird [Pinhanez and Bobick, 1998]. Siehe dazu als Beispiel Abbildung 2.3. Die Komplexität der Schlussfolgerung ist polynomial in der Zeit und dennoch sind Past-Now-Future-

Netzwerke so ausdrucksmächtig, dass sie Situationen im Küchenkontext modellieren können. Von Nachteil ist die Redundanz in der Wissensbasis bei mehrfach auftretenden Teilzusammenhängen.

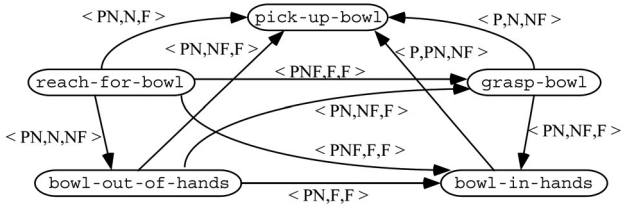


Abbildung 2.3: Past-Now-Future-Netzwerk zu „pick-up bowl“. Die zeitlichen Abhängigkeiten des Allen-Kalküls werden in den Wertebereich $(past, now, future)$ konvertiert. Diese Kanten modellieren eben die zeitlichen Bedingungen zwischen den einzelnen atomaren Aktionen. Abbildung aus [Pinhanez and Bobick, 1998].

Ähnlich [Nevatia et al., 2003], die die Video Event Representation Language einführen um Abläufe auf unterschiedliche Weise zu verknüpfen, wird in [Vu et al., 2003] ein Formalismus (Scenarios) eingeführt, um auf ähnliche Weise Situationen zu modellieren, siehe Abbildung 2.4 für ein Beispiel. Die Scenarios ähneln einem Drehbuch für zu erkennende Situationen, bestehend aus Name, beteiligten Charakteren und Objekten, Teilhandlungen sowie zeitlichen Abfolgebedingungen. In [Patino et al., 2008] werden die beteiligten Charaktere um Objekte mit Kontextrelevanz erweitert.

Eine baumartige Struktur von AND-OR-Graphen präsentiert [Gupta et al., 2009]. Ähnlich dazu die Arbeiten in [Ryoo and Aggarwal, 2006]: Eine kontextfreie Grammatik mit logischen Verküpfungen wie und, $oder$ und $nicht$ charakterisiert das Rahmenwerk. In [Ryoo and Aggarwal, 2009] erfährt dieses Rahmenwerk eine probabilistische Erweiterung, die es erlaubt, auch mit verrauschten Eingabedaten umzugehen. Zusätzlich wurde auch die Halluzination von atomaren Aktionen ermöglicht, ähnlich zu [Minnen et al., 2003].

```
Scenario(Bank_attack,
   Characters((cashier:Person), (robber:Person))
   SubScenarios(
     (cas_at_pos,inside_zone,cashier,"Back_Counter")
     (rob_enters, changes_zone, robber,
           "Entrance_zone", "Infront_Counter")
     (cas_at_safe, inside_zone, cashier, "Safe")
     (rob_at_safe, inside_zone, robber, "Safe") )
   ForbiddenSubScenarios(
     (any_in_branch, inside_zone, any_p,"Branch"))
   Constraints(
     Temporal ((rob_enters during cas_at_pos)
               (rob_enters before cas_at_safe)
               (cas_at_pos before cas_at_safe)
               (rob_enters before rob_at_safe)
               (rob_at_safe during cas_at_safe))
     Atemporal((cashier ≠ robber))
     Forbidden((any_p ≠ cashier) (any_p ≠ robber)
               (any_in_branch during rob_at_safe))))
```

```
Scenario(Bank_attack_1,  # ω¹
   Characters((cashier:Person), (robber:Person))
   SubScenarios(
     (cas_at_pos, inside_zone, cashier,
                                   "Back_Counter")
     (rob_enters, changes_zone, robber,
           "Entrance_zone", "Infront_Counter"))
   Constraints((cas_at_pos during rob_enters)
               (cashier ≠ robber) ))
```

Abbildung 2.4: Das Verhalten „Bank_attack“ wird als Scenario modelliert. Textuell sind Teilverhalten und deren zeitlichen Zusammenhänge beschrieben. Abbildung aus [Vu et al., 2003].

In und um die Arbeitsgruppe von H.-H. Nagel war das angestrebte Ziel die automatische natürlichsprachliche semantische Beschreibung von Bildfolgen. Seit den achtziger Jahren ist man – mit Fokus auf den Diskursbereich Straßenverkehr – diesem Ziel näher gekommen, [Nagel, 1979, 1985, 1988, 1991, Nagel et al., 1995, Nagel, 2000, Arens and Nagel, 2003, Nagel, 2004, 2006, Arens et al., 2008, Gerber and Nagel, 2008, Nagel, 2010]. Nicht nur die Erkennung von Abläufen war dabei Forschungsgegenstand, sondern ebenso die Bereitstellung geeigneter Bildverarbeitungsverfahren, die für die älteren Arbeiten noch nicht zur Verfügung standen.

Der Ansatz ist stark modellgetrieben und versucht eine erschöpfende Beschreibung des beobachteten Diskursbereichs. So wurden beispielsweise in [Cahn von Seelen, 1988] über 80 Bewegungsverben ermittelt und untersucht, wie diese sich auf Trajektorien in der Szene abbilden lassen, um damit Geschehen ableiten zu können; siehe Abbildung 2.5.

Bei der Abbildung von quantitativen Trajektorien zu qualitativen Beschreibungen muss die semantische Lücke überwunden werden. Dabei wurden Schlussfolgerungsmethoden wie die FMTHL [Schäfer and Brzoska, 1996] und Wissensrepräsentationen wie unscharfe Zuordnungsautomaten [Gerber and Nagel, 2008] und Situationsgraphenbäume [Krüger, 1991, Arens, 2004] entwickelt. Hierauf wird in Kapitel 3 genauer eingegangen.

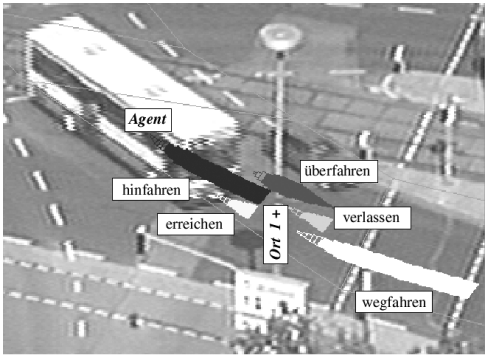


Abbildung 2.5: Bewegungsverben im Straßenverkehr. Abbildung aus [Nagel, 1996].

Die grafische intuitive Erstellung von Hintergrundwissen mit Situationsgraphenbäumen erlaubt es auf einfache Weise, komplexe Situationen begrifflich und zeitlich zu modellieren. Das ist auch ein Grund warum sich diese Form der Wissensrepräsentation bei beschreibungsbasierten Ansätzen bewährt und neben anderen etabliert hat [González et al., 2004, Baiget et al., 2007, González et al., 2009, Fernández Tena, 2010, Bellotto et al., 2012, IJsselmuiden, 2014].

In der Arbeit [IJsselmuiden, 2014] wurden SGTs und eine Wissensbasis aus FMTHL-Regeln entwickelt, um Situationen in einer intelligenten Umgebung (Smart Room) – in diesem Fall einen Übungsraum eines Feuerwehrstabs – kontinuierlich zu analysieren, mit dem Ziel der Protokollierung der gesamten Beobachtungszeit.

Vor- und Nachteile Beschreibungsbasierter Verfahren

Bei beschreibungsbasierten Verfahren wird die Modellierung von Verhaltenswissen oft durch einen Experten durchgeführt. Die Gründe dafür sind durch die oft nur in geringem Umfang vorliegenden Trainingsdaten gegeben. Für komplexe Situationen ist die Menge an Trainingsdaten zwangs-

läufig physikalisch beschränkt. Aus wenigen Trainingsdaten können Lernverfahren in der Regel keine Kausalitäten ableiten. Das in einer Logik o.ä. modellierte Hintergrundwissen liegt formalisiert vor und ist daher maschinenlesbar. Gleichzeitig haftet der Modellierung eine Semantik an. Diese ist dem Menschen direkt verständlich. Durch die Hierarchie wird das Wissen auf verschiedenen Ebenen von oben nach unten konkretisiert. Durch diese hierarchische Modellierung von Wissen kann das Ganze oder auch Teile davon an anderen Stellen gut wiederverwendet werden.

Schritthaltend mit der größeren Ausdrucksmächtigkeit steigt auch die Komplexität der eingesetzten Schlussfolgerungsmaschinen auf den Logiken. Um die Schlussfolgerung dennoch im Hinblick auf Laufzeitaspekte benutzbar zu machen, bedient man sich in der Praxis zusätzlicher Annahmen oder Einschränkungen. Befriedigend ist diese technische Lösung nicht, da damit keine Skalierbarkeit des Verfahrens abgeleitet werden kann.

Eine weitere Herausforderung besteht in der Behandlung von verrauschten und unvollständigen Daten. Wenn eine Beobachtung eines Teilverhaltens fehlt, dann kann das Verhalten nicht weiter ausgeprägt werden. Einige beschreibungsbasierte Ansätze wurden daher um die Fähigkeit von Halluzination fehlender Beobachtungen erweitert.

2.2.4 Hybride Verfahren

Hybride Verfahren verwenden Teilaspekte unterschiedlicher Verfahrensfamilien. Markov-Logic-Networks (MLNs) [Domingos et al., 2006, Richardson and Domingos, 2006] verallgemeinern die Prädikatenlogik erster Ordnung (PL1) und Probabilistische Graphische Modelle. MLNs ergänzen Formeln der PL1 um Gewichte. Diese gewichteten Formeln definieren kollektiv eine Vorlage für die Konstruktion eines Probabilistischen Graphischen Modells, das die Verteilung der möglichen Welten spezifiziert. Die Inferenz leisten Markov-Chain-Monte-Carlo-Verfahren. MLNs haben eine große Ausdrucksmächtigkeit, die in der Praxis die Lern- und Inferenzver-

fahren an ihre Grenzen stoßen lassen oder alternativ starke Approximierungen fordern. MLNs werden bei der Videoüberwachung auf einem Parkplatz [Tran and Davis, 2008], in einer allgemeinen Umgebung [Skarlatidis et al., 2011], im Straßenverkehr [Kembhavi et al., 2010] und bei Sportspielen wie Basketball [Morariu and Davis, 2011] eingesetzt. In der Arbeit [Bär, 2016] wird mit MLNs in einem Personenkraftwagen im Straßenverkehr das Verhalten des Fahrers beobachtet und analysiert, um damit Fahrerassistenzsysteme individuell auf den Fahrer abzustimmen.

Die Bayesian-Logic-Networks [Jain, 2012] stellen gegenüber MLNs hinsichtlich der Komplexität der Lern- und Inferenzverfahren deutlich geringere Anforderungen. Daher stellen sie in dieser Hinsicht einen guten Kompromiss dar. Ein Bayesian-Logic-Network ist ein Metamodell, um Wahrscheinlichkeitsverteilungen aus Fragmenten von Wahrscheinlichkeitsverteilungen und globalen logischen Einschränkungen in PL1 zu konstruieren. Die Ausdrucksmächtigkeit gegenüber MLNs ist geringer.

Die Bayes'schen Kompositionellen Hierarchien [Neumann, 2008] sind eine Aggregathierarchie. Die Gesamtheit der Beschreibung der beobachteten Szene beschreibt die Wurzel, wohingegen die Blätter primitive Objekte repräsentieren: die nicht weiter zerlegbaren Ergebnisse von niederen Verfahren. Die einzelnen Aggregate werden durch eine Wahrscheinlichkeitsverteilung beschrieben. Eine praktische Umsetzung davon findet man in den Arbeiten [Bohlken et al., 2011, Bohlken, 2012]. Dort wird ein generisches System zur Situationsanalyse im Echtzeitbetrieb vorgestellt. Situationen werden mit einer Web-Ontology-Language (OWL)-Ontologie definiert, erweitert durch Semantic-Web-Rule-Language (SWRL)-Regeln, zur Modellierung von Einschränkungen.

Vor- und Nachteile Hybrider Verfahren

Es zeigt sich in der Praxis, dass hybride Verfahren gut funktionieren. Oft stellen sie einen geeigneten Kompromiss zwischen Komplexität der zu

erkennenden Situationen und der Inferenz dar. Allerdings darf nicht außer Acht gelassen werden, dass Lern- und Inferenzverfahren bei MLNs im schlechtesten Falle eine exponentielle Komplexität aufweisen [Koller and Friedman, 2009]. Daher muss man sich oft approximativen Verfahren bedienen.

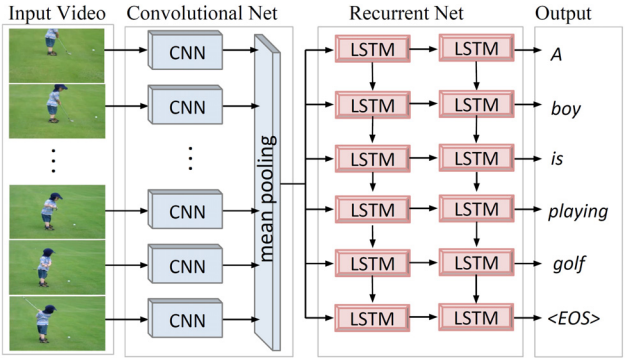


Abbildung 2.6: Die Architektur gliedert sich in Convolutional Neural Networks (CNNs) zur Merkmalsextraktion, ein Meanpooling und schließlich Long-Short-Term-Memory-Networks (LSTMs) zur Erzeugung der sprachlichen Beschreibung des Videoinhalts. Abbildung aus [Venugopalan et al., 2015b].

2.3 Deep Learning basierte Verfahen

Nachdem Deep Convolutional Neural Networks (CNNs) in Anwendungen wie Objektdetektion, semantischer Segmentierung und Aktionserkennung bisherige Ansätze deutlich übertreffen, wurden diese Verfahren auch zur Beschreibung von Bild- bzw. Videodaten übertragen. Bedingt durch den Ansatz per se handelt es sich bei Deep Learning basierten Verfahren nach außen hin um direkte Verfahren, die intern hierarchisch, aber nicht bedeutungstragend sind. Wie bei den direkten Verfahren wird lediglich ein Bezug zwischen Eingabedaten und Ausgabedaten hergestellt. Es ist nicht einfach semantisch interpretierbar, was dazwischen passiert. So kann z.B.

eine Szene als Besprechung erkannt werden ohne aber die einzelnen Personen explizit detektiert zu haben.

Die Arbeit von [Xu et al., 2015] beschreibt die erkannte Situation in Einzelbildern – also ohne zeitliche Information. Die wesentlichen Objekte der Szene und deren Handlungen werden textuell beschrieben. Das Verfahren basiert auf Long-Short-Term-Memory-Network (LSTMs), die um eine Schicht erweitert wurden.

Auch die Arbeit [Venugopalan et al., 2015a] basiert auf LSTMs. Es wird zuerst eine Zwischenrepräsentation der Bilder geschaffen, die dann zu der textuellen Beschreibung der Szene übersetzt wird. Im Gegensatz dazu benutzen die Arbeiten [Donahue et al., 2015, Venugopalan et al., 2015b] sowohl CNNs zur Bilderfassung als auch anschließend LSTMs zur Erzeugung der textuellen Beschreibung der Szene. In Abbildung 2.6 ist exemplarisch die Architektur dieses Ansatzes dargestellt.

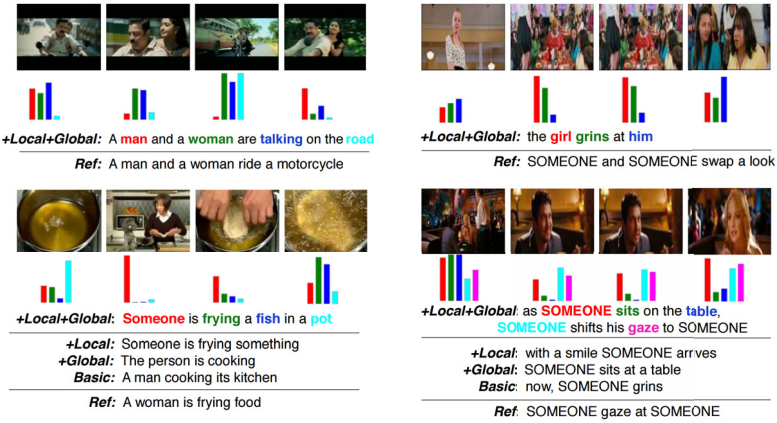


Abbildung 2.7: Textuelle Beschreibung von Videos. Abbildung aus [Yao et al., 2015].

In [Yao et al., 2015] wird zusätzlich zur lokalen Aufmerksamkeit auch die zeitliche Dimension der Aufmerksamkeit berücksichtigt. Dazu werden

spatio-temporal CNNs (3-D-CNN) verwendet. Schließlich werden die betrachteten Videos textuell beschrieben, siehe auch Abbildung 2.7.

Die Fortschritte, die bisher mit Deep CNNs sichtbar wurden, werden in Zukunft weiter voranschreiten. Es stellt sich die Frage, inwieweit sich die Netze in Zukunft einer detaillierten, auf Teilzusammenhänge einer Szene abzielenden Inferenz öffnen werden. Von einer Zeichenfolge, die den Bildinhalt beschreibt, bis zu einem detaillierten Verständnis was, warum, wo, wie und mit wem passiert ist, ist es noch ein weiter Weg.

2.4 Einordnung der Arbeit in den Stand der Forschung

Zur Situationsanalyse im Videoüberwachungskontext gibt es eine große Variabilität der visuellen Information zwischen den beobachteten Szenen innerhalb eines Diskursbereichs. D.h. niedere bildverarbeitende Methoden müssen modular und austauschbar sein, unter der Annahme, dass unter allen Sichtbedingungen funktionierende Bildverarbeitungsmethoden zur z.B. Personendetektion nicht existieren. Direkte Verfahren als Black-Box können das alleine nicht leisten; wohingegen sie als Teilkomponenten für ausdrucksmächtigere Verfahren gut geeignet sind. Die insgesamt ausdrückbare Komplexität von direkten Verfahren ist eingeschränkt und somit können komplexere Sachverhalte nicht abgebildet werden.

Zu den zu beobachtenden Situationen gibt es oft sehr wenige bis keine Trainingsdaten. D.h. das die Situationen modellierende Hintergrundwissen muss oft durch einen Experten erstellt werden, weil es nicht gelernt werden kann. Statistische Ansätze können weniger mit Hintergrundwissen umgehen – hierarchische Ansätze dagegen gut. Bei der Verarbeitung realer Daten treten Effekte wie Rauschen und fehlende Daten auf sowie gleichzeitige Handlungen. Alle diese Eigenschaften lassen sich bei den syntaktischen Verfahren schwer umsetzen.

Komplexere Zusammenhänge, also räumliche, zeitliche und abstrakte Abhängigkeiten, können bei beschreibungsbasierten Verfahren präzise modelliert werden. Gleichzeitig geht die Möglichkeit, komplexere Zusammenhänge zu modellieren, i.d.R. mit steigender Komplexität der Schlussfolgerungsverfahren einher. Bei realen Daten erfordern beschreibungsbasierte Verfahren meist eine explizite Behandlung der auftretenden Fehler sowohl derjenigen aus den Eingabedaten, als auch derjenigen der vorverarbeitenden Bildverarbeitungsmodule. Im Gegensatz zu Deep Learnung basierten Verfahren trägt das bei beschreibungsbasierten Verfahren in den verschiedenen Ebenen repräsentierte Wissen Bedeutung, also können auch Teile davon wiederverwendet werden.

Aus diesen Gründen ist die auf FMTHL und SGTs basierende Situationserkennung für die Situationsanalyse im Videoüberwachungskontext geeignet, bedarf aber einer gesonderten Betrachtung hinsichtlich der Behandlung von realen Daten, der Komplexität und der Übertragbarkeit von Wissen. In Abschnitt 1.4 sind die den Stand der Forschung erweiternden Beiträge dieser Arbeit aufgezählt.

3 Grundlagen

In diesem Kapitel werden die dieser Arbeit zu Grunde liegenden Methoden im Einzelnen vorgestellt. Wie in Abschnitt 2.4 herausgearbeitet wurde, werden in dieser Arbeit zur technischen Realisierung eines CVS die Konzepte der FMTHL und der SGTs verwendet. Die begriffliche formallogische Beschreibung von den Sensordaten über die Basisentitäten bis hin zum zeitlichen Verhalten lässt das Ziel einer natürlichsprachlichen Beschreibung von Sachverhalten bei der Bildfolgenauswertung näher rücken. Ähnlich zu beschreibungsbasierten Verfahren aus Abschnitt 2.2.3 ist die FMTHL die formale Basis. Aufbauend auf ihr werden die SGTs definiert und deren Traversierung beschrieben. Ebenfalls wird in diesem Abschnitt auf den neuen SGTyEditor eingegangen.

3.1 Unscharfe, metrisch-temporale Logik

Die begriffliche Beschreibung von Bildfolgen wird durch eine mehrstufige Kette realisiert [Nagel, 1979, 1985, 1988, 1991, Nagel et al., 1995, Nagel, 2000, 2004, 2006, 2010]. Die Bildfolge wird mit einer Kamera aufgezeichnet, also diskretisiert und quantisiert. Oft wird das erwartete Verhalten vor der Aufzeichnung a priori bekannt sein. Im weiteren Verlauf werden unter Anwendung von Grund- und Szenenwissen die erwarteten dynamischen Verhalten der Agenten abgeleitet. Danach hat man dann a posteriori das Ergebnis. Bereits das Rauschen der Kamera[5] verursacht einen Fehler, der als Zufallsvariable mit entsprechender Verteilung formalisiert werden

[5] Bei modernen Kameras ist dieses Problem untergeordnet.

kann. Dieser Fehler setzt sich durch die verschiedenen Verfahren bis zur Szenenbeschreibung fort. Weit bedeutendere Fehler können auf dem Weg zur Szenenbeschreibung entstehen: Durch Unzulänglichkeiten bei den bildverarbeitenden Verfahren können falsch negative oder falsch positive Fehler entstehen. Diese Fehler werden begünstigt, weil es in einer nicht abgeschlossenen Welt (open world assumption) nicht modellierte Objekte geben kann, also Clutter. Die komplexen Zusammenhänge machen eine analytische Beschreibung der Verteilung der Szenenparameter sehr schwierig, ohne zusätzliche stark einschränkende Annahmen bezüglich der Verteilung der Ausgangsdaten zu treffen. Also ist die Szenenbeschreibung im Allgemeinen mit einem Grad an Unsicherheit behaftet, der nur durch eine experimentelle Bestimmung der vorliegenden Verteilung ermittelt werden kann.

3.1.1 Erweiterung um Unschärfe

Eine Begriffsunschärfe entsteht durch die Zuweisung begrifflicher Beschreibungen zur Szenenkonfiguration durch zugewiesene natürlichsprachliche Begriffe. Die subjektive Interpretation der einzelnen Begriffe wird durch eine formale objektive Definition ersetzt. Um dem Grad der Unsicherheit der Eingangsdaten, der Modellannahmen und der subjektiven Interpretation der Begriffe zu begegnen wird den Begriffen ein gradueller Wahrheitsgehalt zugeordnet. Ein gradueller Wahrheitsgehalt ist wohl stochastisch motiviert, aber trotzdem ein subjektives Maß, das nicht näher formalisiert ist. Die unscharfe Logik (auch bekannt unter dem Namen Fuzzylogik) verwendet diese aus den unscharfen Mengen stammende Eigenschaft, indem einer Aussage $\mathcal{F}$ nicht nur scharf die Wahrheitswerte T oder F zugewiesen werden können, sondern ein Zusicherungsgrad $\mathcal{I}[\mathcal{F}] \in [0,1] \subset \mathbb{R}$. Die Semantik der logischen Operationen der unscharfen Logik ist äquivalent zur Semantik der Prädikatenlogik, spezialisiert auf die Grade 0 und 1. Ein unscharfer Modus Ponens legt fest, wie der Grad der Conclusio berechnet

wird. In [Schäfer, 1996] wird der prädikatenlogische Tableaukalkül derart erweitert, dass die Ableitbarkeit von Teilbeweiszielen durch einen Mindestgrad μ durch Angabe eines abgeleiteten Mindestgrads $\eta \geq \mu$ ermöglicht wird. Ein dort eingeführter Abschwächungsoperator legt fest, ob Atome zum unscharfen kleinsten Herbrandmodell eines unscharfen Programms gehören, indem der Wahrheitswert als Grad zugewiesen wird.

Ganz konkret wird die PL1 zur unscharfen Logik erster Ordnung (FL1) erweitert, indem die Wahrheitswerte 0 und 1 auf das Intervall $[0, 1]$ erweitert werden. Der Operator für unscharfe Abschwächung ist $\downarrow_\kappa$ mit der Definition: Wahrheitswert von $\mathcal{F} \geq \kappa \Rightarrow \downarrow_\kappa$ ist absolut wahr. Analog dazu der Operator für die unscharfe Verstärkung $\uparrow_\lambda$ mit der Definiton: Wahrheitswert von $\mathcal{F}$ absolut wahr $\Rightarrow \uparrow_\lambda \mathcal{F}$ hat Warheitswert λ. Des Weiteren werden die Semantiken *weak*, *medium* und *strong* für die verschiedenen unscharfen Operatoren $\wedge$, $\vee$, $\neg$ und $\leftarrow$ definiert. In Tabelle 3.1 sind die drei gleichzeitig benutzbaren Semantiken dargestellt.

$v \in$	$\{w, m, s\}$	weak	medium	strong
Konjunktion	$x \wedge_v y$	$min\{x, y\}$	$x * y$	$max\{0, x + y - 1\}$
Disjunktion	$x \vee_v y$	$min\{1, x + y\}$	$x + y - x * y$	$max\{x, y\}$
Negation	$\neg_v(x)$	$1 - x^2$	$1 - x$	$1 - \sqrt{x}$
Subjunktion	$y \leftarrow_v x$	$\neg_m x \vee_w y$	$\neg_m x \vee_m y$	$\neg_m x \vee_s y$

Tabelle 3.1: Verschiedene Semantiken der unscharfen Operatoren jeweils für Konjunktion, Disjunktion, Negation und Subjunktion. Bei der Subjunktion wird stets die Standard-Negation $\neg_m$ verwendet, vgl. [Schäfer, 1996].

3.1.2 Erweiterung um Zeit

Die modale Logik lässt alternative Interpretationen in verschiedenen Welten zu, die wiederum durch eine Zugänglichkeitsrelation miteinander verbunden sind. Zusätzliche logische Operatoren ermöglichen den Übergang von einer Welt in eine zugängliche Nachfolgewelt. Ersetzt man „Welt“

durch „Zeitpunkt“ und verwendet im Allgemeinen eine partielle Ordnungsrelation für die Zugänglichkeitsrelation, dann entsteht eine Zeitlogik. Bei einer totalen Ordnung spricht man dann von einer linearen Zeitstruktur. Bei einer dichten Zeitstruktur ist man strukturisomorph zu den reellen Zahlen. Die Basis der Verarbeitungskette ist eine diskrete Erfassung einer Szene. Deshalb ist die hier verwendete Zeitstruktur linear und diskret, also strukturisomorph zu den ganzen Zahlen. Sind nicht nur die Zugänglichkeitsrelationen „Vorgänger“ und „Nachfolger“ definiert, sondern auch arithmetische Operationen wie die Addition, Multiplikation und Subtraktion, dann handelt es sich um eine metrische Zeitstruktur. In einer metrischen Zeitstruktur kann man explizit Aussagen über den Abstand zweier Zeitpunkte treffen.

Ganz konkret wird die PL1 um eine Zeitstruktur $(\mathcal{T}, t_0, \prec)$ und um temporale Operatoren ($\circ$, $\bullet$, $\Box$, $\Diamond$, $\mathcal{S}_S$, $\mathcal{U}_S$) zur metrisch-temporalen Logik (MTL) erweitert. $\mathcal{T}$ ist eine Menge von beliebigen Zeitpunkten, t_0 ist der Bezugszeitpunkt und $\prec$ ist eine Ordungsrelation auf $\mathcal{T}$. Der Operator $\circ$ bezeichnet den nächsten und der Operator $\bullet$ den vorherigen Zeitpunkt. Der temporale Alloperator $\Box$ sagt, dass $\Box\mathcal{F}$ mit $\mathcal{F} \in \mathcal{F}_\Sigma(\mathcal{V})$ immer gültig ist. $\Box_S$ analog, aber nur immer in der Zeitpunktemenge $S \subseteq \mathbb{Z}$. Der temporale Existenzoperator $\Diamond$ sagt, dass $\Diamond\mathcal{F}$ mit $\mathcal{F} \in \mathcal{F}_\Sigma(\mathcal{V})$ irgendwann gültig ist. $\Diamond_S$ analog, aber nur irgendwann in der Zeitpunktemenge $S \subseteq \mathbb{Z}$. Der Operator ‘since’ $\mathcal{F}_1\mathcal{S}_S\mathcal{F}_2$ steht für ‘$\mathcal{F}_1$ immer seit $\mathcal{F}_2$ innerhalb von S’ und der Operator ‘until’ $\mathcal{F}_1\mathcal{U}_S\mathcal{F}_2$ für ‘$\mathcal{F}_1$ immer bis $\mathcal{F}_2$ innerhalb von S’ analog.

3.1.3 Einschränkung auf das Horn-Fragment

Die Entscheidbarkeit der Erfüllbarkeit der Gleichungssysteme von variablen Zeitpunkten muss in der Praxis gegeben sein. Die Presburger-Arithmetik ist zwar NP-vollständig, aber entscheidbar [Enderton, 2001]. Die Komplexität des Entscheidungsproblems der Presburger-Arithmetik ist mindestens doppelt exponentiell und maximal dreifach exponentiell

[Oppen, 1978]. Zur Gewährung der Praxistauglichkeit sollte das Problem mit maximal polynomialen Aufwand gelöst werden können. Schränkt man die logischen Formeln auf die Menge der Formeln in Hornform ein, dann ist der Aufwand polynomial, weil HORNSAT P-vollständig ist [Buning et al., 1995]. Eine Hornformel ist eine konjunktive Normalform von Hornklauseln. Eine Hornklausel ist ein Disjunktionsterm aus Literalen mit maximal einem positiven Literal. Ein Literal ist eine atomare Formel oder die Negation einer atomaren Formel:

$$(A \vee \neg B_1 \vee \ldots \vee \neg B_n) \text{ in PL1 log. äquivalent zu } (A \leftarrow B_1 \wedge \ldots \wedge B_n) \quad (3.1)$$

$$(A \qquad\qquad\qquad\qquad) \text{ in PL1 log. äquivalent zu } (A \leftarrow \qquad\qquad 1) \quad (3.2)$$

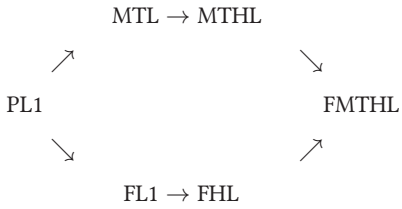
$$(\qquad \neg B_1 \vee \ldots \vee \neg B_n) \text{ in PL1 log. äquivalent zu } (\bot \leftarrow B_1 \wedge \ldots \wedge B_n) \quad (3.3)$$

Die Menge der definiten Hornklauseln (genau ein positives Literal) – siehe Gleichung (3.1) - (3.2) – enthält *Regeln* $\widetilde{\forall}(A \leftarrow B_1 \wedge \ldots \wedge B_n)$ und *Fakten* $\widetilde{\forall} A$. Die Menge der indefiniten Hornklauseln (kein positives Literal) – siehe Gleichung (3.3) – besteht aus *Anfragen* der Form $\widetilde{\exists}(B_1 \wedge \ldots \wedge B_n)$.

Im aussagenlogischen Fall bewirkt die Einschränkung auf Hornformeln, dass in linearer Anzahl von Schritten entscheidbar ist, ob ein Atom aus einer Wissensbasis folgt. Bei der PL1 ist das Erfüllbarkeitsproblem lediglich semi-entscheidbar, wenn man sich auf das Hornfragment einschränkt [Ebbinghaus et al., 2007]. Es ist ein Leichtes, Hornklauseln anzugeben, die zu unendlichen Auswertungssequenzen führen können, wie dieses Beispiel aus der Arithmetik: $\forall x \forall y \; LessThan(succ(x), y) \leftarrow LessThan(x, y)$. Für dieses Beispiel ist eine mögliche unendliche Auswertungssequenz z.B. $(x = 0, y = 0), (x = 1, y = 0), (x = 2, y = 0), \ldots$ Es obliegt also der Verantwortung des Benutzers, unendliche Lösungszweige zu vermeiden, was eben durch die Einschränkung auf das Hornfragment deutlich erleichtert wird, siehe [Brachman and Levesque, 2004].

3.1.4 Zusammenführung zur FMTHL

Durch Zusammenführung der metrisch-temporalen Hornlogik (MTHL) und der unscharfen Hornlogik entsteht die FMTHL.



Zusätzlich ist es jetzt auch möglich, den temporalen Alloperator zeitlich unscharf zu definieren. Der Häufigkeitsoperator $\Box_S^\theta \mathcal{F}$ hat die Bedeutung, dass im Intervall S die Aussage $\mathcal{F}$ mit mindestens dem in $\theta \in [0..1]$ gegebenen Teil wahr ist: „mindestens $100\% \cdot \theta$ oft innerhalb von S gilt $\mathcal{F}$“. Mit diesem Konzept lassen sich dann sehr elegant Begriffe wie *selten*, *oft*, *fast immer*, usw. abbilden.

Die Verwendung von Hornklauseln bietet beispielsweise auch die deklarative Programmiersprache PROLOG. Ähnlich zu PROLOG wurde das Inferenzsystem F-LIMETTE (Fuzzy Logic Programming Integrating Metric Temporal Extensions) geschaffen, das die FMTHL mit u.a. einem Tableaukalkül realisiert, [Schäfer, 1996]. Dabei liegt die Zeitstruktur $(\mathbb{Z}, 0_\mathbb{Z}, <_\mathbb{Z})$ zugrunde.

3.2 Situationsgraphenbäume

Wünschenswert wäre es, in Anfragen variable Zeiten zu verwenden, also $\Box_x \mathcal{F}$ oder $\Diamond_x \mathcal{F}$ für eine Variable x. Damit ließen sich dann Anfragen realisieren, die Belegungen von $\mathcal{F}$ und gleichzeitig aller Zeitpunkte x zurückliefern würden. Leider gibt es diese Operatoren offensichtlich nicht. Eine andere Möglichkeit zur gezielten Betrachtung einzelner Zeitabschnitte in

der Situationsanalyse ist ein SGT [Arens, 2004], der in diesem Abschnitt inkrementell erläutert wird.

Ein Situationsgraphenbaum repräsentiert das Wissen über die zu erwartenden Verhalten von Agenten. Das Wissen in Situationsgraphenbäumen ist i.d.R. für die Anwendung in (bildverarbeitenden) technischen Systemen optimiert [Arens and Nagel, 2003]. Die syntaktische Struktur eines SGT ist ein gerichteter Hypergraph oder eine reguläre Grammatik.

Im Folgenden wird auf die inkrementelle Definition eines SGT und dessen Teilkomponenten eingegangen.

3.2.1 Situationsschema

Bereits in der Arbeit [Nagel, 1988] wird die „generisch beschriebene Situation“ zur Modellierung von Verhalten eingeführt. Der Zustand eines Agenten in seiner Umwelt wird durch ein Situationsschema zusammen mit den Handlungsmöglichkeiten des Agenten modelliert. In Abbildung 3.1 ist exemplarisch die Visualisierung eines Situationsschemas dargestellt.

Situationsbezeichner Jedes Situationsschema besitzt einen eindeutigen Namen, den Situationsbezeichner.

Zustandsschema Damit sich ein Agent in einer von diesem Situationsschema beschriebenen Situation befinden kann, müssen bestimmte Bedingungen erfüllt sein. Das Zustandsschema enthält Bedingungen in Form von logischen Prädikaten. Können alle diese Zustandsprädikate mit einer konkreten Variablenbelegung ausgeprägt werden, dann ist das Zustandsschema erfülllt. Je nach Strategie sind auch mehrere Instanzen eines Situationsschemas möglich.

Handlungsschema Die Handlungsmöglichkeiten eines Agenten werden durch die Handlungsprädikate aufgezeigt. Die Variablen sind i.d.R. nicht frei wählbar, sondern abhängig vom aktuellen Zustand des Agenten.

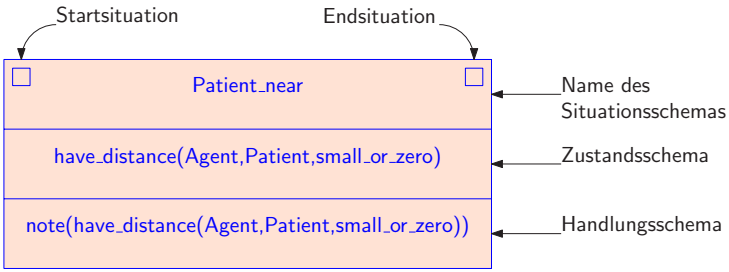


Abbildung 3.1: Ein Situationsschema besitzt einen eindeutigen Namen, den Situationsbezeichner, ein Zustandsschema mit den Vorbedingungen und ein Handlungsschema mit den Nachbedingungen.

3.2.2 Situationsgraph

Das Situationsschema betrachtet lediglich den Zustand eines Agenten zu einem einzigen Zeitpunkt. Hier werden durch temporale Kanten verbundene Situationsschemata als Situationsgraphen bezeichnet, vergleiche [Krüger, 1991]. In Abbildung 3.2 ist ein Situationsgraph schematisch visualisiert.

Prädiktionskante Der Übergang von einem in das zeitlich nächste Situationsschema wird durch Prädiktionskanten realisiert. Selbstprädiktionskanten sind ausdrücklich zulässig, da sie maßgeblich zur Invarianz gegenüber der zeitlichen Abfolge der einzelnen ausprägbaren Situationsschemata beitragen. In früheren Arbeiten wie [Arens, 2004] sind die Prädiktionskanten geordnet. In dieser Arbeit können durch die gleichzeitige Verfolgung aller möglichen Pfade bei der SGT-Traversierung diese Priorisierungen aufgehoben werden.

Bindungsschema Wird von einem ausgeprägten Situationsschema entlang einer Prädiktionskante in ein nachfolgendes Situationschema gewechselt, dann bleiben einmal belegte Variablen des bisherigen Zustandsschemas erhalten. Der Gültigkeitsbereich dieser Variablen ist für jeden Agenten

global. Soll die Möglichkeit bestehen, Variablen beim Übergang in nachfolgende Situationsschemata frei zu geben, dann bietet das Bindungsschma einer Prädiktionskante die explizite Freigabe von Variablen an. Gleichzeitig kann dort auch eine Variable explizit belegt werden. Besonders bei Listenvariablen kommen Bindungsschemata zum Einsatz, um im nachfolgenden Zeitschritt Änderungen der Elemente der Liste berücksichtigen zu können.

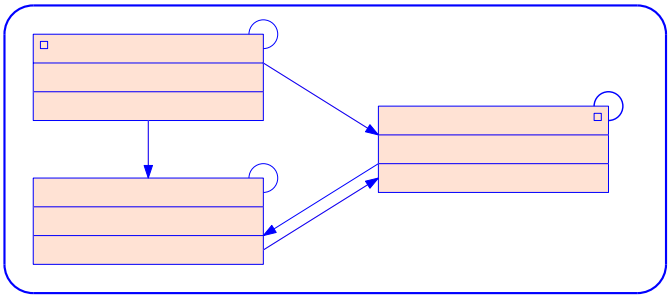


Abbildung 3.2: In einem Situationsgraph können die Situationsschemata mit Prädiktionskanten mit sich selbst und untereinander verbunden sein. Den Prädiktionskanten können Bindungsschemata anhaften.

Start- und Endeigenschaft Bei einer zeitlichen Abfolge von Situationsschemata gibt es ausgezeichnete Situationsschemata mit Starteigenschaft und analog mit Endeigenschaft. Innerhalb eines Situationsgraphen darf es mehrere Situationsschemata mit Start- oder Endeigenschaft sowie Start- und Endeigenschaft geben. Mindestens einmal muss im Situationsgraph eine Start- und Endeigenschaft vorhanden sein.

Ablauf Die Abfolgen in Situationsschemata beginnend mit Starteigenschaft über Prädiktionskanten bis hin zu Situationsschemata mit Endeigenschaft werden als SGT-Ablauf bezeichnet. In Anhang B Definition 2 ist die formale Definition für einen SGT-Ablauf gegeben. Da der Ausgangsgrad

eines Situationsschemas nicht beschränkt ist und durch reflexive Kanten, kann ein Situationsgraph beliebig viele SGT-Abläufe darstellen.

3.2.3 Situationsgraphenbaum

Verschiedene Situationsschemata alleine stehen lediglich zeitlich zueinander in Beziehung. Wie schon in Abschnitt 1.3 erläutert, erfordert die Wissensmodellierung auch Beziehungen zwischen allgemeineren und spezielleren Schemata. Daher wurden in [Krüger, 1991, Arens, 2004] Situationsgraphenbäume eingeführt. In Abbildung 3.3 ist ein SGT skizziert, um das ungefähre grafische Erscheinungsbild zu verdeutlichen.

Verhaltensschema Das Wissen, welches ein SGT repräsentiert, wird als Verhaltensschema bezeichnet. Konkret ist das die Menge aller maximalen und nicht-maximalen SGT-Verhalten (siehe weiter unten).

Wurzelgraph Der Situationsgraph, der kein Situationsschema detailliert, ist der allgemeinste Situationsgraph in diesem Situationsgraphenbaum. Motiviert durch die Baumstruktur wird er deshalb Wurzelgraph genannt.

Detaillierung Eine Detaillierungskante verbindet ein Situationsschema mit einem Situationsgraphen, und zwar immer in die Richtung vom Allgemeinen zum Speziellen, sodass der Situationsgraphenbaum ein Baum bleibt, und kein zyklischer Graph wird. Wird bei einem SGT-Ablauf ein Situationsschema mit einer Detaillierungskante zu einem Situationsgraphen verfeinert, nennt man diesen SGT-Ablauf einen detaillierenden SGT-Ablauf. Ein detaillierender SGT-Ablauf kann also ein allgemeineres Situationsschema **zeitlich verfeinern**. Die Zustandsschemata der allgemeineren Situationsschemata sind dabei weiterhin erfüllend zu belegen. Beim Handlungsschema kann es erforderlich sein, sich nur auf das speziellste zu beschränken und allgemeinere Handlungen eben nicht auszuführen. Analog

zur zeitlichen Verfeinerung kann mit der gleichen Art von Detaillierungskanten auch **terminologisch spezialisiert** werden, indem beispielsweise weitere konkretere Vorbedingungen gefordert werden.

SGT-Verhalten Ein SGT-Ablauf im Wurzelgraphen ist ein SGT-Verhalten. Weitere SGT-Verhalten können daraus konstruiert werden, indem detaillierende SGT-Abläufe einzelne Situationen ersetzen. Siehe auch Anhang B Definition 1 für eine formale Definiton eines SGT-Verhaltens.

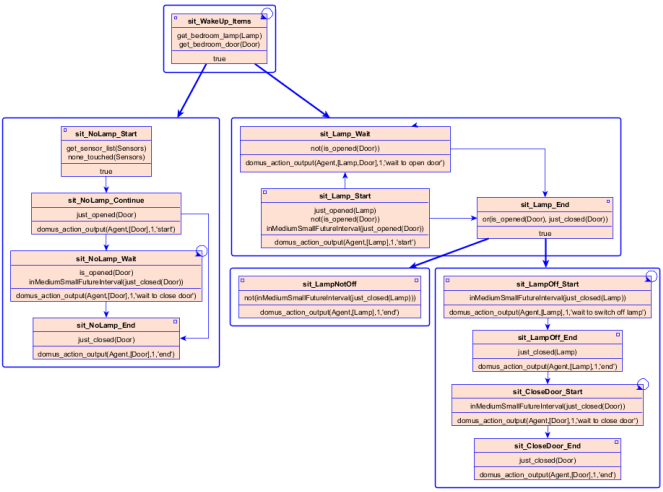


Abbildung 3.3: Exemplarischer SGT zur Visualisierung seiner Struktur.

3.2.4 Syntaktische Struktur eines SGTs

Ein SGT ist ein Hypergraph. In [Arens, 2004] wird gezeigt, dass ein SGT auch durch eine kontextfreie Grammatik ausgedrückt werden kann. In der vorliegenden Arbeit wird in Anhang B nachgewiesen, dass die Struktur eines SGT nicht nur kontextfrei, sondern sogar regulär ist. Dabei wird eine Konstruktionsvorschrift angegeben und bewiesen, dass aus einem Verhal-

tensschema ein äquivalenter endlicher Automat erzeugt werden kann und umgekehrt. Es existiert eine reguläre Sprache, die das Wissen, das ein SGT modelliert, ausdrückt.

Bisher war bekannt, dass das Wort-, Leerheits- und Endlichkeitsproblem der syntaktischen Struktur von SGTs entscheidbar ist. Mit dem Beweis der Regularität ist jetzt auch das Äquivalenz- und Inklusionsproblem entscheidbar. Damit kann man zweifelsfrei entscheiden, ob zwei SGTs äquivalent sind oder ein SGT einen anderen SGT bereits beinhaltet. Ein SGT ist bzgl. Vereinigung, Schnitt, Komplement, Konkatenation, Kleene-Stern und der Differenz abgeschlossen.

3.2.5 SGTyEditor

Das Verhaltenswissen aus der Verhaltens-Repräsentations-Ebene wird in Form von Situationsgraphenbäumen realisiert. Bereits Krüger [Krüger, 1991] entwickelte ein Werkzeug um textuell erstellte SGTs zu visualisieren. Ein neuer SGTEditor mit dem SGTs erstellt, visualisiert und geändert werden können wurde in [Arens and Nagel, 2003] vorgestellt. Dieser SGTEditor basiert auf dem unter GNU General Public License, Version 1.0 stehenden DiaGen Diagrammeditor [Minas and Viehstaedt, 1995]. In diesem Abschnitt wird ein neuer Editor für SGTs vorgestellt, nämlich der SGTyEditor, siehe Abbildung 3.4 und auch [Münch, 2015]. yFiles for Java™ ist die technische Basis für den SGTyEditor. „*yFiles for Java ist eine umfangreiche Java™ Klassenbibliothek, die Algorithmen und Komponenten für die Analyse, die Visualisierung und das automatische Anordnen von Graphen, Diagrammen und Netzwerken zur Verfügung stellt.*“[6].

yFiles for Java™ bietet eine umfangreiche und mächtige Layoutgestaltung, die auch große SGTs handhabbar macht. Die wesentlichen und interessanten Änderungen sind beim SGTyEditor (a) die interne Repräsentation

[6] https://www.yworks.com/de/products/yfiles-for-java-2.x (19.03.2017)

von Wissen, (b) eine instantane Validierung der syntaktischen Struktur des SGTs und (c) die Trennung von Wissensrepräsentation und Inferenz.

Im Zuge des Semantic Web etablieren sich Wissensbasen in Form von Ontologien. In [Bauer, 2012] wurde gezeigt, dass ein SGT in eine Ontologie abgebildet werden kann. In Abbildung 3.5 ist die Ontologie (links) und eine Übersicht über ihre Eigenschaften (rechts) dargestellt. Der Sprache OWL, mit der die Ontologie beschrieben wird, liegt die Beschreibungslogik $\mathcal{SHOIN}(\mathcal{D})$ zugrunde, was zu einer entscheidbaren Untermenge der PL1 äquivalent ist [Baader et al., 2010]. Die interne Repräsentation wurde daher in OWL realisiert.

Um Graphen persistent zu gestalten wird die XML-basierte Sprache GraphML eingesetzt. Um im SGTyEditor SGTs abzuspeichern, werden diese im GraphML Format abgelegt. Darin ist neben der Struktur auch das aktuelle Layout ausgeprägt.

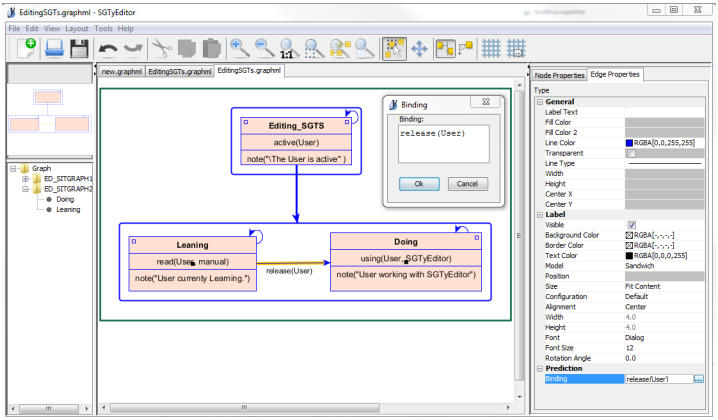


Abbildung 3.4: Grafische Benutzeroberfläche vom SGTyEditor. Gut zu erkennen ist der geteilte Arbeitsbereich: Links die visuelle und strukturelle Übersicht für ein schnelles Navigieren, in der Mitte den stark vergrößerten Arbeitsausschnitt und rechts die Eigenschaften; alternativ auch als Popupbox wie in Abbildung 3.6 (links).

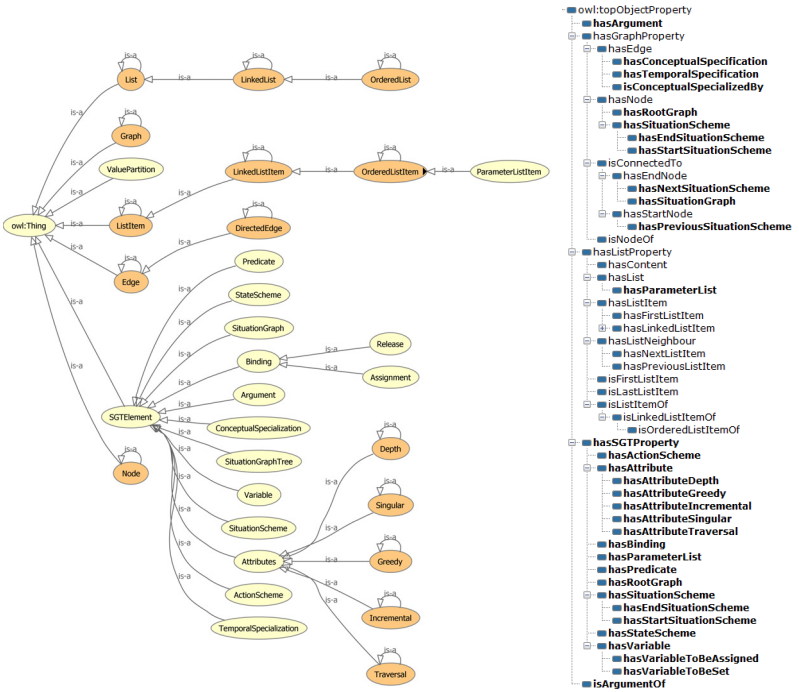


Abbildung 3.5: Ontologie für einen SGT (links) und die entsprechenden Eigenschaften dazu (rechts).

Eine weitere Neuerung bietet die instantane Validierung der syntaktischen Struktur des generierten SGTs. Bei jeder Änderung am SGT wird eine Validierung angestoßen die alle syntaktischen Rahmenbedingungen prüft und bei Fehlschlägen eine entsprechende Fehlermeldung in der grafischen Oberfläche anzeigt, siehe Abbildung 3.6 (rechts). Diese Neuerung ist eine große Erleichterung für den Experten, der Hintergrundwissen in SGTs erstellt, weil ab sofort während der Eingabe auf syntaktische Fehler hingewiesen wird und diese damit vermieden werden können, weil sie auch konkret genannt werden.

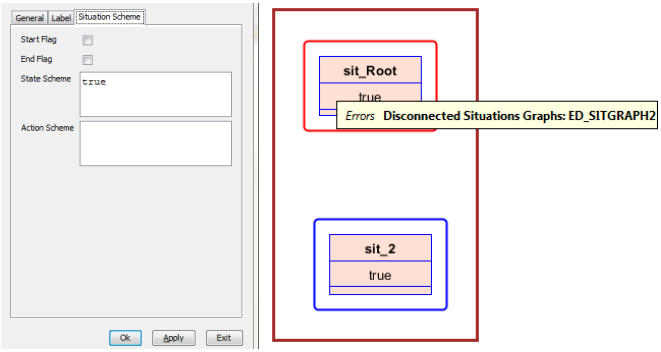


Abbildung 3.6: Ein dediziertes Fenster um Situationsschemata zu modellieren (links). Instantane Validierung der SGT Struktur zeigt Fehler (rechts).

3.3 Situationsanalyse mit SGT-Traversierung

Die Situationsanalyse benutzt Hintergrundwissen in Form von SGTs. Ein SGT repräsentiert das Verhaltenswissen. Um die ausprägbaren Verhalten zu identifizieren müssen diese Verhalten im SGT gefunden werden. Diese Verhaltenserkennung wird durch die Situationsgraphenbaumtraversierung realisiert. Situationsanalyse ist die SGT-Traversierung für alle Agenten zu jedem Zeitpunkt.

Die Situationsgraphenbaumtraversierung läuft wie folgt ab: Für einen konkreten Agenten wird versucht, im Wurzelsituationsgraph alle Situationsschema mit Starteigenschaft auszuprägen. Eine erfolgreiche Ausprägung ist dann gegeben, wenn für das Zustandsschema eine erfüllende Belegung gefunden werden kann. Kann ein Situationsschema ausgeprägt werden, dann wird versucht, es möglichst detailreich auszuprägen. D.h. es wird nach terminologischen Detaillierungskanten gesucht und dann in diesem Situationsgraphen versucht, alle Situationsschema mit Starteigenschaft auszuprägen, bis keine weitere Detaillierung mehr möglich ist.

In diesem Schritt wird also das detaillierteste Situationsschema ermittelt. Algorithmus 1 bildet dieses Verhalten in Zeile 1-8 ab.

Im nächsten Zeitpunkt wird versucht von allen vorher erfolgreich ausgeprägten Situationsschemata alle durch Prädiktionskanten erreichbaren Situationsschemata auszuprägen. In [Arens, 2004] ist die Reihenfolge, in der diese Prädiktionskanten behandelt werden, noch geordnet, siehe Algorithmus 1 Zeile 12. In dieser Arbeit wird diese Priorisierung aufgehoben, siehe Abschnitt 4.2.

Algorithmus 1 : SGT-Traversierung (bisher)

```
   Input : SGT, agent
1  if agent occurs for the first time then
2      G ⟵ SGT root graph;
3      forall s|s ∈ G ∧ s is start situation do
4          if s can be instantiated then
5              forall spec|spec ∈ s ∧ spec is specialization do
6                  s := spec, G := graph containing spec;
7                  start recursion goto line 3;
8              evaluate actions of s;
9  else
10     predOrder = 1;
11     repeat
12         predOrder^th prediction situation predSit of s (the last
             situation of the already known agent);
13         if instantiate predSit successful then
14             s := predSit, G := graph containing predSit;
15             start recursion goto line 5;
16         if predSit is instantiated end situation ∧ predSit ∈ G
             then
17             instantiation successful; return
18         else instantiation failed; predOrder + +;
19     until;
```

Die Traversierung in einem Situationsgraphen ist dann beendet, wenn das erste Situationsschema mit Endeigenschaft erreicht wurde oder das aktuelle Situationsschema nicht ausgeprägt werden kann. In beiden Fällen wird sich entgegen der Detaillierungskante in den nächst abstrakteren Situationsgraph bewegt, was in Algorithmus 1 durch die Rekursion geleistet wird.

Der SGT-Traversierungsalgorithmus ist in FMTHL realisiert und die Traversierung für einen Agenten zu einem Zeitpunkt ist eine einfache Anfrage im Logikprogramm.

Bereits Schäfer [Schäfer, 1996, Seite 190f.] nennt sechs Eigenschaften von Traversierungsstrategien ohne alle vollständig zu realisieren:

- **Absicht der Handlungsmodellierung** Wird die Handlungsmodellierung für ein beobachtendes System eingesetzt oder als auszuführende Stellbefehle eines agierenden Systems?
- **Strategie der Pfadverfolgung** Nach welchen Kriterien wird der Pfad durch den SGT ausgewählt? Dürfen und sollen mehrere Pfade beschritten werden?
- **Anzahl der ausprägbaren Situationen** Welcher Pfad ist zu nehmen, wenn Situationen durch Spezialisierung und Prädiktion verschiedene Pfade ermöglichen? Die möglichst spezifischste Situation oder die allgemeinere?
- **Anzahl der Ausprägungen eines Situationsschemas** Kann sogar ein Situationsschema mehrfach durch verschiedene konkrete Ausprägungen instantiiert werden?
- **Rolle einer Situation im Situationsgraphen** Wird die Spezialisierung verworfen wenn kurzzeitig in eine generellere Situation ausgewichen wurde?

- **Inkrementalität von Handlungen** Wie ist mit Handlungen zu verfahren? Wird nur das Handlungsschema der aktuellen Situation ausgeprägt oder auch die Handlungsschemata der generellereren Situationen?

In dieser Arbeit wird auf die Traversierungsstrategien eingegangen. Im Besonderen wird die Traversierung dahingehend erweitert, dass ein Situationsschema mehrfach ausgeprägt werden kann (siehe Abschnitt 4.1.4), dass mehrere Pfade beschritten werden dürfen (siehe Abschnitt 4.2) und eine kurzzeitige Nichtausprägbarkeit mit Rücksprung zur allgemeinereren Situation umgangen werden kann (zeitliche Lücken und kontrollierte Halluzination, siehe Kapitel 5).

4 Repräsentation von Unschärfe

Wissen ist die Zuordnung von *Werten* zu *Aussagen* [Görz et al., 2013]. Je nach Kalkül können die Begriffe *Werte* und *Aussagen* u.a. durch *Wahrscheinlichkeit von Ereignissen* oder *Wahrheitswert von Zuständen* quasisynonym ersetzt werden.

Impräzision, Unsicherheit und Vagheit sind drei Eigenschaften, die dem Wissen anhaften [Görz et al., 2013]. Präzise Aussagen geben nur einen Wert einer Eigenschaft an, wohingegen impräzise Aussagen eine Menge von Alternativen einer Eigenschaft angeben. Impräzision lässt sich bereits durch die klassische symbolische Logik behandeln. Die Unsicherheit und Vagheit erfordert eine gesonderte Behandlung. Beides zusammen stellen wir unter den Begriff Unschärfe.

4.1 Unschärfe

4.1.1 Vagheit

Begriffe der Sprache sind i.d.R. vage. Extreme lassen sich gut der einen oder der anderen Klasse zuordnen, aber dazwischen sind die Übergänge fließend und werden von jedem Beteiligten und den Umständen entsprechend subjektiv anders ausgelegt. In [Schäfer, 1996] wird Vagheit im Kontext der Situationsanalyse als Begriffsunschärfe erklärt, die durch die subjektive Varianz der Interpretation der Begriffe hervorgerufen wird. Das Sorites-

Paradoxon[7] bringt deutlich zum Ausdruck, dass manche Definitionen nicht klar scharfe Grenzen ziehen können. Nach [Weisbrod, 1996] ist Vagheit der bewusste oder unbewusste Verzicht auf Präzision. Der entscheidende Unterschied zur Unsicherheit besteht darin, dass vage Aussagen nicht auf Unwissen beruhen; sie sind auch nicht von minderer Qualität oder gar nutzlos [Wittgenstein, 2003]. In [Zimmermann, 2001] wird die Vagheit als etwas, das die Bedeutung der semantischen Beschreibung von Aussagen selbst betrifft, der Unsicherheit gegenübergestellt, die den unbekannten zukünftigen Zustand eines wohldefinierten Systems betrifft. Das verwendete Kalkül muss also mit der Vagheit sprachlicher Begriffe umgehen können.

4.1.2 Behandlung von Vagheit

Mit dem Ursprung der Vagheit der menschlichen Sprache entstammend, wird im Folgenden $\widetilde{A}$ als linguistischer Term, $x\ is\ \widetilde{A}$ als Prädikat und x als Variable bezeichnet [Weisbrod, 1996]. Als Konzept zur Repräsention von Vagheit eignen sich unscharfe Mengen (Fuzzy-Mengen). Ein Element charakterisiert seine Zugehörigkeit zu einer Unscharfen Menge mit der charakteristischen Funktion $\mu_A : \mathcal{U} \rightarrow [0, 1]$ Die Operationen Schnitt, Vereinigung und Komplement werden analog zu den Mengenoperatoren definiert. Lediglich die Zugehörigkeitsfunktion muss angegeben werden. Die Bedingungen für Schnitt und Vereinigung umfassen Assoziativität, Kommutativität, ein neutrales Element und Monotonie. Hierfür bieten sich Funktionen an, die eine t-Norm respektive co-t-Norm erfüllen. Beim Komplement muss die Grenzbedingung und Monotonie erfüllt sein. Generell ist bei t-Normen das neutrale Element Eins und es existiert die obere Schranke $t(a, b) = \min\{a, b\}$, die die möglichen Zugehörigkeitsfunktionen einrahmen; dabei ist t eine t-Norm. Bei co-t-Normen verhält es sich komplemen-

[7] Wie viele Sandkörner bilden einen Haufen? http://plato.stanford.edu/entries/sorites-paradox/ (19.03.2017)

tär zu t-Normen; das neutrale Element ist Null und die untere Schranke ist $co - t(a, b) = \max\{a, b\}$.

Zur Schlussfolgerung mit dem verallgemeinerten Modus Ponens in Gleichung (4.1) muss noch die Komposition $\widetilde{Q} \circ \widetilde{R}$ definiert werden. Dabei sind $\widetilde{Q} \in \widetilde{\mathcal{P}}(\mathcal{U} \times \mathcal{V})$ und $\widetilde{R} \in \widetilde{\mathcal{P}}(\mathcal{V} \times \mathcal{W})$ unscharfe Relationen und $\widetilde{Q} \circ \widetilde{R} \in \widetilde{\mathcal{P}}(\mathcal{U} \times \mathcal{W})$. Die Zugehörigkeitsfunktion lautet nach [Weisbrod, 1996]: $\forall (u, v) \in \mathcal{U} \times \mathcal{W} : \mu_{Q \circ R}(u, v) = \underset{v \in \mathcal{V}}{co} - t(t(\mu_Q(u, v), \mu_R(v, w)))$

$$\begin{array}{rl} (x, y)\ is\ \widetilde{\mathcal{R}} & (Regel) \\ x\ is\ \widetilde{A'} & (Fakt) \\ \hline y\ is\ (\widetilde{A'} \circ \widetilde{\mathcal{R}}) & (Schlu\beta) \end{array} \tag{4.1}$$

Die in Abschnitt 3.1.1 vorgestellte unscharfe Logik wird so verwendet, dass sie die in diesem Abschnitt genannten Eigenschaften abbilden kann. Sie geht sogar darüber hinaus und bietet die Praxis betreffend zahlreiche Helfer an, wie z.B. einen Mindestgrad μ.

4.1.3 Unsicherheit

Bei einer sicheren Aussage besteht kein Zweifel an ihrer Gültigkeit. In [Weisbrod, 1996] wird Unsicherheit als das Eingeständnis von fehlendem Wissen beschrieben. In [Zimmermann, 2001] ist die Definition ausführlicher: Unsicherheit drückt aus, dass ein Agent in einer bestimmten Situation nicht über die nötigen Informationen verfügt, die quantitativ und qualitativ erforderlich sind, um deterministisch sein Verhalten oder andere seiner Eigenschaften beschreiben, vorhersagen oder vorschreiben zu können. Er unterscheidet zwischen der Ursache der Unsicherheit, dem Typ der verfügbaren Information, dem Skalierungsniveau numerischer Information und dem Typ der Ausgabeinformation:

Was sind die Gründe für Unsicherheit?

- **Mangel an Information:**
  - quantitativ: Keine Information ist vorhanden.
  - qualitativ: Nur die Wahrscheinlichkeit für das Eintreten eines Zustandes ist bekannt.
  - approximativ: Es wird bewusst auf (zu genaue) Information verzichtet.
- **Überfluss an Information:** Liegt mehr Information als verarbeitet werden kann vor, dann muss zwangsweise gefiltert werden.
- **Widersprüchliche Anhaltspunkte:** Es liegen mindestens zwei Informationen vor, die sich widersprechen; die falsche kann aber nicht lokalisiert werden.
- **Mehrdeutigkeit:** Begrifflichkeiten können in verschiedenen Kontexten unterschiedliche Bedeutungen tragen.
- **Messungenauigkeit:** Unzulänglichkeiten der Sensoren und theoretische Bandbegrenzungen.
- **Überzeugung:** Ein Beobachter erlangt eine subjektive Information durch Überzeugung über objektive Daten.

Zum Einen Unsicherheit und zum Andern Vagheit sind beides Eigenschaften, die bei der Situationsanalyse zur Unschärfe beitragen. Eine Trennung beider ist in der Praxis oft schwer möglich. So kann die Information „ein Kind“ vage sein und durch den Personendetektor bereits Unsicherheit enthalten.

4.1.4 Repräsentation von Unschärfe durch den gesamten Inferenzprozess

Der Umgang mit Vagheit in der Begrifflich-Primitiven-Ebene wie es Abbildung 4.1 visualisiert ist schon immer ein integraler Bestandteil der auf FMTHL und SGTs basierender Situationsanalyse gewesen [Schäfer, 1996]. Dieser Schritt trägt dazu bei, die Semantische Lücke zu schließen, also die quantitativen Zahlen auf semantische Information abzubilden. Dies wird durch Prädikate mit Wahrheitswert realisiert.

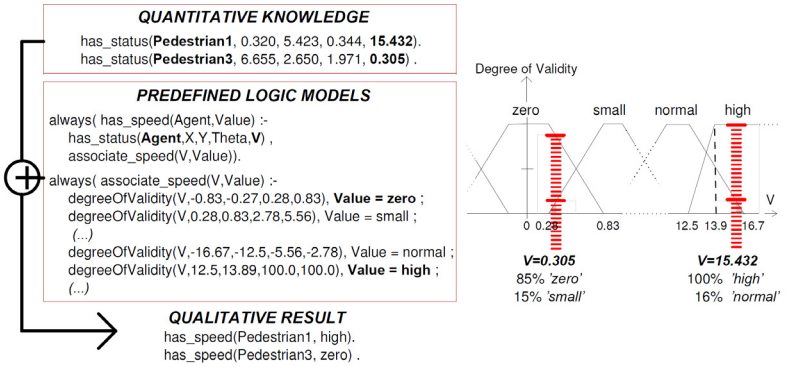


Abbildung 4.1: Abbildung der quantitativen Eingabedaten auf begriffliche Beschreibungen (links). Grafische Visualisierung davon (rechts). Abbildung aus [Fernández Tena, 2010].

Beispielhaft folgt ein Auszug aus dem PETS 2009 Datensatz, siehe auch Abschnitt 7.3.3. Dieser Auszug verdeutlicht den bisherigen Umgang mit Unsicherheit, die lediglich aus der Vagheit der Begrifflichkeiten resultiert. Ab dem Zeitpunkt 39 befinden sich Agent_2 und Agent_0 immer mehr in der Situation *InGroup*. Der Zusicherungsgrad, der jeder erkannten Situation anhaftet resultiert aus den Begrifflichkeiten, die zur Definition einer *InGroup* Situation herangezogen werden; konkret ist das der räumliche Abstand *have_distance(Agent_2,Agent_0,small)* der einzelnen Objekte.

Verändert sich der räumliche Abstand beider Agenten negativ, dann trifft das Konzept *small* immer mehr zu, die beiden Agenten sind dann immer mehr *small* zueinander, was den steigenden Zusicherungsgrad von Zeitpunkt 39 bis 44 zur Folge hat, bis ab Zeitpunkt 45 beide Agenten einen deutlichen Abstand *small* zueinander haben.

```
[0.00 | 38 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup]
0.14 | 39 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.26 | 40 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.42 | 41 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.59 | 42 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.76 | 43 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.89 | 44 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
1.00 | 45 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
1.00 | 46 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
[...]
0.31 | 68 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.34 | 69 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.62 | 70 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.98 | 71 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
1.00 | 72 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
1.00 | 73 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
[...]
```

In [IJsselmuiden et al., 2012, IJsselmuiden, 2014] wurden erste Ansätze zur Betrachtung von Unsicherheit vorgestellt. Mit Unsicherheit behaftete Aussagen haben ihren Ursprung im Quantitativen Teilsystem. Einer Aussage wird z.B. vom Personentracker ein normierter Konfidenzwert zugewiesen. Dieser Konfidenzwert wird dem Prädikat als Wahrheitswert übertragen. Die weitere Behandlung stützt sich jetzt auf Varianten der in Tabelle 3.1 genannten Verknüpfungen. Es werden nun durch den Wahrheitswert die Unschärfe, also die Vagheit und die Unsicherheit ausgedrückt. Bei der Auswertung des Zustandsschemas eines Situationsschemas wird diese Unschärfe explizit betrachtet und als Wahrheitswert dem Situationsschema angefügt.

Der selbe Auszug aus dem PETS 2009 Datensatz – jetzt mit Betrachtung der Unsicherheit der detektierten Personen: Es ist deutlich zu sehen, dass der Wahrheitswert der erkannten Situationen verringert wird. Das ist richtig und erwartet, weil es die tatsächliche Interpretation der Szene näher an

der Realität abbildet, als im o.g. Beispiel, wo die Ergebnisse des Personendetektors mit einem Konfidenzwert größer eines Schwellwerts als absolut richtig (1.00) in das System zugeführt wurden. Hätte der Personendetektor im o.g. Beispiel mehr falsch positive Detektionen, müsste der Schwellwert angehoben werden und viele – auch richtige, aber schwache – Detektionen würden zurückgewiesen werden.

```
[0.00 | 38 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup]
0.10 | 39 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.18 | 40 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.32 | 41 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.45 | 42 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.66 | 43 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.69 | 44 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.61 | 45 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.57 | 46 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
[...]
0.12 | 68 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.25 | 69 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.41 | 70 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.73 | 71 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.30 | 72 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.41 | 73 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
[...]
```

In der Behandlung der Unschärfe bis zur höchsten Ebene, den ausgeprägten Situationsschemata, liegt der Grundstein zur Betrachtung mehrerer gleichzeitig valider Hypothesen.

4.2 Multihypothesen

Bei der Beobachtung einer Szene kann mit den Situationen anhaftenden Wahrheitswerten eine Aussage dazu getroffen werden, bis zu welchem Grad diese Situationen erkannt wurden. Bisher wird, wie in Abschnitt 3.3 beschrieben, der beste mögliche Weg durch den SGT bei der SGT-Traversierung gewählt – entsprechend einer gierigen Tiefensuche. Infolgedessen ist zu der erkannten Situation ein Wahrheitswert verfügbar. Wird

die SGT-Traversierung aus Algorithmus 2 dahingehend geändert, dass alternative Pfade gleichzeitig begangen werden können und sollen – was einer Breitensuche entspricht – dann besteht die Möglichkeit, dass zu einem Zeitpunkt für einen Agenten mehrere Situationsschemata ausprägbar sind. Weiterhin können von ein und demselben Situationsschema auch mehrere Ausprägungen existieren. Das führt zu dem neuen SGT-Traversierungsverfahren in Algorithmus 2. Durch den nun vorhandenen Wahrheitswert jeder ausgeprägten Situation können die zu einem Zeitpunkt t instantiierten Situationsschemata geordnet werden. Die Komplexität steigt allerdings bei einer Vervielfachung der möglichen Traversierungspfade an. Entsprechend [Michaelsen and Meidow, 2014] können dann entsprechend dem gewählten Auswertungszeitfenster pro Zeiteinheit die vielversprechendsten Pfade entsprechend der geordneten instantiierten Situationen verfolgt werden.

Mit dem gleichen Beispiel wie im vorherigen Abschnitt wird die Funktionsweise der Ableitung aller zutreffenden Hypothesen erläutert. Deutlich zu sehen ist die Mehrfachausprägung des Situationsschemas *InGroup* zum Zeitpunkt 68ff. Zum einen wird *InGroup* mit drei beteiligten Objekten und geringerem Wahrheitswert abgeleitet, als auch zum anderen eine Untermenge von *InGroup* mit zwei Agenten und höherem Wahrheitswert. Zu der Zeit, zu der sich Agent_1 immer mehr der Gruppe bestehend aus Agent_2 und Agent_0 nähert (*Approach*), wird zusätzlich *InGroup* bestehend aus Agent_0 und Agent_2 erkannt.

```
[0.00 | 38 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup]
0.10 | 39 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.18 | 40 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
[...]
0.69 | 44 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.61 | 45 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.57 | 46 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
[...]
0.12 | 68 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.43 | 68 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.91 | 68 | Agent_1 | [...] | [Agent_2,Agent_0] | Approach
```

```
0.25 | 69 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.55 | 69 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.87 | 69 | Agent_1 | [...] | [Agent_2,Agent_0] | Approach
0.41 | 70 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.67 | 70 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.32 | 70 | Agent_1 | [...] | [Agent_2,Agent_0] | Approach
0.53 | 71 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.77 | 71 | Agent_2 | [...] | [Agent_2,Agent_0] | InGroup
0.30 | 72 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
0.41 | 73 | Agent_2 | [...] | [Agent_2,Agent_1,Agent_0] | InGroup
[...]
```

Algorithmus 2 : Fuzzy Situation Graph Tree Traversal

Input : SGT, *object*

1 **if** *object occurs for the first time* **then**
2 $G \longleftarrow$ SGT root graph;
3 **forall** $s | s \in G \wedge s$ *is start situation* **do**
4 **if** s *can be instantiated* **then**
5 **forall** $spec | spec \in s \wedge spec$ *is specialization* **do**
6 $s := spec, G :=$ graph containing $spec$;
7 start recursion goto line 3;
8 evaluate actions of s;
9 **else**
10 **forall** $predSit | predSit$ *is prediction situation of* s *(all the last situation of the already known* $object$*)* **do**
11 **if** *instantiate* $predSit$ *successful* **then**
12 $s := predSit, G :=$ graph containing $predSit$;
13 start recursion goto line 5;
14 **else**
15 **if** $predSit$ *is instantiated end situation* $\wedge predSit \in G$ **then**
16 instantiation successful;
17 **else**
18 instantiation failed;

4.3 Zusammenfassung

Der Formalismus der begrifflichen auf FMTHL und SGTs basierenden Situationserkennung wurde um die Behandlung von Unschärfe und Ableitbarkeit aller zutreffenden Hypothesen erweitert. Diese Erweiterung ermöglicht es, die Eingabedaten entsprechend der Realität abzubilden und durch den gesamten Inferenzprozess bis zu den Situationen die Unsicherheit zu propagieren. Des Weiteren haben die untersuchten Diskursbereiche gezeigt, dass sich ein beobachteter Agent gleichzeitig in mehr als einer Situation befinden kann. Daher erlaubt die Ableitung aller zu diesem Zeitpunkt zutreffenden Situationen die erschöpfende Beschreibung der beobachteten Szene.

5 Repräsentation und Behandlung von Unvollständigkeit

Künstlich erzeugte Eingabgedaten sind initial per se perfekt – d.h. vollständig und frei von Rauschen. Bei anhand von Sensoren beobachteten und mit Bildverarbeitungsverfahren prozessierten realen Daten liegen in der Regel immer kleine Fehler – wie etwa bei der Lokalisierung – vor. Gelegentlich werden die Daten auch – etwa durch kurzzeitige Verdeckung – unvollständig sein. Auch werden wegen der open world assumption nicht modellierte Dinge auftreten (Clutter). Diese Effekte müssen – sofern z.B. zum Zwecke der Entwicklung robuster Verfahren erwünscht – in den künstlichen Daten ebenfalls künstlich erzeugt werden. In diesem Kapitel wird auf die Behandlung von verrauschten, fehlerhaften und unvollständigen Daten eingegangen. Das Problem wird zweifach angegangen: Zum einen werden auf Ebene der Eingabedaten Vorverarbeitungsfilter unter Berücksichtigung und Beibehaltung der systemimmanenten Unschärfe eingeführt. Zum anderen wird zur Behandlung von Unvollständigkeiten in der Verhaltens-Repräsentations-Ebene das Konzept der kontrollierten Halluzination eingeführt.

5.1 Behandlung von Rauschen

Bei der Erfassung von Bildsignalen wird das Bildsignal durch die Sensoren und deren Nachverarbeitung rechnerlesbar diskretisiert abgespeichert. Es wird also zu jedem Bild $g(x)$ sowohl der Ort x als auch der Bildwert g diskretisiert. Dieser Vorgang ist mit einem deutlichen Informationsver-

lust verbunden; es ist immer ein Kompromiss notwendig zwischen grober Diskretisierung, also wenig Speicherplatz und dadurch auch wenig weiter zu verarbeitender Information und dadurch geringer Rechenzeit und feiner Diskretisierung, also geringer Informationsverlust und dadurch Bilder von hoher Güte und somit viel weiter zu verarbeitender Information. Es stellt sich immer die Frage, ob die wesentlichen Signalinhalte noch vorhanden sind. So kann eine komplette Szene, die auf nur wenige Pixel abgebildet ist, nicht interpretiert werden, weil ein Großteil der Information beim Prozess der Diskretisierung verloren gegangen ist. Andererseits kann diese Aussage sehr schnell getroffen werden, weil wenig Information zu verarbeiten ist.

Fasst man die Bilderfassung und darauf aufbauende Bildverarbeitungsmethoden, wie in Kapitel 7 unter Abschnitt 7.2 vorgestellt, als System $\mathcal{S}$ auf, dann ist diesem eine Störgröße überlagert. Dieses Rauschen betrachten wir in unserem Fall als weißes Rauschen. Dieses ideale Signalmodell ist mittelwertfrei, normalverteilt und hat ein konstantes Spektrum. Mit einer endlichen Beobachtung der Zustände kann das Rauschen nicht eliminiert werden.

Methoden der Bildverarbeitung liefern beispielsweise als Eingabedaten für die Situationsanalyse Personentracks. Diese wiederum basieren auf Personendetektionen. Durch die o.g. Effekte bei der Bilderfassung und zusätzlich durch systemimmanente Phänomene der Bildverarbeitungsverfahren wird der von der Bildverarbeitung generierte Personentrack ebenfalls im Ortsbereich verrauscht sein. Die hier verwendete FMTHL kann, wie bereits in Abschnitt 3.1.1 erläutert, diese Effekte teilweise abbilden, jedoch werden bei Unsicherheiten in allen Dimensionen die getroffenen Aussagen bei der Schlussfolgerung immer weniger robust. Somit besteht die Notwendigkeit, diese verrauschten Eingebedaten für die Situationsanalyse geeignet vorzuverarbeiten. Die Notwendigkeit, diese Effekte zu behandeln, wird in [Harland, 2011] als wichtige zukünftige Herausforderung herausgearbeitet.

5.1.1 Vollständige, verrauschte Daten

Unter der Annahme, dass im Zeitbereich keine Fehler vorliegen, also zu jedem erwarteten Zeitpunkt ein verrauschtes Datum zum Ort im Bild oder in der Szene existiert, dann können zeitliche Tiefpassfilter, im Besonderen Glättungsfilter, einen großen Beitrag zur Verbesserung der Extraktion von Information aus den Daten leisten.

Die wesentlichen Ziele der Anwendung der Filter sind aufschlussreichere Daten – und zwar weniger die originären Eingabedaten als geglättete Daten, die näher an der tatsächlichen Trajektorie liegen und die Schätzung dieser präziser macht. Ein lokaler Glättungsfilter ist durch die Faltung $K * g$ beschrieben. Mit geeignetem Kern K. Dabei ist W das Fenster über dem Kern K.

$$(K * g)(t) = \sum_{(i) \in W} g(i) K(t - i) \tag{5.1}$$

In dieser Arbeit werden die beiden zeitlichen Tiefpassfilter Rechteckfilter und Gaußfilter verwendet. Beide erfüllen die Bedingungen, dass sie nichtnegativ sind, Mittelwert eins haben ($\sum_{n=-\infty}^{\infty} k_n = 1$) und Gradheit vorliegt ($k_n = -k_n$). Weil die Eingabedaten zeitlich äquidistant sind, werden nur ungerade Kerngrößen eingesetzt und der Gaußfilter durch eine diskrete Näherung, den Binominalfilter ersetzt.

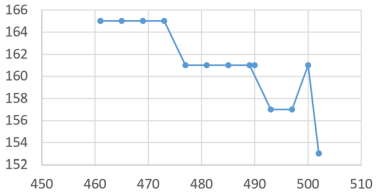
Beispiel: Aus dem PETS2009 Datensatz, siehe Abschnitt 7.3.3, wird in Abbildung 5.1 (a,b) eine Trajektorie geliefert. Ohne Beschränkung der Allgemeinheit betrachten wir hier Koordinaten im Bildbereich, statt transformiert in Weltkoordinaten. `t` steht für den Zeitpunkt, (x,y) für die Position der Personendetektion im Bild, (w,h) für das die Detektion umgebende Rechteck und `c` für einen Konfidenzwert.

In Abbildung 5.1 (c) bzw. (d) ist die Position (x,y) nach Tiefpassfilterung mit fünfdimensionalem Rechteckfilter mit Kern $K = [\frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}]^\intercal$ bzw. diskreter Gaußfilteralternative Binomialtiefpass mit Kern $K = [\frac{1}{16}\ \frac{1}{4}\ \frac{3}{8}\ \frac{1}{4}\ \frac{1}{16}]^\intercal$

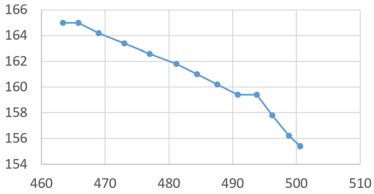
visualisiert. Es ist deutlich zu sehen, dass das Rauschen reduziert und die Trajektorie glatter wurde.

t	x	y	w	h	c
0	502	153	26	79	2.01
1	500	161	23	70	3.69167
2	497	157	26	79	4.01199
3	493	157	26	79	3.92423
4	489	161	25	74	3.45918
5	490	161	26	79	3.60182
6	485	161	26	79	3.46147
7	481	161	26	79	3.58387
8	477	161	26	79	3.78188
9	473	165	26	79	3.56313
10	469	165	26	79	3.55279
11	465	165	26	79	3.59965
12	461	165	26	79	3.57899

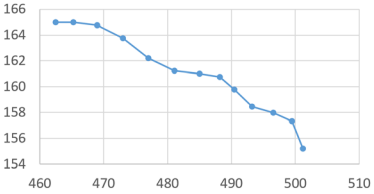
(a)



(b)



(c)



(d)

Abbildung 5.1: (a) Trajektorie einer Person. (b) Visualisierung von (x,y). (c) (x,y) mit Rechteckfilter. (d) (x,y) mit Binominalfilter.

Mit der Verwendung einer zeitlichen Tiefpassfilterung geht ein zeitlicher Versatz von der halben Fensterbreite zwischen dem Zeitpunkt der Detektion durch das Erkennungsverfahren und dem Zeitpunkt der Bereitstellung der gefilterten Daten für die Situationsanalyse einher. Möglicherweise verwischt die Tiefpassfilterung stark dynamische Bewegungen.

In diesem Abschnitt wurden Filter vorgestellt, welche die Eingabedaten vorverarbeiten, um sie für die Logik sinnvoll benutzbar zu machen. Im folgenden Abschnitt wird zusätzlich darauf eingegangen, wie Informationslücken auf der Ebene der Eingabedaten geschlossen werden können.

5.2 Behandlung von fehlenden Daten

In diesem Abschnitt wird beschrieben, wie auf Eingabeebene unvollständige Daten behandelt werden. Um zu einem Zeitpunkt t Inferenz zu betreiben, kann eine Regel nur dann angewandt werden, wenn alle dafür benötigten Fakten vorliegen. Ein Prädikat kann nur dann belegt werden, wenn eine passende Detektion vorliegt. Wenn also die Detektion ausbleibt – etwa wegen partieller Verdeckung – dann kann für das Prädikat keine erfüllende Belegung gefunden werden. Allerdings erfordern die meisten Regeln mehrere verschiedene Eingabedaten und diese von verschiedenen Zeitpunkten. Man betrachte beispielsweise die folgende Regel ***move_direction_is***:

$$
\begin{aligned}
\Box\{\textbf{\textit{move\_direction\_is}}(Agent, Direction) \leftarrow \\
\Diamond_{-3}\, as\_ground\_geometry(Agent, X_{-3}, Y_{-3}) \\
\wedge \Diamond_{-2}\, has\_ground\_geometry(Agent, X_{-2}, Y_{-2}) \\
\wedge \Diamond_{-1}\, has\_ground\_geometry(Agent, X_{-1}, Y_{-1}) \\
\wedge \Diamond_{1}\, has\_ground\_geometry(Agent, X_{1}, Y_{1}) \\
\wedge \Diamond_{2}\, has\_ground\_geometry(Agent, X_{2}, Y_{2}) \\
\wedge \Diamond_{3}\, has\_ground\_geometry(Agent, X_{3}, Y_{3}) \\
\wedge Y = Y_3 + Y_2 + Y_1 - Y_{-1} - Y_{-2} - Y_{-3} \\
\wedge X = X_3 + X_2 + X_1 - X_{-1} - X_{-2} - X_{-3} \\
\wedge atan2(Y, X, Rad) \\
\wedge convert\_angle(Rad, Direction)\}
\end{aligned}
\tag{5.2}
$$

Bei dieser Regel werden Eingabedaten aus dem Intervall $[t - \Delta t_1, t + \Delta t_2]$ mit $\Delta t_1 = \Delta t_2 = 3$ benötigt. Liegt auch nur ein benötigtes Datum in diesem Intervall bei der Inferenz nicht vor, dann kann diese Regel für den Zeitpunkt t nicht ausgeprägt werden. Dies liegt besonders daran, dass Regeln im Rumpf konjunktiv verknüpfte Hornformeln sind.

Dabei gilt es zu beachten, dass Daten entweder fehlen, weil die Sensoren und Bildverarbeitungsverfahren fehlgeschlagen sind oder weil die intendierte Situation eben nicht vorliegt. Treffen wir immer die Annahme, dass die Daten vorliegen, dann kann es zu einem Fehler kommen. Es würde ein Datum geschaffen werden, welches eine Ausprägung von Situationen ermöglichen würde, die zu diesem Zeitpunkt eigentlich gar nicht existieren. Im Folgenden wird eine Möglichkeit vorgestellt, wie mit unvollständigen Eingabedaten umgegangen und ein Kompromiss gefunden werden kann.

Beim Versuch, eine erfüllende Belegung für Regeln zu finden, müssen in einem Intervall $[t - \Delta t_1, t + \Delta t_2]$ die entsprechenden Daten vorliegen. Je nach Regel und Regeltiefe variieren Δt_1 und Δt_2. Geht man über alle Regeln des Begrifflichen Teilsystems für den gerade betrachteten Diskursbereich, kann ein maximaler Wert für Δt_1 und Δt_2 gefunden werden. Liegen im Intervall $[t - \Delta t_1, t + \Delta t_2]$ unvollständige Daten vor, d.h. gibt es Lücken einzelner Daten, dann können diese Lücken geschlossen werden. Ohne Beschränkung der Allgemeinheit setzen wir den aktuell zu betrachtenden Zeitpunkt t in das Intervall $[t - \Delta t_1, t + \Delta t_2]$. Damit geht einher, dass die Erfassung der Eingabedaten mindestens um die Zeitspanne Δt_2 dem Zeitpunkt t vorauseilt.

Durch Interpolation (und auch Extrapolation) können diese Lücken geschlossen werden. Es gibt unterschiedliche Typen von Daten, die eine unterschiedliche Interpolation erfordern.

Kumulieren von Daten

Metrische Daten Wenn Daten von t bis $t + \Delta t_0$ fehlen, dann wird jedes $x(t')$ mit $t' \in [t, t + \Delta t_0]$ berechnet als Mittel über alle prädizierten Werte $T_p = [t - \Delta t_1, t - 1]$ und $T_f = [t + \Delta t_0 + 1, t + \Delta t_0 + \Delta t_2]$ (erweiterte lineare Interpolation):

$$x(t') = \frac{1}{|T_p| + |T_f|} \sum_{t_p \in T_p} (w_{x(t_p)} \cdot x(t_p)) + \sum_{t_f \in T_f} (w_{x(t_f)} \cdot x(t_f)). \quad (5.3)$$

Mit den Gewichten $w_{x(t_p)}$ und $w_{x(t_f)}$ kann der Einfluss von T_p resp. T_f erhöht werden, wenn sich t' näher am Anfang oder Ende von $[t, t + \Delta t_0]$ befindet. Werden die Gewichte $w_{x(t_p)}$ und $w_{x(t_f)}$ auf Eins gesetzt, also nicht benutzt, erhalten alle $x(t')$ mit $t' \in [t, t + \Delta t_0]$ den selben Wert, was einer Rechteckfilterung entspricht.

Nominal skalierte Daten Sind einzelne Merkmale der Daten nominalskaliert, dann macht es keinen Sinn, davon einen Mittelwert zu bilden, weil sonst die Homomorphie verletzt werden würde. Beispielsweise sind *person*, *car* und *bike* Merkmale, die der Typ eines Objekts annehmen kann. Solche Lücken können und sollen nicht geschlossen werden.

Winkelmaße Bei Daten aus zirkulären Mengen, die für Winkel mit einem Wertebereich von $[0°, 360°)$ verwendet werden, darf der Mittelwert nicht auf naive Weise gebildet werden. Eine gängige Praxis ist die Abbildung von Winkeln auf den Einheitskreis, also ein Winkel α wird zu den Koordinaten $(\cos\alpha, \sin\alpha)$. Mit den Winkeln $\alpha_0, ..., \alpha_n$ berechnet sich das Mittel zu

$$\overline{\alpha} = atan2\left(\sum_{i=0}^{n} \sin\alpha_i, \sum_{i=0}^{n} \cos\alpha_i\right).$$

Äquivalente Alternativen bestehen mit Arkustangens über eine Fallunterscheidung oder mit komplexen Zahlen.

Behandlung Zusicherungsgrad etc.

Bei der Bestimmung des Zusicherungsgrads beim Schließen von Lücken von t bis $t + \Delta t_0$ werden der vorherige Zusicherungsgrad η_{t-1} und der nächste $\eta_{t+\Delta t_0+1}$ betrachtet. Der Zusicherungsgrad η für den Wert an der Lücke t berechnet sich über alle zum aktuellen Zeitpunkt aktiven Objekte obj zu $\eta_t = \min_{obj}(1 - \frac{step\_to\_border(obj)}{I+1})$. Dabei ist die ungerade Zahl I die maximale Breite einer Lücke und $step\_to\_border(obj)$ der maximale

Abstand zum nächsten Datum von Objekt obj. In Worten bedeutet das, dass der Zusicherungsgrad in der Mitte einer zu schließenden Lücke am größten gesetzt wird, weil der maximale Abstand zum Rand der Lücke minimal ist. Je weiter man sich von der Mitte einer Lücke entfernt, desto geringer wird der Zusicherungsgrad gesetzt, weil der maximale Abstand zum Rand der Lücke immer größer wird. Diese Art der Gewichtung rührt daher, dass die Lücke keine Lücke einer Trajektorie sein muss, sondern auch zwischen zwei verschiedenen Trajektorien liegen kann.

5.3 Kontrollierte Halluzination

Bisher wurde in Abschnitt 5.1 auf Unvollständigkeit auf Ebene der Eingabedaten eingegangen. In diesem Abschnitt wird auf Unvollständigkeit auf semantischer Ebene eingegangen. Bei einem SGT-Ablauf kann eine Prädiktionskante passiert werden, wenn im nachfolgenden Situationsschema das Verhaltensschema erfolgreich ausgeprägt werden kann. Die Prädikate im Verhaltensschema stützen sich i.d.R. direkt auf Ergebnisse aus dem Quantitativen Teilsystem. Schlagen Bildverarbeitungsverfahren fehl, weil z.B. ein Objekt nicht detektiert wird, dann kann ein Prädikat, das dessen Existenz voraussetzt, nicht erfüllend belegt werden. Damit würde bei der SGT-Traversierung dieser SGT-Ablauf abgebrochen werden.

Es wird nun die SGT-Traversierung dahingehend erweitert, dass bei der Prädiktion im Situationsgraphen ein Übergang in zeitlich nachfolgende Situationsschema ermöglicht wird, auch wenn einzelne Prädikate nicht erfüllend belegt werden konnten, weil keine Evidenz vorhanden war. Es heißt dann, dass diese Prädikate halluziniert wurden, siehe auch Abbildung 5.2 für eine Visualisierung. Technische Gründe für unvollständige Daten sind beispielsweise Verdeckungen oder Unzulänglichkeiten der Bildverarbeitungsverfahen.

Detaillierende Kanten werden bei der kontrollierten Halluzination nicht beeinflusst. Das Zeil der kontrollierten Halluzination ist die erfolgreiche

Durchführung von SGT-Abläufen und nicht eine mögliche weitere Detaillierung. Würden auch detaillierende Kanten halluziniert werden, dann würde das einer Generierung aller möglicher Zustände der Welt, repräsentiert durch das Hintergrundwissen eines SGTs, entsprechen. Dieses Vorgehen wäre kombinatorisch nicht handhabbar. Dadurch, dass in einem Situationsgraphen ausschließlich temporale Kanten halluziniert werden können, ist der Mehraufwand linear in der Anzahl zeitlich nachfolgender Situationsschemata.

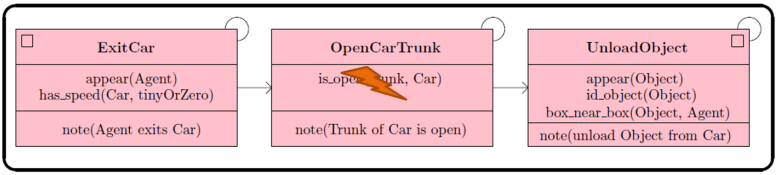


Abbildung 5.2: Visualisierung der kontrollierten Halluzination bei der Verhaltenserkennung. Kann in einem SGT-Ablauf das nachfolgende Situationsschema (**OpenCarTrunk**) nicht ausgeprägt werden, weil ein Prädikat im Verhaltensschema (*is_open(Trunk,Car)*) nicht erfüllend belegt werden kann, dann wird trotzdem dieses Situationsschema instantiiert.

5.3.1 Umsetzung der kontrollierten Halluzination

Die Halluzination kann durch eine Erweiterung des SGT-Traversierungsalgorithmus realisiert werden. Obwohl die Darstellung in prozeduralem Pseudocode kompakt erscheint, ist die Umsetzung des SGT-Traversierungsalgorithmus als deklaratives Logikprogramm umfangreich.

Die Erweiterung bedeutet, dass alle weiterführenden Prädiktionskanten im Situationsgraphen passiert werden sollen, und wenn ein Zustandsschema auf den ersten Versuch nicht ausgeprägt werden kann, dann soll es halluziniert werden und mit sehr niedrigem Zusicherungsgrad ausgeprägt werden. In Pseudocode werden im Algorithmus 2 in Zeile 10 die folgenden Anweisungen eingefügt:

if $\neg(predSit$ *is start situation* $) \wedge predSit \in G$ **then**
  continue instantiation with Degree of Validity 0.01;

Diese Änderung zieht auch eine weitere Behandlung des Zusicherungsgrads bei ausgeprägten Situationsschemata nach sich. Damit ist es dann möglich, SGT-Verhalten mit halluzinierten Elementen von SGT-Verhalten ohne diese zu unterscheiden und ggf. getrennt zu behandeln.

5.3.2 Exemplarisches Beispiel zur kontrollierten Halluzination

Zur Verdeutlichung der kontrollierten Halluzination dient beispielhaft eine Szene aus dem VIRAT-Datensatz VIRAT_S_000002 ab Frame 3933ff., wo sich eine Person einem Auto nähert, den Kofferraum öffnet, ein Objekt auslädt, den Kofferraum wieder schließt und sich dann zusammen mit dem Objekt wieder vom Auto entfernt. Siehe Abbildung 5.3 für die Szene.

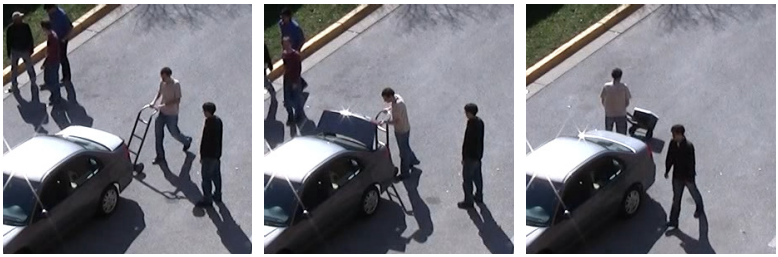


Abbildung 5.3: Szene aus dem VIRAT-Datensatz VIRAT_S_000002 ab Frame 3933ff., die das Ausladen eines Objektes darstellt.

Die Schwierigkeit für Objektdetektionsverfahren ist es hier, die Objekte zu erkennen, die entweder verdeckt sind oder schlecht sichtbar. Zu Gunsten der Situationserkennung wurden für alle Objekte – außer Personen – die Grundwahrheit – sofern vorhanden – verwendet. Das erklärt auch den

Wahrheitswert 1.00 bei den Objekten. Im Folgenden ein Auszug der ausgeprägten Prädikate:

```
[...]
0.08 | 131 | appear(Agent_17)
1.00 | 131 | has_speed(Car_5,tinyOrZero)
0.32 | 132 | appear(Agent_17)
1.00 | 132 | has_speed(Car_5,tinyOrZero)
0.61 | 133 | appear(Agent_17)
1.00 | 133 | has_speed(Car_5,tinyOrZero)
[...]
[0.01 | [...] | is_open(Trunk, Car_5)]
[...]
0.21 | 142 | appear(Object_3)
1.00 | 142 | is_object(Object_3)
1.00 | 142 | box_near_box(Object_3,Agent_17)
[...]
```

Entsprechend einem möglichen SGT-Ablauf zum Situationsgraphen aus Abbildung 5.2 wurde das Öffnen des Kofferraums nicht beobachtet (in diesem Fall, weil dazu keine Daten vorhanden waren). Somit müsste der mit *Agent_17* und *Car_5* begonnene SGT-Ablauf abgebrochen werden. Mit den Mitteln der kontrollierten Halluzination wird der SGT-Ablauf weiterverfolgt und kann auch, nachdem in Zeitschritt 142 (Frame 4263) das Situationsschema **UnloadObject** ausgeprägt werden kann, erfolgreich zu Ende geführt werden.

5.4 Zusammenfassung

Reale Daten sind fehlerbehaftet. Verrauschte Eingabedaten werden zeitlich gefiltert. Zur Kompensation von Unvollständigkeit interpolieren Vorverarbeitungsfilter die Lücken in den Daten. Auf semantischer Ebene können Lücken von Beobachtungen dazu führen, dass einzelne Situationen nicht ausgeprägt werden können und die konsekutiven Situationen somit niemals erreicht werden können. Das Konzept der kontrollierten Halluzination wurde entwickelt, um auf semantischer Ebene diese Unvollständigkeiten zu kompensieren.

6 Komplexitätsreduktion

Inferenz mit vielen Objekten, die lokal und temporal zueinander nahe sind, ist aufwendig. Die Modellierung von Situationen mit mehreren Objekten ist umfangreich und unübersichtlich, wenn binäre Relationen verwendet werden [Arens et al., 2008, Münch et al., 2011a]. In diesem Kapitel werden für beide Probleme Lösungen vorgestellt.

6.1 Semantische Vorfilterung

Bei der Videoüberwachung im Innen- und Außenbereich ist oft weniger das Verhalten von einzelnen Personen als das Verhalten von mehreren Personen von Interesse – besonders dann, wenn die Personen in einer Gruppe auftreten. Das Konzept Gruppe kann von dem hier verwendeten Ansatz modelliert werden. Es stützt sich dabei auf Personen und deren zeitlichen und örtlichen Abstand zueinander. Jedoch steht dieser Modellierung eine hohe kombinatorische Komplexität gegenüber. Diese rührt daher, dass die Suche einer optimalen Partition einer Menge von n Individuen nach Gruppen zu einer Suche in der Potenzmenge der Individuen führt. In einer solchen Situation tendieren auf Logik basierende Systeme – mit ihrem Schwerpunkt auf der Korrektheit – zu einer tiefen exponentiell verzweigenden Suche.

Um dem Problem der Komplexität zu begegnen wird an dieser Stelle der Suchraum von Beginn an methodisch reduziert. Einfache geometrische, quantitative Prädikate, wie beispielsweise *have_distance* erlauben die Anwendung von Methoden des Maschinellen Lernens, um diese Prädikate

nicht nur räumlich, sondern auch zeitlich zusammenzufassen. Konkret kann also mit geringem Aufwand eine Ballungsbildung dieser Prädikate durchgeführt werden. Der bestehende Ansatz bleibt methodisch unangetastet und arbeitet dann nicht mehr mit den originären Eingabedaten, sondern nur noch mit den in diesem Schritt erzeugten reduzierten vorverarbeiteten Daten. Damit sinkt die kombinatorische Komplexität und damit der Gesamtaufwand.

Um Situationen, die Gruppen von Personen beinhalten, zu erkennen führt [Lan et al., 2011] ein latentes Variablenmodellrahmenwerk ein. Kontextbezogene Informationen werden benutzt, um einen neuen kontextbezogenen Aktionsdeskriptor einzuführen. Ein hierarchischer beschreibungsbasierter probabilistischer Ansatz erlaubt es, auch komplexere Situationen zu detektieren [Ryoo and Aggarwal, 2009]. Dieser Ansatz wurde in [Ryoo and Aggarwal, 2011] um die Detektion von Gruppenaktivitäten erweitert. Einige grundlegende und einfache Aktivitäten führen zu einer übergeordneten Gruppenaktivität. Diese einfachen Aktivitäten sind die Aktivitäten einer einzelnen Person, die Interaktion zwischen Personen oder schon Gruppenaktivitäten. Manuell erstellte Regeln in Form einer kontextfreien Grammatik werden im hierarchisch probabilistisch Bayes'schen Ansatz verwendet. Markov-Chain-Monte-Carlo-Verfahren nähern dabei die multivariate Verteilung an.

Das Verfolgen von Gruppen und die Interaktion in Gruppen wurden in [Zaidenberg et al., 2011] erarbeitet. Dort werden Gruppeneigenschaften definiert, wie z.B. den durchschnittlichen Abstand für einige aufeinanderfolgende Einzelbilder. Außerdem führen der Mittelwert der Standardabweichung der Geschwindigkeit eines Agenten und der Mittelwert der Standardabweichung der Richtungen zu einem gewichteten *groupCoherence*-Faktor entsprechend der Definition einer Gruppe ebendort: "*two or more people who are spatially and temporally close to each other and have similar direction and speed of movement for a minimum duration*".

In [Ge et al., 2012] wird eine hierarchische Ballungsbildung vorgestellt, die eine verallgemeinerte symmetrische Hausdorff-Metrik verwendet, basierend auf paarweiser Distanz und Geschwindigkeit der einzelnen Objekte von Interesse. Dieser Ansatz demonstriert anschaulich, wie die kombinatorischen Beschränkungen von regelbasierten Systemen durch nichtdeklarative Verfahren wie die Ballungsbildung in einem Vorverarbeitungsschritt effizient unterstützt werden können. Der deklarative Aspekt bleibt dabei bestehen.

6.1.1 Mean-Shift-Ballungsbildung

Ursprünglich wurde Mean-Shift in [Cheng, 1995] als Algorithmus vorgestellt, der Modi einer Wahrscheinlichkeitsdichtefunktion sucht. In [Comaniciu and Meer, 2002] wurde Mean-Shift dann auf weitere Probleme aus der Computer Vision angewandt. K-Meanssky, siehe [Lloyd, 1982], oder auch Mixtures of Gaussians sind parametrische Verfahren für die Wahrscheinlichkeitsdichteschätzung. Die Anzahl der Modi muss für solche Verfahren im Voraus bekannt sein, was aber oft nicht der Fall ist. Im Gegensatz dazu nutzt Mean-Shift ein parameterloses Modell um eine beliebige Anzahl beliebig geformter Modi zu finden.

Gegeben sind n Werte $\mathbf{x}_i$ mit $i = 1, ..., n$ aus einem d-dimensionalen Merkmalsraum R^d. Der multivariate Kerndichteschätzer mit Kern $K(\mathbf{x})$ und Bandbreite h für den Punkt $\mathbf{x}$ lautet:

$$f(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{N} K(\frac{\mathbf{x} - \mathbf{x}_i}{h}). \tag{6.1}$$

Wie schon [Comaniciu and Meer, 2002] beschreiben, sind radialsymmetrische Kerne geeignet und vorteilhaft für die weitere Berechnung. Es wird der Kern $K(\mathbf{x}) = c_{k,d}k(\|\mathbf{x}\|^2)$ definiert mit der Konstante $c_{k,d}$, sodass gilt $\int_{R^d} K(\mathbf{x})\mathrm{d}\mathbf{x} = 1$. Die Modi der Wahrscheinlichkeitsdichtefunktion sind genau dort, wo gilt $\nabla f(\mathbf{x}) = 0$. Mit der geschickten Wahl eines Gauß-

kerns für $G(\mathbf{x})$ sind $G(\mathbf{x})$ und dessen Shadowkernel $K(\mathbf{x})$ identisch. Der Gradient von Gleichung 6.1 lautet:

$$\nabla f_{h,K}(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2)}{\sum_{i=1}^{n} g(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2)} - \mathbf{x} \right] \tag{6.2}$$

Der erste Teil in Gleichung 6.2 ist proportional zu Gleichung 6.1. Der zweite Teil ist der dem Verfahren namensgebende „Mean-Shift“:

$$\mathbf{m}_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2)}{\sum_{i=1}^{n} g(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2)} - \mathbf{x}. \tag{6.3}$$

Mit anderen Worten ist $\mathbf{m}_{h,G}$ die Differenz zwischen dem gewichteten Mittel und $\mathbf{x}$, dem Zentrum des Kerns. Formt man Gleichung 6.3 weiter um sieht man, dass an Stelle $\mathbf{x}$ der mit Kern G berechnete „Mean-Shift“-Vektor proportional zum normalisierten Wahrscheinlichkeitsdichtegradienten ist, der wiederum mit Kern K berechnet wurde. Dieser Vektor zeigt immer in die Richtung der größten Zunahme der Wahrscheinlichkeitsdichte.

Ein Schritt beim iterativen Mean-Shift-Verfahren besteht daher aus zwei alternierenden Teilschritten:

- Zuerst wird der Mean-Shift-Vektor $\mathbf{m}_{h,G}(\mathbf{x})$, siehe Gleichung (6.3), berechnet.
- Anschließend verschiebt man den Kern $G(\mathbf{x})$ um $\mathbf{m}_{h,G}(\mathbf{x})$:

$$\mathbf{x}_{i+1} = \mathbf{m}_{h,G}(\mathbf{x}) + \mathbf{x}_i \tag{6.4}$$

Die Normalisierung in Gleichung 6.3 hat den Vorteil, dass in Regionen mit wenigen Merkmalen die Schritte des Mean-Shift groß sind, wohingegen in Regionen mit vielen Merkmalen und Nebenmaxima die Schritte kleiner und damit die Abtastung feingranularer sind. In diesem Sinne verhält sich

Mean-Shift wie ein adaptives Gradientenaufstiegsverfahren. Die Konvergenz wird in [Comaniciu and Meer, 2002] bewiesen.

Der entscheidende Vorteil besteht aber nicht nur in der Parameterlosigkeit, sondern in der Berechnung der Wahrscheinlichkeitsdichtefunktion. Diese wird beim Mean-Shift-Verfahren nämlich gar nicht berechnet. Es interessieren nur die Modi[8] der Wahrscheinlichkeitsdichtefunktion.

6.1.2 Mean-Shift-Ballungsbildung zur semantischen Vorfilterung

Einerseits ist das in der Begrifflich-Primitiven-Ebene repräsentierte Wissen bezüglich des betrachteten Diskursbereichs allgemeingültig wie z.B. *distance_is(Agent, Patient, Distance)*, andererseits handelt es sich dabei um Begrifflichkeiten auf einer niederen Ebene mit geringer Komplexität. Das Beispiel *have_distance(agent2,agent6,small)* lässt sich so interpretieren, dass der räumliche Abstand von *agent2* und *agent6 small* ist.

Die Grundannahme sind eine in- und extrinsisch kalibrierte Kamera und eine Abbildung der realen beobachteten Szene zur Bildebene. Im Allgemeinen ist es nicht sinnvoll, mit den Bildkoordinaten zu arbeiten, weil das absolute Verhältnis in Bildkoordinaten dem Verhältnis in der realen Szene nicht entsprechen muss. Zur weiteren Verarbeitung werden die von der Bildebene in die reale Szene zurückprojizierten Koordinaten verwendet. Zu Visualisierungszwecken kann diese Abbildung wieder umgekehrt werden und damit können reale Koordinaten in der Bildebene dargestellt werden.

Der Merkmalsraum, in dem Mean-Shift zur semantischen Vorverarbeitung eine Ballungsbildung durchführt, kann sich über drei Ebenen erstrecken: (i) geometrisch, quantitativ, (ii) basisbegrifflich und (iii) situativ. Der einfache Fall (i) behandelt die quantitativen Eingabedaten zur Situationsanalyse direkt auf dem Bild. Bei der Ballungsbildung im basisbegrifflichen

[8] Die Mean-Shift-Ballungsbildung ist ein multivariates strukturentdeckendes Verfahren.

Merkmalsraum (ii) werden auf Ebene der Begrifflich-Primitiven-Ebene Prädikate geclustert. Schließlich können auch instantiierte Situationen (iii) geclustert werden. Dies ist besonders dann hilfreich, wenn die Situationsanalyse viele konkurrierende Hypothesen gestattet. In dieser Arbeit wird in Abschnitt 6.1.3 exemplarisch die Anwendbarkeit von (ii) gezeigt:

Abbildung 6.1: Ergebnisse des Mean-Shift-Verfahrens basierend auf der Basisregel *distance_is*. Aus dem BEHAVE-Datensatz (siehe Abschnitt 7.3.1) exemplarisch Frame 5370 (links), 5415 (mittig) und 5445 (rechts). Man kann deutlich sehen, wie sich nahe stehende Objekte zusammengefasst werden und gleichzeitig die einzelne passierende Person nicht hinzugefügt wird.

Die Idee ist nun, einfache Basisprädikate aus der Begrifflich-Primitiven-Ebene zu clustern. Exemplarisch wird hier die Verteilung aus der räumlichen und zeitlichen Nähe der jeweiligen Objekte betrachtet: sekundär ist das *have_distance* und primär wird *distance_is(Agent, Patient, Distance)* über *Distance* gelustert. Mean-Shift berechnet darauf die Modi, die von einer Häufung von zueinander nahen Objekten verursacht werden. Man beachte, dass man sich hier nicht wie in (i) im Bildbereich befindet, sondern dass es sich um die in die reale Szene zurückprojezierten Koordinaten handelt. In Abbildung 6.1 ist ein qualitatives Ergebnis dargestellt: Von den fünf Personen in der beobachteten Szene werden korrekterweise zwei Gruppen mit jeweils zwei Personen detektiert und die einzelne Person, die die anderen passiert, wird auch als solche erkannt. Das Mean-Shift-Verfahren hat keine Parameter betreffend der Anzahl der Elemente eines Clusters und kann somit beliebig große Gruppen identifizieren.

Die Anwendung von Mean-Shift zur semantischen Vorfilterung bietet drei entscheidende Vorteile.

Erstens entspricht diese Beschreibung von Gruppen in etwa dem, wie Menschen intuitiv Gruppen von Objekten identifizieren. Die Gruppe von Objekten ist nicht quantitativ – z.B. durch einen Abstand in Meter – bestimmt, sondern durch eine örtliche und zeitliche Häufung.

Zweitens wurde bisher die agentenbasierte Situationserkennung pro Agent durchgeführt. Das wiederum bedeutet, dass alle Gruppensituationen mehrfach erkannt werden, jeweils pro Agenten. Konkret sind die Reflexivitäts- und Symmetrieeigenschaft erfüllt. Die Transitivität kann allerdings in diesem Fall verletzt werden. Seien beispielsweise zwei Agenten von einem dritten gleich weit entfernt und es wird aus Sicht des Dritten eine Gruppe erkannt, so können die beiden ersten Agenten so weit auseinander stehen, dass sie nur eine Gruppe mit dem Dritten eingehen, siehe auch [IJsselmuiden, 2014]. Somit liegt die Äquivalenz der aus der Sicht eines jeden Agenten erkannten Situationen nicht vor und es können im weiteren Verlauf der Situationserkennung verschiedene Gruppen für die in Wirklichkeit gleiche Gruppe verwendet werden. Bei der Vorfilterung mit Mean-Shift wird diese Beschränkung aufgehoben. Es steht nicht mehr der Agent im Zentrum der Situationserkennung, sondern die von Mean-Shift geschätzten Häufungen. Diese Häufungen kann man nun als Gruppe bezeichnen und darauf aufbauend mit diesen Gruppen als Superagenten die Situationserkennung durchführen. Gegebenenfalls ist eine ausdrückliche Behandlung von Situationen innerhalb der Superagenten erforderlich.

Drittens ergibt die praktische Reduzierung von vielen Agenten auf wenige Superagenten eine drastische Reduktion des Lösungsraums und somit erhebliche Speicher- und Laufzeitreduzierungen durch die geringere Komplexität.

In diesem Abschnitt wurde im Wesentlichen die semantische Vorfilterung an einem Beispiel verdeutlicht. Die geometrisch und quantitativen Fälle aus (i) und einfache Prädikate aus (ii) sind damit abgedeckt. Komplexere Prädikate aus (ii) und Situationen aus (iii) bedürfen einer weiteren detaillierteren Untersuchung.

6.1.3 Auswertung Mean-Shift-Ballungsbildung zur semantischen Vorfilterung

Die Anwendung der Mean-Shift-Vorfilterung wurde hinsichtlich der Laufzeit auf dem BEHAVE-Datensatz (siehe Abschnitt 7.3.1) ausgewertet. Die Quantisierung der theoretischen Komplexität und deren Reduktion sind nur schwer möglich. Um die im BEHAVE-Datensatz vorkommenden Situationen zu erkennen reicht es aus, die Datenrate auf ein Fünftel gemittelt zu reduzieren.

In Abbildung 6.2 sind die Laufzeiten in Sekunden auf dem BEHAVE-Datensatz dargestellt: In Rot die Länge der Teilsequenz, in Grün die Dauer der Situationserkennung ohne die Mean-Shift-Ballungsbildung als vorfilternden Schritt und in Blau die Dauer der Situationserkennung pro Teilsequenz mit der Mean-Shift-Ballungsbildung als Vorfilterung. Die Ergebnisse wurden mit einem Standardcomputer (Intel i7 3.3GHz, 8GB Ram, Win7 x64) erzeugt. Daraus kann man erkennen, dass in diesem Fall die Auswertung der einzelnen Teilsequenzen deutlich weniger Zeit erfordert als die Teilsequenzen lang sind und somit eine Echtzeitverarbeitung der Situationserkennung realistisch erscheint.

In Abbildung 6.3 und 6.4 werden qualitative Ergebnisse mit und ohne Mean-Shift-Ballungsbildung als vorfilternden Schritt gezeigt. Die Sensitivität (Recall) ist sowohl mit als auch ohne die Vorverarbeitung im Prinzip gleich, aber die Genauigkeit (Precision) ist durch die Vorverarbeitung höher. Das rührt daher, dass eine Gruppe von Personen als Ganzes betrachtet wird und nicht mehr Situationen mit Teilen von Gruppen gebildet wer-

den können. Es werden in den Abbildungen die Situationen *WalkTogether* (blau), *InGroup* (gelb), *Approach* (rot) und *Split* (violett) dargestellt.

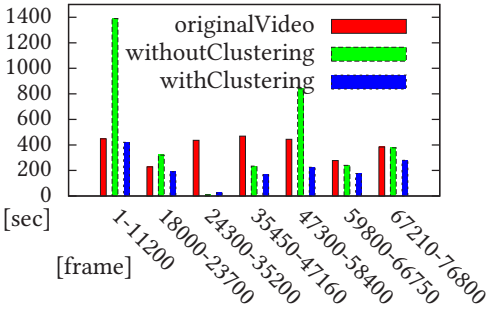
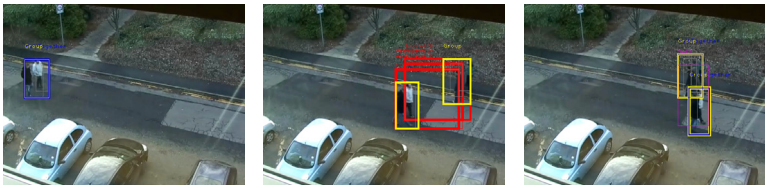


Abbildung 6.2: Die Laufzeit der Situationserkennung auf den einzelnen Teilsequenzen gegenüber der Laufzeit mit und ohne Mean-Shift-Ballungsbildung als vorfilternden Schritt.

6.2 Mengenbasierte Inferenz und Erweiterung der Regelbasis

Mit der Ballungsbildung von u.a. einfachen Basisprädikaten kann die Komplexität und Laufzeit der gesamten Situationserkennung deutlich reduziert werden. Weiterhin besteht das Problem, Situationen mit vielen Abhängigkeiten zwischen den einzelnen Objekten zu modellieren, auch innerhalb von „Gruppen“ oder in Teilen davon. Jede zu modellierende Abhängigkeit muss bisher meist binär für alle möglichen Szenarien modelliert werden. Das macht die Modellierung von Situationen v.a. im Gruppenkontext nicht handhabbar. Historisch betrachtet stellte das kein Problem dar, da die betrachteten Situationen nur sehr wenige Objekte und keine Situationen im Gruppenkontext umfassten.



(a) (b) (c)

Abbildung 6.3: Sequenz vom BEHAVE-Datensatz bestehend aus Frame 19285, 19395, 19420. Hier wurde keine Mean-Shift-Ballungsbildung als vorverarbeitenden Schritt durchgeführt. In (a) sind zwei zusammen gehende Personen zu sehen. „WalkTogether“ (blau) von beiden Personen wird erkannt, ebenso wird auch erkannt, dass die beiden „InGroup“ (gelb) sind. In (b) gehen zwei Gruppen von Personen aufeinander zu. Durch die jeden Agenten betrachtende Situationsanalyse werden viele nicht deckungsgleiche „Approach“-Situationen (rot) detektiert. In (c) verhält es sich ähnlich zu (b). Zwar werden beide Gruppen als „InGroup“ und „WalkTogether“ korrekterweise erkannt, aber die Situationen zwischen beiden Gruppen (violett) umfassen nicht alle Personen.

(a) (b) (c)

Abbildung 6.4: Sequenz vom BEHAVE-Datensatz bestehend aus Frame 19285, 19395, 19420. Hier wurde Mean-Shift-Ballungsbildung als vorverarbeitenden Schritt durchgeführt. (a) Innerhalb des Superagenten werden „WalkTogether“ (blau) und „InGroup“ (gelb) erkannt. Im Vergleich zu Abbildung 6.3 (b) wird „Approach“ (rot) jetzt mit der vollständigen Menge an Agenten instantiiert. (c) analog zu (b).

Diese aufwendige Modellierung der Abhängigkeiten zwischen allen beteiligten Objekten wird durch die Verwendung von Filtern auf Mengen aufgehoben. Ein Agent modelliert nicht mehr alle Abhängigkeiten bzgl. aller anderen Objekte einzeln, sondern bzgl. der gesamten Menge aller anderen Objekte. Dabei spielt es dann keine Rolle mehr, wie viele Elemente diese Menge enthält. Konkret wird ein- und dieselbe Regel für Mengen beliebiger Kardinalitäten angegeben.

Prolog ist eine logische Programmiersprache. Aus technischer Sicht ist sie F-LIMETTE sehr ähnlich. Sowohl Prolog als auch F-LIMETTE basieren auf Horn-Klauseln und auf dieser Datenbasis wird die Inferenz durch Resolution durchgeführt. Für Mengen gibt es in Prolog allerdings keine ausgezeichnete Datenstruktur. Jedoch ist für endliche Mengen die Datenstruktur der Liste geeignet. Listen sind ein wesentlicher Bestandteil von Prolog. Diese Verwandschaft erlaubt eine Übertragung etablierter Konzepte aus Prolog in F-LIMETTE. An dieser Stelle führen wir nun das Konzept der Liste in die auf FMTHL und SGT basierenden Situationserkennung ein, mit dem Ziel, Operationen auf endlichen Mengen auszuführen.

Dazu wurden die Basisregeln in der Begrifflich-Primitiven-Ebene erweitert. Der Kern bildet dabei das Konzept des Filters. Allgemein gesprochen wendet ein Filter ein Prädikat nicht nur auf ein einzelnes Objekt an, sondern auf eine Menge von Objekten. In Gleichung (6.5)-(6.7) ist der rekursiv definierte ***truefilter*** dargestellt. Die Implementierung in FMTHL teilt ihn in drei Prädikate auf, die sich rekursiv aufrufen. Die nach außen zu benutzende Signatur lautet ***truefilter***$(in, Fun, agent, \ parameter, res)$. Durch den temporalen Alloperator $\Box$ wird aus dem Prädikat eine Regel. Wird nun versucht das Prädikat in Gleichung (6.5) auszuprägen, wird Gleichung (6.6) „aufgerufen“. ***truefilter_*** ruft sich dabei rekursiv auf, bis alle Elemente der Eingabemenge verarbeitet wurden und die Abbruchbedingung (6.7) erreicht wird. Die Liste `res` enthält alle Elemente aus der Liste

in, für die *Fun(agent, elem, parameter)* eine erfolgreiche Belegung gefunden hat. Weitere Filter werden analog definiert.

$$\Box\{\textbf{\textit{truefilter}}(in, Fun, agent, parameter, res) \leftarrow \\ truefilter\_(in, Fun, agent, parameter, res) \wedge\ res <> []\} \tag{6.5}$$

$$\Box\{\textbf{\textit{truefilter\_}}([elem|in], Fun, agent, parameter, res) \leftarrow \\ functor = ..[Fun, agent, elem, parameter] \\ \wedge [(call(functor) \wedge\ res = [elem|new] \wedge !) \vee\ res = new] \\ \wedge truefilter\_(in, Fun, agent, parameter, new)\} \tag{6.6}$$

$$\Box\textbf{\textit{truefilter\_}}([], \_, \_, \_, []) \tag{6.7}$$

Aus Sicht der Theorie könnte *call/N* auf die gesamte Eingabeliste angewandt werden. Allerdings wird in [Naish, 1996] argumentiert, dass aus Komplexitätsgründen auf *call/N* verzichtet werden sollte. Dahingehend wurde die Implementierung auf *call/3* umgestellt, indem das Filter rekursiv definiert wurde, siehe Gleichungen (6.5)-(6.7).

Mit der Einführung von Filtern gehen zwangsweise weitere Listenoperationen einher, um in FMTHL auf endlichen Mengen zu arbeiten:
truefilter$(+In, +Fun, +Agent, +Parameter, ?Res)$
Res enthält alle Elemente von In für die $Fun(Agent, Elem, Parameter)$ zu wahr ausgewertet wurde.
falsefilter$(In, Fun, Agent, Parameter, Res)$
Res enthält alle Elemente von In für die $Fun(Agent, Elem, Parameter)$ zu falsch ausgewertet wurde.
truerevfilter$(+In, +Fun, +Agent, +Parameter, ?Res)$
Res enthält alle Elemente von In für die $Fun(Elem, Agent, Parameter)$ zu wahr ausgewertet wurde. Man beachte, dass $Elem$ und $Agent$ vertauscht wurden.

falserevfilter$(In, Fun, Agent, Parameter, Res)$
Res enthält alle Elemente von In für die $Fun(Elem, Agent, Parameter)$ zu falsch ausgewertet wurde. Man beachte, dass $Elem$ und $Agent$ vertauscht wurden.
rev$(+List, ?Res)$
Ist wahr, wenn die Elemente von Res und $List$ dieselben sind, aber entgegengesetzt sortiert.
member_of$(?Elem, +List)$
Ist wahr, wenn $Elem$ ein Element von $List$ ist.
cut_of_last_element$(+List, ?Elem, ?Res)$
Ist wahr, wenn Res die Liste $List$ ohne das letzte Element $Elem$ hat.
concat_lists$(+List1, +List2, ?Res)$
Ist wahr, wenn Res die Konkatenation von $List1$ und $List2$ ist.
list_length$(+List, ?Res)$
Ist wahr, wenn Res die Länge der Liste $List$ ist.
flatten_list$(+List, ?Res)$
Ist wahr, wenn Res keine verschachtelte Liste von $List$ ist.
delete_doubles$(+List, ?Res)$
Res enthält eine Liste der Elemente von $List$ nachdem alle mehrfach vorkommende Elemente auf ein einziges Vorkommen reduziert wurden.
order_consistent$(+List1, +List2)$
Ist wahr, wenn $List1$ ist Teilmenge von $List2$ und für jedes Tupel $(l11, l12)$ von $List1$ mit $index(l11) < index(l12)$ in $List1$ gilt auch in $List2$ $index(l11) < index(l12)$.
nth_elem$(+List, +Elem, ?N)$/***nth_elem***$(+List, ?Elem, +N)$
Position N entspricht der Position von $Elem$ in $List$. Die Zählung beginnt bei 1. Element $Elem$ entspricht dem Element an Position N in $List$.
replace_elem$(+List, +Elem, +N, ?Res)$
Res enthält die Liste $List$, bei der das Element an Position N mit dem Element $Elem$ ersetzt wurde.

reset_nth_elem$(+List, +N, ?Res)$
Res enthält die Liste $List$, in der eine konkrete Belegung an Position N zurückgesetzt wurde.
common_head$(+List1, +List2, ?Res)$
Res enthält diejenigen Elemente, die von vorne sowohl in $List1$ als auch $List2$ identisch sind.
common_tail$(+List1, +List2, ?Res)$
Res enthält diejenigen Elemente, die von hinten sowohl in $List1$ als auch $List2$ identisch sind.
delete_empty_lists$(+List, ?Res)$
Res enthält alle Elemente von $List$, außer den leeren Listen auf gleicher Ebene.
sub_list$(+List, +StartAt, +EndAt, ?Res)$
Res enthält einen Teil der Liste $List$ und zwar von Element an Position $StartAt$ bis $EndAt$.
has_exactly_one_element$(+List, +Elem)$
Ist wahr, wenn $Elem$ das einzige Element der Liste $List$ ist.

Durch die Einführung der Inferenz auf Mengen von Objekten und dessen Umsetzung ist es jetzt möglich Situationen zu modellieren, bei denen mehr als zwei Objekte beteiligt sind. Gleichzeitig ist die dadurch geschaffene Repräsentation des Wissens lesbar und anschaulich.

6.3 Diskussion

Die mengenbasierte Inferenz ermöglicht das Modellieren von Situationen, die ein Zusammenspiel von mehr als zwei Objekten erfordern; und zwar auf eine einfache und übersichtliche Art und Weise. Damit einher geht auch die Umgestaltung von Hintergrundwissen. Da die Anwendung von Prädikaten auf ganze Listen – statt auf einzelne Variablen – eine grundlegende Änderung des Architekturparadigmas ist, müssen bereits existierende

Regeln aus der Begrifflich-Primitiven-Ebene an die neuen Gegebenheiten angepasst werden. Auch in der Verhaltens-Repräsentations-Ebene kann es erforderlich sein, das Hintergrundwissen an die neuen Gegebenheiten anzupassen. Grundsätzlich geht dies bei gleichbleibender Ausdrucksmächtigkeit mit einem deutlich geringeren Modellierungsaufwand einher.

Bei der mengenbasierten Inferenz wird nicht nur – wenn auch oft – die Gruppe von Agenten als Menge von Interesse verwendet, sondern alle in der Szene vorkommenden Objekte kommen dafür in Frage.

Die kompaktere Darstellung des Hintergrundwissens bei der mengenbasierten Inferenz darf nicht mit einer Einschränkung der ausdrückbaren Komplexität verwechselt werden. Ganz im Gegenteil: durch die Erweiterungen in Kapitel 4 werden alle ausprägbaren Situationen auf der Menge und deren Teilmengen gefunden. Insbesondere fallen darunter auch Situationen mit Beziehungen zwischen mehr als zwei Objekten.

In Abschnitt 7.5 Mengenbasierte Inferenz wird eine Auswertung auf dem BEHAVE-Datensatz mit dem in Abbildung 6.5 gezeigten Hintergrundwissen durchgeführt. In den Verhaltensschemata des SGT ebendort wird exzessiv von der mengenbasierten Inferenz Gebrauch gemacht.

Ein Vergleich der Laufzeit ohne und mit mengenbasierter Inferenz bei gleicher Ausdrucksmächtigkeit des verwendeten Hintergrundwissens ist nicht möglich. Für den BEHAVE-Datensatz und VIRAT-Datensatz liegt Hintergrundwissen ohne mengenbasierte Inferenz und mit mengenbasierter Inferenz vor. Die ausdrückbare Komplexität mit mengenbasierter Inferenz ist aber höher. Dennoch ist die Laufzeit über den gesamten betrachteten Datensatz bei VIRAT um 13% und bei BEHAVE um 2% reduziert.

6.4 Zusammenfassung

In einem ersten Schritt wird die Mean-Shift-Vorfilterung angewandt. In einem zweiten Schritt wird mit und innerhalb dieser Zwischenergebnisse die auf FMTHL und SGT basierende Situationserkennung ausgeführt.

Durch die neu eingeführte Modellierbarkeit auf Mengen ist der Prozess der Wissensmodellierung übersichtlicher und weniger umfangreich. Die Komplexität wird somit insgesamt auf zwei verschiedene Arten reduziert; zum einen kombinatorisch bezüglich der Inferenz und zum andern methodisch bezüglich der Modellierbarkeit. Es können nun mit geringerem Aufwand komplexere Sachverhalte modelliert und erkannt werden.

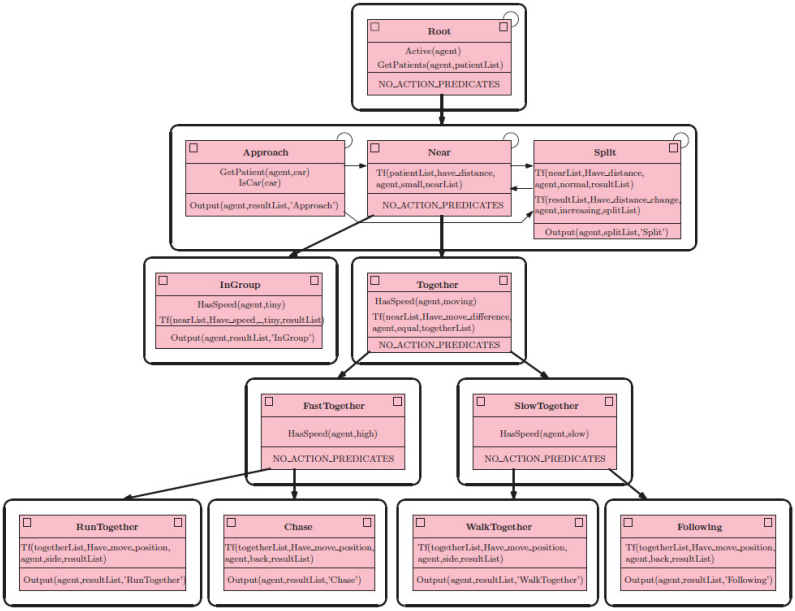


Abbildung 6.5: Das als SGT modellierte Verhaltenswissen für den BEHAVE-Datensatz. Zur besseren Lesbarkeit wurde ***truefilter*** durch Tf abgekürzt. Situationen **Approach**, **Split**, **InGroup**, **RunTogether**, **Chase**, **WalkTogether**, und **Following**.

7 Exemplarische Umsetzung und Auswertung

In diesem Kapitel wird die konkrete Umsetzung des Gesamtsystems der auf FMTHL und SGT basierenden Situationserkennung vorgestellt und konkrete Bildverarbeitungsmodule und die verwendeten Auswertungsmethoden erläutert. Anschließend werden die oben genannten methodischen Erweiterungen in den Formalismus der begrifflichen auf FMTHL und SGTs basierenden Situationserkennung ausgewertet.

7.1 Systemarchitektur

In Abbildung 1.1 ist die formale Architektur des CVS dargestellt, die ein kognitives Bildverarbeitungssystem realisiert. In diesem Abschnitt wird eine Systemarchitektur vorgestellt, die das CVS konkret technisch umsetzt. Eine Übersicht über die Systemarchitektur gibt Abbildung 7.1.

Im Begrifflichen Teilsystem sind die Werkzeuge zur Wissensmodellierung und die Inferenzmaschine zwei eigenständige Teile, was einerseits eine getrennte Ent- und Weiterentwicklung sowie eine bessere Wartbarkeit ermöglicht. Andererseits können Experten das Wissen für Diskursbereiche modellieren, ohne über die Inferenzmaschine Bescheid zu wissen. Diese Eigenschaft ist eine Voraussetzung, um die auf FMTHL und SGT basierende Situationserkennung diskursbereichsübergreifend einzusetzen.

Die modulare Systemarchitektur besteht im Wesentlichen aus den folgenden Komponenten:

- **Relationales Datenbankmanagementsystem (RDBMS)**: Zwischen Quantitativem und Begrifflichen Teilsystem wird über ein RDBMS kommuniziert. Alle Daten, die dem Quantitativen Teilsystem entstammen, wie z.B. Trajektorien, werden hier festgehalten. Ebenso werden alle Daten, die aus dem Begrifflichen Teilsystem kommen, wie z.B. ausgeprägte Situationen, abgelegt. Es gibt keine Einschränkung an Eingabedaten, solange sie in das erwartete Format konvertiert werden können. Beispielsweise wurden bei der Generierung vom Datensatz aus Abschnitt 7.3.5 die Daten vom lokalen Positionsmesssystem (LPM) adaptiert.
- **Wissensmodellierung**: Die Generierung von Hintergrundwissen als SGTs wird mit dem in Kapitel 3.2.5 beschriebenen SGTyEditor durchgeführt. Auch die Regelbasis wird hier entwickelt.
- **Inferenz**: Hier wird das Hintergrundwissen in Form von SGTs und die Regelbasis geladen, ebenso Zwischenergebnisse, und dann der Inferenzmaschine F-LIMETTE übertragen. Ein Adapter um F-LIMETTE startet den Inferenzprozess und verarbeitet die Ergebnisse.
- **Auswertung**: Sobald ein Experiment beendet ist, können mit der zugehörigen Grundwahrheit und den Ergebnissen die Maße zur Bewertung des Systems berechnet werden. Ein weiterer Bestandteil ist die Visualisierungskomponente, die bereits zur Laufzeit Grundwahrheit und Ergebnis darstellt.
- **Polling**: Im Realzeit- oder auch im Echtzeitbetrieb wird die Datenbank ständig nach neuen Daten durchsucht, die dann dem Begrifflichen Teilsystem zugänglich gemacht werden.

Durch den Ablauf der Kommunikation über das RDBMS ist diese Architektur sehr flexibel und günstig erweiterbar. Moderne RDBMS sind so leistungsfähig, dass sie gegenüber der deutlich aufwendigeren Inferenz in der vorgestellten Architektur nie den Flaschenhals darstellen werden.

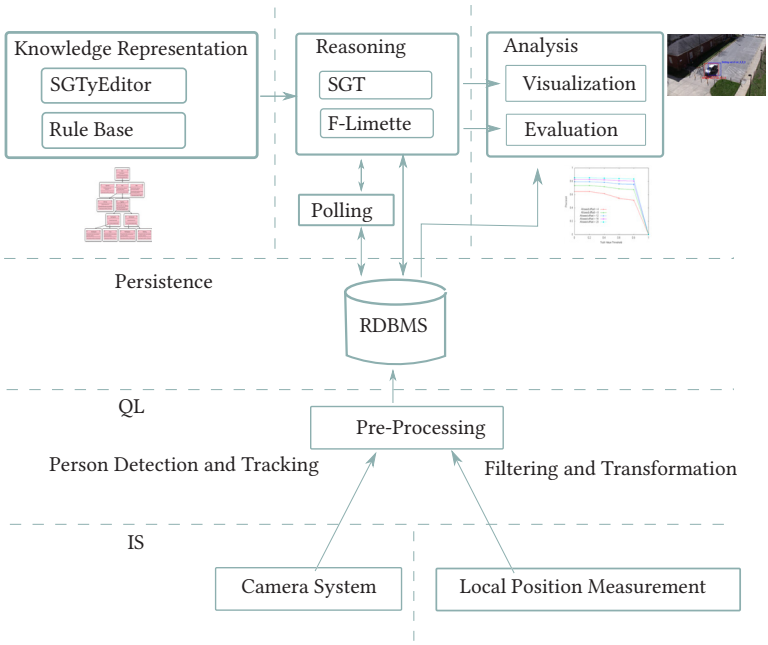


Abbildung 7.1: Systemarchitektur, die das CVS umsetzt. Die unterste Ebene dient der Datengewinnung und entspricht dem Interaktiven Teilsystem. Im vorliegenden Fall werden exemplarisch Kameras und ein LPM System zur Erzeugung von Trajektorien eingesetzt. In der Quantitativen Ebene sind die Bildverarbeitungsmodule wie Personendetektion und -tracking bzw. Filterung von Daten angesiedelt. Auf der höchsten Ebene befinden sich Module zur Wissensmodellierung, Inferenz und Auswertung. Alle Module kommunizieren über ein Relationales Datenbankmanagementsystem.

7.2 Bildverarbeitungsmodule

7.2.1 Objektdetektion

Um in Bildern Objekte zu erkennen bedient man sich eines Objektdetektors. In unserem Fall fokussieren wir uns auf Personen. Der verwendete Personendetektor basiert auf einer Variante von Integral Channel Features [Dollár et al., 2009], siehe Abbildung 7.2, bei dem jedoch nicht das Bild skaliert wird, sondern wie bei [Benenson et al., 2012] das gelernte Personenmodel.

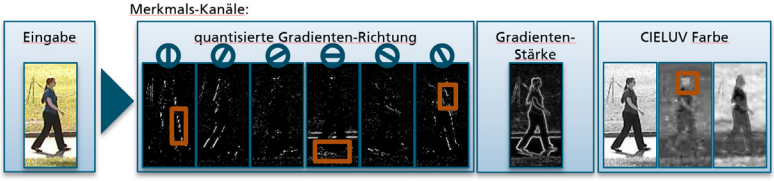


Abbildung 7.2: Exemplarische Visualisierung verwendeter Integral Channel Features als Basis für die Personendetektion.

Eine Verallgemeinerung aller auf Integral Channel Features basierenden Verfahren liefert [Zhang et al., 2015]. Dort wird die allgemeine Struktur des Klassifikators prägnant erklärt: „*The input image is transformed into a set of feature channels (also called feature maps), the feature vector is constructed by sum-pooling over a (large) set of rectangular regions. This feature vector is fed into a decision forest learned via Adaboost. The split nodes in the trees are a simple comparison between a feature value and a learned threshold. Commonly only a subset of the feature vector is used by the learned decision forest. Adaboost serves both for feature selection and for learning the thresholds in the split nodes.*“ Zur Detektion wird das gelernte Personenmodell skaliert und anhand eines Sliding Window Verfahrens über das gesamte Bild hinweg nach Personen gesucht. Einen Überblick über den allgemeinen Stand der Personendetektion liefert [Benenson et al., 2015].

7.2.2 Objektverfolgung

Eine Detektion in einem Einzelbild liefert zu einem Zeitpunkt eine Information. Mehr Information kann gewonnen werden, wenn der Zusammenhang zwischen Detektionen in Bildfolgen und konkretem Objekt konsistent hergestellt wird. Die Zuordnung einer Detektion an einem Ort und zu einer Zeit zu einem Objekt, das sich im Ortsraum bewegt, heißt Trajektorie. Die Trajektorie in einer Bildfolge entspricht der Objektbewegung in der beobachteten Szene. Die Auswahl[9] möglicher unterschiedlicher Objektverfolgungsverfahren ist groß, und zwar aus dem Grund, da viele unterschiedliche Verfahren in unterschiedlichen Anwendungsbereichen unter unterschiedlichen Randbedingungen unterschiedlich gut funktionieren.

Im Allgemeinen reicht pure Objektdetektion nicht aus, um Trajektorien zu erzeugen, da die Assoziationen der Objektdetektionen von Bild zu Bild nicht hergestellt werden können. Deshalb wird das von [Kieritz et al., 2016] vorgestellte Verfahren verwendet, da es einerseits nahtlos mit dem Objektdetektionsverfahren zusammenarbeitet, weil es auf der selben Merkmalsbasis aufbaut. Andererseits wendet es Onlinelearning zur ansichtsbasierten Diskriminierung verschiedener Objekte an. Das online gelernte ansichtsbasierte Modell bekommt eine besondere Bedeutung, wenn keine Detektionen mehr vorliegen und dann hiermit das Objekt weiter verfolgt werden kann und wird.

7.2.3 Aktionserkennung

Um Aussagen über die direkten Aktionen einer Person zu treffen, kann es notwendig sein, zusätzlich Aktionserkennung anzuwenden. Der Autor hat ein auf der generalisierten Hough-Transformation basierendes Verfahren zur Erkennung von sechs verschiedenen Aktionen (walk, jump, run, sit, wave and box) in Echtzeit umgesetzt [Münch et al., 2014, Zepf, 2012,

[9] https://motchallenge.net (19.03.2017)

Ramirez Fandiño, 2013, Ramirez Fandiño and Münch, 2014]. Diese Arbeiten wurden fortgeführt [Hilsenbeck et al., 2016b,a], indem sie auf Hough-Forests methodisch erweitert und in den thermischen Infrarotbereich übertragen wurden; ebenfalls wurde die Menge an zu erkennenden Aktionen vergrößert zu bend, handwave, jump, jump-in-place, run und walk und den kombinierten Aktionen sit-stand-up, run-fall, walksit und run-jump-walk. Gleichzeitig wurde ein Datensatz mit diesen Aktionen geschaffen.

7.3 Datensätze

Im Folgenden sind die in dieser Arbeit verwendeten Datensätze aufgeführt. Das System wurde absichtlich nicht nur an einem Datensatz ausgewertet, sondern an verschiedenen, die auch in unterschiedlichen Szenarien eines Diskursbereichs angesiedelt sind, um die Vielseitigkeit zu zeigen. Zusätzlich stehen dieser Arbeit eigene am Fraunhofer IOSB entwickelte Datensätze, sowie ein Onlinesystem, zur Verfügung. Sie dazu auch Anhang A.

Die Anzahl der freien, öffentlich verfügbaren Datensätze ist groß[10,11,12]. Dennoch ist der Großteil davon nicht für diese Arbeit geeignet, weil sie nicht ausreichend komplexe Situationen beinhalten. Deshalb wurden die im Folgenden näher beschriebenen Datensätze ausgewählt.

7.3.1 BEHAVE Interactions Test Case Scenarios

Inhaltlich finden sich im BEHAVE Datensatz[13] [Blunsden and Fisher, 2010] Situationen, in die mehrere Personen involviert sind. In Abbildung 7.3 sind Situationen des Datensatzes dargestellt. Insgesamt beinhaltet der Datensatz zehn verschiedene Situationen, die in Tabelle 7.1 aufgezählt sind.

[10] http://www.computervisiononline.com/datasets (19.03.2017)
[11] http://www.cvpapers.com/datasets.html (19.03.2017)
[12] http://datasets.visionbib.com/index.html (19.03.2017)
[13] http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/index.html (19.03.2017)

(a) (b) (c)

Abbildung 7.3: Drei beispielhafte Szenen des BEHAVE Datensatzes [Blunsden and Fisher, 2010]. In (a) ist Einzelbild 9260, in (b) Einzelbild 10789 und in (c) Einzelbild 44422 dargestellt. In (a) sind zwei Gruppen mit den Situationen „WalkTogether“, in (b) trennt sich die Gruppe gerade in zwei kleinere Gruppen „Split“ und in (c) rennt die gesamte Gruppe durch die Szene „RunTogether“.

Situation	Beschreibung
InGroup (IG)	Personen stehen zusammen als Gruppe.
Approach (A)	Zwei Personen oder Gruppen nähern sich einander.
WalkTogether (WT)	Personen gehen gemeinsam.
Meet (M)	Treffen von zwei oder mehreren Personen.
Split (S)	Weggehen von einer oder mehreren Personen von anderen Personen.
Ignore (I)	Die andere Person nicht beachten.
Chase (C)	Eine oder mehrere Personen jagen eine andere Person oder mehrere Personen.
Fight (FI)	Zwei oder mehrere Personen kämpfen miteinander.
RunTogether (RT)	Eine Gruppe von Personen rennt zusammen.
Following (FO)	Eine Person wird von einer anderen Person verfolgt.

Tabelle 7.1: Die verschiedenen Situationen beim BEHAVE Datensatz.

Sequenz	Grundw.	Trackingdaten	Situation Grundw.
1-11200	BEHAVE1	BEHAVE1_tracking	IG, A, WT, I, S, FO
11500-17450	n/a	BEHAVE2_tracking	WT, A, IG, S
18000-23700	BEHAVE3	BEHAVE3_tracking	S, WT, A, IG
24300-35200	BEHAVE4	BEHAVE4_tracking	n/a
35450-47160	BEHAVE5	BEHAVE5_tracking	C,RT,I,IG,WT,FI,S
47300-58400	BEHAVE6	BEHAVE6_tracking	A,M,RT,IG,WT,FI,S
59800-66750	BEHAVE7	BEHAVE7_tracking	A,IG,WT,FI,S
67210-76800	BEHAVE8	BEHAVE8_tracking	n/a

Tabelle 7.2: Videosequenzen aus BEHAVE Clip1 mit vorhandener Grundwahrheit und vorkommenden Situationen.

Der BEHAVE Datensatz, der eine zusammenhängende Aufzeichnung von über einer Stunde umfasst, besteht aus vier Clips, davon betrachten je zwei dieselbe Szene, jedoch aus einer anderen Kameraposition. Im Folgenden wird nur Clip1 (fight_margaret_1_24_01_2007.wmv) behandelt, weil nur für diesen Clip Grundwahrheiten vorhanden sind. Clip1 steht aufgeteilt in acht Sequenzen zur Verfügung, siehe dazu Tabelle 7.2. Insgesamt besteht Clip1 aus über 75.000 Einzelbildern in einer Größe von 640×480 Bildpunkten, die mit einer Frequenz von 25 Hz aufgezeichnet wurden. Insgesamt kommen 163 Instanzen von Situationen mit einem Umfang von 29.196 Einzelbildern vor. Konkrete Daten zur Bestimmung einer Abbildung von Bildpunkten auf die Bodenebene der Szene werden mitgeliefert.

7.3.2 VIRAT Video Dataset

Der VIRAT Release 1.0 Datensatz besteht aus 16 Szenen im Außenbereich. Alle Situationen werden von normalen Personen – keinen Schauspielern –

in der realen Welt dargestellt. Mit über sechs Stunden Videomaterial bietet dieser Datensatz ausreichend Material für eine gründliche Auswertung. Die Vorteile des VIRAT Datensatzes liegen in der Vielfalt der einzelnen Situationen innerhalb der Klassen, einer Vielzahl an unterschiedlichen Personen, sowohl stationärer als auch bewegter Aufzeichnungen [Oh et al., 2011], an unterschiedlichen Orten in den Vereinigten Staaten aufgenommene Daten, einer enormen Menge an aufgezeichneten Daten und in unterschiedlichen Bildgrößen und Aufnahmefrequenzen. In einer erweiterten zweiten Version des VIRAT Datensatzes kamen außerdem aus der Luft aufgenommene Videos ohne Grundwahrheit hinzu. In Abbildung 7.4 sind zwei beispielhafte Situationen des Datensatzes dargestellt. Insgesamt beinhaltet der Datensatz 13 verschiedene Situationen, die in Tabelle 7.3 aufgezählt sind.

In Tabelle 7.4 sind die in dieser Arbeit untersuchten Videos aufgeführt. Darin sind die Situationen „Objekt einladen“, „Objekt ausladen“, „Einsteigen“ und „Aussteigen“ von Interesse.

(a) (b)

Abbildung 7.4: Zwei Szenen aus dem VIRAT Datensatz, [Oh et al., 2011]. Parkplatz *VIRAT_S_000002* (a) und Parkplatz *VIRAT_S_000200* (b). Typische Situationen in diesem Kontext sind das Ein- und Aussteigen aus dem Kraftfahrzeug sowie das Ein- und Ausladen von Objekten in und von Kraftfahrzeugen.

Situation	Beschreibung
0	unbekannt
1	Person lädt Objekt in ein KFZ.
2	Person lädt ein Objekt aus einem KFZ aus.
3	Öffnen einer Autotür bzw. des Kofferraums.
5 resp. ***6***	In KFZ einsteigen resp. aussteigen.
7	Person gestikuliert.
8	Person gräbt.
9	Person trägt ein Objekt.
10	Person rennt.
11 resp. ***12***	Person betritt resp. verlässt ein Gebäude.

Tabelle 7.3: Die verschiedenen Situationen des VIRAT Datensatzes.

Sequenz	Grundwahrheit	Situation Grundwahrheit
VIRAT_S_000001	VIRAT001	1, 2, 5, 6, 9
VIRAT_S_000002	VIRAT002	2, 3, 4, 5, 6, 9
VIRAT_S_000003	VIRAT003	1, 2, 3, 4, 5, 6, 9
VIRAT_S_000004	VIRAT004	2, 3, 4, 5, 6
VIRAT_S_000006	VIRAT006	2, 3, 4, 5, 6
VIRAT_S_000200_06 _001693_001824	VIRAT200	3, 4, 6
VIRAT_S_000202_00 _000000_000977	VIRAT202	3, 4, 6

Tabelle 7.4: Videosequenzen aus dem VIRAT Datensatz mit vorhandener Grundwahrheit und darin vorkommenden Situationen.

7.3.3 PETS 2009 dataset

Für die Serie der „Performance Evaluation of Tracking and Surveillance“ Arbeitstreffen wurden verschiedene Datensätze geschaffen[14]. Inhaltlich basieren die meisten auf derselben Datenbasis, jedoch mit unterschiedlichen Schwerpunkten bezüglich der Auswertung des entsprechenden Treffens. Hier wird auf den PETS2009 Datensatz [Ellis et al., 2009, Ferryman and Shahrokni, 2009] näher eingegangen: Abbildung 7.5 zeigt einige Beispielsituationen aus dem Teil „S0: Training Data and Set city center“ und zwar „Time_12-34“ und „View_001“. Hauptsächlich bewegen sich viele Personen durch das Video und stellen einfache Situationen nach, wie „Treffen mehrerer Personen“ oder „Personen begeben sich in Bereiche abseits des Weges“. Diese Sequenz wurde ausgewählt, weil Grundwahrheit verfügbar ist und da viele Personen unterschiedliche Situationen nachstellen.

(a) (b) (c)

Abbildung 7.5: Drei beispielhafte Szenen des PETS 2009 Datensatzes [Ellis et al., 2009]. In (a) ist Einzelbild 36, in (b) Einzelbild 223 und in (c) Einzelbild 565 aus dem Teil „S0: Training Data und Set city center“ Time_12-34 und View_001 dargestellt. In (a) sind zwei Gruppen mit den Situationen „InGroup“ und „WalkTogether“ dargestellt, in (b) gehen einzelne Personen durch die Szene und eine „WalkTogether“ Situation kann identifiziert werden und in (c) betreten zwei Personen einen Bereich neben dem zulässigen Weg.

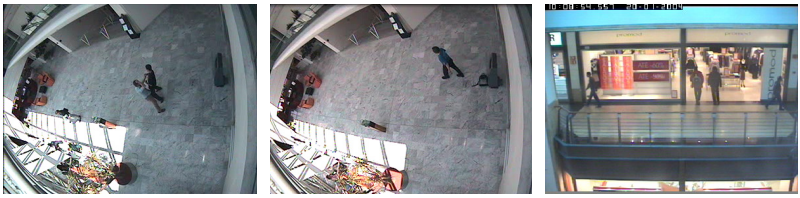
[14] http://ftp.cs.rdg.ac.uk/ (19.03.2017)

7.3.4 CAVIAR Test Case Scenarios

Der CAVIAR Datensatz [CAVIAR, 2001] besteht aus zwei großen Teilen: Zum einen aus einem Lobbybereich und zum andern aus dem Gang eines Kaufhauses. Die Videos wurden mit einer Größe von 384×288 Bildpunkten und mit einer Frequenz von 25 Hz aufgezeichnet. Insgesamt besteht der Datensatz aus weit über 100.000 Einzelbildern. Konkrete Daten zur Bestimmung einer Abbildung von Bildpunkten auf die jeweilige Bodenebene der beiden Szenen werden mitgeliefert.

Die in dem Datensatz vorkommenden Situationen umfassen „Walk“, „Browsing“, „Sturz“, „LeaveBag“, „InGroup“, „WalkTogether“, „SplitUp“, „Fight“, „EnterShop“, and „LeaveShop“. Sowohl die einzelnen Personen als auch die Situationen mit den jeweils beteiligten Personen sind weitestgehend annotiert. In Abbildung 7.6 sind drei beispielhafte Situationen aus dem CAVIAR Datensatz dargestellt.

Tabelle A.1 listet die verwendeten Sequenzen aus dem zweiten Teil des CAVIAR Datensatzes auf.



(a) (b) (c)

Abbildung 7.6: Drei beispielhafte Szenen des CAVIAR Datensatzes [CAVIAR, 2001]. In (a) sind zwei Personen mit der Situation „Fight“ dargestellt, in (b) geht eine einzelne Personen durch die Szene und stellt einen Koffer ab „LeaveBag“ und in (c) betreten zwei Personen ein Geschäft „EnterShop“.

7.3.5 VCA-Datensatz

Im Rahmen der Masterarbeit [Katumba, 2013] wurde am Fraunhofer IOSB ein eigener Datensatz zur Untersuchung von Situationen von mehreren Personen im Videoüberwachungskontext aufgezeichnet. Die konkreten Details des VCA-Datensatzes sind in Anhang A.2.1 beschrieben. Ähnlich zum BEHAVE Datensatz aus Abschnitt 7.3.1 behandelt der VCA-Datensatz über zwanzig verschiedene Situationen, siehe Tabelle A.2, die in ihrer Anzahl der beteiligten Aktoren grob in die Kategorie Einzelpersonen, Personenpaar und Gruppe eingeteilt werden können. Der Unterschied zu bestehenden Datensätzen wie z.B. in Abschnitt 7.3 besteht sowohl in der gleichzeitigen Aufzeichnung von Grundwahrheit der Personen, um damit eine spätere Annotierung zu vermeiden, als auch an der Komplexität und Menge der Situationen, die Gruppen betreffen.

7.4 Auswertungsmethoden

Die Auswertungsmethodik, die in die Architektur aus Abschnitt 7.1 integriert wurde, verwendet den intervallbasierten Ansatz aus [Oh et al., 2011].

7.4.1 Übereinstimmungskriterien auf Situationsebene

Die Übereinstimmungskriterien auf Situationsebene spezifizieren die Bedingungen, die erfüllt sein müssen, damit eine Detektion (D) einer Situation mit einer Situation (G) der Grundwahrheit übereinstimmt. Die Kriterien lauten wie folgt:

i) **Örtliche Übereinstimmung:** Die Detektion D wird als Übereinstimmung mit der Grundwahrheit G betrachtet, wenn das Verhältnis der Schnittmenge aller Paare von umgebenden Rechtecken pro Bild größer als 10% ist, Wert nach [Oh et al., 2011]. In Gleichung 7.1 ist die Berechnungsvorschrift definiert:

$$\text{Intersection Ratios} = \frac{\text{\# of intersected pixels}}{\text{Total \# of pixels in Bounding Box}} \tag{7.1}$$

ii) **Zeitliche Übereinstimmung:** Vor- und nachgelagerte zeitliche Schnitte zwischen D und G müssen größer als 10% sein, Wert nach [Oh et al., 2011]. Eine örtliche Übereinstimmung ist eine Vorbedingung für eine zeitliche Übereinstimmung. Das Verhältnis der temporalen Schnittmenge wird berechnet als Dauer der Schnittmenge geteilt durch Dauer der Vereinigung von D und G.

iii) **Begriffliche Übereinstimmung:** Eine weitere zusätzliche Bedingung ist die Übereinstimmung des Typs von D und G. Wenn z.B. D vom Typ *InGroup* ist, dann muss G das auch sein.

Mit diesen Kriterien können die von der Situationsanalyse generierten Ergebnisse in richtig positive, falsch positive, falsch negative und richtig negative Klassen eingeteilt werden.

7.4.2 Auswertungsmetriken

Aufbauend auf den oben genannten Übereinstimmungskriterien auf Situationsebene werden die folgenden Auswertungsmetriken berücksichtigt:

i) **Precision:** Die Precision gibt das Verhältnis der richtig positiv erkannten Situationen bzgl. aller erkannten Situationen an:

$$\text{Precision} = \frac{\text{\# of true positives (TPs)}}{\text{Total \# of Detections (TDs)}} \tag{7.2}$$

ii) **Recall:** Der Recall (PD) gibt das Verhältnis der richtig positiv erkannten Situationen bzgl. aller richtigen Situationen an:

$$\text{Recall} = \frac{\text{\# of true positives (TPs)}}{\text{Total \# of Ground Truth Events (T)}} \tag{7.3}$$

iii) **F-Score:** Die F-Score ist das gewichtete harmonische Mittel von Precision und Recall. Durch dieses kombinierte Maß sind verschiedene Ergebnisse leichter vergleichbar:

$$\text{F-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7.4}$$

Die Ausgabe, sowohl als reine Textdatei als auch im JSON-Format, erlaubt es, Schaubilder und andere Analysen durchzuführen. Die Skalierbarkeit ist gegeben, eine Erweiterung mit anderen Datensätzen und Auswertungsmetriken ist möglich.

7.4.3 Korrekte Detektionen und Falschalarme

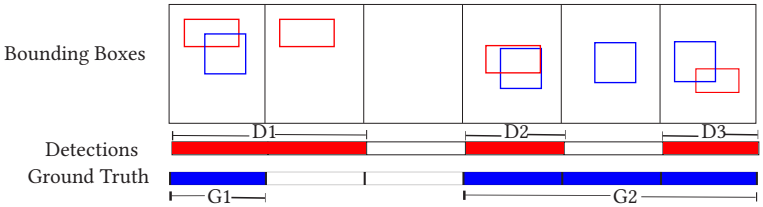
In der folgenden Diskussion repräsentieren rote umgebende Rechtecke detektierte Situationen und blaue umgebende Rechtecke die Grundwahrheit. Detektionen werden als D bezeichnet und sind fortlaufend nummeriert. Mit G wird die Grundwahrheit bezeichnet. Die in Abschnitt 7.4.1 genannten Übereinstimmungskriterien auf Situationsebene müssen im Folgenden erfüllt sein.

i) **Korrekte Detektionen**

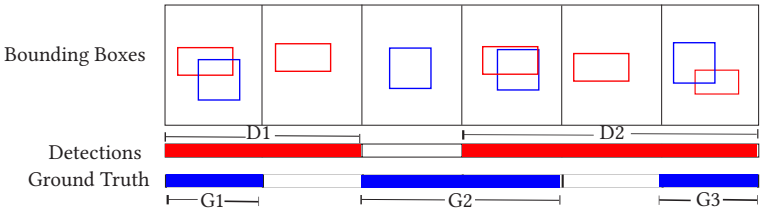
   a) Eine Situation aus der Grundwahrheit passt zu einer detektierten Situation. Dies zählt als korrekte Detektion für die Situation, siehe Abbildung 7.7 (a), dort G1 zu D1 und G2 zu D2 und G2 zu D3.

   b) Wie in Abbildung 7.7 (b) dargestellt, kann eine detektierte Situation auch zu mehreren Situationen passen, dort G2 und G3 zu D2. In diesem Fall trägt eine Detektion zu mehreren Situationen bei.

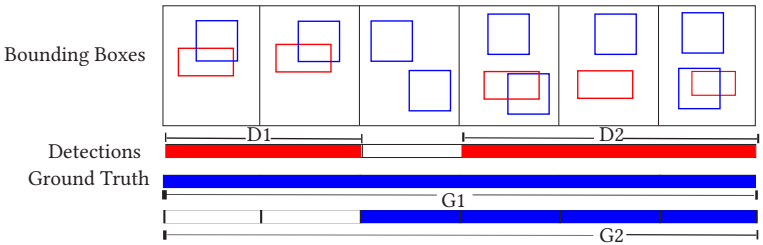
   c) Mehrere Detektionen und mehrere Situationen können zueinander passen wie in Abbildung 7.7 (c) dargestellt, dort G2 und G2 zu D2 und G1 zu D1 und D2. Hier wird die Auswertung entsprechend (a) und (b) durchgeführt.



(a) Eine Situation passt zu einer oder mehreren detektierten Situationen.



(b) Eine detektierte Situation passt zu mehreren Situationen.



(c) Mehrere detektierte Situationen und mehrere Situationen passen zusammen.

Abbildung 7.7: Kriterien zur Zählung korrekter Detektionen. Umgebende Rechtecke für die Grundwahrheit sind blau, die der detektierten Situationen rot [Oh et al., 2011].

d) Im Fall überlappender umgebender Rechtecke treten die folgenden in Abbildung 7.8 visualisierten Fälle auf:

Fall 1: Eine einzelne detektierte Situation überdeckt zwei Situationen: zwei Detektionen zählen als korrekte Detektion.

Fall 2: Eine einzelne detektierte Situation überlappt zwei Situationen: zwei Detektionen zählen als korrekte Detektion.

Fall 3: Zwei detektierte Situation überdecken zwei Situationen: zwei Detektionen zählen als korrekte Detektion.

Fall 4: Drei detektierte Situation überdecken zwei Situationen: zwei Detektionen zählen als korrekte Detektion. Keine Strafe für zu viele Detektionen.

ii) **Falschalarme** treten auf, wenn detektierte Situationen zu keiner existierenden Grundwahrheit passen und die in Abschnitt 7.4.1 genannten Übereinstimmungskriterien auf Situationsebene nicht erfüllt sind. In Abbildung 7.9 ist dieser Fall dargestellt.

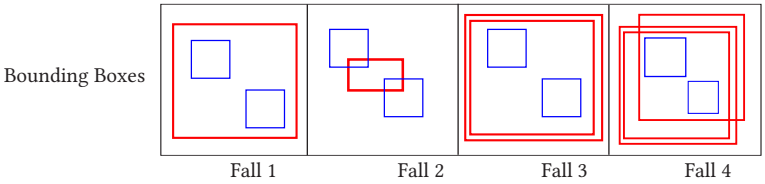


Abbildung 7.8: Kriterien zur Zählung korrekter Detektionen bei überlappenden umgebenden Rechtecken. Umgebende Rechtecke für die Grundwahrheit sind blau, die der detektierten Situationen rot [Oh et al., 2011].

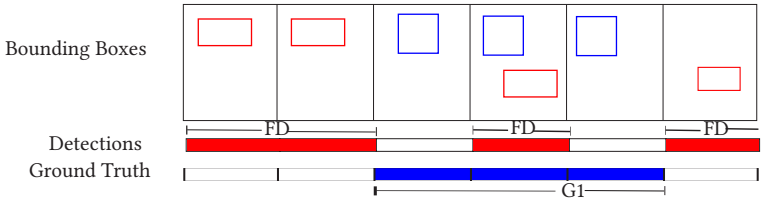


Abbildung 7.9: Basis zur Erfassung von Falschalarmen [Oh et al., 2011].

7.5 Auswertung

In diesem Abschnitt werden anhand von Experimenten die Auswirkungen der oben genannten methodischen Erweiterungen in den Formalismus der begrifflichen auf FMTHL und SGTs basierenden Situationserkennung ermittelt. Im Idealfall würden alle Auswertungen auf allen Datensätzen durchgeführt werden. Die oben genannten Datensätze sind allerdings sehr unterschiedlich und nicht alle Erweiterungen lassen sich auf allen Datensätzen sinnvoll zeigen. Zudem existieren nicht für alle Datensätze alle verschiedenen Arten von Hintergrundwissen – ohne/mit mengenbasierter Inferenz, ohne/mit Unschärfe, etc. Deshalb wurden einzelne geeignete Datensätze ausgewählt um einzelne Erweiterungen systematisch und exemplarisch auszuwerten.

Unschärfe

Die in Kapitel 4 eingeführte explizite Behandlung der Unsicherheit als Teil der Unschärfe wird auf dem PETS 2009 Datensatz (Abschnitt 7.3.3) ausgewertet. Siehe auch [Münch et al., 2011b].

In Tabelle 7.5 sind die Auswertungen der originären und der um Unschärfe erweiterten Situationserkennung auf dem PETS 2009 Datensatz aufgelistet.

	GT	Baseline				Unschärfe			
η	> 0.3	> 0.3	> 0.6	> 0.9	$= 1.0$	> 0.3	> 0.6	> 0.9	$= 1.0$
Precision	0.91	0.63	0.72	0.78	1	0.79	0.75	0.67	undef.
Recall	0.94	0.79	0.70	0.55	0.15	0.91	0.36	0.06	0
F-Score	0.93	0.70	0.71	0.64	0.26	0.85	0.49	0.11	undef.

Tabelle 7.5: Auswertung der originären und der um Unschärfe erweiterten Situationserkennung auf dem PETS 2009 Datensatz.

Der Datensatz ist nicht sehr schwierig, wie die Spalte „GT“ zeigt. Bei perfekten Eingabedaten sind Precision sowie Recall hoch. Bei der Anwendung des originären Verfahrens auf Personentrackingdaten werden die Ergebnisse der Situationserkennung nur mit η größer des angegebenen Schwellwertes betrachtet. Insgesamt sind die Ergebnisse schlechter als auf perfekten Daten. Stützt man sich bei der Auswertung auf Ergebnisse mit einem größeren Wahrheitswert η, dann nimmt zwar die Precision deutlich zu, geht aber mit einer wesentlichen Verschlechterung des Recalls einher. Bei der um die Unsicherheit als Teil der Unschärfe erweiterten Version sind die Wahrheitswerte insgesamt niedriger – was auch erwartet wird, weil die Eingebedaten nicht mehr mit Wahrheitswert Eins in die Situationserkennung eingehen. Allerdings ist eine signifikante Verbesserung von Precision und Recall zu beobachten bei kleinen η. Tatsächlich bildet sich in diesem Ergebnis der erwartete Fall ab: Schwache „falsche“ Eingabedaten werden eben in der erweiterten Version auch mit ihrer Unsicherheit berücksichtigt und nicht als total wahr betrachtet. Die Precision bei der um Unschärfe erweiterten Version sinkt zwar mit steigendem η; der tatsächliche Grund sind aber insgesamt niedrigere Wahrheitswerte, weil durch die Konfidenzwerte des Personentrackings initial kleinere Werte in die Situationserkennung eingehen und somit der Wahrheitswert des Gesamtergebnises auch niedriger ausfällt als bei perfekten Daten (GT) mit Wahrheitswert eins. Der Schwellwert η sollte immer entsprechend der Anwendung gesetzt werden. Es müssen immer die beiden konkurrierenden Ziele „nichts verpassen und

Fehlalarme akzeptieren“ und „lieber weniger aber alles richtig“ gegeneinander abgewogen werden.

Multihypothesen

Ebenfalls in Kapitel 4 wurde die erschöpfende Situationsgraphenbaumtraversierung engeführt, was eine Ausprägbarbeit von mehreren verschiedenen Situationen gleichzeitig und Mehrfachausprägungen desselben Situationsschemas erlaubt. Die Motivation dazu war eine möglichst vollständige Erfassung aller vorkommenden Situationen.

	GT	Baseline				Multihypothesen			
η	> 0.3	> 0.3	> 0.6	> 0.9	$= 1.0$	> 0.3	> 0.6	> 0.9	$= 1.0$
Precision	0.91	0.63	0.72	0.78	1	0.58	0.57	0.55	undef.
Recall	0.94	0.79	0.70	0.55	0.15	0.94	0.52	0.15	0
F-Score	0.93	0.70	0.71	0.64	0.26	0.72	0.54	0.24	undef.

Tabelle 7.6: Auswertung der originären und der um Multihypothesen erweiterten Situationserkennung auf dem PETS 2009 Datensatz.

In Tabelle 7.6 sind die Ergebnisse der Situationserkennung, erweitert um Multihypothesenfähigkeit, dargestellt. Auf den ersten Blick ist im Hinblick auf fast alle Kennzahlen eine Verschlechterung gegenüber der Baseline und gegenüber der in Tabelle 7.5 dargestellten Ergebnisse aufgetreten. Auf den zweiten Blick muss diese Aussage korrigiert werden, denn es wurden absolut mehr richtig positive Situationen erkannt, gleichzeitig aber auch mehr falsch positive Situationen. D.h. der Recall ist hoch – es werden weniger Situationen verpasst – aber gleichzeitig werden mehr falsch positive Situationen erkannt. Es muss beachtet werden, dass die Multihypothesenfähigkeit nicht nur performancegetrieben ist, sondern auch eine konzeptionelle Vorbedingung für bsp. die mengenbasierte Inferenz, um dort mit verschiedenen Teilmengen von Objekten gleichzeitig zu arbeiten.

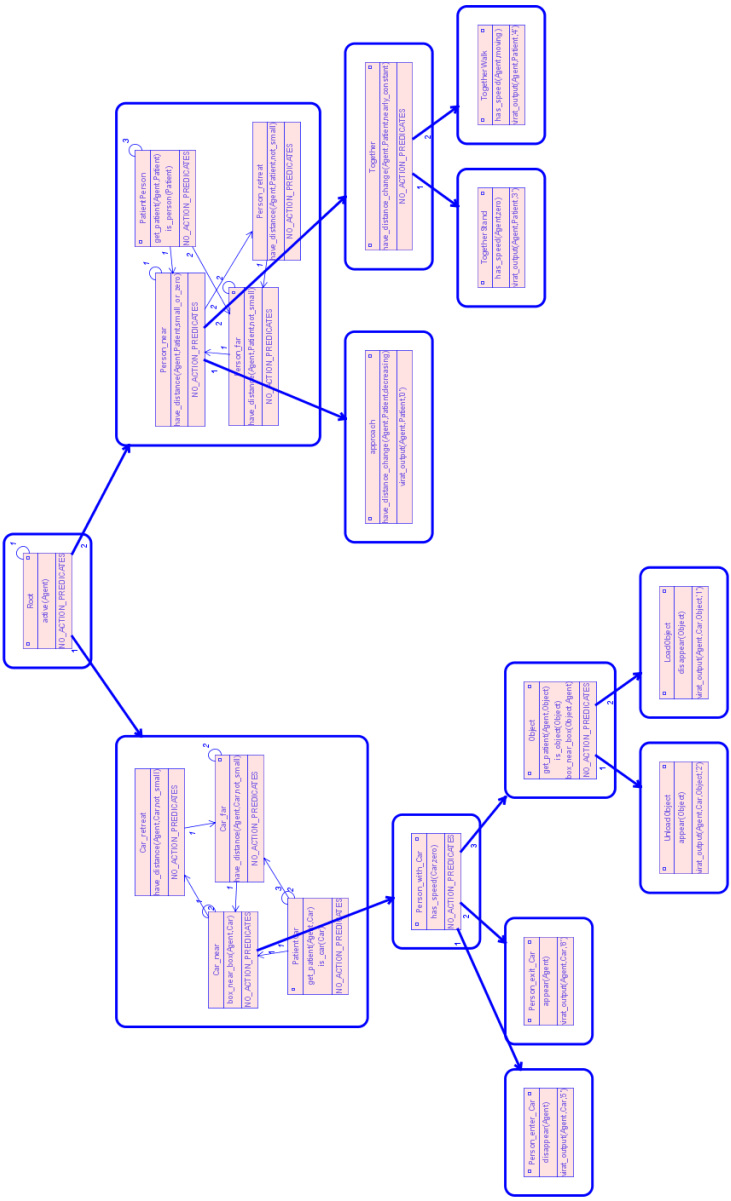


Abbildung 7.10: SGT, der die für den VIRAT Datensatz relevanten Situationen modelliert.

Vollständige verrauschte Daten

In Abbildung 7.10 ist das für den VIRAT Datensatz (Abschnitt 7.3.2) modellierte Hintergrundwissen über die zu erwartenden Situationen als SGT visualisiert. Die Experimente wurden mit den Sequenzen VIRAT_S_000002, VIRAT_S_000003 und VIRAT_S_000004 durchgeführt. Datenbasis sind die Grundwahrheiten, die dann schrittweise modifiziert werden. Im Gegensatz zu [Münch et al., 2012a] ist in den folgenden Experimenten die mengenbasierte Inferenz bereits eingeführt. Dadurch unterscheidet sich die Modellierung und infolgedessen die Ergebnisse.

	Baseline			Filter		
	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$
Precision	0.91	0.95	0.96	0.92	0.97	0.98
Recall	0.48	0.43	0.43	0.50	0.46	0.46
F-Score	0.62	0.60	0.59	0.65	0.63	0.62

Tabelle 7.7: Auswertung auf den vollständigen rohen und mit einem Rechteckfilter zeitlich gefilterten Daten aus VIRAT_S_000002.

	Baseline			Filter		
	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$
Precision	0.95	0.97	0.98	0.95	0.98	0.99
Recall	0.53	0.49	0.49	0.57	0.52	0.51
F-Score	0.68	0.66	0.65	0.71	0.68	0.68

Tabelle 7.8: Auswertung auf den vollständigen rohen und mit einem Rechteckfilter zeitlich gefilterten Daten aus VIRAT_S_000003.

In den Tabellen 7.7-7.9 sind die einzelnen Ergebnisse der Sequenzen VIRAT_S_000002/3/4 aufgelistet; und in Tabelle 7.10 zusammen aggregiert. Als Baseline dienen die rohen Grundwahrheitsdaten der Personen und Objekte. Die mit einem zeitlichen Rechteckfilter der Dimension fünf

gefilterten Daten führen zu gering besseren Ergebnissen - auch wenn absolut oft die falsch positiven Ergebnisse dadurch ansteigen. Insgesamt ist die Precision konstant hoch und der Recall zwischen 0.39 und 0.57.

Die Sequenzen des VIRAT Datensatzes werden mit $1Hz$ verarbeitet. Ein Filter der Dimension fünf erstreckt sich zeitlich absolut über fünf Sekunden. Über den VIRAT Datensatz hinweg hat sich ein Wert von fünf als brauchbar erwiesen. Tabelle 7.11 mit Filterdimension drei und sieben bestätigt dies exemplarisch auf VIRAT_S_000002. Zu große Filter glätten die Daten zu stark, sodass zu viel Information verloren geht und Situationen nicht mehr erkannt werden können.

	Baseline			Filter		
	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$
Precision	0.89	0.96	0.97	0.91	0.97	0.97
Recall	0.46	0.44	0.34	0.48	0.44	0.39
F-Score	0.61	0.60	0.51	0.64	0.60	0.52

Tabelle 7.9: Auswertung auf den vollständigen rohen und mit einem Rechteckfilter zeitlich gefilterten Daten aus VIRAT_S_000004.

	Baseline			Filter		
	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$
Precision	0.92	0.96	0.98	0.94	0.97	0.99
Recall	0.50	0.46	0.42	0.53	0.48	0.46
F-Score	0.65	0.62	0.59	0.67	0.64	0.62

Tabelle 7.10: Aggregierte Ergebnisse von Tabellen 7.7-7.9.

Unvollständige verrauschte Daten

Unvollständige Daten wurden entsprechend [Münch et al., 2012a] erzeugt. In den folgenden Experimenten der Tabellen 7.12-7.14 wurden 20% der Da-

ten zufällig entfernt. In Tabelle 7.15 sind alle drei zusammen aggregiert. Eine Auswertung, die die Menge der entfernten Daten untersucht, findet man in [Münch et al., 2012a].

	Filter Dimension 3			Filter Dimension 7		
	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$
Precision	0.92	0.96	0.97	0.90	0.93	0.97
Recall	0.49	0.49	0.45	0.43	0.37	0.35
F-Score	0.64	0.61	0.61	0.58	0.53	0.51

Tabelle 7.11: Auswertung auf den vollständigen rohen und mit einem Rechteckfilter zeitlich gefilterten Daten aus VIRAT_S_000002. Vgl. gegen Tabelle 7.7.

	Baseline			Interpolation		
	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$
Precision	0.74	0.87	0.93	0.87	0.95	0.98
Recall	0.28	0.22	0.21	0.48	0.43	0.43
F-Score	0.40	0.35	0.34	0.62	0.60	0.59

Tabelle 7.12: Auswertung auf den unvollständigen rohen und interpolierten Daten aus VIRAT_S_000002.

	Baseline			Interpolation		
	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$
Precision	0.88	0.95	0.99	0.93	0.97	1.00
Recall	0.40	0.35	0.34	0.55	0.51	0.50
F-Score	0.55	0.51	0.50	0.69	0.67	0.66

Tabelle 7.13: Auswertung auf den unvollständigen rohen und interpolierten Daten aus VIRAT_S_000003.

In allen drei Experimenten ist bei den unvollständigen rohen Daten ein signifikanter Performanceverlust gegenüber den vollständigen rohen Da-

ten sichtbar. Die Lücken in den Daten wirken sich deutlich auf die Performance der Logik aus. Demgegenüber leistet die Interpolation entsprechend Abschnitt 5.2 einen signifikanten Beitrag zur Steigerung der Performance bei unvollständigen Daten und führt im Allgemeinen zu deutlich robusteren Ergebnissen.

	Baseline			Interpolation		
	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$
Precision	0.85	0.93	0.96	0.90	0.96	0.98
Recall	0.37	0.32	0.28	0.47	0.43	0.39
F-Score	0.51	0.48	0.43	0.61	0.59	0.56

Tabelle 7.14: Auswertung auf den unvollständigen rohen und interpolierten Daten aus VIRAT_S_000004.

	Baseline			Interpolation		
	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$	$\eta > 0.3$	$\eta > 0.6$	$\eta > 0.9$
Precision	0.84	0.93	0.96	0.91	0.97	0.99
Recall	0.36	0.31	0.29	0.50	0.46	0.44
F-Score	0.50	0.47	0.44	0.65	0.62	0.61

Tabelle 7.15: Aggregierte Ergebnisse von Tabellen 7.12-7.14.

Kontrollierte Halluzination

Die kontrollierte Halluzination entfaltet ihre Stärke bei zeitlich ausgedehnten Situationsgraphen, wie es auch in Abschnitt 5.3.2 beispielhaft erläutert wird. Die hier verwendeten Datensätze sind in der Menge an zeitlich ausgedehnten komplexen Situationen arm. Daher kann die kontrollierte Halluzination auf den hier zur Verfügung stehenden Datensätzen nicht sinnvoll quantitativ ausgewertet werden.

Semantische Vorfilterung

In Tabelle 7.16 sind die Konfusionsmatrizen der Auswertungen auf BEHAVE2 (siehe Anhang 7.3.1) dargestellt. Der Recall ist bei beiden konstant hoch, d.h. es werden kaum Situationen verpasst. Dem gegenüber stehen aber falsch positive, die sich negativ auf die Precision auswirken. Ohne die semantische Vorfilterung ist – in diesem Beispiel – die Precision bei **InGroup** deutlich reduziert. Das rührt daher, dass eine Gruppe von Personen als Ganzes betrachtet wird und nicht mehr Situationen mit Teilen von Gruppen gebildet werden können. Der reduzierten Komplexität steht also eine u.U. geringere modellierbare Ausdrucksmächtigkeit gegenüber.

	WalkTogether	RunTogether	InGroup	Approach	Split	Chase	Following
WalkToget.	.45	.1	.3	.1	0	0	.05
InGroup	.15	.1	.6	0	.1	0	.05
Approach	.05	0	.3	.6	.05	0	0
Split	0	0	.25	.05	.7	0	0

	WalkTogether	RunTogether	InGroup	Approach	Split	Chase	Following
WalkToget.	.45	.1	.3	.1	0	0	.05
InGroup	0	0	.9	.05	.05	0	0
Approach	.05	0	.3	.6	.05	0	0
Split	0	0	.25	.05	.7	0	0

Tabelle 7.16: Konfusionsmatrix auf dem BEHAVE2 Datensatz ohne (oben) und mit (unten) semantischer Vorfilterung vom Prädikat ***distance_is***. Auf der linken Seite die Grundwahrheit, oben die erkannte Situation.

Mengenbasierte Inferenz

Eine Auswertung zur mengenbasierten Inferenz ist im Wesentlichen deshalb von der Baseline verschieden, weil das Hintergrundwissen jeweils anders modelliert werden muss. Was die Erkennungsleistung betrifft, sind sie theoretisch gleich – auch wenn die Modellierung bei der nicht mengenbasierten Inferenz unhandhabbar ist. Der Einfluss der mengenbasierten Inferenz wird in Abschnitt 6.3 diskutiert.

Eine Gesamtauswertung ohne bestimmte einzelne isolierte Teile zu untersuchen ist in den Abbildungen 7.11(a)-(d) mit Precision, Recall, F-Score und ROC-Kurve vom BEHAVE1 Datensatz dargestellt. Der erlaubte temporale Offset bei den erkannten Situationen variiert zu 1, 21, 41 und 61 Frames. Das entspricht 0.04s, 0.84s, 1.64s und 2.44s. Ein temporaler Offset darf nicht mit der zeitlichen Übereinstimmung in Abschnitt 7.4.1 verwechselt werden, sondern der hier eingeführte zeitliche Offset erlaubt, dass Detektion und Grundwahrheit um den zeitlichen Offset auseinander liegen dürfen. Im Graphen und in den weiteren Auswertungen für den BEHAVE-, CAVIAR- und VCA-Datensatz im Anhang C sieht man, dass oft kleinere Werte des Wahrheitswertes sowohl eine hohe Precision als auch einen hohen Recall verursachen. Der Einsatz eines Offsets hängt stark vom Datensatz und der Dauer seiner Situationen ab. Bei eher länger andauernden Situationen wie im BEHAVE Datensatz ist ein hoher Offset oft kontraproduktiv.

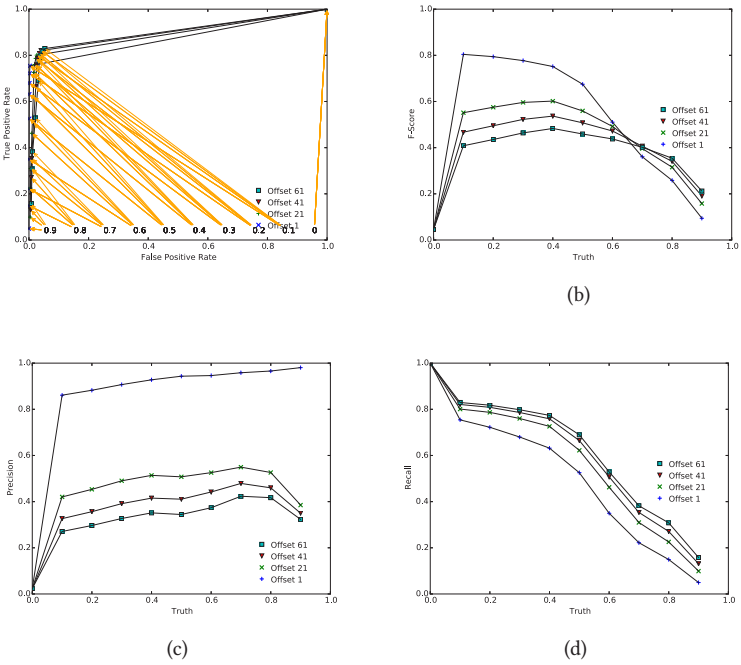


Abbildung 7.11: Gesamtauswertung auf BEHAVE1 Datensatz. (a) ROC (b) F-Score (c) Precision (d) Recall. Der erlaubte temporale Offset bei den erkannten Situationen variiert zu 1, 21, 41 und 61 Frames. Das entspricht 0.04s, 0.84s, 1.64s und 2.44s. Ein temporaler Offset darf nicht mit der zeitlichen Übereinstimmung in Abschnitt 7.4.1 verwechselt werden, sondern der hier eingeführte zeitliche Offset erlaubt, dass Detektion und Grundwahrheit um den zeitlichen Offset auseinander liegen dürfen.

8 Generische Anwendbarkeit

Bisher wurde in dieser Arbeit die Anwendbarkeit der auf FMTHL und SGTs basierenden Situationserkennung im Diskursbereich der Videoüberwachung im Innen- und Außenbereich gezeigt. Wünschenswert wäre, sie auch in anderen Diskursbereichen erfolgreich einsetzen zu können. Diese können sich sowohl inhaltlich als auch technisch-konzeptionell durch andersgeartete Sensoren und Aktoren von der beobachteten Szene unterscheiden. In diesem Kapitel soll die Frage beantwortet werden, ob eine generische Anwendbarkeit möglich ist. Dafür wird der vorliegende Ansatz erstmalig für die automatische Erkennung von Aktivitäten des täglichen Lebens (ADL) in intelligenten Umgebungen angewandt.

Von den öffentlich verfügbaren ADL Datensätzen werden zwei ausgewählt, die sich zum Einen dadurch auszeichnen, dass sie sehr umfangreich sind - sowohl in der Anzahl von Daten, als auch in der Anzahl vorkommender Situationen - und zum Andern eine ganz andere Art von Sensoren benutzen. Nachdem die beiden Datensätze unserem Ansatz zugeführt wurden und die Schnittstelle zur Datenerfassung angepasst worden ist, können die in diesen Datensätzen vorhandenen Situationen modelliert werden. Es hat sich gezeigt, dass durch eine personenspezifisch optimierte Wissensbasis die Erkennungsleistung gesteigert werden kann, indem lernende Verfahren eingesetzt werden um Verhalten personenspezifisch zu identifizieren und um schließlich die Modellierung von Hintergrundwissen zu unterstützen. Insgesamt kann das aus dem Diskursbereich Videoüberwachung vorhandene Basiswissen in der Begrifflich-Primitiven-Ebene in vollem Umfang, bis auf Schnittstellenanpassungen, wiederverwendet werden. In der

Verhaltens-Repräsentations-Ebene muss das Wissen über die zu erwartenden Situationen neu modelliert werden, zum einen wegen der völlig neuen Situationen, zum andern wegen der völlig unterschiedlichen Sensoren. Innerhalb des Diskursbereiches kann dagegen Wissen weitestgehend übertragen werden.

8.1 Aktivitäten des täglichen Lebens

Die Erkennung von Aktivitäten des täglichen Lebens in intelligenten Umgebungen ist von besonderem Interesse für unterstützende Anwendungen wie zum Beispiel die Verhaltenserkennung von Personen mit besonderen Bedürfnissen oder die Unterstützung von demenziell erkrankten Personen. In letzterem Fall könnte eine unterstützende Anwendung beispielsweise die Situation der Person erkennen, um über einen längeren Zeithorizont hinweg Unregelmäßigkeiten in der Abfolge aller Situationen zu identifizieren und um dann der Person entsprechende Unterstützung anzubieten; beispielsweise nur der akustische Hinweis, dass die Person sich nun zum zweiten Mal am selben Tag ein Frühstück zubereiten wolle. In der Anwendung möchte man rein aus praktischer Sicht auf am Körper getragene Sensoren verzichten. Ebenfalls sollen zur Gewährung der Privatsphäre möglichst keine Kameras eingesetzt werden. Infolgedessen werden dann Bewegungssensoren, Türkontakte, Drucksensoren, Wasserflusssensoren, Sensoren zur Messung der elektrischen Energie, usw. vorzugsweise eingesetzt.

8.2 Verwendete Datensätze

Es existiert eine Vielzahl von Datensätzen mit Aktivitäten des täglichen Lebens in intelligenten Umgebungen, siehe Tabelle D.1. Jedem dieser Datensätze liegt initial ein bestimmtes Forschungsziel zugrunde und daher unterscheiden sich die Datensätze deutlich voneinander. Wir stellen daher diese Anforderungen an einen Datensatz:

- Aktivitäten des täglichen Lebens von einzelnen Personen.
- Mehrfaches Vorkommen einzelner Situationen.
- Andere bzw. zusätzliche Sensorkonzepte zu Video-/Audiosensoren.
- Vollständig annotierte und korrekt aufgezeichnete Daten.
- Ausreichend viele Situationen und Daten.

Unter allen in Tabelle D.1 aufgelisteten Datensätzen erfüllen „DOMUS“ und „vanKasteren“ diese Anforderungen am besten.

8.2.1 DOMUS-Datensatz

Der DOMUS-Datensatz der Universität Sherbrooke wurde von [Chikhaoui et al., 2010] und [Kadouche et al., 2010] vorgestellt. Er umfasst sieben unterschiedliche typische morgendliche Situationen aus dem ADL Kontext. Der DOMUS-Datensatz ist in zwei Serien aufgeteilt: Eine Serie bestehend aus sechs Tagen pro Person und eine weitere Serie aus fünf Tagen pro Person für insgesamt sechs unterschiedliche Personen. Das Vorgehen bei der Annotation der Daten wird nicht erläutert.

Vorkommende Situationen

Die erste Serie enthält die Situationen **Wake Up**, **Use Toilet**, **Prepare Breakfast**, **Have Breakfast**, **Wash Dishes** und **Other**. Die zweite Serie enthält zusätzlich noch **Prepare Tea**. Die Situation **Other** wurde von uns nicht modelliert, weil keine Information über ihren Inhalt existiert. Zusätzlich unterscheiden sich die Sensordaten, die mit **Other** annotiert wurden, von Person zu Person und sogar von Tag zu Tag bei derselben Person.

Verfügbare Sensoren

Die Daten wurden von folgenden unterschiedlichen binären Sensoren aufgezeichnet:

- 5 Passive-Infrarot-Bewegungsmelder: Küche Spüle, Küche Backofen, Küche Toaster, Esszimmer und Wohnzimmer;
- 5 Lichtschalter: Küche, Esszimmer, Wohnzimmer, Schlafzimmer und Bad;
- 2 Türkontakte: Schlafzimmer und Bad;
- 4 Wasserflusssensoren: Bad Waschbecken, Bad Toilette, Küche Warmwasserhahn, Küche Kaltwasserhahn;
- 18 Türkontakte für die Küchenschränke und
- 1 Sensor für die Mikrowelle.

Die Sensoren liefern binäre Werte, und zwar `Open` oder `Close`. Nach [Kadouche et al., 2010] ändert jeder Sensor seinen Wert genau dann, wenn eine Handlung einer Person durchgeführt wird, also insbesondere wechselt ein Sensor nicht seinen Zustand von selbst. Abbildung 8.1 gibt einen Überblick vom Grundriss der Räume.

8.2.2 Der vanKasteren-Datensatz

Der vanKasteren-Datensatz wurde von [van Kasteren et al., 2011] vorgestellt. Im Gegensatz zu DOMUS haben die Sensoren ununterbrochen Daten aufgezeichnet. Der Datensatz gliedert sich in drei Teile (House A, House B und House C) mit unterschiedlichen Personen, Sensorkonzepten und Annotationsvorgehen; siehe Tabelle 8.1 in [van Kasteren et al., 2011]. Die Grundwahrheit zu den aktuell ausgeführten Situationen wurde live von den Personen selbst von Hand in ein Notizbuch oder per Sprache über ein Funk Headset aufgezeichnet. Daher kann man davon ausgehen, dass die Annotationen leicht unpräzise sind und ggf. kleinere Verzögerungen beinhalten. Weil die Annotationen im Notizbuch zu ungenau sind, wird

House B in unserem Fall nicht beachtet. Weil House C deutlich mehr Annotationen beinhaltet als House A, wird im Folgenden nur auf House C eingegangen. Der Grundriss ist in Abbildung 8.2 dargestellt.

Abbildung 8.1: Der Grundriss der Umgebung beim DOMUS-Datensatz. Eine Visualisierung der Positionen der Sensoren existiert nicht. Abbildung aus [Kadouche et al., 2010].

Vorkommende Situationen

Die folgenden Situationen werden modelliert: **Leave House**, **Eat**, **Use Toilet Downstairs**, **Take Shower**, **Brush Teeth**, **Use Toilet Upstairs**, **Shave**, **Go to Bed**, **Get Dressed**, **Take Medication**, **Prepare Breakfast**, **Prepare Lunch**, **Prepare Dinner**, **Get Snack**, **Get Drink** und **Relax**. Die ebenfalls in [van Kasteren et al., 2011] beschriebenen Situationen werden nicht modelliert, weil sie im gesamten Datensatz nicht oder maximal nur einmal vorkommen: **Take Bath**, **Put Items in Dishwasher**, **Unload Dishwasher**, **Store Groceries**, **Grooming**, **Put Clothes in Washing Machine**, **Unload Washing Machine**, **Receive Guest**, **Watch TV**, **Read Paper** und **Other**.

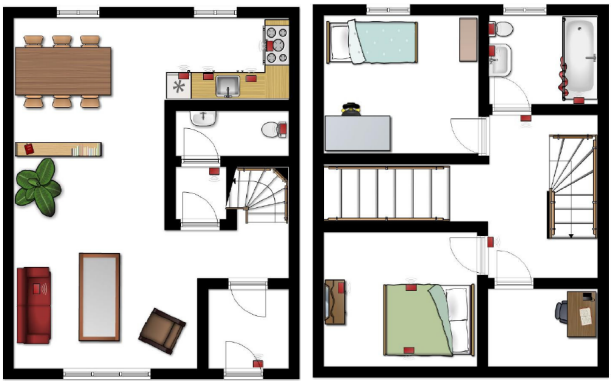


Abbildung 8.2: Der Grundriss der Umgebung beim vanKastern Datensatz. Abbildung aus [van Kasteren et al., 2011].

Verfügbare Sensoren

Folgende Sensoren zeichneten die Daten auf:

- 3 Drucksensoren: zwei im Bett und einer auf dem Sofa im Wohnzimmer;
- 11 Reedschalter: Vier an bekannten Schränken in der Küche, einer jeweils an Mikrowelle, Kühlschrank, Gefrierschrank, Badezimmertür, Toilettentür unten, Wohnungstür;
- 3 Schubladenbewegungsdetektoren: zwei für das Besteck und einer für die Schlüssel;
- 2 Passive-Infrarot-Bewegungsmelder: Schlafzimmer Ankleide und Bad Badewanne und
- 3 Wasserflusssensoren: Bad Toilette, Bad Waschbecken und Toilette oben.

Sensordaten werden in Bezug zur Aktivierungszeit aufgezeichnet. Dabei ist zu beachten, dass die Aktivierung der verschiedenen Sensoren unterschiedlich ist: ein aktiver Sensor an einer Tür bedeutet beispielsweise, dass diese geschlossen ist, wohingegen ein aktiver Wasserflusssensor am Waschbecken anzeigt, dass Wasser fließt. Alle Sensordaten sind über den gesamten Zeitraum von 19 Tagen durchgängig vorhanden.

8.3 Modellierung von Hintergrundwissen

In diesem Abschnitt wird das zur Erkennung von Situationen erforderliche Hintergrundwissen modelliert. Beginnend bei allgemeinen Basisregeln der Begrifflich-Primitiven-Ebene gelangt man zu den SGTs für den DOMUS und vanKasteren-Datensatz. Es werden auftretende Schwierigkeiten untersucht und durch lernende Verfahren gelöst.

8.3.1 Basisregeln

Die Zugänglichkeit zu den verschiedenen Sensoren wird in der Begrifflich-Primitiven-Ebene durch FMTHL-Prädikate modelliert. Im Folgenden die Signaturen:

- *getBedroomLamp(Lamp)* etc.
- *getKitchenLockerList(ItemList)*
- *morningTime* (analog: *forenoonTime*, ...)
- *isOpened(Sensor)* (analog: *isClosed*, *isTouched*)

Des Weiteren werden weitere Regeln der Basistaxonomie hinzugefügt:

- *inSmallInterval(Pred)*
  (analog: *inMediumSmallInterval*, ..., *inSmallFutureInterval*, ...)
- *ofteninMediumSmallFutureInterval*
  (analog: *sometimesinMediumSmallFutureInterval*, ...)
- *almostAlwaysInMediumSmallFutureInterval*
- *futureActionSequence*

- *domusActionOutput* (analog: *vkActionOutput*)
- *getBathroomItem(Item)*
- *justOpened(Sensor)* (analog *justClosed*, *justTouched*)
- *someJustOpened(SensorList)*
- *getSensorOutsideBathroom*
- *actionInKitchen, actionOutsideBathroom*

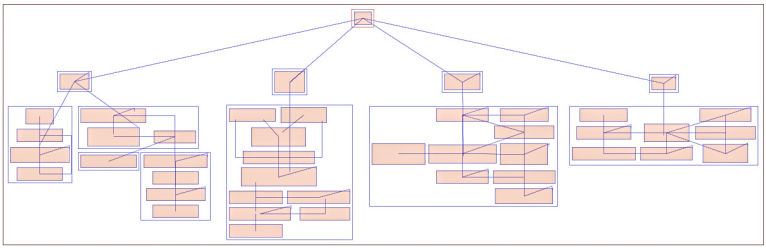


Abbildung 8.3: Übersicht der Struktur des SGTs für DOMUS. Deutlich sichtbar in der Struktur sind die vier disjunkten Teilzusammenhänge, die im Wurzelknoten zusammengeführt werden.

8.3.2 Situationsmodellierung für den DOMUS-Datensatz

Um das Verhalten der Personen aus dem DOMUS-Datensatz zu modellieren werden die temporale Modellierung von Teilschritten und die raumweise Erfassung von Sensordaten kombiniert. Die Notwendigkeit ergibt sich aus der relativ dünnen Sensorabdeckung der gesamten Szene. Man beachte, dass die Situation **Prepare Tea** im ersten Teil des Datensatzes nicht enthalten ist. Daher ist auch der SGT für diesen Teil um diese Situation reduziert. Abbildung 8.3 gibt eine Übersicht über die Struktur des SGT für DOMUS.

Der SGT enthält vier disjunkte Teilgraphen, wobei einer die Situation **Wake Up**, einer die Situation **Use Toilet**, einer die Situation **Prepare Breakfast** und einer die Situationen **Have Breakfast** und **Wash Dishes**

modelliert. Die Wurzelsituation fordert die agentenzentrierte Situationserkennung, indem das Zustandsschema *active(Agent)* erfüllt werden muss. Im Folgenden wird der Teilgraph - wiederum ein SGT - dargestellt auf der rechten Seite in Abbildung 8.3 näher betrachtet. Die weiteren SGTs sind in Anhang D.1 zu finden.

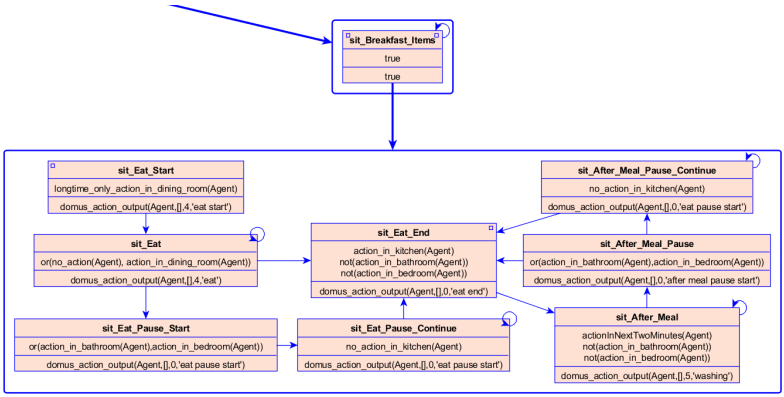


Abbildung 8.4: Der Teilgraph vom DOMUS SGT aus Abbildung 8.3, der die Situationen **Eat Breakfast** und **Wash Dishes** modelliert.

Die beiden Situationen Eat Breakfast und Wash Dishes

Die Modellierung des Hintergrundwissens für die Situationen **Eat Breakfast** und **Wash Dishes** (manchmal auch mit **Cleaning** bezeichnet) visualisiert Abbildung 8.4.

Man beachte, dass *not(actionInNextTwoMinutes(Agent))* nicht genug Information liefern würde, um den Beginn des Frühstücks zu beschreiben, weil es eine Vielzahl an weiteren erfolgreichen Belegungen für dieses Prädikat gibt, beispielsweise das Verharren auf der Toilette für mehr als zwei Minuten oder das Warten vor dem Wasser bis es kocht. Das Prädikat *longtime_only_action_in_dining_room(Agent)* beschreibt eine zweckmäßige Vorbedingung fürs Frühstück unter der Zusatzbedingung, dass sich

die Person im Esszimmer befindet. Weil in diesem Szenario keine Notwendigkeit besteht, zwischen **Prepare Tea** und **Washing Dishes** zu unterscheiden, ist es ausreichend, die Situationen **sit_After_Meal** recht grob wie in Abbildung 8.4 dargestellt zu modellieren.

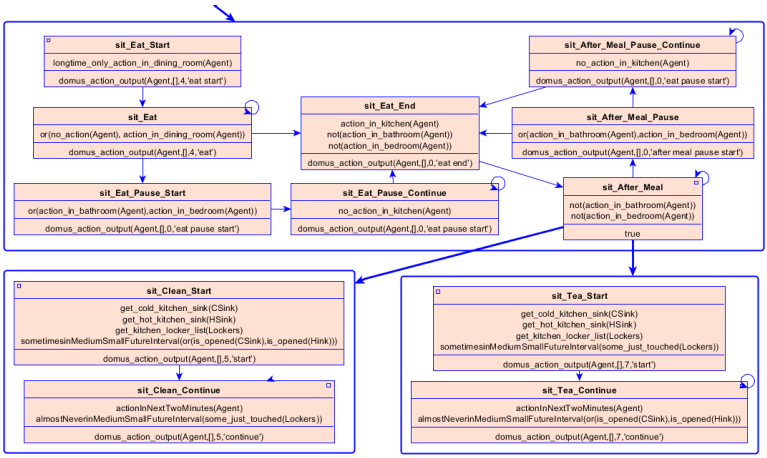


Abbildung 8.5: Ein Teilgraph des DOMUS SGT, umfassend die Situationen **Eat Breakfast**, **Wash Dishes** und **Prepare Tea**.

Die Situation Prepare Tea: Erster Versuch

Entsprechend der Beschreibung in Abschnitt 4.2 in [Chikhaoui et al., 2010] besteht die Situation **Prepare Tea** aus dem Erlernen eines Teezubereitungsrezepts, welches weniger als zehn Minuten dauern sollte. Gewöhnlich wird dies nach dem Frühstück durchgeführt, sodass lediglich die Situation **sit_After_Meal** (siehe Abbildung 8.4) angepasst werden muss.

Die Idee, die hinter dieser Modellierung steht, ist, dass die Person während der Teezubereitung normalerweise einige Küchenschränke öffnen und schließen muss, um an die einzelnen Gegenstände wie Tee, Tasse, oder Zucker zu gelangen, wohingegen beim Abwasch eher der Wasserhahn der

Spüle und der Passive-Infrarot-Bewegungsmelder ebendort aktiviert werden. Eine erste Auswertung anhand der Modellierung aus Abbildung 8.5 wird in Abbildung 8.6 (a) gezeigt. Die Ergebnisse für **Wake Up**, **Use Toilet**, **Prepare Breakfast** und **Have Breakfast** sind hervorragend. Jedoch gibt es offensichtlich Schwierigkeiten bei den beiden Situationen **Wash Dishes** und **Prepare Tea**. Diese Beobachtung gilt nach weiteren Auswertungen für den gesamten DOMUS-Datensatz. Im Folgenden wird eine mögliche Lösung vorgestellt, wie **Prepare Tea** und **Wash Dishes** besser diskriminiert werden können.

Zeitreihenanalyse

Bei der Zeitreihenanalyse aus dem Data-Mining möchte man nichttriviale 'interessante' Muster in zeitlich geordneten Daten entdecken. Eine interessante Übersicht darüber findet man in [Laxman and Sastry, 2006].

In den beiden hier untersuchten Datensätzen liegen zeitlich geordnete Sensordaten vor. Deshalb können wir das in [Mannila et al., 1997] vorgestellte Verfahren anwenden, bei dem Sequenzen von Events mit deren Eventtyp und Zeitpunkt beachtet werden. Bei diesem Verfahren sollen häufige Eventmuster in Eventsequenzen identifiziert werden. Das Vorkommen von Episoden wird nicht-überlappend gezählt, wie es auch in [Laxman, 2006] gehandhabt wird.

Formal ist ein *Event* ein Tupel (E, t) mit einem *Eventtyp* $E \in \mathcal{E}$ und einem *Zeitpunkt* $t \geq 0$. Hier steht $\mathcal{E}$ für die Menge aller möglichen Eventtypen. In unserem Fall bei Sensordaten in Intelligenten Umgebungen benutzen wir $\mathcal{E} = S \times \Sigma$, dabei bezeichnet S die Menge aller Sensoren und Σ die Menge aller möglichen Sensorzustände. Beispielsweise ist $(door, open)$ ein Eventtyp. Eine *Eventsequenz* ist eine Menge $\mathcal{S} = \{(E_1, t_1), ..., (E_N, t_N)\}$, sodass $E_i \in \mathcal{E}$ und $t_i \leq t_{i+1}$. Dabei bezeichnet N die Länge der Eventsequenz. Eine Eventsequenz der Länge 3 ist zum Beispiel:

$$\{(door, open, 1), (light, on, 10), (fridge, open, 16)\} \tag{8.1}$$

Eine *serielle Episode* ist ein Tupel $\alpha = (V_\alpha, \leq_\alpha, g_\alpha)$, mit einer Menge von *Knoten* V_α, einer totalen Ordnung $\leq_\alpha$ auf der Menge der Knoten und einer Abbildung $g_\alpha \colon V_\alpha \to \mathcal{E}$. Die Länge von α wird durch die Kardinalität von V_α bestimmt. Eine serielle Episode der Länge M kann man in der Form schreiben:

$$(v_1 \to v_2 \to ... \to v_M),$$

mit $V_\alpha = \{v_1, ..., v_M\}$ und $v_i \leq_\alpha v_{i+1}$ für alle i. Man beachte, dass diese Repräsentation nicht eindeutig sein muss. Man nehme die folgende serielle Episode als Beispiel:

$$((door, open) \to (fridge, open)). \tag{8.2}$$

Gegeben sei eine Eventsequenz Σ der Länge N, dann ist das *Vorkommen* einer Episode α in S eine Abbildung $h \colon V_\alpha \to \{1, ..., n\}$ so dass gilt $t_{h(v)} \leq t_{h(w)}$ für alle $v, w \in V_\alpha$ mit $v \leq_\alpha w$ und so dass $g_\alpha(v) = E_{h(v)}$ für alle $v \in V_\alpha$. Eine Episode α *kommt vor* in S genau dann, wenn es ein Vorkommen von α in S gibt. Als Beispiel siehe (8.2), wo die serielle Episode in der Eventsequenz (8.1) vorkommt. Man beachte besonders, dass das Konzept des Vorkommens nicht verlangt, dass die verbundenen Eventtyp aufeinander folgend vorkommen müssen, sondern dass auch andere Events dazwischenliegen können. Zwei Vorkommen h_1 und h_2 von α sind *nicht überlappend*, wenn entweder

1. $h_2(v_i) > h_1(v_j) \forall v_i, v_j \in V_\alpha$ oder
2. $h_1(v_i) < h_2(v_j) \forall v_i, v_j \in V_\alpha$ gilt.

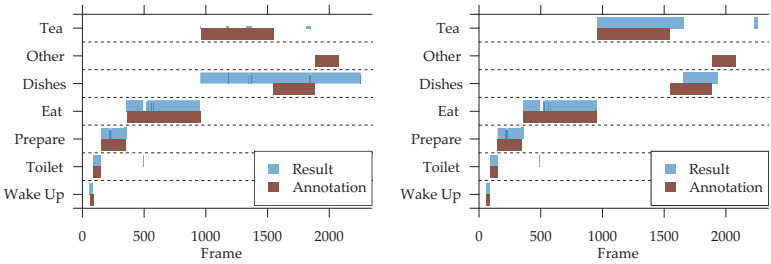
Weil sich in den bearbeiteten Szenarien die Situationen der einzelnen Personen gegenseitig ausschließen, sind wir nur an nicht überlappenden Vorkommen interessiert. Eine Menge von Vorkommen wird nicht überlappend genannt, wenn die sich darin befindenden Vorkommen paarweise nicht überlappen. Die Frequenz f für eine Episode α in S wird dann definiert

als die Kardinalität der größten Menge an nicht überlappenden Vorkommen von α in S. Eine Episode wird *häufig* genannt, wenn ihre Frequenz einen gewissen Schwellwert λ übersteigt. Ein effizienter Algorithmus zur Identifikation von häufigen Episoden mit nicht überlappenden Frequenzen wird in [Laxman, 2006] vorgestellt. Dieser Algorithmus basiert auf einem inkrementellen schrittweisen Vorgehen, in dem in jedem Schritt l häufige Muster $\mathcal{F}_l$ der Länge l basierend auf $\mathcal{F}_{l-1}$ und einer Menge von Kandidaten $\mathcal{C}_l$ für Schritt l berechnet werden, siehe auch [Laxman, 2006]. Eine Implementierung des Verfahrens bietet das Werkzeug TDMiner[15] von [Patnaik et al., 2008].

Die Situation Prepare Tea: Personenspezifisch gelernt

Die Durchführung der beiden Situationen **Prepare Tea** und **Wash Dishes** unterscheidet sich deutlich zwischen verschiedenen Personen: Manche Personen bereiten ihren Tee mit Milch zu, andere verzichten auf Milch, wieder andere spülen ihr Geschirr vor dem Essen, andere danach usw. Deshalb ist es nicht verwunderlich, dass die bisher auf personenunabhängigen Prädikaten aufbauende Modellierung mit diesen personenspezifischen Unterschieden nicht generalisierend funktioniert. In diesem Abschnitt wird der oben beschriebene, auf Zeitreihenanalyse aus dem Datamining basierende Ansatz, verwendet um diesem Problem zu begegnen. Die grundlegende Idee ist, basierend auf häufig Eventmustern, personenabhängige Prädikate des Zustandsschemas zu lernen. Diese gelernten personenabhängigen Prädikate erlauben nun die Diskriminierung zwischen **Prepare Tea** und **Wash Dishes**. In [Chikhaoui et al., 2010] werden Methoden der Zeitreihenanalyse zur Situationserkennung in intelligenten Umgebungen vorgestellt. Die dort verwendeten Verfahren werden auf das einfachere Problem der Klassifikation bei situationshomogenen Sensordaten angewendet.

[15] https://code.google.com/p/tdminer/ (19.03.2017)



(a) Erster Versuch.

(b) Personenspezifisch gelernt.

Abbildung 8.6: Ergebnisse ohne (a) und mit (b) der personenspezifischen Modellierung beim DOMUS set 2, user 1, day 5.

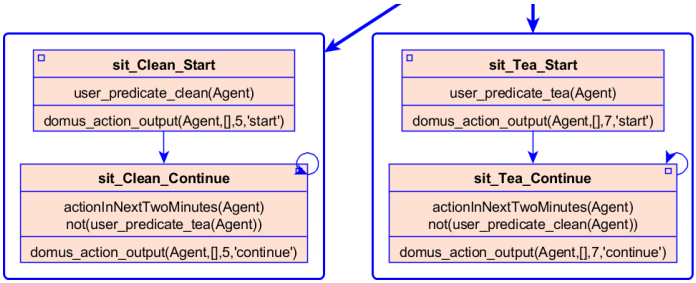


Abbildung 8.7: Die modifizierten Teile des personenspezifischen SGT, die die Situationen **Wash Dishes** und **Prepare Tea** modellieren.

Lediglich die spezialisierenden Teilgraphen unter **sit_After_Meal** vom SGT aus Abbildung 8.5 müssen modifiziert werden. Die Änderungen zeigt Abbildung 8.7. Die weiteren Teilgraphen verbleiben beim personenzentrierten Ansatz unangetastet. Mit den nun personenspezifisch gelernten Prädikaten kann sogar die zeitliche Bedingung, nämlich dass **Eat Breakfast** vor **Prepare Tea** stattfindet, fallen gelassen werden, eben weil die unterscheidenden Zustandspraedikate unabhängig vom zeitlichen Rahmen gelernt werden. Der SGT ist für alle Personen gleich, aber die *Definition*

der personenspezifischen Prädikate unterscheidet sich deutlich zwischen den verschiedenen Personen basierend auf den gelernten sequenziellen Mustern. Aktuell werden die personenspezifisch modellierten Prädikate in separaten Basistaxonomien gespeichert, die für jede Person jeweils geladen werden. In Abbildung 8.6 (b) sind die Ergebnisse dieses verbesserten Ansatzes dargestellt, die zeigt, dass die Diskriminierung zwischen **Prepare Tea** und **Wash Dishes** nun funktioniert.

8.3.3 Situationsmodellierung für den vanKasteren-Datensatz

In diesem Abschnitt wird das Hintergrundwissen der im vanKasteren-Datensatz erwarteten Situationen beschrieben. Wie bereits in Abschnitt 8.2.2 geschrieben, werden 16 von 27 Situationen betrachtet, da die verbleibenden elf Situationen nie oder maximal einmal im Datensatz auftreten. Im Vergleich zu dieser größeren Anzahl an verschiedenen Situationen ist die Sensorabdeckung, die ebendort beschrieben ist, recht dünn. Beispielsweise ist in der Küche und im Wohnzimmer kein Passiver-Infrarot-Bewegungsmelder, sodass der Modellierung der dort vorkommenden Situationen besondere Beachtung geschenkt werden muss. Die grundlegende Struktur des SGT für den vanKasteren-Datensatz gibt Abbildung 8.8.

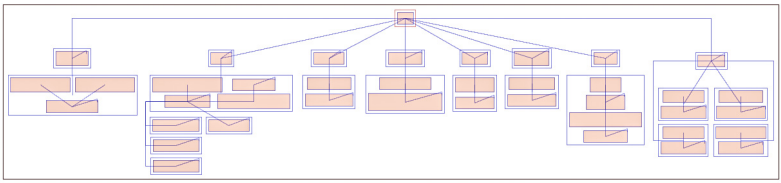


Abbildung 8.8: Übersicht der Struktur des SGT für vanKasteren. Deutlich sichtbar in der Struktur sind die acht disjunkten Teilzusammenhänge, die im Wurzelknoten zusammengeführt werden.

An dieser Stelle werden zwei Teilgraphen aus Abbildung 8.8 näher diskutiert. Alle anderen Teile sind im Anhang D.2 zu finden.

In Zusammenhang mit Mahlzeiten stehende Situationen

Abbildung 8.9 zeigt die Modellierung der Situationen, die im Zusammenhang mit Mahlzeiten stehen. Die initiale Vorbedingung zur Zubereitung einer Mahlzeit wird durch die Situation **sit_Prepare_Meal_Start** modelliert. Die Unterscheidung zwischen verschiedenen Arten der zubereiteten Mahlzeiten (Breakfast, Lunch, Dinner und Snack) wird durch die Spezialisierungen der Situation **sit_Prepare_Meal_Continue** übernommen, in der die Tageszeit betrachtende Prädikate (siehe Abschnitt 8.3.1) verwendet werden. Im Vergleich zu DOMUS gibt es kein Esszimmer sowie keinen Passiven-Infrarot-Bewegungsmelder im Wohnzimmer. Da der Drucksensor im Sofa der einzige Sensor im Wohnzimmer ist, wird damit der Beginn einer Mahlzeit betreffenden Situation in **sit_Eat_Start** modelliert.

Situationen im Badezimmer

Abbildung 8.10 visualisiert die Situationen, die sich im Badezimmer abspielen. Die Prädikate der Zustandsschemata modellieren die entsprechenden Voraussetzungen zum Start der einzelnen Situationen. Die darauf zeitlich folgenden Situationen werden sich gegenseitig ausschließend modelliert, sodass die Zustandsschemata mit **_continue_predicate* erfüllt bleiben, solange kein Event außerhalb des aktuellen Raumes stattfindet und kein anderes Zustandsschema mit **_start_predicate* im aktuellen Raum erfüllend belegt werden kann. Die Definition von **_start_predicate* und **_continue_predicate* befindet sich ausführlich in Anhang D.3. Das Prädikat *shower_start_predicate* verlangt eine signifikante Aktivierung des Sensors in der Badewanne, was genau dann gegeben ist, wenn eine Person sich in der Badewanne bewegt. Das Prädikat *toilet_start_predicate* setzt lediglich die Benutzung des Wasserflusses am Toilettenspülkasten voraus.

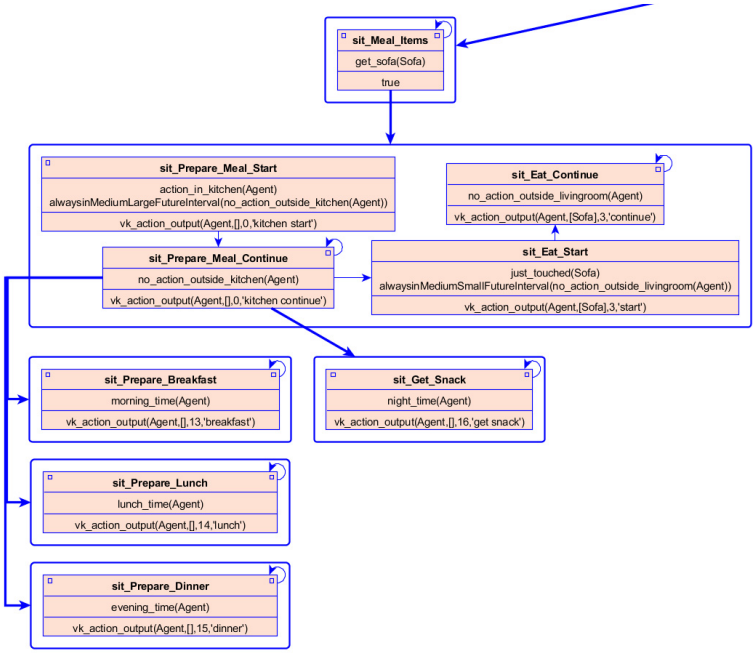


Abbildung 8.9: Ein Teilgraph vom vanKasteren SGT, der folgenden Situationen umfasst: 13: **Prepare Breakfast**, 14: **Prepare Lunch**, 15: **Prepare Dinner**, 16: **Get Snack** and 3: **Eat**.

Im vanKasteren-Datensatz gibt es keine Sensoren, die es erlauben würden den Schrank, die Zahnbürste oder den Rasierer zu identifizieren. Daher ist es dem Experten auf den ersten Blick nicht klar, wie die Situationen **Brush Teeth** und **Shave** zu modellieren sind. Eine Möglichkeit wäre die Benutzung des Waschbeckens, jedoch ist dies stark von den Gewohnheiten einer Person abhängig und ohne Vorwissen über die Personen und ihre speziellen Verhaltensweisen am Waschbecken nur schwer möglich. Daher werden diese Prädikate mit dem oben vorgestellten Verfahren der Zeitreihenanalyse aus dem Datamining versucht zu lernen.

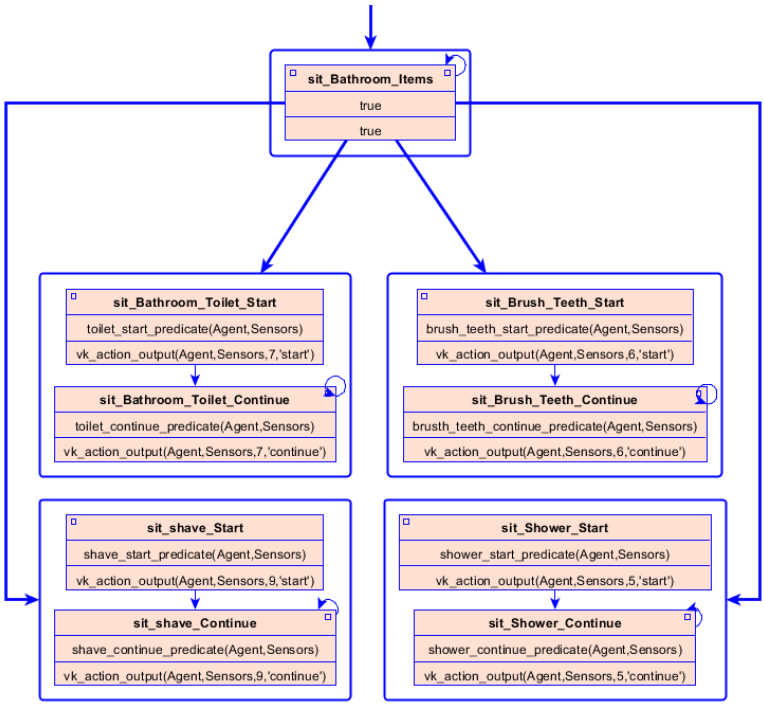


Abbildung 8.10: Teilgraph vom vanKasteren SGT mit den Situationen: 5: **Take Shower**, 6: **Brush Teeth**, 7: **Use Toilet Upstairs** und 9: **Shave**.

Im vanKasteren-Datensatz wird die in Abschnitt 8.3.2 vorgestellte Zeitreihenanalyse an den Tagen 11-19 durchgeführt. Überraschenderweise ist es nicht möglich, für die Situation **Brush Teeth** brauchbare Ergebnisse zu erzielen. Nach einem tieferen Blick in die Daten wäre eine Erklärung, dass sich die Personen beim Zähneputzen durch die Wohnung bewegen und sich manchmal sogar auf das Sofa setzen. Die führt dazu, dass keine sinnvolle Modellierung von **Brush Teeth** möglich ist. Eine weitere mögliche Erklärung könnte auch die recht ungenaue Annotation des Datensatzes sein, die dazu führt, dass die Zeitspanne des tatsächlichen Zähneputzens und der an-

notierten Situation **Brush Teeth** nicht übereinstimmt. Daher wird **Brush Teeth** nicht in die Modellierung der Situationen mit aufgenommen und das Prädikat *brush_teeth_start_predicate* bleibt undefiniert. Demgegenüber steht die Situation **Shave**, bei der das Lernen von häufigen Eventmustern erfolgreich ist, was dazu führt, dass das Prädikat *shave_start_predicate* auf Events vom Wasserflusssensor des Wasserhahns basiert.

8.4 Auswertung

In Tabelle 8.1 ist die Gesamtperformance beim DOMUS-Datensatz dargestellt. Die obere Tabelle zeigt die kummulierten Ergebnisse beim ersten Versuch und die untere Tabelle zeigt die kummulierten Ergebnisse bei personenspezifisch gelernten Hintergrundwissen. Insgesamt sieht man, dass die F-Score von 0.76 auf 0.9 deutlich ansteigt.

Activity	1	2	3	4	5	7	Overall
Frames in GT	33	307	734	1683	3164	2166	8340
Precision	1	0.98	0.98	0.95	0.85	0.09	0.85
Recall	1	0.92	1	1	0.93	0.02	0.69
F-Score	1	0.95	0.99	0.97	0.89	0.03	0.76

Activity	1	2	3	4	5	7	Overall
Frames in GT	33	307	734	1683	3164	2166	8340
Precision	1	0.98	0.98	0.95	1	0.92	0.94
Recall	1	0.92	1	1	0.68	1	0.87
F-Score	1	0.95	0.99	0.97	0.81	0.96	0.9

Tabelle 8.1: Vergleich der kummulierten Ergebnisse für Tag 4 und 5 der Einwohner 1 und 2 in Set 2. (oben) Kummulierte Ergebnisse für den ersten Versuch. (unten) Kummulierte Ergebnisse für personenspezifisch gelernt.

Tabelle 8.2 listet die Gesamtperformance beim vanKasteren-Datensatz auf. Einige wenige Situationen werden schlecht erkannt. Das liegt oft am geringen Vorkommen im Datensatz und an der fehlenden Beschreibung der Situationen.

Activity	Frames in GT	Precision	Recall	F-Score
Relax	62766	0.37	0.33	0.35
Get Drink	706	0	0	⊥
Get Snack	584	0.03	0.6	0.05
Prepare Dinner	10107	0.3	0.74	0.43
Prepare Lunch	2664	0.19	0.88	0.32
Prepare Breakfast	6152	0.59	0.92	0.72
Take Medication	424	⊥	0	⊥
Get Dressed	1427	0.98	0.69	0.81
Go to Bed	259817	0.98	1	0.99
Shave	1442	0.27	0.2	0.23
Toilet Upstairs	2156	0.3	0.83	0.44
Brush Teeth	3063	⊥	0	⊥
Take Shower	3765	0.59	1	0.74
Toilet Downstairs	3483	0.24	0.75	0.36
Eat	26210	0.28	0.8	0.42
Leave house	308063	0.99	1	0.99
Overall	692829	0.8	0.91	0.85

Tabelle 8.2: Kummulierte Ergebnisse für Days 1-10 vom vanKasteren-Datensatz House C.

8.5 Übertragbarkeit von Wissen

Wenn man den DOMUS-Datensatz mit dem vanKasteren-Datensatz vergleicht, findet man einige Unterschiede. Erstens enthalten die beiden Datensätze unterschiedliche Situationen, bis auf **Use Toilet** und **Prepare Breakfast**. Zweitens verwenden beide Datensätze unterschiedliches Hintergrundwissen: So gibt es beispielsweise bei DOMUS als Mahlzeit lediglich **Prepare Breakfast**, während bei vanKasteren zwischen Frühstück, Mittagessen, Abendessen und Imbiss unterschieden wird. Drittens unterscheiden sich beide Datensätze im verwendeten Sensorkonzept. Daraus folgt, dass das für beide Datensätze verwendete Hintergrundwissen Unterschiede aufweisen wird. Zum Beispiel existiert beim vanKasteren-Datensatz kein Wasserflusssensor in der Küche oder kein Sensor für den Lichtschalter auf der Toilette. Das fordert eine unterschiedliche Modellierung von **Use Toilet** in den Datensätzen.

Andererseits haben beide Datensätze viele Gemeinsamkeiten: Erstens sind beide Datensätze aus dem Diskursbereich ADL. Zweitens sind in beiden Datensätzen dieselben Basisregeln aus der Begrifflich-Primitive-Ebene anwendbar, siehe dazu Abschnitt 8.3.1: Die binären Sensorprädikate, die raumweisen Aktionsprädikate, die zeitlichen Intervallprädikate, die Ausgabeprädikate, die Tageszeit betrachtenden Prädikate sind unabhängig vom Datensatz und können für weitere Modelle verwendet werden. Drittens verwenden beide Wissensbasen ähnliche zeitliche Abfolgen der betrachteten Situationen: Die Situation **Eat** wird in beiden Datensätzen als temporale Nachfolgesituation von Essen vorbereiten modelliert. Des Weiteren verwenden beide das Konzept der ortsbasierten Schlussfolgerung. In beiden Datensätzen ist die grundlegende Idee der modellierten Situation **Use Toilet** Event im Badezimmer und kein Event außerhalb des Badezimmers für eine kurze Zeit (siehe Abbildung D.2). Besonders für diese Situation ist die Schlussfolgerung entsprechend der Modellierung präziser, weil es einen Sensor für den Lichtschalter im Badezimmer gibt. Ohne die explizite

Modellierung des Sensors für den Lichtschalter würde die Situationserkennung für **Use Toilet** ebenfalls funktionieren, aber mit geringerer Erkennungsleistung. Die ortsbasierte Schlussfolgerung ist ein sehr generisches Konzept für die Modellierung von Hintergrundwissen.

Das hier geschaffene Hintergrundwissen ist eine Vorlage für weitere intelligente Umgebungen. Um es an neue Anwendungen anzupassen, müssen die jeweiligen Unterschiede der Situationen, des Hintergrundwissens und des Sensorkonzepts berücksichtigt werden.

9 Zusammenfassung und Ausblick

9.1 Zusammenfassung

Diese Arbeit beschäftigt sich mit der Erkennung komplexer Situationen in Bildfolgen im Videoüberwachungskontext. Konzeptionell liegt dem in Form von FMTHL und SGTs umgesetzten Ansatz das Cognitive Vision System zugrunde. Dies transformiert u.a. Bildsensordaten in semantische Beschreibungen, die dann zur Schlussfolgerung über das Auftreten von erwarteten Situationen der durch die Bildsensoren beobachteten Szene verwendet werden können.

Das Ziel dieser Arbeit war die automatische Erkennung von Situationen aus Videodaten. Bei der Behandlung von Daten aus natürlichen Umgebungen ergeben sich verschiedene Schwierigkeiten, wie z.B. Mess-, Quantisierungs- und Verfahrensfehler. Diese sind unvermeidbar. Daher muss die Situationserkennung explizit mit diesen Effekten umgehen können. Die Frage, die diese Arbeit beantwortet hat, lautete, wie die Robustheit der Erkennung von komplexen Situationen in natürlichen Umgebungen aufrecht erhalten werden kann, in denen fehlerbehaftete, unvollständige und verrauschte Daten verarbeitet werden.

Im Folgenden werden die einzelnen Beiträge dieser Arbeit aufgezählt: Die SGT-Traversierung und deren zugrunde liegenden Konzepte wurden um die Behandlung von *Unschärfe* und die *Ableitbarkeit aller zutreffenden Hypothesen* in Kapitel 4 erweitert. Damit ist jetzt eine erschöpfende Beschreibung aller gleichzeitig möglichen Situationen gegeben sowie das Propagieren und Benutzen von Unschärfe vom Sensor bis zur Situation. In

Kapitel 5 wurden *Vorverarbeitungsfilter* auf verschiedenen Ebenen eingeführt. Zusätzlich wurde das Konzept der *kontrollierten Halluzination* entwickelt um auf der Verhaltens-Repräsentations-Ebene Unvollständigkeiten zu kompensieren. Die Modellierbarkeit von Abhängigkeiten von Objekten wurde auf *mengenbasierte Konzepte* in Kapitel 6 erweitert. Mit reduziertem Aufwand können nun komplexere Situationen modelliert werden. Eine weitere Reduktion der Komplexität wird durch eine *Ballungsbildung von ähnlichen semantischen Informationen* ermöglicht. Es wurden verschiedene öffentlich verfügbare *Datensätze* der Videoüberwachung im Innen- und Außenbereich als auch ein eigener Datensatz verwendet. Die für diesen Diskursbereich notwendige Basistaxonomie in Form von FMTHL Prädikaten wurde erweitert und das modellierte Hintergrundwissen als SGTs wurde neu geschaffen und auf jedes Szenario entsprechend angepasst. Die generische Anwendbarkeit wird in Kapitel 8 gezeigt, indem der Ansatz auf die Erkennung von Aktivitäten des täglichen Lebens in intelligenten Umgebungen eingesetzt wurde. Ebenfalls an dieser Stelle wird gezeigt, dass lernende Verfahren die Modellierung von Hintergrundwissen sinnvoll unterstützen.

Insgesamt tragen die einzelnen Beiträge dazu bei, dass die Robustheit der Situationserkennung bei natürlichen Szenarien trotz der dort auftretenden Komplexität und der fehlerbehafteten, unvollständigen und verrauschten Informationen aufrecht erhalten werden kann.

9.2 Ausblick

Während der Durchführung dieser Arbeit wurden Anknüpfungspunkte für weitere Arbeiten identifiziert. Der Ansatz kann auf weitere Diskursbereiche übertragen werden und die hier vorgestellten Methoden auch dort eingesetzt werden.

In Kapitel 8 wurde gezeigt, dass lernende Verfahren zur Identifikation von aussagekräftigen Merkmalen eingesetzt werden können. In der Arbeit

[IJsselmuiden, 2014] wurden lernende Verfahren eingesetzt um Parameter für bekannte Regeln zu lernen. Die beiden Ansätze versuchen die diskriminativen und bedeutungstragenden Merkmale zu identifizieren bzw. zu schärfen. Bei Teilaspekten, bei denen ausreichend Trainingsdaten verfügbar sind, sollte weiter untersucht werden, wie diese Merkmale zu identifizieren sind um die systemimmanente Semantik nicht zu verlieren. Auf dieser Ebene, wo die quantitativen Daten zueinander in Beziehung gesetzt werden, spricht konzeptionell nichts gegen lernende, unterstützende Verfahren.

Davon zu unterscheiden sind höhere Konzepte, wie sie in der Verhaltens-Repräsentations-Ebene, aber auch schon in der Begrifflich-Primitiven-Ebene, modelliert werden. Die Frage, die bei modellbasierten Verfahren, die Regeln verwenden, immer gestellt wird, lautet, ob diese Regeln nicht gelernt werden könnten. Cum hoc ergo propter hoc, oder mit anderen Worten die Korrelation bzw. Koinzidenz von Ereignissen lässt nicht ohne gesonderte Betrachtung einen Kausalschluss zu. Interessante Ansätze bezüglich dieser grundsätzlichen Fragestellung findet man in den Arbeiten [Pearl, 2009, Pearl et al., 2016].

Ein interessanter Aspekt wäre auch zu untersuchen, inwieweit das in SGTs repräsentierte Verhaltenswissen für die Generierung von Verhalten – wie beispielsweise in Spielen oder bei autonomen Fahrzeugen – eingesetzt werden kann. In [Arens and Nagel, 2003] wurden dazu erste Überlegungen angestellt. Dort wird die *Controlled Imagery Generation* als inverses Problem der Situationsanalyse beschrieben, aufbauend auf dem selben Hintergrundwissen.

Es wurde in dieser Arbeit gezeigt, dass die syntaktische Struktur von SGTs regulär ist. Die darauf abgeleiteten formalen Eigenschaften erlauben starke Ableitungen über Ausdrucksmächtigkeit und algorithmischer Komplexität bezüglich Speicher und Laufzeit. Wie lassen sich diese Eigenschaften effizient ausnutzen um einen minimalen SGT zu bestimmen, der das Hintergrundwissen redundanzfrei und laufzeitvorteilhaft modelliert?

Über der gesamten Arbeit steht außerdem die Frage, inwieweit modellbasierte Ansätze ihre Berechtigung haben. Einerseits sind Methoden des Deep Learnings erfolgreich und funktionieren in der Praxis gut, andererseits bauen sie auf einer großen Trainigsdatenbasis auf. Folglich kommen diese Methoden genau dann an ihre Grenzen, wenn zu einem Verhalten nicht viele Trainingsdaten existieren. Das ist in der Regel immer dann der Fall, wenn die Verhalten komplexer sind, d.h. nicht eine einzelne Aktion umfassen, sondern zusammengesetzte und interaktive Handlungen zwischen verschiedenen Objekten sind oder eben auch, wenn ein Verhalten selten ist. Zeitlich ausgedehnte und komplexe Verhalten, die ein Verständnis des kausalen Zusammenhangs voraussetzen, werden weiterhin modelliert werden müssen. Zukünftige Fragestellungen lauten daher: Bis zu welcher semantischen Ebene können Methoden aus der raffinierten Statistik die Situationserkennung sinnvoll unterstützen?

A Datensätze

Im Folgenden sind weitere Details zu den in dieser Arbeit verwendeten Datensätzen aufgeführt.

A.1 Frei verfügbare Datensätze

Die meisten der frei verfügbaren Datensätze werden in Abschnitt 7.3 vorgestellt. Tabelle A.1 listet den zweiten Teil des CAVIAR Datensatzes auf.

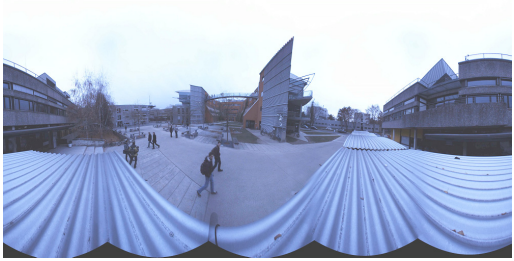
A.1.1 Virtual PTZ (vPTZ)

Der vPTZ Datensatz ist kein Datensatz im eigentlichen Sinne, sondern besteht lediglich aus zwei hochaufgelösten Videosequenzen als Datenbasis für die Implementierung einer virtuellen PTZ-Kamera. In [Possegger et al., 2012] wird eine der Realität entsprechende virtuelle PTZ-Kamera inklusive dem dazugehörigen Programmcode vorgestellt. Damit ist es möglich auf einem Übersichtsvideo mit hoher Auflösung eine oder mehrere PTZ-Kameras zu simulieren. Dies ermöglicht erstmals auch die offline Auswertung von Verfahren, die PTZ-Kameras benutzen.

Die beiden Videosequenzen, siehe auch Abbildung A.1, wurden mit 5400×2700 Bildpunkten und 15 Hz mit einer Ladybug® 3 von Point-Grey aufgezeichnet. Eine der beiden Videosequenzen wurde im Außenbereich aufgezeichnet und beinhaltet vorwiegend weniger komplexe Situationen wie „WalkTogether“, „InGroup“, „SplitUp“ und „PassBy“. Die andere Videosequenz wurde in einer Sporthalle aufgezeichnet. Die Grundwahrheit der Personen und der stattfindenden Situationen wurde manuell ergänzt.

Sequenz und Person GT	Tracking-daten [Bäuml et al., 2012]	Trackingdaten	Situation Grund-wahrheit
EnterExitCrossingPaths1cor	CAVIAR01	CAVIAR01_tracking	GT_CAVIAR01.txt
EnterExitCrossingPaths2cor	CAVIAR02	CAVIAR02_tracking	N/A
OneLeaveShop1cor	CAVIAR03	CAVIAR03_tracking	GT_CAVIAR03.txt
OneLeaveShop2cor	CAVIAR04	CAVIAR04_tracking	GT_CAVIAR04.txt
OneLeaveShopReenter1cor	CAVIAR05	CAVIAR05_tracking	GT_CAVIAR05.txt
OneLeaveShopReenter2cor	CAVIAR06	CAVIAR06_tracking	GT_CAVIAR06.txt
OneShopOneWait1cor	CAVIAR07	CAVIAR07_tracking	GT_CAVIAR07.txt
OneShopOneWait2cor	CAVIAR08	CAVIAR08_tracking	GT_CAVIAR08.txt
OneStopEnter1cor	CAVIAR09	CAVIAR09_tracking	GT_CAVIAR09.txt
OneStopEnter2cor	CAVIAR10	CAVIAR10_tracking	GT_CAVIAR10.txt
OneStopMoveEnter1cor	CAVIAR11	CAVIAR11_tracking	GT_CAVIAR11.txt
OneStopMoveEnter2cor	CAVIAR12	CAVIAR12_tracking	GT_CAVIAR12.txt
OneStopMoveNoEnter1cor	CAVIAR13	CAVIAR13_tracking	GT_CAVIAR13.txt
OneStopMoveNoEnter2cor	CAVIAR14	CAVIAR14_tracking	GT_CAVIAR14.txt
OneStopNoEnter1cor	CAVIAR15	CAVIAR15_tracking	N/A
OneStopNoEnter2cor	CAVIAR16	CAVIAR16_tracking	GT_CAVIAR16.txt
ShopAssistant1cor	CAVIAR17	CAVIAR17_tracking	GT_CAVIAR17.txt
ShopAssistant2cor	CAVIAR18	CAVIAR18_tracking	GT_CAVIAR18.txt
ThreePastShop1cor	CAVIAR19	CAVIAR19_tracking	GT_CAVIAR19.txt
ThreePastShop2cor	CAVIAR20	CAVIAR20_tracking	GT_CAVIAR20.txt
TwoEnterShop1cor	CAVIAR21	CAVIAR21_tracking	GT_CAVIAR21.txt
TwoEnterShop2cor	CAVIAR22	CAVIAR22_tracking	GT_CAVIAR22.txt
TwoEnterShop3cor	CAVIAR23	CAVIAR23_tracking	GT_CAVIAR23.txt
TwoLeaveShop1cor	CAVIAR24	CAVIAR24_tracking	GT_CAVIAR24.txt
TwoLeaveShop2cor	CAVIAR25	CAVIAR25_tracking	GT_CAVIAR25.txt
EnterExitCrossingPaths1front	CAVIAR26	CAVIAR26_tracking	N/A
EnterExitCrossingPaths2front	CAVIAR27	CAVIAR27_tracking	GT_CAVIAR27.txt
OneLeaveShop1front	CAVIAR28	CAVIAR28_tracking	N/A
OneLeaveShop2front	CAVIAR29	CAVIAR29_tracking	N/A
OneLeaveShopReenter1front	CAVIAR30	CAVIAR30_tracking	N/A
OneLeaveShopReenter2front	CAVIAR31	CAVIAR31_tracking	N/A
OneShopOneWait1front	CAVIAR32	CAVIAR32_tracking	GT_CAVIAR32.txt
OneShopOneWait2front	CAVIAR33	CAVIAR33_tracking	GT_CAVIAR33.txt
OneStopEnter1front	CAVIAR34	CAVIAR34_tracking	N/A
OneStopMoveEnter1front	CAVIAR36	CAVIAR36_tracking	GT_CAVIAR36.txt

Tabelle A.1: Videosequenzen aus der zweiten Hälfte des CAVIAR Datensatzes mit Trackingdaten und darin vorkommenden Situationen.



(a)

(b)

Abbildung A.1: Zwei beispielhafte Szenen des vPTZ Datensatzes [Possegger et al., 2012]. In (a) sind mehrere Personenpaare mit der Situation „WalkTogether“ dargestellt, in (b) gehen die unstrukturiert über das Spielfeld.

A.2 Eigene Datensätze

Frei verfügbare erzeugte Datensätze decken oftmals einen großen Bereich von Interesse ab. Im Spezialfall um die besonderen Stärken und Schwächen eines konkreten Systems zu verdeutlichen fehlen dann aber oft bestimmte Daten um gezielt besondere Charakteristiken aufzuzeigen. Dann ist die Erfassung eigener Daten meist unumgänglich. Ein weiterer Grund für eigene Daten wird dann ersichtlich, wenn man live am eigenen System dessen Eigenschaften demonstrieren möchte.

A.2.1 VCA-Datensatz

Im Rahmen der Masterarbeit [Katumba, 2013] wurde am Fraunhofer IOSB ein eigener Datensatz zur Untersuchung von Situationen von mehreren Personen im Videoüberwachungskontext aufgezeichnet. Die konkreten technischen Details zur Aufzeichnung des VCA-Datensatzes sind in Abschnitt A.2.1 Technische Details beschrieben. Ähnlich zum BEHAVE Datensatz aus Abschnitt 7.3.1 behandelt der VCA-Datensatz über zwanzig verschiedene Situationen, siehe Tabelle A.2, die in ihrer Anzahl der beteiligten Aktoren, grob in die Kategorie Einzelpersonen, Personenpaar und Gruppe eingeteilt werden können. Der Unterschied zu bestehenden Datensätzen wie z.B. in Anhang A.1 besteht sowohl in der gleichzeitigen Aufzeichnung von Grundwahrheit der Personen um damit eine spätere Annotierung zu vermeiden, als auch an der Komplexität und Menge der Situationen Gruppen betreffend.

Erfassung der Grundwahrheit der Personen

Eine besondere Eigenschaft des VCA-Datensatzes ist die automatische Erfassung der Grundwahrheit der beteiligten Personen. Das wurde dadurch realisiert, dass die Videoaufzeichnung synchron mit dem kabellosen lokalen Positionsmesssystem (LPM) von der inmotiotec GmbH der abatec group durchgeführt wurde. In Abbildung A.2 (b) ist der schematische Aufbau des LPM beschrieben. Das LPM System nutzt eine Kombination von Basisstationen, deren Position genau mit dem Tachymeter eingemessen wurden, und Transpondern, von denen jede Person zwei trägt, um die exakten Positionen der Personen in der Szene zu messen. Detaillierte technische Informationen sind in Abschnitt A.2.1 Lokales Positionsmesssystem (LPM) aufgeführt.

Die Vorteile einer exakten Aufzeichnung von Grundwahrheit sind wie folgt:

- Keine zweitaufwändige manuelle Annotierung der Grundwahrheit notwendig.
- Keine Fehler im Datensatz durch fehlerhafte manuelle Annotierung.
- Der Datensatz kann zur einfacheren Auswertung für Situationserkennung und andere bildverarbeitenden Verfahren wie z.B. Personendetektion und -verfolgung benutzt werden.

Zwei Personen, die nebeneinander durch die beobachtete Szene rennen, sind in Abbildung A.3 (a) zu sehen. Man beachte auch die aufgebauten schwarzen Basisstationen des LPM am Rand der Szene, als auch die große Tiefe der Szene, die diesen Datensatz mit seiner großen Skalierungsunterschiede besonders herausfordernd macht.

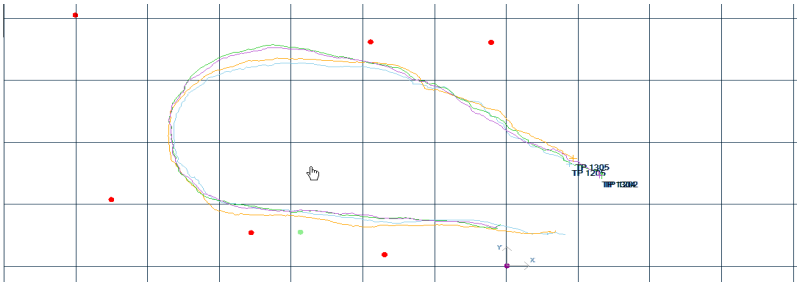
Nachdem die Videoaufzeichnungen abgeschlossen und die Spurdaten vom LPM übernommen worden waren, konnte die Nachbearbeitung des Datensatzes durchgeführt werden, im Einzelnen umfasste sie:

- Stabilisierung der LPM Daten durch Anwendung eines Kalman Filters.
- Überprüfung und ggf. Anpassung der Synchronisation der Einzelbilder und der LPM Daten.
- Bestimmung einer Abbildung von den Weltkoordinaten des LPM in die Bildkoordinaten der Videoaufzeichnung um die Position der einzelnen Personen in Bildkoordinaten zu kennen damit diese ggf. visuell markiert werden können.

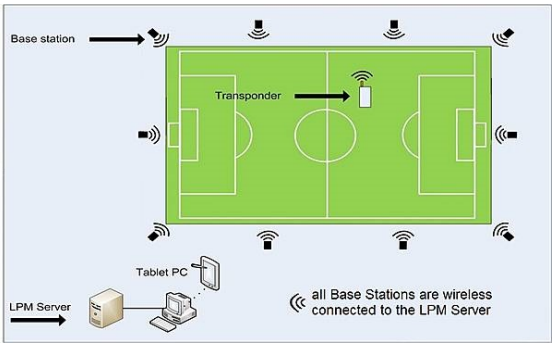
Eine Übersicht dieser Nachbearbeitungsschritte ist in Abbildung A.4 dargestellt.

Sequenz	Grundwahrheit	Trackingdaten	Situationen
ACT1s1*	ACT1s1*	ACT1s1*_tracking	Herumlungern
ACT1s2*	ACT1s2*	ACT1s2*_tracking	Tor passieren und weiter gehen
ACT1s3*	ACT1s3*	ACT1s3*_tracking	Zur Kamera gehen und zurück
ACT1s4*	ACT1s4*	ACT1s4*_tracking	Zur Kamera gehen zickzack
ACT1s6*	ACT1s6*	ACT1s6*_tracking	Zaun besteigen
ACT2s1d	ACT2s1d	ACT2s1d_tracking	Zwei Personen gehen zusammen
ACT2s2a	ACT2s2a	ACT2s2a_tracking	Zwei Personen rennen zusammen
ACT2s2b	ACT2s2b	ACT2s2b_tracking	Zwei Personen rennen zusammen
ACT2s3a	ACT2s3a	ACT2s3a_tracking	Zwei Personen gehen hintereinander
ACT2s3b	ACT2s3b	ACT2s3b_tracking	Zwei Personen gehen hintereinander
ACT2s3c	ACT2s3c	ACT2s3c_tracking	Zwei Personen gehen hintereinander
ACT2s4a	ACT2s4a	ACT2s4a_tracking	Zwei Personen rennen hintereinander
ACT2s5a	ACT2s5a	ACT2s5a_tracking	Zwei Personen treffen sich
ACT2s5c	ACT2s5c	ACT2s5c_tracking	Zwei Personen treffen sich
ACT2s6b	ACT2s6b	ACT2s6b_tracking	Zwei Personen gehen aneinander vorbei
ACT2s6c	ACT2s6c	ACT2s6c_tracking	Zwei Personen gehen aneinander vorbei
ACT3s1	ACT3s1	ACT3s1_tracking	Gruppenbildung
ACT3s2	ACT3s2	ACT3s2_tracking	Gruppenbildung, zwei Gruppen
ACT3s4a	ACT3s4a	ACT3s4a_tracking	Herumlungern und dann wegrennen
ACT3s4b	ACT3s4b	ACT3s4b_tracking	Herumlungern und dann wegrennen
ACT3s5a	ACT3s5a	ACT3s5a_tracking	Herumlungern und dann wegrennen
ACT3s5c	ACT3s5c	ACT3s5c_tracking	Herumlungern und dann wegrennen
ACT3s6	ACT3s6	ACT3s6_tracking	Diverses
ACT3s7a	ACT3s7a	ACT3s7a_tracking	Kurzzeitiges Verstecken
ACT3s7b	ACT3s7b	ACT3s7b_tracking	Kurzzeitiges Verstecken
ACT3s8	ACT3s8	ACT3s8_tracking	Gruppenbildung
ACT4s1a	ACT4s1a	ACT4s1a_tracking	Medizinischer Notfall
ACT4s1b	ACT4s1b	ACT4s1b_tracking	Medizinischer Notfall
ACT5s1a	ACT5s1a	ACT5s1a_tracking	Zwei Personen treffen sich (Innen)
ACT5s1e	ACT5s1e	ACT5s1e_tracking	Zwei Personen treffen sich (Innen)
ACT5s2a	ACT5s2a	ACT5s2a_tracking	Zwei Personen gehen zusammen (Innen)
ACT5s3a	ACT5s3a	ACT5s3a_tracking	Zwei Personen gehen zusammen (Innen, IR)
ACT5s3a	ACT5s3a	ACT5s3a_tracking	Zwei Personen gehen durch Tür (IR)
ACT5s3b	ACT5s3b	ACT5s3b_tracking	Zwei Personen gehen durch Tür (IR)
ACT5s3c	ACT5s3c	ACT5s3c_tracking	Zwei Personen gehen durch Tür (IR)

Tabelle A.2: Videosequenzen des VCA-Datensatzes mit vorhandener Grundwahrheit und vorkommenden Situationen.



(a)

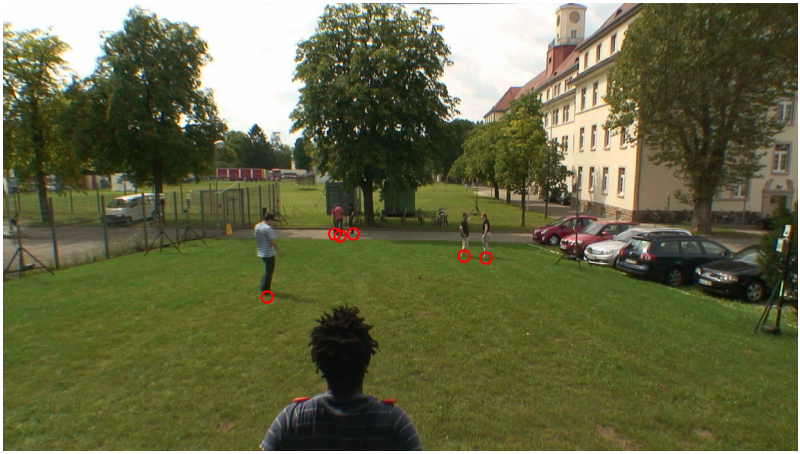


(b)

(c)

Abbildung A.2: Übersicht der verwendeten Werkzeuge um den VCA-Datensatz aufzuzeichnen. Abbilding (a) repräsentiert eine beispielhafte Spur des LPM während den Aufzeichnungen von zwei Personen, die nebeneinander rennen, siehe dazu auch Abbildung A.3 (a). Man beachte, dass jede Person zwei Transponder trägt. Die schematische Darstellung des LPM ist in (b) zu sehen (Quelle: http://www.abatec-ag.com/typo3temp/pics/c5cee2f3d4.jpg (19.03.2017)) und Abbildung (c) zeigt das verwendete Kamerasystem.

(a)



(b)

Abbildung A.3: (a) Beispielbild aus dem VCA-Datensatz mit der Situation „Run-Together“. (b) Visualisierung der Lage der einzelnen Personen in Bildkoordinaten.

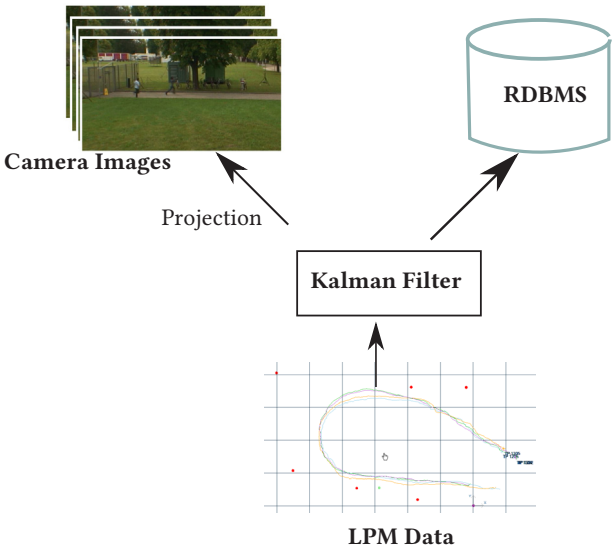


Abbildung A.4: Nachbearbeitung des VCA-Datensatzes. Zuerst werden die LPM Daten durch Anwendung eines Kalman Filters stabilisiert, dann in einer Datenbank für die spätere Nutzung zur Personenverfolgung oder Situationserkennung gespeichert. Zusätzlich wird eine Abbildung von den Weltkoordinaten des LPM in die Bildkoordinaten bestimmt und dann die Position der einzelnen Personen im Bild markiert.

Weil die Basisstationen präzise mit dem Tachymeter eingemessen wurden stehen zum einen diese 3D Weltkoordinaten zur Verfügung, als auch die korrespondierenden Pixelkoordinaten in den 2D Bildkoordinaten. Die Transformation von den Weltkoordinaten des LPM in die Bildkoordinaten wurde mit einem perspective-n-point (PnP) Verfahren [Fischler and Bolles, 1981] bestimmt. Eine Visualisierung der Projektion der Lage der einzelnen Personen ist in Abbildung A.3 (b) dargestellt.

Technische Details

In den folgenden Abschnitten wird auf die technischen Details der Hard- und Software eingegangen, die zur Aufzeichnung des VCA-Datensatzes verwendet wurden.

Kameras Sowohl für Aufzeichnungen in Innenbereichen als auch für Aufnahmen in Außenbereichen wurde bei Tageslicht eine Axis Q1755 Netzwerkkamera verwendet, siehe dazu Abbildung A.5 mit den technischen Details der Kamera. Aufnahmen bei Nacht im Außenbereich wurden mit einer Axis P5534 PTZ Kamera durchgeführt. Das Datenblatt zu dieser Kamera ist in Abbildung A.6 hinterlegt.

Camera	
Models: Indoor	AXIS Q1755 60 Hz; AXIS Q1755 50 Hz
Models: Outdoor	AXIS Q1755-E 60 Hz; AXIS Q1755-E 50 Hz
Image sensor	1/3" progressive scan CMOS 2 megapixel
Lens	f=5.1 - 51 mm, F1.8 - 2.1, autofocus, automatic day/night Horizontal angle of view: 48.1° - 5.1° M37x0.75 mounting thread for optional lens adaptor
Minimum illumination	Color: 2 lux at 30 IRE, F1.8 B/W: 0.2 lux at 30 IRE, F1.8
Shutter time	1/10000 s to 1/2 s
Zoom	10x optical and 12x digital, total 120x

Abbildung A.5: Axis Q1755 Netzwerkkamera technische Details.

Name	BS2	BS4	BS8	BS10	RT	BS1	BS3	BS5	BS9
x[m]	-8.54	-27.6	-24.94	-1.12	-14.40	0.00	-17.83	-30.08	-9.51
y[m]	0.90	5.34	33.14	18.01	2.71	0.00	2.67	20.20	18.04
z[m]	2.15	1.77	1.49	1.36	1.17	3.00	0.88	1.22	1.90

Tabelle A.3: Die mit dem Tachymeter genau eingemessenen Positionen der Basisstationen. Der Koordinatenursprung wurde in BS1 gelegt.

Camera	
Models	AXIS P5534 60 Hz; AXIS P5534 50 Hz
Image sensor	1/3" progressive scan CCD 1.3 megapixel
Lens	f=4.7 – 84.6 mm, F1.6 – 2.8, autofocus, automatic day/night Horizontal angle of view: 55.2° - 3.2°
Minimum illumination	Color: 0.74 lux at 30 IRE F1.6 B/W: 0.04 lux at 30 IRE F1.6
Shutter time	1/10 000 s to 1/4 s
Pan/tilt/zoom	E-flip, Auto-flip, 100 preset positions Pan: 360° (with Auto-flip), 0.2° - 300°/s Tilt: 180°, 0.2° – 300°/s 18x optical zoom and 12x digital zoom, total 216x zoom
Pan/tilt/zoom functionalities	Limited guard tour Control queue On-screen directional indicator

Abbildung A.6: Axis P5534 PTZ Netzwerkkamera technische Details.

Lokales Positionsmesssystem (LPM) Initial muss das LPM kalibriert werden. Dazu müssen die Koordinaten für die acht Basisstationen (BS) und eine Referenzstation (RT) präzise mit z.B. einem Tachymeter eingemessen werden. Die acht BS werden um die Szene herum aufgestellt, siehe dazu beispielsweise Abbildung A.2 (a); BS sind rot, die RT türkis. Die Ergebnisse der Einmessung sind in Tabelle A.3 aufgeführt. Eine ausgezeichnete BS, die sogenannte RT, sendet periodisch ein Signal um die Zeiten der einzelnen BS zu synchronisieren [Resch et al., 2012]. Nachdem alle BS synchronisiert sind, können die Transponder (MT) aktiviert werden. In allen Szenen wurde jede Person mit zwei MT, jeweils auf der linken und der rechten Schulter, ausgestattet. Die Distanz der MT bzgl. den BS wird zu jeder Zeit an das Auswertungssystem geschickt, das durch diese Distanzen die Spuren der einzelnen MT berechnen kann.

Datenformat Die Struktur des Datenformats des LPM wird ein einem Verbund der Programmiersprache C/C++ definiert:

```
typedef struct _LPMLongLine
{
  // 26 int: allgemeine Daten; 6 int: Daten zu je einer BS
  // mit 20 BS ergeben sich 146 int = 584 Byte
  int Timestamp;  // in 0.1 milliseconds
  int TranspID;
  int Quality;
  int Telemetry;
  int PosX;       // in mm
  int PosY;       // in mm
  int PosZ;       // in mm
  int SpeedX;
  int SpeedY;
  int SpeedZ;
  int AccelX;
  int AccelY;
  int AccelZ;
  int RawX;       // in mm
  int RawY;       // in mm
  int RawZ;       // in mm
  int Heading;
  int Roll;
  int Pitch;
  int TrackStatus;
  int FilterMode;
  int W_Offset_Bancroft;
  int W_Offset_EKF;
  int W_Point_EKF;
  int tNumberBS;
  int CellID;
  int HUBPort_[20];
  int PowerLevel_[20];
  int PeakQuality_[20];
  int Chi2_[20];
  int TimeDiff_[20];
  int BSTelemetry_[20];
} LPMLongLineStruct;
```

A.2.2 Onlinesystem

Das in Abschnitt A.2.3 beschriebene Mehrkamerasystem, siehe Abbildung A.7, ist die Grundlage eines Onlinesystems zur Erkennung von Situationen live und in Echtzeit. Die Initialisierung des Onlinesystems geht in wenigen

Schritten vor sich. Zuerst wird die starre Übersichtskamera auf die zu beobachtende Szene gerichtet; und zwar so, dass entsprechend den Randbedingungen der vorhandenen bildverarbeitenden Verfahren entsprochen wird. Anschließend wird die Autokalibrierung, siehe Abschnitt A.2.3, für das Mehrkamerasystem durchgeführt. Schließlich wird eine Abbildung von den Realweltkoordinaten in die Bildkoordinaten definiert und für die Situationserkennung hinterlegt. Nach Abschluss dieser kurzen Initialisierungsphase werden sowohl die bildverarbeitenden Module, wie z.B. die Personendetektion, als auch die Situationserkennung mit dem Hintergrundwissen für die zu erwartenden Situationen gestartet. Die Kommunikation geschieht dabei aus Gründen der Skalierbarkeit, dem Verteilten Rechnen und der Persistenz über eine relationale Datenbank.

A.2.3 Autokalibrierung und Steuerung von Sensoren

Bei der videobasierten Situationserkennung werden Beobachtungen benutzt, um Aussagen über die Verhalten in der beobachteten Szene abzuleiten. Die auf FMTHL und SGT basierende Situationserkennung ist hierarchisch aufgebaut. Die zu erkennenden Situationen eines SGT beginnen mit der allgemeinsten Situation und werden dann semantisch verfeinert. Dieses Prinzip hat sich ebenfalls bei der Sensor- und Verfahrensauswahl bewährt: Zuerst wird mit einem bildgebenden Sensor eine möglichst gute Übersicht der Szene geschaffen, um dann mit weiteren Sensoren einzelne Bereiche in der Szene zu fokussieren und dort mehr Details zu erfassen. Diese Funktionalität wird von einem Master-Slave-Kamerasystem realisiert, welches im Folgenden vorgestellt und auf dessen Kalibrierung eingegangen wird. Ein Master-Slave-Kamerasystem besteht aus einer Masterkamera mit Weitwinkeloptik und aktiv steuerbaren Schwenk-Neige-Zoom-Kameras (engl. pan-tilt-zoom camera (PTZ)). Die Masterkamera liefert einen groben Überblick über die gesamte beobachtete Szene, wohingegen die PTZ-Kamera gezielt einzelne Details der Szene erfassen kann. Um im Bildbereich der

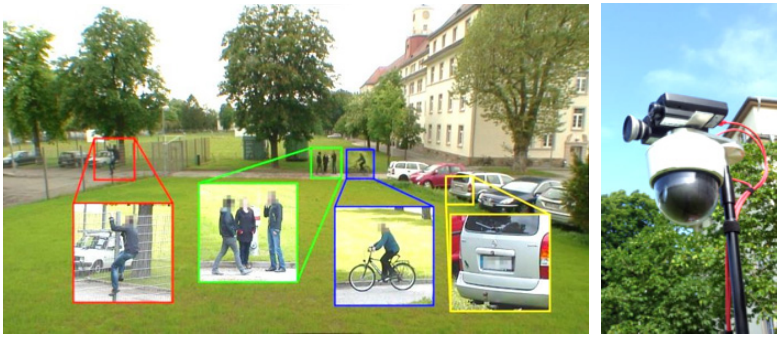
Masterkamera ausgewählte Orte mit der PTZ-Kamera anzusteuern wird eine Abbildung $\mathbb{N} \times \mathbb{N} \mapsto \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ benötigt, die die Bildkoordinaten der Masterkamera in Motorkoordinaten der PTZ-Kamera abbildet. Eine initiale grobe Abbildung wird durch Bildregistrierung gewonnen. Im weiteren Verlauf wird diese Abbildung verbessert, im Besonderen durch die in der Szene vorkommenden bewegten und texturierten Objekte.

(a)

(b)

Abbildung A.7: (a) Das Mehrkamerasystem bestehend aus starrer Übersichtskamera und PTZ-Kamera. (b) Der Überblick über die Szene und die detaillierte Betrachtung der einzelnen interessanten Regionen.

Abbildung A.8 beschreibt den Sensoraufbau (b) und ein mögliches Anwendungsszenario (a), bei dem detailliertere Informationen über Personen, Interaktionen oder andere Objekte automatisch mit Detailaufnahmen durch die Slavekamera erfasst werden. Um präzisere Aussagen über das Verhalten in einer Szene zu treffen, sind diese Detailaufnahmen notwendig, da mit der Masterkamera häufig die Auflösungsgrenzen unterschritten werden, die als untere Schranke für die Detektion von Objekten notwendig sind [Dollár et al., 2012]. Weitere Notwendigkeiten für Detailaufnahmen kommen von der Posenrekonstruktion [Brauer et al., 2012] für die Aktionserkennung bei Personen. Zur Identifikation von einzelnen Personen werden noch weitere Informationen benötigt, die eine Gesichtsdetektion leisten kann [Bellotto et al., 2012]. Eine weitere Anwendungsmöglichkeit bietet sich bei der Situationserkennung, die semantische Rückmeldungen an das Kamerasystem gibt um damit gezielt interessante Bildbereiche weiter zu detaillieren und somit präzisere Aussagen über die im Moment zutreffenden Situationen einer Szene zu treffen.



(a) (b)

Abbildung A.8: (a) Der Überblick über die Szene und die detaillierte Betrachtung der einzelnen interessanten Regionen. (b) Das Mehrkamerasystem bestehend aus starrer Übersichtskamera und PTZ-Kamera.

In der Literatur finden sich Mehrkamerasysteme, die hauptsächlich wohldefiniert und kalibriert sind [Hu et al., 2004]. In dieser Arbeit hingegen betrachten wir Mehrkamerasysteme in unstrukturierten Umgebungen ohne extrinsische Kalibrierung und ohne genaues Wissen über die Geometrie der Kameras zueinander.

Mehrkamerasysteme können unterschieden werden durch ihre aktiven Sensoren. In [Bellotto et al., 2012] kooperieren mehrere PTZ-Kameras in einer durch eine Masterkamera beobachteten Szene. In [Szwoch et al., 2011] werden mehrere Masterkameras und PTZ-Kameras eingesetzt. In einem Innenraumszenario, wie beispielsweise einem intelligenten Kontrollraum, sind dutzende Kameras, darunter Kameras mit Fischaugenoptiken und zwei PTZ-Kameras [IJsselmuiden and Stiefelhagen, 2010]. Es gibt außerdem Arbeiten, die ausschließlich PTZ-Kameras verwenden [Del Bimbo et al., 2010].

Die Motivation und Konzeption der Kamerasteuerung ist bei verschiedenen Mehrkamerasystemen unterschiedlich: In [Everts et al., 2007] werden Objekte über mehrere kooperierende Kameras verfolgt. Informationen über bestimmte Objekte wie Kennzeichen [Tian et al., 2008] oder Personenidentifikation [Yi et al., 2008] werden gewonnen. In [Bellotto et al., 2012] wird Mehrkameraverfolgung von Objekten mit Detailaufnahmen kombiniert. Dort wird die Kamerasteuerung auf drei Ebenen betrachtet: Auf Bildebene (Picture Domain Camera Control) spielen nur Informationen aus dem zweidimensionalen Bild eine Rolle, auf Szenenebene (Scene Domain Camera Control) liegt ein 3D-Modell der beobachteten Szene zugrunde und auf der begrifflichen Ebene (Conceptual Level Camera Control) wird abgeleitetes Szenenwissen zur Steuerung der Kameras eingesetzt.

In einem Mehrkamerasystem ist die Abbildung von einem Punkt in der einen Kamera auf den korrespondierenden Punkt in der anderen Kamera unentbehrlich. Dazu wird eine Kalibrierung benötigt, wobei man zwischen starken und schwachen Kalibrierungen unterscheiden kann.

Eine starke Kalibrierung setzt voraus, dass sowohl die intrinsischen als auch die extrinsischen Kameraparameter aller Kameras bekannt sind. Damit lassen sich 2D-Korrespondenzen im Bildbereich über 3D-Weltkoordinaten bestimmen. Diese Kalibrierung kann für jede Kamera [Tsai, 1987] oder als eine paarweise Stereokonfiguration [Horaud et al., 2006] durchgeführt werden. Bei Anwendungen im Außenbereich wird oft ein Raumbezug zu Geobasisdaten hinzugefügt [Szwoch et al., 2011]. Einfache Kameras erfordern eine erweiterte Kalibrierung [Jain et al., 2006], im Gegensatz dazu verwenden andere ein vereinfachtes Kameramodell [Sinha and Pollefeys, 2006]. Es gibt auch Verfahren, die händische Unterstützung benötigen [Davis and Chen, 2003].

Bei der schwachen Kalibrierung wird die Abbildung von Bildbereich zu Bildbereich nicht über die 3D-Weltkoordinaten durchgeführt, sondern es wird eine direkte Lookup-Tabelle (LUT) zwischen den 2D-Bildkoordinaten und den Motorkoordinaten der PTZ-Kamera generiert. Die frühen Ansätze [Zhou et al., 2003] benutzen noch händisch ausgewählte Korrespondenzen oder annotierte Personen [Mohanty and Gellaboina, 2011]. Die Interpolation von dünn besetzten LUTs kann geometrisch [Liao et al., 2011] oder mit Splines [Badri et al., 2007] durchgeführt werden. Bestimmte Eigenschaften einer Szene wie Fahrbahnmarkierungen oder Fluchtpunkte können ebenfalls verwendet werden. Methoden, die auf lokalen Bildmerkmalen arbeiten wie [Liao et al., 2011] vermeiden im Allgemeinen eine Abhängigkeit zur beobachteten Szene. Weil das Sichtfeld der PTZ-Kamera normalerweise nur ein kleiner Ausschnitt des Sichtfelds der Masterkamera ist, besteht auch die Möglichkeit, ein ganzes Bildmosaik von Bildern der PTZ-Kamera und deren korrespondierenden Motorkoordinaten abzuspeichern. [Wu and Radke, 2012] verzichten in ihrer Arbeit ganz auf die Bilder und speichern nur die Bildmerkmale.

Das hier vorgestellte automatische Verfahren verbindet mehrere oben genannten Vorteile in ein Verfahren um das Master-Slave-Kamerasystem in Abbildung A.8 zu kalibrieren. Das Verfahren führt eine schwache

Kalibrierung durch unter der Annahme, dass beide Kameras einen ähnlichen Standpunkt besitzen. Im Allgemeinen brauchen keine weiteren Annahmen über die Kamerageometrie getroffen zu werden. Im Gegensatz zu einer Mosaikbildung wird eine dünn besetzte LUT mit linearer Regression interpoliert. Weil es bei schwach texturierten Regionen nicht möglich ist Korrespondenzen zu finden, wird mit zur Laufzeit in dieses Regionen vorhandenen Objekten eine Verbesserung der LUT durchgeführt.

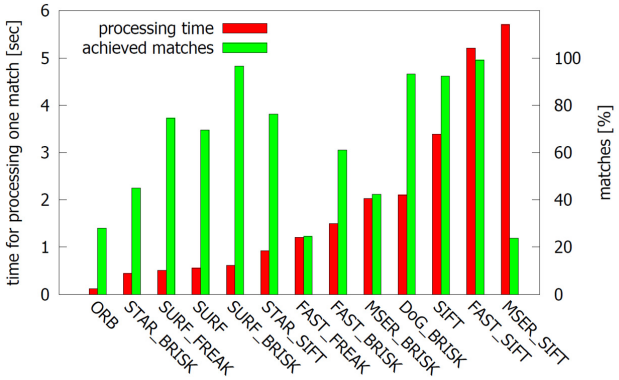
A.2.4 Methoden

Im Folgenden wird die Verarbeitungskette zur schwachen Kalibrierung des Master-Slave-Kamerasystems vorgestellt.

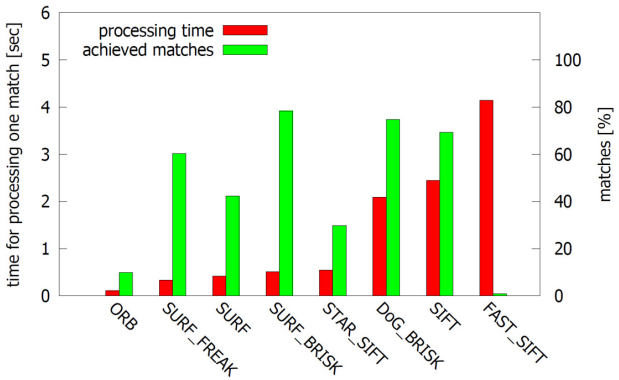
In einem ersten Schritt werden Korrespondenzen zwischen den extrinsisch nicht kalibrierten Kameras gesucht: Lokale Bildmerkmale [Li and Allinson, 2008, Mikolajczyk and Schmid, 2005] sind dafür geeignet. In [Münch et al., 2013a] wurden verschiedene Kombinationen von Detektoren und Deskriptoren experimentell untersucht. Für unseren Anwendungsfall hat sich gezeigt, dass Speeded Up Robust Features (SURF) im Verhältnis von Genauigkeit und Laufzeit allen anderen Bildmerkmalen überlegen ist, siehe Abbildung A.9. Kombiniert man den SURF-Detektor mit dem Binary Robust Invariant Scalable Keypoints (BRISK)-Deskriptor kann die Genauigkeit noch einmal deutlich gesteigert werden, insbesondere wenn das Blickfeld von Master- und PTZ-Kamera weit auseinander liegen. Außer der intrinsischen Kameraparameter sind keine weiteren Annahmen oder Vorverarbeitungsschritte erforderlich.

Ein grober Überblick über die Verarbeitungskette zeigt Abbildung A.10. Um die Konfiguration zu starten werden einige Korrespondenzen benötigt. Dafür bewegt sich die PTZ-Kamera spiralförmig (a) im Schwenk-Neige-Raum, um möglichst den Sichtbereich der Masterkamera zu überdecken. Für jedes Bild der PTZ-Kamera werden Bildmerkmale extrahiert, wird versucht, Korrespondenzen herzustellen und die abbildende Homographie zu

schätzen (b). Es werden auf feste Schwellwerte und Nächste-Nachbarn-Suche verzichtet, stattdessen wird Nearest Neighbor Distance Ratio verwendet, weil es in unseren Szenarien besser funktioniert [Mikolajczyk and Schmid, 2005].



(a)



(b)

Abbildung A.9: Auswertung der Laufzeit und Performance verschiedener Detektoren. (a) ohne Zoom. (b) mit 5x Zoom.

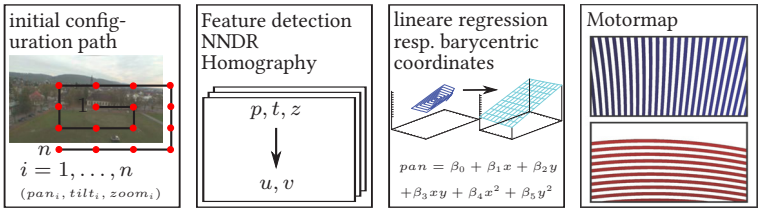


Abbildung A.10: Überblick über die gesamte Verarbeitungskette um die LUT zu bestimmen. (a) Eine grobe initiale Menge an Korrespondenzen wird ermittelt. (b) Es wird eine Abbildung von den ausgewählten Bildpunkten (u, v) zu den Motorkoordinaten der PTZ-Kamera gebildet. (c) Eine lineare Regression erzeugt eine dichte LUT. (d) Visualisierung der LUT, Pan und Tilt getrennt, gerundet auf ganze Zahlen und alternierend gefärbt.

Schließlich wird die Homographie zwischen beiden Bildern berechnet, siehe Abbildung A.11 (a). Dabei stehen $\mathbf{p}'_m = [x'_m, y'_m, z'_m]^\intercal$ für die Motorkoordinaten der PTZ-Kamera, $\mathbf{p}_s = [x_s, y_s, 1]^\intercal$ für die Bildkoordinaten der Masterkamera und $\mathbf{H}_{sm}$ für die Abbildungsmatrix:

$$\begin{bmatrix} x'_m \\ y'_m \\ z'_m \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} \Leftrightarrow \mathbf{p}'_m = \mathbf{H}_{sm}\mathbf{p}_s, \quad \mathbf{p}_m = \frac{\mathbf{p}'_m}{z'_m} \tag{A.1}$$

Um in diesem Schritt unplausible Homographien zu filtern, wird die Transformationsmatrix H_{sm} zu $detH'_{sm} = 1$ transformiert:

$$H'_{sm} = H_{sm} * \frac{sgn(detH_{sm})}{\sqrt[3]{|detH_{sm}|}},$$

siehe auch [Begelfor and Werman, 2005].

Als Element der speziellen linearen Gruppe $H'_{sm} \in SL_3(\mathbb{R})$ können Schwellwerte angegeben werden, um starke Ausreißer zu filtern, konkret: $-5{,}0 < h'_{11} < 5{,}0, \quad -5{,}0 < h'_{22} < 5{,}0, \quad 0{,}55 < h'_{33} < 2{,}5.$

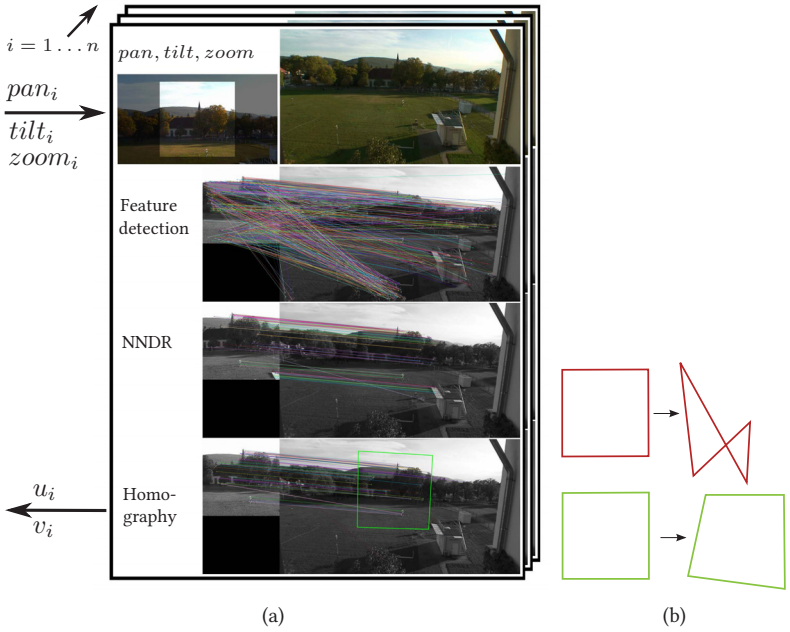


Abbildung A.11: (a) Visualisierung der Merkmalsdetektion und Korrespondenzfindung im Master- und Slave-Kamerabild und Schätzung der Homographie. (b) Filterung von unplausiblen Homographien.

Bisher ist die LUT noch spärlich besetzt, siehe Abbildung A.15. Es wird eine dichte LUT benötigt, somit ist ein Interpolationsschritt erforderlich. Im allgemeinen Fall sollte die Abbildung in Gleichung A.1 bijektiv sein.

Aufgrund technisch-mechanischer Ungenauigkeiten wird die Abbildung in unserem Fall lediglich injektiv sein. Eine lineare Regression, getrennt für Pan und Tilt, wird durchgeführt um die Gleichungen

$$pan = a_{p0} + a_{p1} * x + a_{p2} * y + a_{p3} * x * y + a_{p4} * x^2 + a_{p5} * y^2 \quad \text{(A.2)}$$

und

$$tilt = a_{t0} + a_{t1} * x + a_{t2} * y + a_{t3} * x * y + a_{t4} * x^2 + a_{t5} * y^2 \quad \text{(A.3)}$$

zu bestimmen. Dabei sind die zu bestimmenden Koeffizienten a_{pi} bzw. a_{ti}. Es wurden verschiedene Grade und Polynome auf ihren Fehler ausgewertet und es hat sich gezeigt, dass die Polynome in Gleichung A.2 und A.3 den geringsten Fehler verursachen. Aufgrund der sphärischen Kinematik der PTZ-Kamera ist dieses Ergebnis plausibel.

Die Strukturgleichung des linearen Modells lautet:

$$(p_1 \cdots p_n)^T = \boldsymbol{X}\boldsymbol{a_p} + \boldsymbol{\epsilon_p} \quad \text{(A.4)}$$

$$(t_1 \cdots t_n)^T = \boldsymbol{X}\boldsymbol{a_t} + \boldsymbol{\epsilon_t} \quad \text{(A.5)}$$

mit $\boldsymbol{X}$ entsprechend Polynom A.2 bzw. A.3 ergibt sich unter der Verwendung der Methode der kleinsten Quadrate die geschätzte Struktur zu:

$$\hat{\boldsymbol{a}}_{\boldsymbol{p}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T (p_1 \cdots p_n)^T = \boldsymbol{X}^+ (p_1 \cdots p_n)^T \quad \text{(A.6)}$$

$$\hat{\boldsymbol{a}}_{\boldsymbol{t}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T (t_1 \cdots t_n)^T = \boldsymbol{X}^+ (t_1 \cdots t_n)^T \quad \text{(A.7)}$$

mit $X^+ = (X^T X)^{-1} X^T$, die Moore-Penrose-Pseudoinverse.

Bei diesem Schritt wird der RANSAC-Algorithmus angewandt um noch vorhandene Ausreißer möglichst zu ignorieren. Das Ergebnis zeigt Abbildung A.12.

Mit den nun vollständig vorliegenden Gleichungen A.2 und A.3 wird eine dichte LUT erstellt. In unserem Fall wird entsprechend des eingangs

definierten Abbildungsraum der Bild- und Motorkoordinaten auf ganze Zahlen gerundet. Es ergibt sich beispielsweise eine LUT in Abbildung A.13. Somit liegen jetzt zu jedem Punkt im Bild der Masterkamera die korrespondierenden Motorkoordinaten der PTZ-Kamera vor.

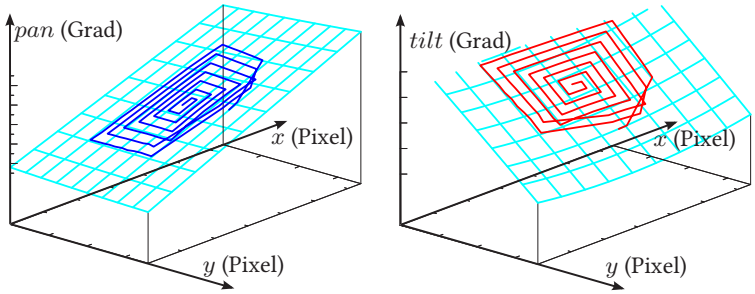


Abbildung A.12: Polynom zweiten Grades als Näherung für die gewonnenen Pan- respektive Tilt-Werte in Abhängigkeit der Bildkoordinaten der Masterkamera. In blau resp. rot die interpolierten Messwerte und darübergelegt die Funktionen.

A.2.5 Auswertung

Das hier vorgestellte Mehrkamerasystem kann ausschließlich online ausgewertet werden. Dazu wurde die Leistung in Form von Genauigkeit in verschiedenen Anwendungsgebieten ausgewertet. In Abbildung A.14 sind diese verschiedenen Anwendungsgebiete dargestellt.

Der Ablauf der Auswertung beginnt mit der Autokalibrierung des Master-Slave-Kamerasystems. Im Anschluss werden viele Marker in der jeweiligen Szene verteilt. Jeder Marker wird manuell durch die PTZ-Kamera fokussiert um die Grundwahrheit der Motorkoordinaten für diesen Marker zu bestimmen. Dann wird der Marker durch die Autokalibrierung angesteuert. Die Auswertung der Genauigkeit erfolgt durch den Abstand in Grad von der durch die Autokalibrierung angesteuerten Motorkoordinaten zur Grundwahrheit, siehe Abbildung A.15 (b).

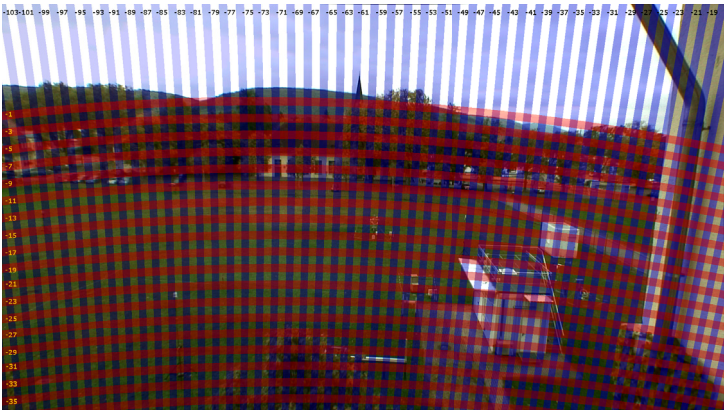
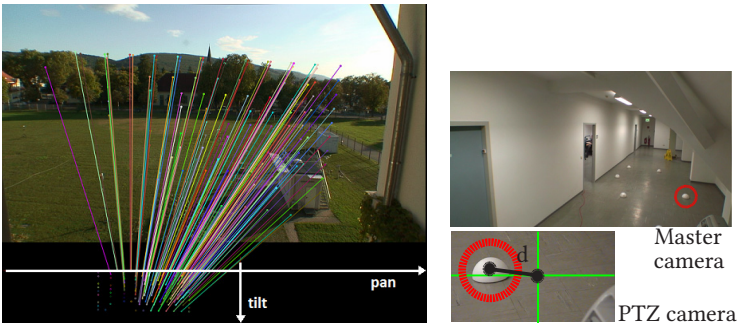


Abbildung A.13: Dichte Abbildung der Pan-Tilt-Werte für das Bild der Masterkamera. Die Kinematik der Kamera ist mechanisch beschränkt, deshalb keine Tilt-Werte im oberen Bereich.

(a) (b) (c)

(d) (e) (f)

Abbildung A.14: Das Master-Slave-Kamerasystem wurde in verschiedenen Diskursbereichen ausgewertet. In einem Überwachungsszenario mit menschenähnlicher Perspektive (a,d), in einem großräumigen Überwachungsszenario (b,c), bei einem Perimeterschutzszenario bei Nacht (e) und in Gebäuden (f).



(a) (b)

Abbildung A.15: (a) Registrierung der Motorkoordinaten auf das Masterkamerabild. (b) Auswertung des Fehlers d (die Distanz der Grundwahrheit der Motorkoordinaten zu den gelernten Motorkoordinaten) der gelernten Master-Slave-Konfiguration.

In Abbildung A.16 sind die Ergebnisse der Szene (b) aus Abbildung A.14 dargestellt. Über hundert Marker wurden in der von der Masterkamera beobachteten Szene plaziert und ausgewertet, siehe Abbildung A.15 (b). In Abbildung A.16 (a) bzw. (b) sind die Fehler in Grad für Pan bzw. Tilt der PTZ-Kamera dargestellt. Auf der horizontalen Achse sind die Pixel in horizontaler (a) bzw. vertikaler (b) Richtung aufgetragen. Wie man nun deutlich sieht, ist im unteren linken Bereich der LUT der Fehler besonders groß. Mit Abbildung A.15 (a) findet man dafür eine Erklärung: Es wurden an dieser Stelle keine Korrespondenzen gefunden. In jeder Anwendungsdomäne treten eigene Herausforderungen auf. Die ausführlichen Auswertungen sind in [Grosselfinger, 2012] beschrieben.

Es zeigt sich, dass das vorgestellte Verfahren in schwach texturierten Umgebungen nicht gut funktioniert. In diesen Regionen werden nicht genügend Korrespondenzen gefunden, was dazu führt, dass durch die fehlerbehaftete LUT die PTZ-Steuerung ungenau wird.

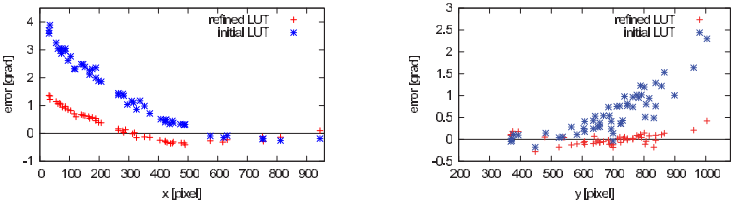
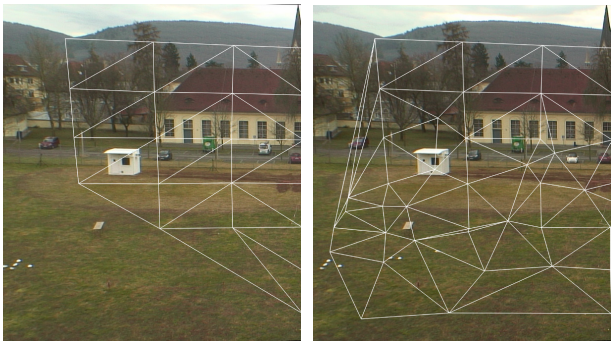


Abbildung A.16: Auswertung der Genauigkeit von Pan (links) und Tilt (rechts) im Szenario aus Abbildung A.14 (b) mit der initialen LUT (blau) und der automatisch online verbesserten LUT (rot)

Um diese Einschränkungen aufzuheben, treffen wir folgende Annahme: Wenn der wenig texturierte Bereich von Interesse ist, dann wird zu einer bestimmten Zeit dort ein Objekt wie beispielsweise eine Person oder ein Auto anzutreffen sein. Dieses Objekt liefert dann im Allgemeinen ausreichend Textur, um zu diesem Zeitpunkt an diesem Ort eine Korrespondenz herzustellen. Diese zur Laufzeit gewonnene Korrespondenz wird nun zu den bereits vorhandenen hinzugefügt und die LUT erneut berechnet.

In Abbildung A.17 (a) ist eine Visualisierung der initialen Korrespondenzen zu sehen. In (b) wurden nach kurzer Laufzeit bereits im bisher schwachen linken unteren Bildbereich weitere Korrespondenzen hinzugefügt. Die Auswertung ist ebenfalls in Abbildung A.16 aufgetragen (rot). Man sieht deutlich, wie der Gesamtfehler um den Faktor fünf reduziert wurde und die Autokalibrierung auch in wenig texturierten Gebieten von Interesse anwendbar macht.

Zusammenfassend wurde ein Master-Slave-Kamerasystem vorgestellt, welches sich unter unkooperativen und wenig texturierten Bedingungen selbst kalibriert. Die Anwendbarkeit wurde in verschiedenen Anwendungsgebieten nachgewiesen.



(a) (b)

Abbildung A.17: Triangulation der initial gelernten LUT (a); Verfeinert mit zusätzlich zur Laufzeit gefundenen Korrespondenzen (b).

A.3 Vergleichende Zusammenfassung der Datensätze

In Tabelle A.4 werden die oben genannten und in dieser Arbeit verwendeten Datensätze vergleichend gegenübergestellt. Es wird sofort ersichtlich, dass jeder der verwendeten Datensätze seine ganz eigenen Charakteristika hat. So variiert die Bildfrequenz bei der Datenaufnahme von 7 bis 30 Hz. Bei der Situationserkennung spielt diese feingranulare Abstimmung eine untergeordnete Rolle, da die zu erkennenden Situationen auf höheren und zeitlich ausgedehnteren Konzepten basieren. Die Kameraauflösung ist hingegen von Bedeutung, da, um mit Objektdetektoren robuste Ergebnisse zu erzielen, eine Objektmindestauflösung gefordert ist.

In allen Datensätzen ist die Grundwahrheit für die Personen bekannt; außer im Onlinesystem. Somit ist man bei der Auswertung von der tatsächlichen Bildverarbeitung unabhängig und könnte und kann auch ohne Bildverarbeitungsverfahren mit den vorhandenen Datensätzen arbeiten.

Datensatz	BEHAVE	VIRAT	PETS 2009	CAVIAR	vPTZ	VCA	Onlinesystem
Bildfrequenz [Hz]	25	$23 - 30$	7	25	15	15	10
Bildauflösung [px]	640×480	1920×1080 1280×720	768×576	384×288	5400×2700	1920×1080	1920×1080 1280×720
Umfang	ca. 1h	ca. 6h, hier 66min	ca. 2min	ca. 1h	ca. 7min	ca. 90min	∞
Anzahl verschiedener Situationen	10	13, hier 7	7	10	$8 + 2$	20	∞
Anzahl aller Situationen	163	241, hier 86	33	> 60	> 200	> 100	∞
Personen annotiert	✓	✓	✓	✓	✓(manuell)	✓(LPM)/✗	✗
Situationen annotiert	✓	✓	✓(manuell)	✓	✓(manuell)	✓/✗	✗
Abbildung Bild- - Weltkoordinaten	✓	✓	✗	✓	✗	✓/✗	✓/✗

Tabelle A.4: Vergleichende Zusammenfassung der verwendeten und in Abschnitt 7.3 und Anhang A beschriebenen Datensätze.

B Verhältnis von Verhaltensschema und regulärer Grammatik

Die Hypergraphstruktur von Situationsgraphenbäumen erlaubt eine benutzerfreundliche und logische Modellierung von Hintergrundwissen. Rein syntaktisch kann diese Hypergraphenstruktur in eine reguläre Sprache transformiert werden. Die Struktur von Situationsgraphenbäumen ist regulär. Dies erlaubt, das Wortproblem mit einer Zeitkomplexität von $\mathcal{O}(n)$ und Speicherkomplexität $\mathcal{O}(1)$ zu lösen.

B.1 Situationsgraphenbäume als formale Sprache

In [Arens, 2004] Abschnitt 3.5 und Anhang B.2 wird für Situationsgraphenbäume (SGTs) gezeigt, dass sie mindestens kontextfrei sind, indem eine kontextfreie Grammatik für SGTs angegeben wird. Im folgenden Abschnitt wird nun gezeigt, dass SGTs sogar regulär sind, indem ein dem SGT syntaktisch äquivalenter nichtdeterministischer endlicher Automat angegeben und dessen Äquivalenz bewiesen wird. In Abschnitt B.4 wird dann die Konstruktionsvorschrift in die andere Richtung angegeben, wie aus einem endlichen Automaten ein äquivalenter SGT wird und bewiesen.

B.2 Konstruktionsvorschrift zur Erzeugung eines endlichen Automaten aus einem SGT

Ein SGT $\mathcal{B}$ ist regulär $\Leftrightarrow$ es existiert ein endlicher Automat $\mathcal{M}_{\mathcal{B}}$, dessen Sprache $\mathcal{L}(\mathcal{M}_{\mathcal{B}})$ der Menge der durch $\mathcal{B}$ definierten SGT-Verhalten entspricht.

Definition 1 (SGT-Verhalten, siehe [Arens, 2004] Definition 7.2)**.** *Gegeben ein SGT $\mathcal{B}$. Ein SGT-Verhalten ist induktiv definiert durch folgende Bedingungen:*

- *jeder durch den Wurzelgraphen von $\mathcal{B}$ beschriebene SGT-Ablauf ist ein SGT-Verhalten.*
- *Aus jedem SGT-Verhalten E kann ein weiteres SGT-Verhalten dadurch konstruiert werden, dass eine Situation S in E durch einen diese Situation detaillierenden SGT-Ablauf ersetzt wird.*

Definition 2 (SGT-Ablauf, siehe [Arens, 2004] Definition 7.1)**.** *Jede Folge E von Situationen innerhalb eines Situationsgraphen G, welche einen Pfad von einer Startsituation zu einer Endsituation entlang von Prädiktionskanten bildet, wird von G beschriebener SGT-Ablauf genannt. Detailliert G einen Situationsschema S, so wird E detaillierender SGT-Ablauf von S genannt.*

Der endliche Automat $\mathcal{M}_{\mathcal{B}}$ ist durch ein Tupel $\mathcal{M}_{\mathcal{B}} = (Q, \Sigma, \delta, s, \mathcal{F})$ beschrieben mit Q als der Menge aller Zustände, Σ als das Eingabealphabet mit $Q \cap \Sigma = \emptyset$, $\delta : Q \times \Sigma \to Q$ ist die Überführungsfunktion, $s \in Q$ der Startzustand und $\mathcal{F} \subseteq Q$ die Menge der Endzustände.

Im Folgenden wird nun eine Konstruktionsvorschrift angegeben, mit der zu jedem SGT ein syntaktisch äquivalenter nichtdeterministischer endlicher Automat erstellt werden kann:

(1) $Q = \emptyset, \Sigma = \emptyset, \delta = \emptyset$

(2) Füge für alle Situationsschemata SS_i einen eigenen Eintrittsknoten s_SS_i und Austrittsknoten F_SS_i hinzu:
$\forall i : Q = Q \cup \{SS_i, s\_SS_i, F\_SS_i\}$

(3) Füge für alle Situationsgraphen SG_j einen eigenen Eintrittsknoten s_SG_j und Austrittsknoten F_SG_j hinzu:
$\forall j : Q = Q \cup \{s\_SG_j, F\_SG_j\}$

(4) Füge globalen Startzustand S und Endzustand F ein:
$Q = Q \cup \{S, F\}$
$s = S$
$\mathcal{F} = \{F\}$
Verbinde S mit dem Eintrittsknoten s_SG_{root} des Wurzelsituationsgraphen und F_SG_{root} mit F.
$\delta = \delta \cup \{S \times \varepsilon \rightarrow s\_SG_{root}, F\_SG_{root} \times \varepsilon \rightarrow F\}$.

(5) $\forall j \forall i$: Verbinde Eintrittsknoten s_SG_j von Situationsgraph SG_j mit allen Eintrittsknoten s_SS_i der Situationsschemata $SS_i \in SG_j$ mit SS_i ist Startsituationsschema:
$\forall j : \forall i : \delta = \delta \cup \{s\_SG_j \times \varepsilon \rightarrow s\_SS_i\}$

(6) $\forall j \forall i$: Verbinde alle Austrittsknoten F_SS_i der Situationsschemata $SS_i \in SG_j$ mit SS_i ist Endsituationsschema mit dem Austrittsknoten F_SG_j von Situationsgraph SG_j :
$\forall j : \forall i : \delta = \delta \cup \{F\_SS_i \times \varepsilon \rightarrow F\_SG_j\}$

(7) $\forall j, i, k$: Verbinde für alle temporale Kanten (SS_i, SS_k) mit SS_i und $SS_j \in SG_j$ F_SS_i mit s_SS_k:
$\forall j, i, k : \delta = \delta \cup \{F\_SS_i \times \varepsilon \rightarrow s\_SS_k\}$

(8) $\forall i \forall j$: Verbinde für alle spezialisierenden Kanten (SS_i, SG_j) s_SS_i mit s_SG_j und F_SG_j mit F_SS_i unter der Bedingung $SS_i \notin SG_j$ und $\nexists$ Kante (SS_l, SG_j) mit $i \neq l$:
$\forall i : \forall j : \delta = \delta \cup \{s\_SS_i \times \varepsilon \rightarrow s\_SG_j, F\_SG_j \times \varepsilon \rightarrow F\_SS_i\}$

(9) $\forall i$: Verbinde alle Eintrittsknoten s_SS_i mit dem korrespondierenden Situationsschema SS_i und dieses wiederum mit dem Austrittsknoten F_SS_i:
$\forall i : \Sigma = \Sigma \cup s_i$
$\forall i : \delta = \delta \cup \{s\_SS_i \times s_i \rightarrow SS_i, SS_i \times \varepsilon \rightarrow F\_SS_i\}$

B.3 Ein SGT-Verhalten entspricht einem Wort der Sprache des endlichen Automaten

Sei $v = \langle s_1, ..., s_n \rangle$ ein durch $\mathcal{B}$ definiertes SGT-Verhalten. Dabei ist ein Situationsschema SS_i mit seinem Bezeichner s_i definiert. $\forall v \in V(\mathcal{B})$: $v \in \mathcal{L}(\mathcal{M}_\mathcal{B})$ Es ist nun zu zeigen $\exists s \in Q$ mit s ist Startzustand : $\exists F \in \mathcal{F}$: $s \xrightarrow{s_1 \cdots s_n} F$.

Das ist gleichbedeutend mit der Aussage, dass es im endlichen Automaten $\mathcal{M}_\mathcal{B}$ einen Weg vom Startzustand s zu einem Endzustand F mit der Kantenbeschriftung $s_1 \cdots s_n$ gibt.

In [Arens, 2004] steht dazu: "Wenn $v = \langle s_1, ..., s_n \rangle$ ein SGT-Verhalten darstellt, folgt hieraus [...], dass diese Folge entweder einen SGT-Ablauf innerhalb des Wurzelgraphen darstellt oder aber sich aus einem solchen SGT-Ablauf ableiten lässt, indem sukzessive Situationschemata innerhalb diese (sic!) Ablaufs durch detaillierende SGT-Abläufe ersetzt werden. Man beachte, dass die hier genannten Abläufe Situationsfolgen bezeichnen, welche innerhalb eines Situationsgraphen von einer Start- zu einer Endsituation entlang von Prädiktionskanten führen und dies insbesondere auch Schritte entlang von Selbstprädiktionen beinhalten kann. Es ist jedoch leicht einsichtig, dass zu jedem SGT-Ablauf mit Selbstprädiktionen ein entsprechender SGT-Ablauf existieren muss, welcher die gleichen Situationsschemata jeweils nur einmal aufweist. Es kann daher für jedes SGT-Verhalten $\langle s_1, ..., s_n \rangle$ angenommen werden, dass eine Folge von SGT-Verhalten $\langle S_0, ..., S_m \rangle = \langle \langle s_1^0, ..., s_{n_0}^0 \rangle, \ldots, \langle s_1^m, ..., s_{n_m}^m \rangle \rangle$ derart existiert, dass die folgenden Aussagen wahr sind:

(a) S_0 ist ein SGT-Ablauf (siehe Definition 1) ohne Schritte entlang von Selbstprädiktionen innerhalb des Wurzelgraphen W von $\mathcal{M}_\mathcal{B}$.

(b) $\forall S_i, S_j$ mit $i = j - 1$ gilt:

(i) S_j entsteht aus S_i durch Detaillierung eines Situationsschemas aus S_i, also oBdA.: $s_k^i \rightsquigarrow \langle s_l^j, \ldots, s_{l+o}^j \rangle$ mit $1 \leq l, l + o \leq n_j$ und $\langle s_l^j, \ldots, s_{l+o}^j \rangle$ ist detaillierender SGT-Ablauf von s_k^i. Man beachte, dass der Spezialfall $m = 0$ dem Fall entspricht, in welchem $\langle s_l^j, \cdots, s_{l+o}^j \rangle$ sowohl ein SGT-Verhalten als auch einen SGT-Ablauf innerhalb eines Wurzelgraphen von $\mathcal{B}$ darstellt. Oder

(ii) S_j entsteht aus S_i durch Vervielfältigung eines Situationsschemas aus S_i entsprechend einer Selbstprädiktion, welche dieses Schema besitzt, also oBdA.:
$S_i = \langle s_1^i, \ldots, s_{n_i}^i \rangle$ und $S_j = \langle s_1^i, \ldots, s_l^i, \ldots, s_l^i, \ldots, s_{n_i}^i \rangle$.

(c) $\langle s_1^m, \ldots, s_{n_m}^m \rangle \equiv \langle s_1, \ldots, s_n \rangle$."

Wir zeigen nun nacheinander die Aussagen (a), (b) und (c).

Zu (a): Wegen (4) und (2), (3), (5) wird der SGT-Ablauf S_0 vom Startzustand s aus erreicht. Wegen (4) und (6) wird vom SGT-Ablauf S_0 der Endzustand F erreicht. Die Wege von s zu S_0 und von S_0 zu F haben wegen (4), (5), (6) die Kantenbeschriftung ε.

S_0 ist ein SGT-Ablauf weil er wegen (7) alle Situationsschemata $s_1^0, \ldots, s_{n_0}^0$ in dem Wurzelsituationsgraphen entsprechend dessen temporalen Kanten verbindet. Alle Kantenbeschriftungen sind dabei ε. Mit (9) werden nun alle Situationsschema intern von ihrem Startknoten zu ihrem Endknoten verbunden. Die Kantenbeschriftung ist dabei s_i. Somit ist S_0 ein SGT-Verhalten im Wurzelgraphen, das vom Startzustand s aus den Endzustand F erreicht. Es ist gezeigt: W ist Wurzelgraph von $\mathcal{B}$ und $\langle s_1, \ldots, s_n \rangle$ ist SGT-Ablauf. Es gilt daher auch $s \xrightarrow{s_1 \cdots s_n} F$.

Zu (b): a) Spezialisierung: Der Übergang von S_i nach S_j mit $i = j - 1$ besteht in der Ersetzung von s_k^i in S_i durch die Folge $s_l^j, \ldots, s_{l+o}^j$ mit $s_k^i \rightsquigarrow s_l^j, \ldots, s_{l+o}^j$. Mit (8) und (2),(3),(5),(6) existieren solche Kanten von s_k^i zu S_i und wieder von S_i zurück zu s_k^i. S_i ist ein SGT-Ablauf weil er wegen (7) alle Situationsschemata in dem Situationsgraphen entsprechend den temporalen Kanten verbindet. Alle Kantenbeschriftungen sind dabei ε. Mit (9) werden nun alle Situationsschemata intern von ihrem Startknoten zu ihrem Endknoten verbunden. Die Kantenbeschriftung ist dabei s_i. Somit ist auch S_j ein SGT-Verhalten, das s_k^i durch $s_l^j, \ldots, s_{l+o}^j$ ersetzt, weil von s_l^j aus s_{l+o}^j erreicht wird.
b) Temporale reflexive Kante: Wegen (7).

Zu (c): Wegen 9.

Aus (a), (b) und (c) folgt, dass das Wort $s_1 \cdots s_n$ in der Sprache von $\mathcal{M}_\mathcal{B}$ enthalten ist, weil vom Startzustand s über einen Weg mit Kantenbeschriftung $s_1 \cdots s_n$ der Endzustand F erreicht werden kann.

B.4 Konstruktionsvorschrift zur Erzeugung eines SGT aus einem endlichen Automaten

Der endliche Automat $\mathcal{M}_\mathcal{B}$ ist durch ein Tupel $\mathcal{M}_\mathcal{B} = (Q, \Sigma, \delta, s, \mathcal{F})$ beschrieben mit Q als der Menge aller Zustände, Σ als das Eingabealphabet mit $Q \cap \Sigma = \emptyset$, $\delta : Q \times \Sigma \to Q$ ist die Überführungsfunktion, $s \in Q$ der Startzustand und $\mathcal{F} \subseteq Q$ die Menge der Endzustände.

Im Folgenden wird nun einen Konstruktionsvorschrift angegeben, mit der zu jedem endlichen Automaten ein syntaktisch äquivalenter SGT konstruiert werden kann.

(1) Eliminiere alle ε-Übergänge in dem Automaten.

(2) Für alle Kanten (i, j) mit Kantenbeschriftung s des endlichen Automaten wird jeweils ein Situationschema mit Bezeichner SS_{ij} (eindeutig!) und Zustandsschema s erzeugt, siehe Abbildung B.2.

(3) $\forall i : (i, j)$ mit i ist Startsituation, füge Situationsschema SS_{ij} die Starteigenschaft hinzu.

(4) $\forall j : (i, j)$ mit j ist Endsituation, füge Situationsschema SS_{ij} die Endeeigenschaft hinzu.

(5) $\forall j : \forall i : \forall k$: verbinde Situationsschema SS_{ij}, falls eine Kante (j, k) mit Kantenbeschriftung s_k existiert, mit allen Situationsschemata SS_{jk}, die s_k im Zustandsschema haben. Siehe Abbildung B.2 für ein Beispiel.

(6) Alle Situationsschemata sind Teil des Wurzelsituationsgraphen.

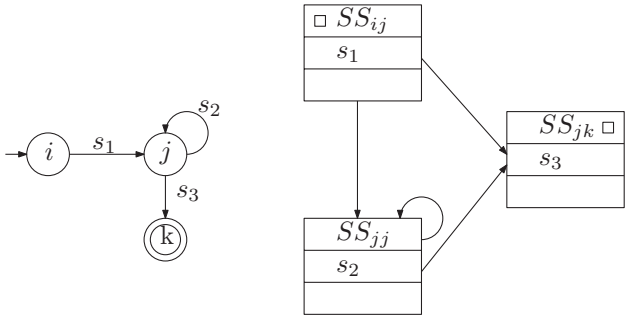


Abbildung B.1: Beispiel einer Umformung von einem endlichen Automaten in einen SGT.

Abbildung B.2: Konstruktion eines Situationsschemas aus einer Kante.

B.5 Ein Wort der Sprache des endlichen Automaten entspricht einem SGT-Verhalten

$s_1 \cdots s_n$ sei ein Wort der Sprache $\mathcal{L}(\mathcal{M}_\mathcal{B})$. Es ist nun zu zeigen, dass die Folge $\langle s_1, \ldots, s_n \rangle$ ein durch $\mathcal{B}$ definiertes SGT-Verhalten darstellt. Es wird nun im Folgenden gezeigt, dass alle Wege von Startzuständen $s \in Q$ zu Endzuständen $F \subseteq \mathcal{F}$ nur solche Kantenbeschriftungen haben, welche SGT-Abläufen entsprechen. Es wird nun gezeigt, dass

(a) Der SGT besitzt keine Spezialisierungen und

(b) alle Kantenübergänge mit Kantenbeschriftungen, die SGT-Verhalten entsprechen, produzieren korrekte SGT-Verhalten, sofern sie anwendbar sind.

Zu (a): Der gesamte endliche Automat $\mathcal{M}_\mathcal{B}$ wird in einen Situationsgraphen $\mathcal{B}$ transformiert und zwar einen einzigen, den Wurzelsituationsgraphen. Dieser besteht nur aus Situationsschemata, die mit temporalen Kanten verbunden sind.

Zu (b): Weil bei dem endlichen Automaten oBdA alle unerreichbaren Knoten entfernt werden können, existieren nach obiger Konstruktionsvorschrift in dem endlichen Automaten nur Pfade mit Kantenbeschriftungen s_i vom Startzustand s zum Endzustand F. Wie man leicht sieht wird die temporale Struktur des endlichen Automaten bei der Transformation in

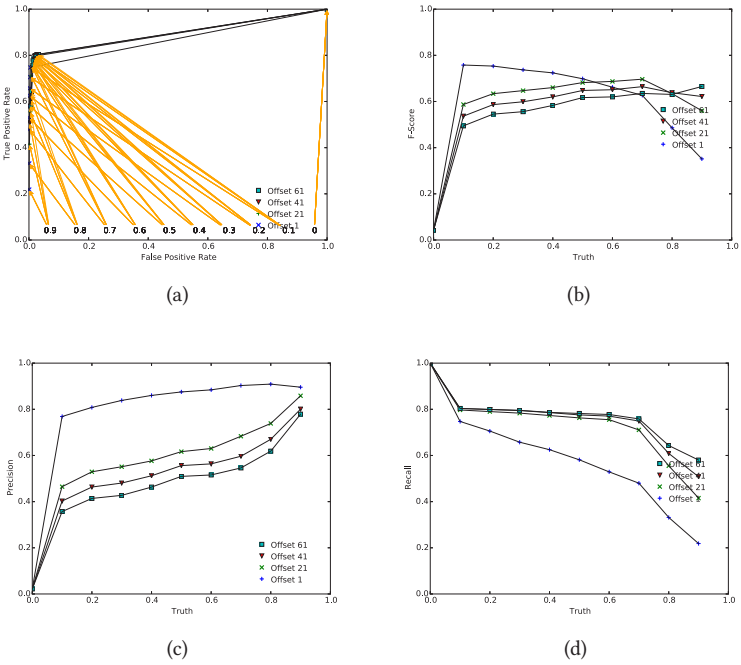
einen Situationsgraphen aufrechterhalten. Somit sind auch alle Pfade von Startsituationsschemata zu Endsituationsschemata korrekte SGT-Abläufe. Weil der SGT B nur aus einem einzigen Situationsgraphen, dem Wurzelsituationsgraphen W besteht, sind die SGT-Abläufe in W auch gleich korrekte SGT-Verhalten.

Aus (a) und (b) folgt, dass alle erstellten endlichen Automaten von Startzustand s nur über einen Weg mit Kantenbeschriftung $s_1 \cdots s_n$ zu einem Endzustand F führen.

Insgesamt haben wir gezeigt, dass jedes Wort der Sprache $\mathcal{L}(\mathcal{M}_{\mathcal{B}})$ einem durch $\mathcal{B}$ definierten SGT-Verhalten entspricht und umgekehrt.

C Weitere Auswertungen

In diesem Kapitel sind ergänzend zu Abschnitt 7.5 weitere Gesamtauswertungen dargestellt. Der erlaubte temporale Offset bei den erkannten Situationen variiert zu 1, 21, 41 und 61 Frames. Ein temporaler Offset darf nicht mit der zeitlichen Übereinstimmung in Abschnitt 7.4.1 verwechselt werden, sondern der hier eingeführte zeitliche Offset erlaubt, dass Detektion und Grundwahrheit um den zeitlichen Offset auseinander sein darf.



(a) (b)

(c) (d)

Abbildung C.1: BEHAVE BEHAVE3 (a) ROC (b) F-Score (c) Precision (d) Recall.

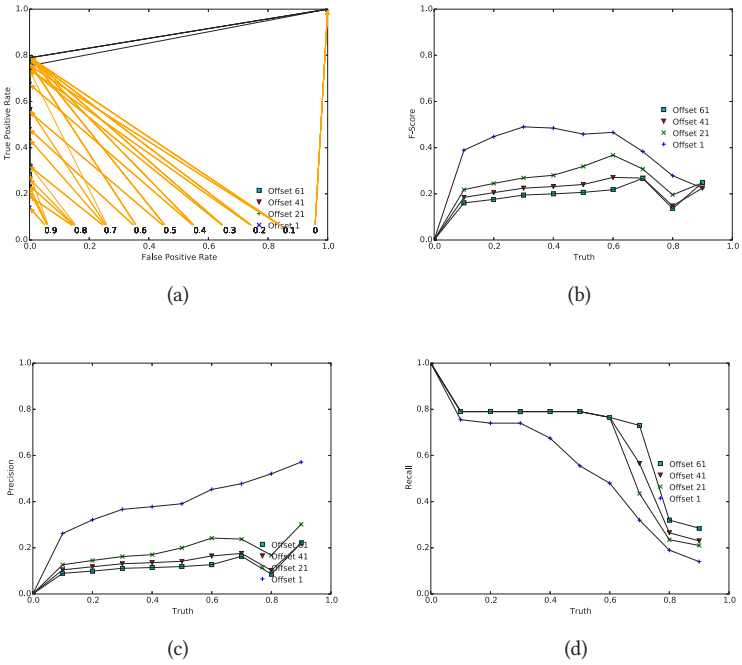
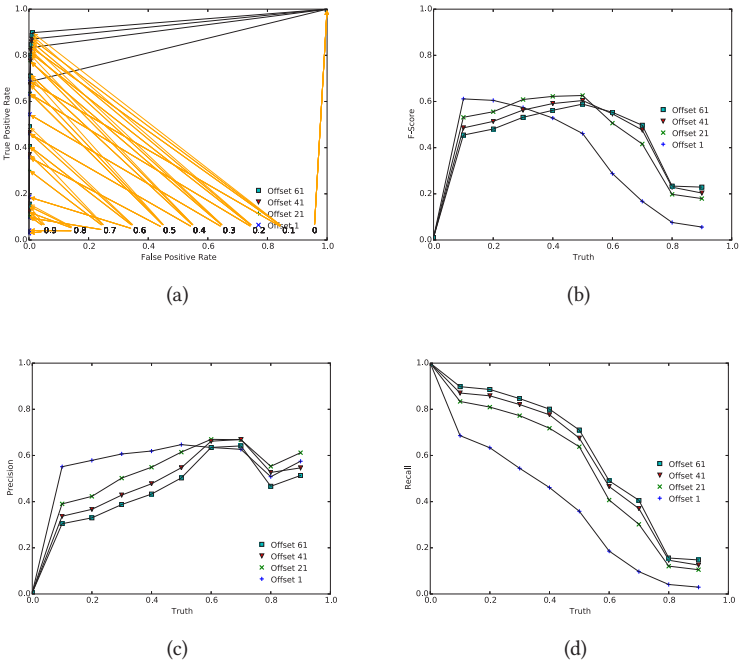


Abbildung C.2: BEHAVE BEHAVE5 (a) ROC (b) F-Score (c) Precision (d) Recall.



(a) (b)

(c) (d)

Abbildung C.3: BEHAVE BEHAVE6 (a) ROC (b) F-Score (c) Precision (d) Recall.

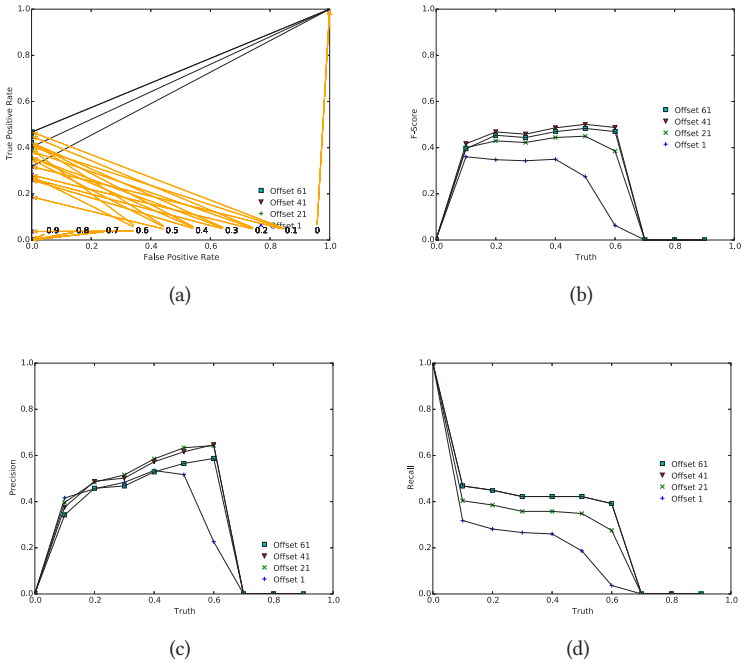


Abbildung C.4: BEHAVE BEHAVE7 (a) ROC (b) F-Score (c) Precision (d) Recall.

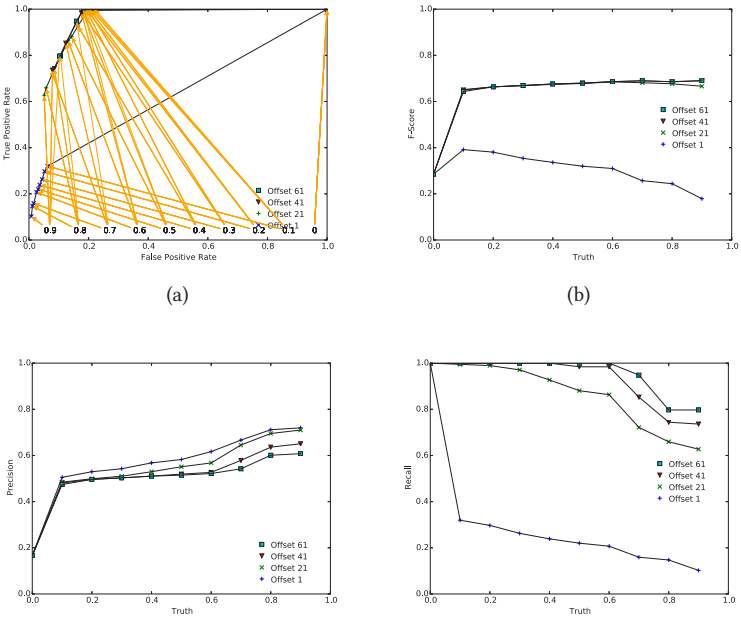


Abbildung C.5: CAVIAR CAVIAR01 (a) ROC (b) F-Score (c) Precision (d) Recall.

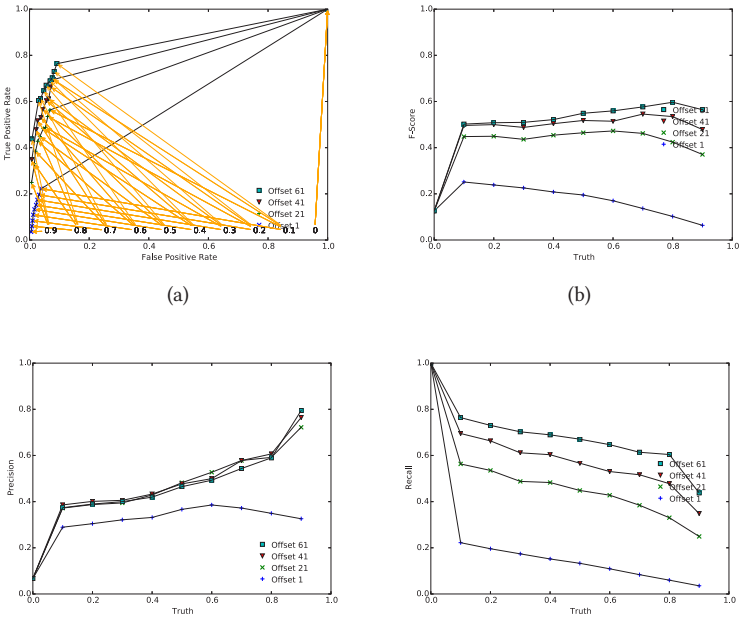


Abbildung C.6: CAVIAR CAVIAR08 (a) ROC (b) F-Score (c) Precision (d) Recall.

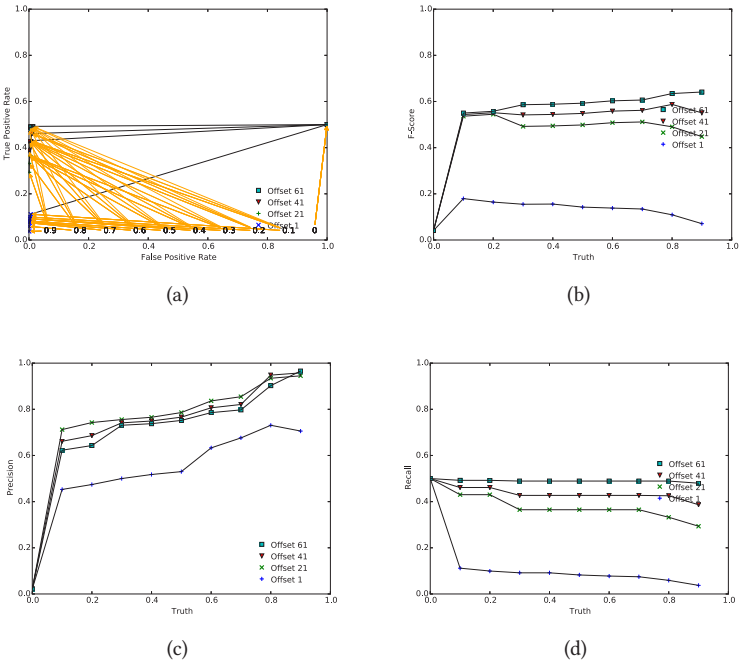


Abbildung C.7: CAVIAR CAVIAR11 (a) ROC (b) F-Score (c) Precision (d) Recall.

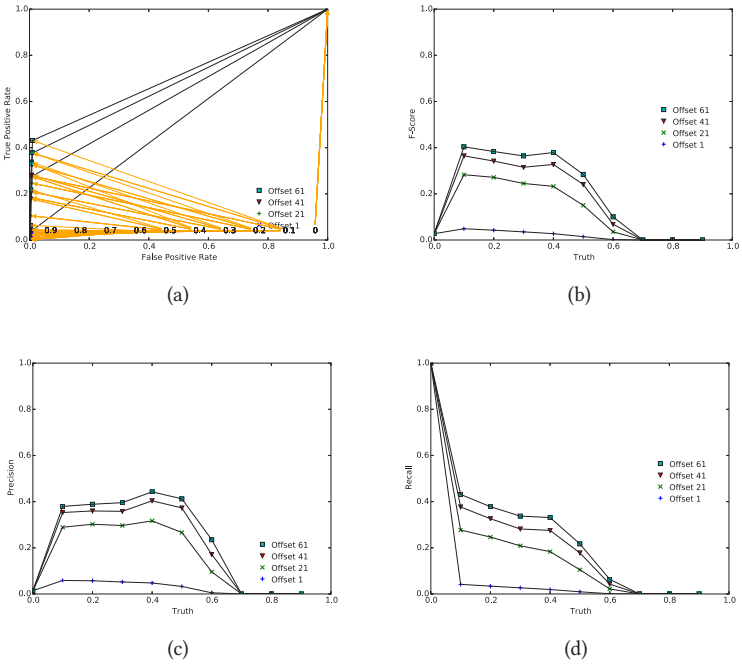


Abbildung C.8: CAVIAR CAVIAR12 (a) ROC (b) F-Score (c) Precision (d) Recall.

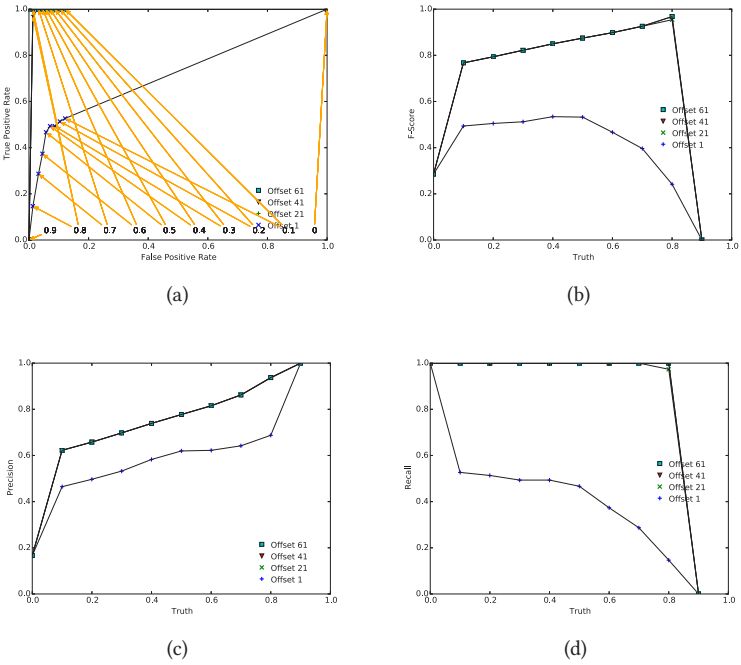


Abbildung C.9: CAVIAR CAVIAR18 (a) ROC (b) F-Score (c) Precision (d) Recall.

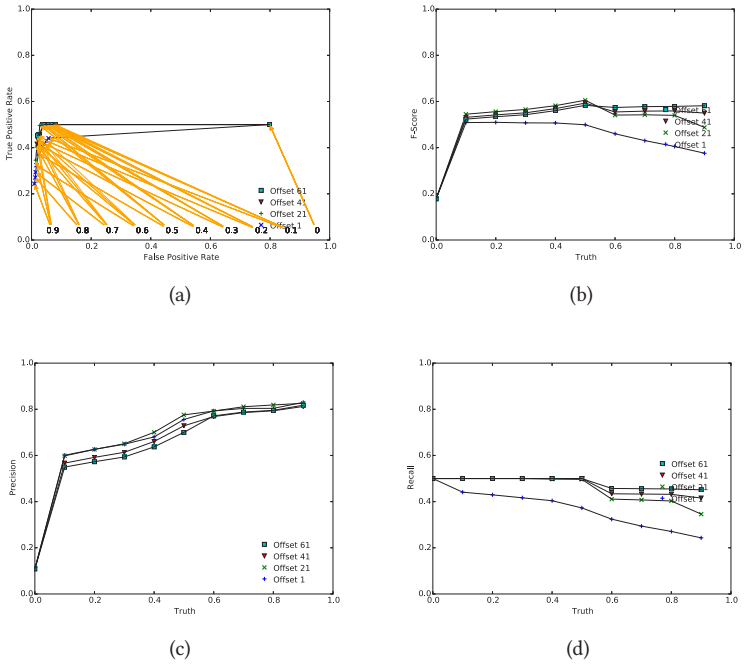


Abbildung C.10: VCA ACT2s2a (a) ROC (b) F-Score (c) Precision (d) Recall.

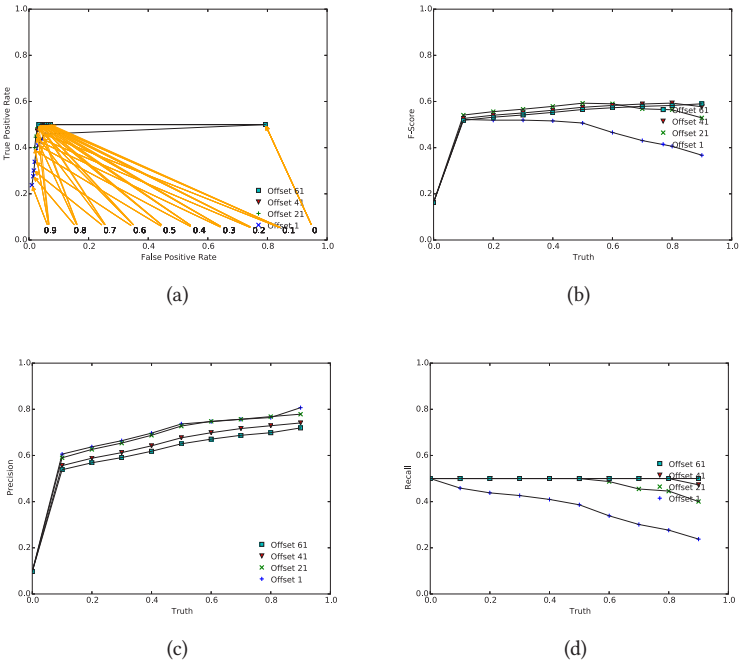


Abbildung C.11: VCA ACT2s2b (a) ROC (b) F-Score (c) Precision (d) Recall.

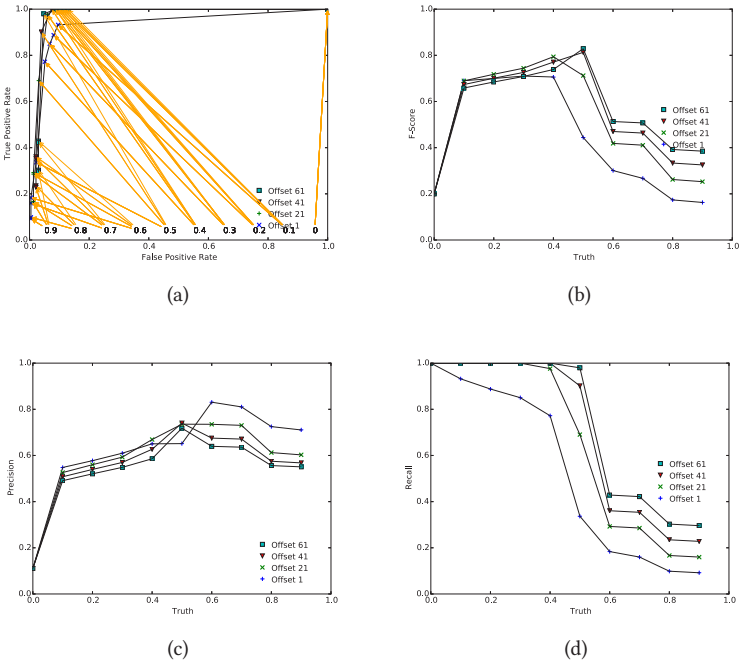
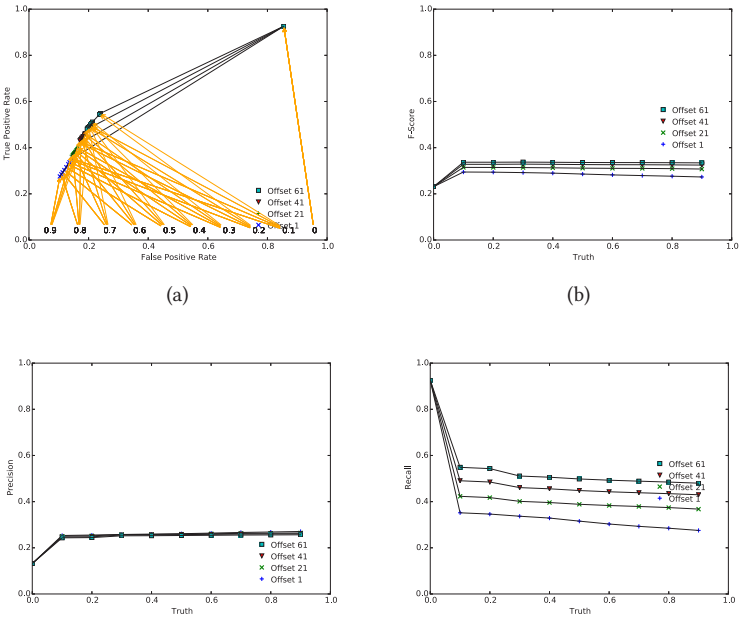


Abbildung C.12: VCA ACT2s4a (a) ROC (b) F-Score (c) Precision (d) Recall.



(a) (b)

(c) (d)

Abbildung C.13: VCA ACT3s1 (a) ROC (b) F-Score (c) Precision (d) Recall.

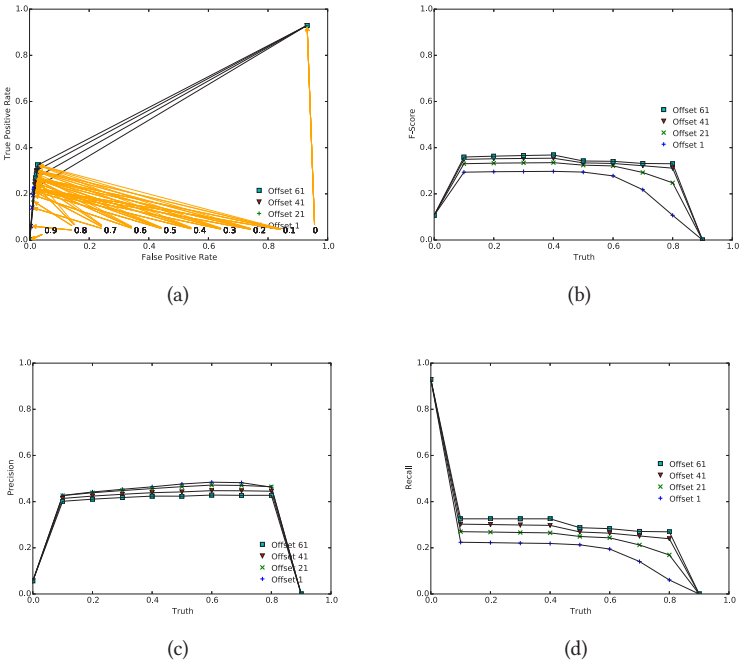
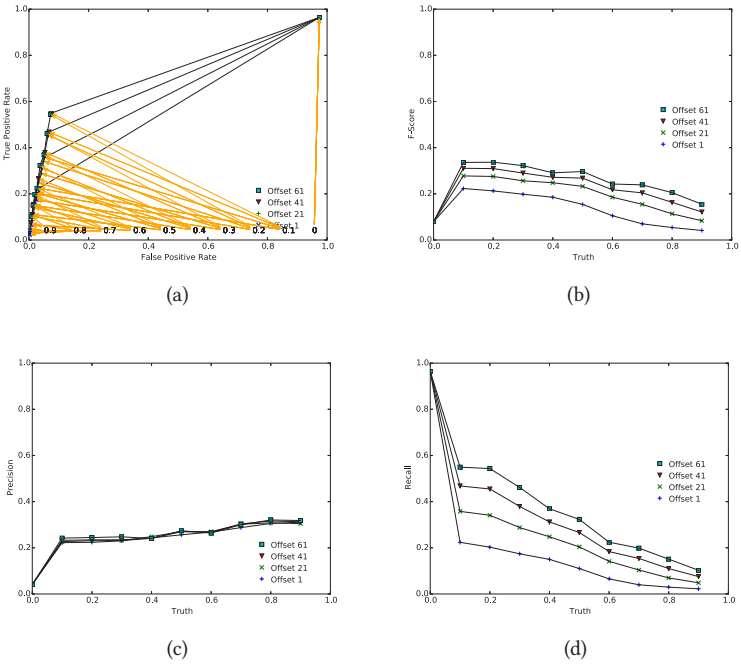


Abbildung C.14: VCA ACT3s4a (a) ROC (b) F-Score (c) Precision (d) Recall.



(a) (b)

(c) (d)

Abbildung C.15: VCA ACT3s4b (a) ROC (b) F-Score (c) Precision (d) Recall.

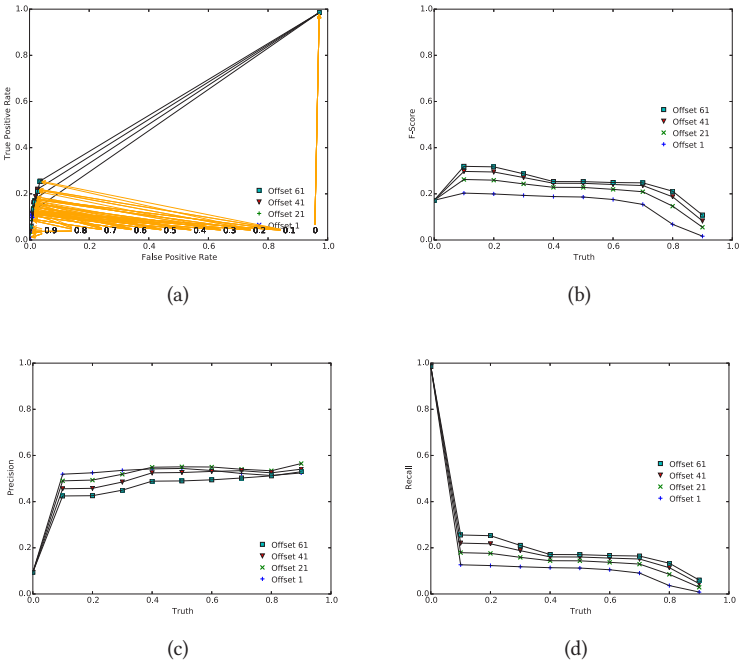


Abbildung C.16: VCA ACT3s5a (a) ROC (b) F-Score (c) Precision (d) Recall.

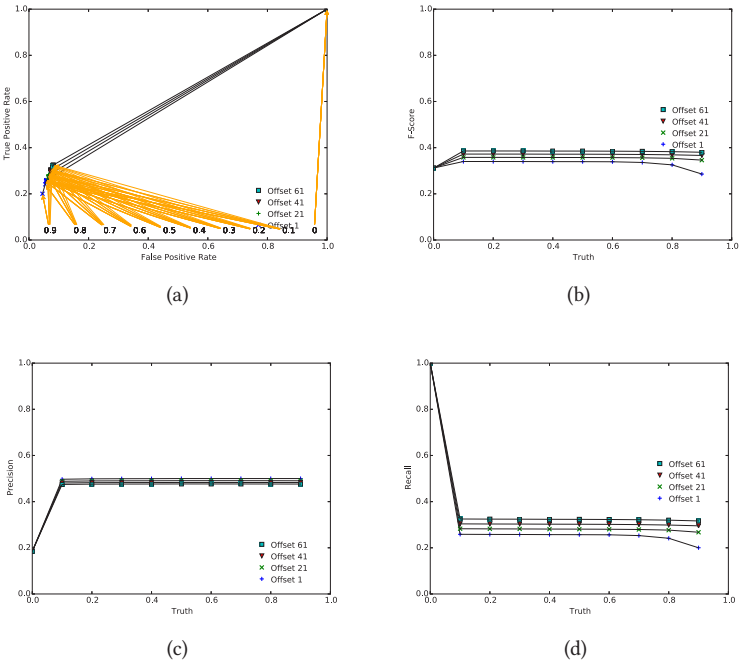


Abbildung C.17: VCA ACT3s5c (a) ROC (b) F-Score (c) Precision (d) Recall.

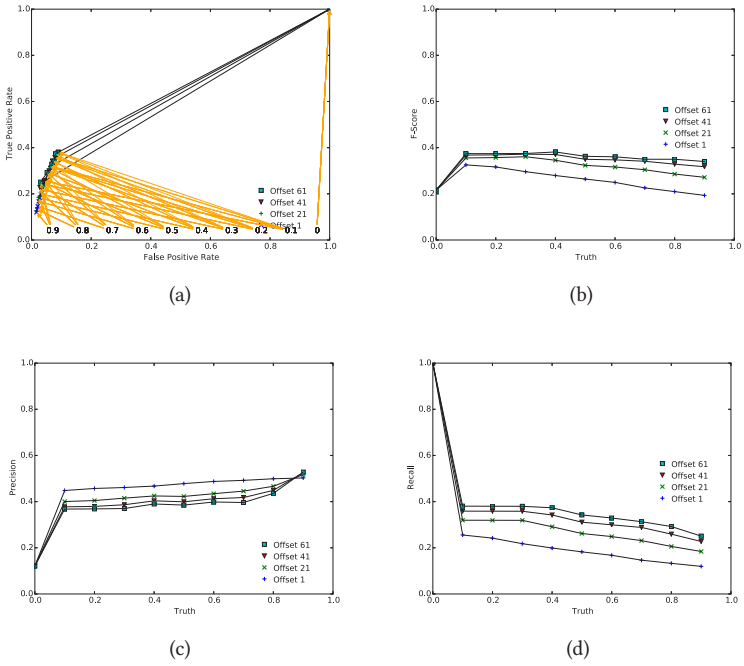


Abbildung C.18: VCA ACT3s7a (a) ROC (b) F-Score (c) Precision (d) Recall.

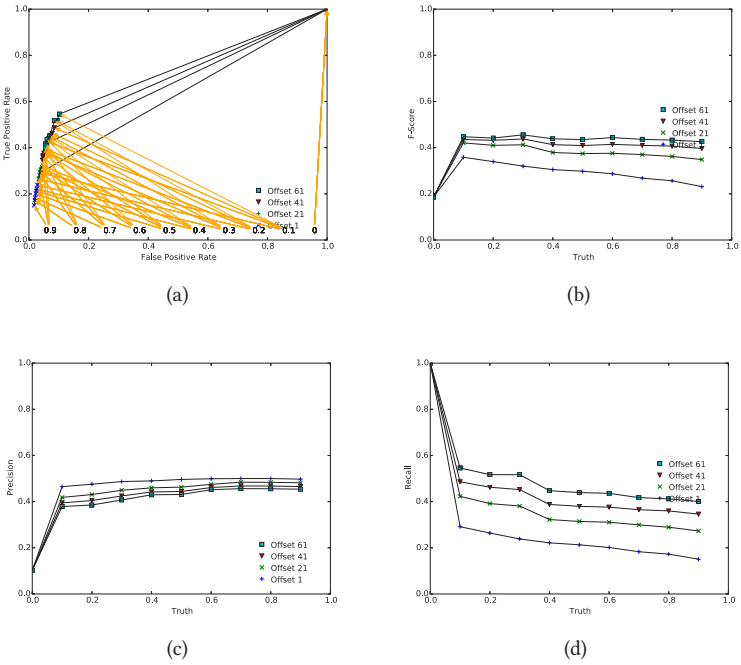


Abbildung C.19: VCA ACT3s7b (a) ROC (b) F-Score (c) Precision (d) Recall.

D Hintergrundwissen

D.1 Weitere Situationsmodellierung für den DOMUS-Datensatz

Wake Up

In Abbildung D.1 ist die Modelleriung der Situation **Wake Up** dargestellt. Sie entspricht dem linken Teil der Abbildung 8.3 Dieser SGT modelliert alle Aktivitäten vom Aufwachen bis zum Verlassen des Schlafzimmers. Unsere Modellierung unterscheidet zwischen zwei möglichen Szenarien: Anknipsen der Schlafzimmerlampe nach dem Aufwachen (siehe rechte Seite der Verfeinerung der Situation **sit_WakeUp_Items**) und Nichtbenutzung der Schlafzimmerlampe nach dem Aufwachen (siehe linke Seite der Verfeinerungder Situation **sit_WakeUp_Items**).

Man beachte, dass alle Sensoren, die in diesem Aktivitätsmodell involviert sind, zur besseren Lesbarkeit aus der Situation **sit_WakeUp_Items** eingeholt werden.

Use Toilet

Die Modellierung der Situation **Use Toilet** ist in Abbildung Figure D.2 dargestellt. Sie bezieht sich auf den zweiten Teil des Überblicks-SGT aus Abbildung Figure 8.3.

Die Definition des Prädikats *no_action_outside_bathroom(Agent)* benutzt schlicht die Ortseigenschaft der vorhandenen Sensoren. Allerdings für die Situation **sit_WaitInBathroom** benutzen wir dieses Prädikat nicht, sondern das differenziertere Prädikat *only_action_in_bathroom(Agent)* das zu-

Name	Year	Participants	Duration	Category	Annotated	Errors	Interweaving	Multiresident	Sensor Types	Num. Sensors	Tasks	File format	URL
CASAS\01-adlnormal	2008	24	1	ADL	y	n	n	Single	BM, I, F, CS	39	5	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\02-adlerror	2008	20	1	ADL	y	y	n	Single	BM, I, F, CS	39	5	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\03-adlinterweave	2008	20	1	ADL	y	n	y	Single	BM, I, F, T, CS	78	8	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\04-adlmr	2008	26	1	ADL	y	n	y	Pair	BM, I, F, T, D, LC, CS	99	15	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\05-wsuk1	2007	2	219	ADL	n	n	n	Single	BM, T	40	0	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\06-apartment	2008	2	3	ADL	n	n	n	Pair	BM, F	52	0	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\07-twor2009	2009	2	54	ADL	partially	n	y	Pair	BM, I, F, T, D, ES	3+n/a	13	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\08-twor.summer2009	2009	2	63	ADL	partially	n	y	Pair	BM, I, F, T, D, ES	3+n/a	13	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\09-tulum2009	2009	2	120	ADL	y	n	y	Pair	BM, T	20	10	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\10-tulum2	2010	2	184	ADL	y	n	y	Pair	BM, T	36	16	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\11-twor2010	2010	2	250	ADL	partially	n	y	Pair	BM, I, F, T, D, ES	3+n/a	25	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\12-actigraph	n/a	2	1	ADL	y	n	n	Single	T, A	2	7	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\14-cairo	2009	2	56	ADL	partially	n	y	Pair	BM, T	n/a	13	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\15-milan	2009	1	82	ADL	partially	n	n	Single	BM, T	n/a	15	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\17-aruba	2011	1	219	ADL	partially	n	n	Single	BM, T, D	n/a	15	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\18-puck	2011	40	1	ADL	partially	n	n	Single	BM, I, F, T, D, LC, CS, ES, A, S, W	1+n/a	6	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\21-assessmentdata	n/a	400	1	ADL	y	y	n	Single	BM, I, F, T, D, ES	3+n/a	24	TSV	http://ailab.wsu.edu/casas/datasets/
CASAS\synthetic_dataset	n/a	24	1	ADL	y	n	n	Single	BM, I, F, CS	39	5	TSV	http://ailab.wsu.edu/casas/datasets/
MITthesis_data	2003	2	14	ADL	y	n	n	Single	I, F, D, LC, CS, ES, W, AV	14+n/a	36	Matlab	http://courses.media.mit.edu/2004fall/mas622j/04.projects/home/
CMU	2009	43	1	Cooking	y	partially	n	Single	ACC, W, AV	19	5	n/a	http://kitchen.cs.cmu.edu/
Intel Lab Data	2004	n/a	23	Workplace	n	n	n	Multiple	T, ES, H, LS	n/a	0	TSV	http://db.csail.mit.edu/labdata/labdata.html
MERLSense	2008	n/a	365	Workplace	n	n	y	Multiple	BM	200	0	TSV	http://www.merl.com/wmd/
MavPad2004	2004	1	16	ADL	n	n	n	Single	BM, I, F, T, D, CS, H, LS	90+n/a	0	.txt table	http://ailab.wsu.edu/casas/datasets/mavpad2004.zip
MavPad2005	2005	1	39	ADL	n	n	n	Single	BM, I, F, T, D, CS, H, LS	90+n/a	0	.txt table	http://ailab.wsu.edu/casas/datasets/mavpad2005.zip
MavLab	2003	6	57	Workplace	n	n	n	Multiple	BM, I, T, H, LS	n/a	0	TSV	http://ailab.wsu.edu/casas/datasets/mavlab.zip
INRIA PRIMA	2007	n/a	1	Movement	y	n	n	Single	AV	1	5	XML	http://ailab.wsu.edu/casas/datasets/inria.zip
Multicom DOMUS	2011	24	1	ADL	y	partially	n	Single	BM, I, F, T, D, LC, ES, H, LS, FE	92	3	CSV	http://domuslab.fr/datasets/
Berkeley WARD	2008	20	1	Movement	n/a	partially	n	Single	W	5	13	Matlab	http://www.eecs.berkeley.edu/ yang/software/WAR/index.html
TIMC-IMAG\Distress	2008	10	1	Emergency	y	n	n	Single	BM, T, D, W, AV	17	5	n/a	https://sites.google.com/site/thonyfleury/health-smart-home-his-datasets [OFFLINE]
TIMC-IMAG\ADL	2008	15	1	ADL	y	n	n	Single	BM, T, D, W, AV	17	7	n/a	https://sites.google.com/site/thonyfleury/health-smart-home-his-datasets [OFFLINE]
UMassTrace\HomeA	2012	3	92	ADL	n	n	y	Multiple	BM, I, T, D, LC, ES, H	n/a	0	CSV	http://traces.cs.umass.edu/index.php/Smart/Smart
UMassTrace\HomeB	2012	4	93	ADL	n	n	y	Multiple	F, T, CS, ES, H	2+n/a	0	CSV	http://traces.cs.umass.edu/index.php/Smart/Smart
UMassTrace\HomeC	2011	n/a	91	ADL	n	n	y	Multiple	F, T, CS, ES, H	2+n/a	0	CSV	http://traces.cs.umass.edu/index.php/Smart/Smart
UMassTrace\Microgrid	2011	443	1	ADL	n	n	y	Multiple	ES	n/a	0	CSV	http://traces.cs.umass.edu/index.php/Smart/Smart
VanKasteren\HomeA	2010	1	25	ADL	y	n	n	Single	BM, I, F	n/a	10	Matlab	https://sites.google.com/site/tim0306/datasets
VanKasteren\HomeB	2010	1	14	ADL	y	n	n	Single	BM, I, F	n/a	13	Matlab	https://sites.google.com/site/tim0306/datasets
VanKasteren\HomeC	2010	1	19	ADL	y	n	n	Single	BM, I, F	n/a	16	Matlab	https://sites.google.com/site/tim0306/datasets
AMPds2	2013	n/a	730	ADL	n	n	n/a	n/a	F, ES	26	0	n/a	http://ampds.org/ (User account needed!)
DOMUS Sherbrooke	2008	6	1	ADL	y	n	n	Single	BM, I, F, D, LC	32	7	VNA	http://domus.usherbrooke.ca/jeux-de-donnees/

Tabelle D.1: Überblick von SmartHome Datensätzen. Sensorabkürzungen: A=activity monitor, ACC=accelerometer, AV=audio/video recording, BM=binary motion, CS=communication/speech, D=doors and windows, ES=electricity, F=facilities, H=humidity, FE=feeling tracking, I=items, LC=light controller, LS=light sensors, S=smart phone, T=temperature, W=wearable.

sätzlich etwas Rauschen des Küchensensors toleriert, wie es in einigen Teilen des Datensatzes vorkommt. Das macht die Situationsanalyse robuster.

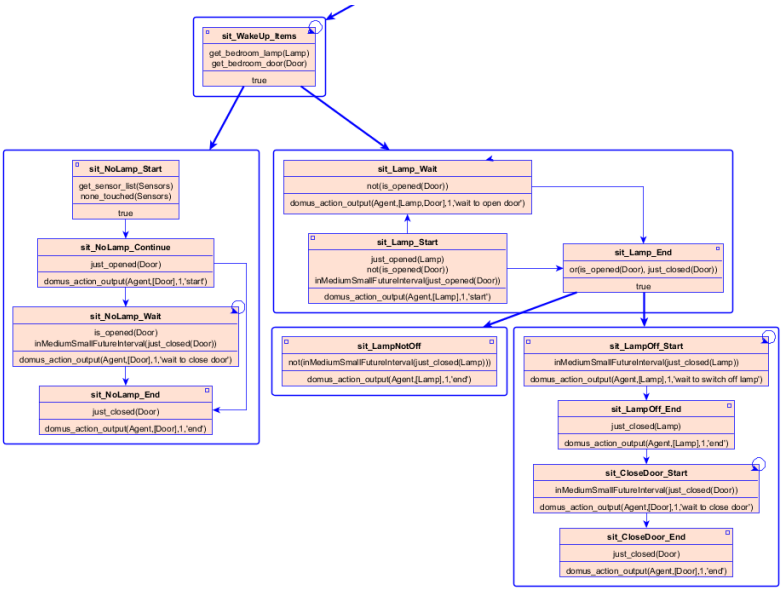


Abbildung D.1: Der erste Teil des DOMUS SGT modelliert Aktivität 1: **Wake Up**.

Prepare Breakfast

Abbildung D.3 veranschaulicht die Modellierung der Situation **Prepare Breakfast**. Es entspricht dem dritten Teil des Überblicks-SGT aus Abbildung 8.3.

Das Prädikat *actionInNextTwoMinutes(Agent)* stellt fest ob innerhalb der nächsten 120 Frames eine Aktivität eines *beliebigen* Sensors auftritt. Aufgrund seiner Einfachheit (besonders wegen der nicht vorhandenen Unschärfe) wird es aus Laufzeitgründen vorberechnet. Wir brauchen es aus mehreren Gründen: In der DOMUS Umgebung gibt es für das gesamte Esszimmer nur einen Infrarot-Bewegungs-Sensor. Deshalb gibt es während

des Frühstücks oft Sequenzen von mehr als zwei Minuten ohne Sensordaten (in der der Bewohner ruhig am Esstisch frühstückt und deshalb den Infrarot Sensor nicht aktiviert). Diese Eigenschaft wird in der Situation **sit_Prepare** in Abbildung D.3 verwendet und bildet eine implizite Verbindung zwischen den Situationen **Prepare Breakfast** und **Have Breakfast**, sodass in diesem Kontext keine Prädiktionskante erforderlich ist.

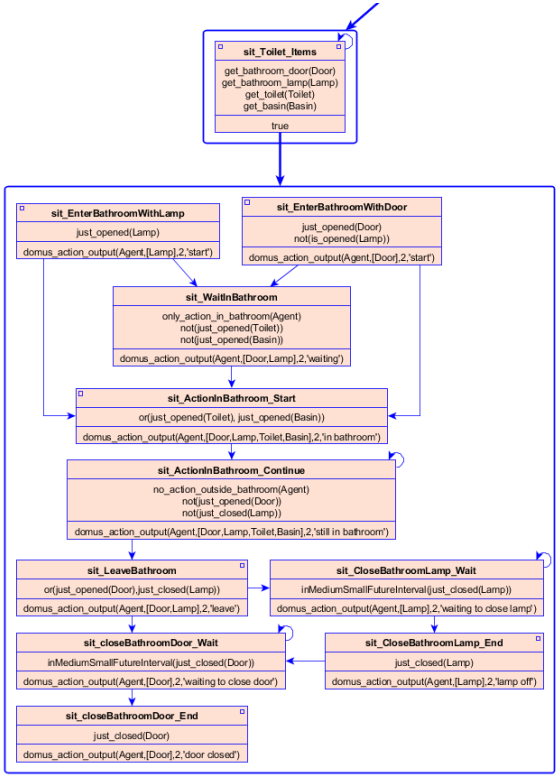


Abbildung D.2: Der zweite Teil des DOMUS SGT modelliert Aktivität 2: **Use Toilet**.

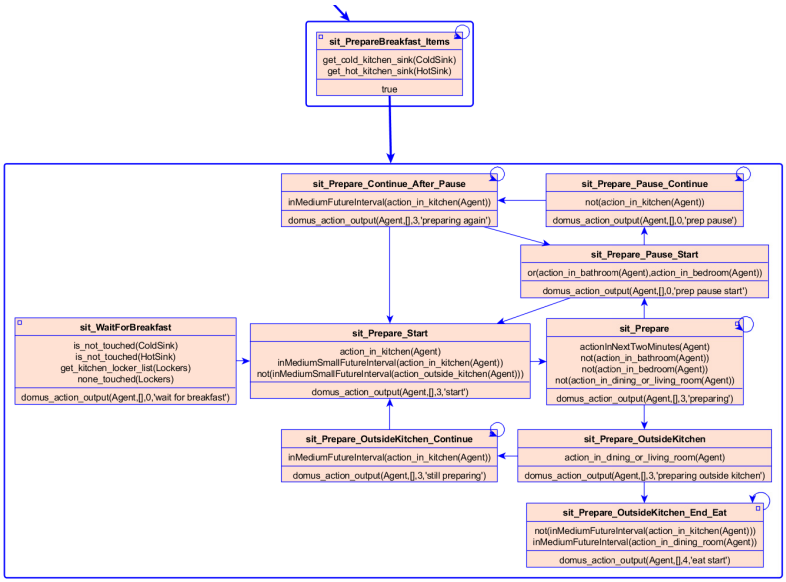
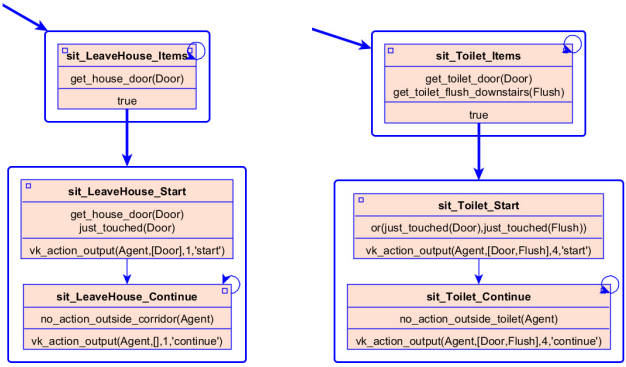


Abbildung D.3: Der dritte Teil des DOMUS SGT modelliert Aktivität 3: **Prepare Breakfast**.

D.2 Weitere Situationsmodellierung für den vanKasteren-Datensatz

Leave House

Abbildung D.4 (a) zeigt die Modellierung der Aktivitaet **Leave House**. Sie umfasst die gesamte Zeitspanne zwischen dem Verlassen und der Rückkehr in die Wohnung. Unser Modell verlässt sich lediglich auf die Sensoren auf dem Fußboden (in unseren Daten Korridor genannt). Für die Modellierung der Situation **sit_LeaveHouse_Continue** verwenden wir ein raumweises Handlungsprädikat, wie es bereits für den DOMUS-Datensatz diskutiert wurde.



(a) Activity 1: **Leave House**. (b) Activity 4: **Use Toilet Downstairs**.

Abbildung D.4: Zwei Teil-SGTs vom vanKasteren SGT.

Use Toilet Downstairs

Abbildung D.4 (b) veranschaulicht die Modellierung der Situation **Use Toilet Downstairs**. Wie man erkennt, ist die Modellierung dieser Situation in der vanKasteren Umgebung einfacher als die selbe Situation **Use Toilet** in der DOMUS Umgebung. Dafür gibt es mehrere Gründe: Zunächst ist die Sensordichte der Toilette der DOMUS Umgebung höher als die der vanKasteren Umgebung, da es bei DOMUS zusätzliche Sensoren im Waschbecken und in der Lampe gibt. Dies muss im Blick auf die präzise zeitliche Modellierung bedacht werden. Diese zusätzliche Information gibt es in der vanKasteren Umgebung nicht. Außerdem ist im Hinblick auf Rauschen zu beachten, dass in der DOMUS Umgebung ein Infrarotsensor der Küche direkt neben dem Badezimmer installiert ist. In der vanKasteren Umgebung ist das nicht der Fall, was die Modellierung wiederum einfacher macht. Man beachte, dass es möglich gewesen wäre, diesen Teil der vanKasteren Modellierung für den DOMUS-Datensatz zu verwenden - jedoch mit weitaus schlechteren Ergebnissen.

Go to Bed

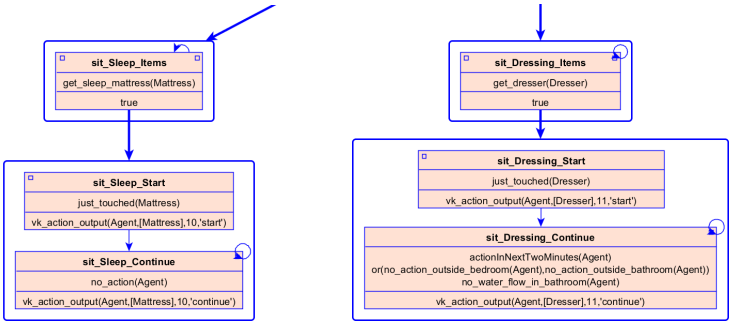
Abbildung D.5 zeigt die Modellierung der Situation **Go to Bed**. Diese ist ein relativ einfaches Vorgehen, das nur auf dem Drucksensor des Bettes basiert. In unserem Modell wird die Situation **Go to Bed** so lange erfasst, als es keine weitere widersprüchliche Information gibt (beispielsweise ein Sensorsignal der Schlafzimmertür, woraus man ableiten kann, dass der Bewohner nun nicht mehr schläft)[16] Wie haben dieses Paradigma bereits ausführlich für die Situationsmodellierung beim DOMUS-Datensatz verwendet.

Nun könnte man argumentieren, dass eine berührte Matratze allein nicht ausreicht, um daraus **Go to Bed** schlusszufolgern, da durchaus denkbar ist, dass der Bewohner sich nur für fünf Minuten im Bett ausruht. Um solche Fehler allerdings auszuschließen, müsste man über das derzeit betrachteten Zeitintervall unserer SGT-Modellierung hinausgehen. Dies ist weder im Hinblick auf die Laufzeit der Inferenz noch im Blick auf eine Anwendung in Echtzeit wünschenswert.

Get Dressed

Abbildung D.5 veranschaulicht die Modellierung der Situation **Get Dressed**. Der Bewohner zieht sich wahrscheinlich entweder im Schlafzimmer oder im Badezimmer an. Um jedoch zu vermeiden, das die Aktivität fälschlicherweise angezeigt wird während der Bewohner duscht oder sich rasiert, fordern wir außerdem mit dem Prädikat *no_water_flow_in_bathroom* explizit, dass im Badezimmer keine Wasserhahn aktiviert ist.

[16] Hier wird deutlich, dass unser Modell keine Einbrecher erfasst. Dieses Szenario würde zur Situationsanalyse in einer Umgebung mit mehreren Bewohnern mit verflechteten Situationen gehören, was hier nicht diskutiert wird.



(a) Activity 10: **Go to Bed**. (b) Activity 11: **Get Dressed**.

Abbildung D.5: Zwei Teil-SGTs vom vanKasteren SGT.

D.3 FMTHL Regeln

Beim vanKasteren-Datensatz gibt es neben den gewöhnlichen FMTHL Regeln zwei besondere Regeltypen. Das sind die **_start_predicate* und **_continue_predicate* Regeln. Bei den **_start_predicate* werden die Vorbedingungen modelliert, wie z.B. bei *toilet_start_predicate*

```
always (toilet_start_predicate(Agent,List) :-
 get_toilet_flush_upstairs(Flush) ,
 List = [Flush] ,
 inMediumLargeFutureInterval(just_touched(Flush))
).
```

Die **_continue_predicate* Regeln sind alle gegenseitig ausschließend modelliert. Hier das Beispiel *toilet_continue_predicate*:

```
always (toilet_continue_predicate(Agent,List) :-
  (no_action_outside_bathroom(Agent)) ,@W
  not@W (shower_start_predicate(Agent,ShowerList)) ,@W
  not@W (brush_teeth_start_predicate(Agent,Brushlist)) ,@W
  not@W (shave_start_predicate(Agent,Shavelist))
).
```

Die Definition der Regeln für **Take Shower**, **Brush Teeth** und **Shave** verläuft analog.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

J.K. Aggarwal and M.S. Ryoo. Human Activity Analysis: A Review. *ACM Computing Surveys*, 43(3):16:1–16:43, 2011. Doi: 10.1145/1922649.1922653. (Zitiert auf Seiten 13 und 14).

J.F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983. Doi: 10.1145/182.358434. (Zitiert auf Seite 21)

J.F. Allen and G. Ferguson. Actions and Events in Interval Temporal Logic. *Journal of Logic and Computation*, 4(5):531–579, 1994. Doi: 10.1093/logcom/4.5.531. (Zitiert auf Seite 21)

Y. Aloimonos, G. Guerra, and A. Ogale. *Human-Centric Interfaces for Ambient Intelligence*, chapter The Language of Action: A New Tool for Human-Centric Interfaces., pages 95–129. Elsevier, 2009. Doi: 10.1016/B978-0-12-374708-2.00005-X. (Zitiert auf Seite 19)

J.R. Anderson, D. Bothell, M.D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An Integrated Theory of the Mind. *Psychological review*, 111(4):1036, 2004. Doi: 10.1037/0033-295X.111.4.1036. (Zitiert auf Seite 8)

M. Arens. *Repräsentation und Nutzung von Verhaltenswissen in der Bildfolgenauswertung*. Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Dissertationen zur Künstlichen Intelligenz (DISKI) 287, Akademische Verlagsgesellschaft Aka GmbH, 2004. (Zitiert auf Seiten 4, 9, 23, 37, 38, 40, 41, 46, 169, 170 und 172).

M. Arens and H.-H. Nagel. Behavioral Knowledge Representation for the Understanding and Creation of Video Sequences. In *Proceedings of the 26th German Conference on Artificial Intelligence (KI 2003)*, pages 149–163. Springer, 2003. Doi: 10.1007/978-3-540-39451-8_12. (Zitiert auf Seiten 23, 37, 42 und 139).

M. Arens, R. Gerber, and H.-H. Nagel. Conceptual representations between video signals and natural language descriptions. *Image and Vision Computing*, 26(1):53–66, 2008. Doi: 10.1016/j.imavis.2005.07.026. (Zitiert auf Seiten 2, 6, 23 und 71).

F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2 edition, 2010. ISBN: 9780521150118. (Zitiert auf Seite 43)

J. Badri, C. Tilmant, J.-M. Lavest, Q.-C. Pham, and P. Sayd. Camera-to-Camera Mapping for Hybrid Pan-Tilt-Zoom Sensors Calibration. In *Proceedings of the 15th Scandinavian Conference (SCIA 2007)*, volume 4522 of *LNCS*, pages 132–141. Springer, 2007. Doi: 10.1007/978-3-540-73040-8_14. (Zitiert auf Seite 157)

P. Baiget, C. Fernández, X. Roca, and J. Gonzàlez. Automatic learning of conceptual knowledge for the interpretation of human behavior in video sequences. In *3rd Iberian Conference on Pattern Recognition and Image Analysis (Ibpria 2007)*, Girona, Spain, 2007. Springer LNCS. Doi: http://dx.doi.org/10.1007/978-3-540-72847-4_65. (Zitiert auf Seite 24)

T. Bär. *Analyse des situativen Fahrerverhaltens zur benutzeradaptiven Assistenz*. PhD thesis, Fakultät für Informatik der Universität Karlsruhe (TH), 2016. (Zitiert auf Seite 26)

D.P. Barrett, A. Barbu, N. Siddharth, and J.M. Siskind. Saying What You're Looking For: Linguistics Meets Video Search. *IEEE Transactions on Pat-*

tern Analysis and Machine Intelligence, 38(10):2069–2081, 2016. Doi: 10.1109/TPAMI.2015.2505297. (Zitiert auf Seite 19)

S. Bauer. Comparing Ontologies and Situation Graph Trees in Cognitive Vision Systems. Diplomarbeit, Fakultät für Informatik. Institut für Anthropomatik (IFA). KIT, 2012. (Zitiert auf Seite 43)

M. Bäuml, M. Tapaswi, A. Schumann, and R. Stiefelhagen. Contextual Constraints for Person Retrieval in Camera Networks. In *IEEE Conference on Advanced Video and Signal-based Surveillance (AVSS 2012)*, pages 221–227. IEEE, 2012. Doi: 10.1109/AVSS.2012.28. (Zitiert auf Seite 142)

E. Begelfor and M. Werman. How to Put Probabilities on Homographies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10): 1666–1670, 2005. Doi: 10.1109/TPAMI.2005.200. (Zitiert auf Seite 160)

N. Bellotto, B. Benfold, H. Harland, H.-H. Nagel, N. Pirlo, I. Reid, E. Sommerlade, and C. Zhao. Cognitive Visual Tracking and Camera Control. *Computer Vision and Image Understanding*, 116(3):457–471, 2012. Doi: 10.1016/j.cviu.2011.09.011. (Zitiert auf Seiten 24, 155 und 156).

R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pages 2903–2910. IEEE, 2012. Doi: 10.1109/CVPR.2012.6248017. (Zitiert auf Seite 90)

R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten Years of Pedestrian Detection, What Have We Learned? In *European Conference on Computer Vision (ECCV 2014 Workshops)*, volume 8926, pages 613–627. Springer, 2015. Doi: 10.1007/978-3-319-16181-5_47. (Zitiert auf Seite 90)

J.O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics. Springer, 1985. Doi: 10.1007/978-1-4757-4286-2. (Zitiert auf Seite 16)

S. Blunsden and R. Fisher. The BEHAVE video dataset: ground truthed video for multi-person behavior classification. *Annals of the BMVA*, 2010 (4):1–11, 2010. (Zitiert auf Seiten 92 und 93).

W. Bohlken. *Realzeit-Szeneninterpretation mit ontologiebasierten Regeln*. Dissertation, Universität Hamburg, Fakultät für Informatik, 2012. (Zitiert auf Seite 26)

W. Bohlken, B. Neumann, L. Hotz, and P. Koopmann. Ontology-Based Real-time Activity Monitoring Using Beam Search. In *Proceedings of the 8th International Conference on Computer Vision Systems (ICVS 2011)*, volume 6962 of *LNCS*, pages 112–121. Springer, 2011. ISBN 978-3-642-23968-7. Doi: 10.1007/978-3-642-23968-7_12. URL ok. (Zitiert auf Seite 26)

R. Brachman and H. Levesque. *Knowledge Representation and Reasoning*. Artificial Intelligence. Morgan Kaufmann, 2004. ISBN 1558609326. (Zitiert auf Seite 35)

J. Brauer, W. Hübner, and M. Arens. Generative 2D and 3D Human Pose Estimation with Vote Distributions. In *8th International Symposium (ISVC 2012)*, volume 7431 of *LNCS*, pages 470–481. Springer, 2012. Doi: 10.1007/978-3-642-33179-4_45. (Zitiert auf Seite 155)

H.K. Buning, M. Karpinski, and A. Flogel. Resolution for Quantified Boolean Formulas. *Information and Computation*, 117(1):12–18, 1995. Doi: 10.1006/inco.1995.1025. (Zitiert auf Seite 35)

U. Cahn von Seelen. Ein Formalismus zur Beschreibung von Bewegungsverben mit Hilfe von Trajektorien. Diplomarbeit, Universität Karlsruhe, Fakultät für Informatik, 1988. (Zitiert auf Seite 23)

CAVIAR. EC Funded CAVIAR project / IST 2001 37540, http://homepages.inf.ed.ac.uk/rbf/CAVIAR/, 2001. (Zitiert auf Seite 98)

Y. Cheng. Mean Shift, Mode Seeking, and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995. Doi: 10.1109/34.400568. (Zitiert auf Seite 73)

B. Chikhaoui, S. Wang, and H. Pigot. A New Algorithm Based On Sequential Pattern Mining For Person Identification In Ubiquitous Environments. In *4th International Workshop on Knowledge Discovery from Sensor Data (KDD 2010)*, pages 19–28. ACM, 2010. (Zitiert auf Seiten 117, 124 und 127).

D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002. Doi: 10.1109/34.1000236. (Zitiert auf Seiten 73 und 75).

P. Dai, H. Di, L. Dong, L. Tao, and G. Xu. Group Interaction Analysis in Dynamic Context. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(1):275–282, 2008. Doi: 10.1109/TSMCB.2007.909939. (Zitiert auf Seite 17)

J. Davis and X. Chen. Calibrating pan-tilt cameras in wide-area surveillance networks. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 1, pages 144–149. IEEE, 2003. Doi: 10.1109/ICCV.2003.1238329. (Zitiert auf Seite 157)

A. Del Bimbo, F. Dini, G. Lisanti, and F. Pernici. Exploiting distinctive visual landmark maps in pan-tilt-zoom camera networks. *Computer Vision and Image Understanding*, 114(6):611–623, 2010. Doi: 10.1016/j.cviu.2010.01.007. Special Issue on Multi-Camera and Multi-Modal Sensor Fusion. (Zitiert auf Seite 156)

P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. In *Proceedings of the British Machine Vision Conference (BMVC 2009)*, pages 91.1–91.11. BMVC Press, 2009. Doi: 10.5244/C.23.91. (Zitiert auf Seite 90)

P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian Detection: An Evaluation of the State of the Art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012. Doi: 10.1109/TPAMI.2011.155. (Zitiert auf Seite 155)

P.M. Domingos, S. Kok, H. Poon, M. Richardson, and P. Singla. Unifying Logical and Statistical AI. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006)*, volume 1, pages 2–7. AAAI Press, 2006. ISBN: 978-1-57735-281-5. (Zitiert auf Seite 25)

L. Donahue, J.and Anne Hendricks, S. Guadarrama, M. Rohrbach, S.i Venugopalan, K. Saenko, and T. Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, pages 2625–2634. IEEE, 2015. Doi: 10.1109/TPAMI.2016.2599174. (Zitiert auf Seite 28)

H.D. Ebbinghaus, J. Flum, and W. Thomas. *Einführung in die mathematische Logik*. Springer Spektrum, 5 edition, 2007. ISBN 9783411156030. ISBN: 978-3-8274-1691-9. (Zitiert auf Seite 35)

A. Ellis, A. Shahrokni, and J.M. Ferryman. PETS2009 and Winter-PETS 2009 Results: A Combined Evaluation. In *Proceedings of the Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-Winter 2009)*, pages 1–8. IEEE, 2009. Doi: 10.1109/PETS-WINTER.2009.5399728. (Zitiert auf Seite 97)

H.B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 2 edition, 2001. ISBN: 978-0122384523. (Zitiert auf Seite 34)

I. Everts, N. Sebe, and G. Jones. Cooperative Object Tracking with Multiple PTZ Cameras. In *Proceedings of the 14th International Conference on Image Analysis and Processing (ICIAP 2007)*, pages 323–330. IEEE, 2007. Doi: 10.1109/ICIAP.2007.4362799. (Zitiert auf Seite 156)

C. Fernández Tena. *Understanding Image Sequences: the Role of Ontologies in Cognitive Vision*. Dissertation, Universitat Autònoma de Barcelona. Departament de Ciències de la Computació, 2010. (Zitiert auf Seiten 24 und 53).

J. Ferryman and A. Shahrokni. PETS2009: Dataset and Challenge. In *12th IEEE International Workshop on Performance Evaluation of Tracking*

and Surveillance (PETS-Winter 2009), pages 1–6. IEEE, 2009. Doi: 10.1109/PETS-WINTER.2009.5399556. (Zitiert auf Seite 97)

Y. Fischer. A Top-Down-View on Intelligent Surveillance Systems. In *Proceedings of the Seventh International Conference on Systems (ICONS 2012)*, pages 43–48. IARIA, 2012. (Zitiert auf Seite 17)

Y. Fischer and J. Beyerer. Defining Dynamic Bayesian Networks for Probabilistic Situation Assessment. In *Proceedings of the 15th International Conference on Information Fusion (FUSION 2012)*, pages 888–895. IEEE, 2012. ISBN: 978-1-4673-0417-7. (Zitiert auf Seite 17)

M.A. Fischler and R.C. Bolles. 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. Doi: 10.1145/358669.358692. (Zitiert auf Seite 149)

W. Ge, R.T. Collins, and R.B. Ruback. Vision-based Analysis of Small Groups in Pedestrian Crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1003 –1016, 2012. Doi: 10.1109/TPAMI.2011.176. (Zitiert auf Seite 73)

C.W. Geib. Delaying Commitment in Plan Recognition Using Combinatory Categorial Grammars. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence (AI 2009)*, pages 1702–1707. Morgan Kaufmann, 2009. (Zitiert auf Seite 19)

A. Geiger. *Probabilistic Models for 3D Urban Scene Understanding from Movable Platforms.* Dissertation, Fakultät für Maschinenbau. Institut für Mess- und Regelungstechnik. KIT, 2013. (Zitiert auf Seite 3)

R. Gerber and H.-H. Nagel. Representation of Occurrences for Road Vehicle Traffic. *Artificial Intelligence*, 172(4-5):351–391, 2008. Doi: 10.1016/j.artint.2007.07.001. (Zitiert auf Seite 23)

S. Gong and T. Xiang. Recognition of Group Activities Using Dynamic Probabilistic Networks. In *Proceedings of the 9th IEEE International Conference on Computer Vision*, volume 2, pages 742 –749. IEEE, 2003. Doi: 10.1109/ICCV.2003.1238423. (Zitiert auf Seite 17)

S. Gong and T. Xiang. *Visual Analysis of Behaviour. From Pixels to Semantics*. Springer, 2011. Doi: 10.1007/978-0-85729-670-2. (Zitiert auf Seite 13)

J. González, J. Varona, FX Roca, and JJ Villanueva. Situation graph trees for human behavior modeling. *Recent Advances In Artificial Intelligence Research And Development*, 1:85, 2004. (Zitiert auf Seite 24)

J. González, D. Rowe, J. Varona, and F.X. Roca. Understanding dynamic scenes based on human sequence evaluation. *Image and Vision Computing*, 27(10):1433–1444, 2009. Doi: 10.1016/j.imavis.2008.02.004. Special Section: Computer Vision Methods for Ambient Intelligence. (Zitiert auf Seite 24)

G. Görz, J. Schneeberger, and U. Schmid, editors. *Handbuch der künstlichen Intelligenz*. De Gruyter, 5 edition, 2013. ISBN: 978-3486713077. (Zitiert auf Seite 49)

B. Gottfried. Behaviour Monitoring and Interpretation. A Computational Approach to Ethology. In *Proceedings of the 32nd Annual Conference on AI (KI 2009)*, volume 5803 of *LNCS*, pages 572–580. Springer, 2009. Doi: 10.1007/978-3-642-04617-9_72. (Zitiert auf Seite 13)

A.-K. Grosselfinger. Automatische Konfiguration eines Mehrkamerasystems zur Verfolgung von Objekten in unstrukturierten Umgebungen. Diplomarbeit, Fakultät für Informatik. Institut für Anthropomatik (IFA). KIT, 2012. (Zitiert auf Seite 165)

G. Guerra-Filho. *A Sensory-Motor Linguistic Framework for Human Activity Understanding*. phdthesis, University of Maryland, 2007. ISBN: 978-0-549-15936-0. (Zitiert auf Seite 20)

G. Guerra-Filho and Y. Aloimonos. A Language for Human Action. *Computer*, 40(5):42 –51, 2007. Doi: 10.1109/MC.2007.154. (Zitiert auf Seite 19)

A. Gupta, P. Srinivasan, J. Shi, and L.S. Davis. Understanding Videos, Constructing Plots Learning a Visually Grounded Storyline Model from Annotated Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 2012 –2019. IEEE, 2009. Doi: 10.1109/CVPR.2009.5206492. (Zitiert auf Seiten 21 und 22).

H. Harland. *Nutzung logikbasierter Verhaltensrepräsentationen zur natürlichsprachlichen Beschreibung von Videos.* Dissertation, KIT, 2011. (Zitiert auf Seite 60)

B. Hilsenbeck, D. Münch, A.-K. Grosselfinger, W. Hübner, and M. Arens. Action Recognition in the Longwave Infrared and the Visible Spectrum using Hough Forests. In *Proceedings of the IEEE International Symposium on Multimedia (ISM 2016)*, 2016a. (Zitiert auf Seite 92)

B. Hilsenbeck, D. Münch, H. Kieritz, W. Hübner, and M. Arens. Hierarchical Hough Forests for View-Independent Action Recognition. In *Proceedings of 23rd International Conference on Pattern Recognition (ICPR 2016)*, 2016b. (Zitiert auf Seite 92)

R. Horaud, D. Knossow, and M. Michaelis. Camera cooperation for achieving visual attention. *Machine Vision and Applications*, 16(6):331–342, 2006. Doi: 10.1007/s00138-005-0182-9. (Zitiert auf Seite 157)

W. Hu, T. Tan, L. Wang, and S. Maybank. A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):334–352, 2004. Doi: 10.1109/TSMCC.2004.829274. (Zitiert auf Seite 156)

J. IJsselmuiden. *Interaction Analysis in Smart Work Environments through Fuzzy Temporal Logic.* Dissertation, Fakultät fúr Informatik. Institut für Anthropomatik und Robotik (IAR). KIT, 2014. (Zitiert auf Seiten 3, 24, 54, 77 und 139).

J. IJsselmuiden and R. Stiefelhagen. Towards High-Level Human Activity Recognition through Computer Vision and Temporal Logic. In *Proceedings of the 33rd Annual German Conference on Advances in Artificial Intelligence (KI 2010)*, volume 6359 of *LNCS*, pages 426–435. Springer, 2010. Doi: 10.1007/978-3-642-16111-7_49. (Zitiert auf Seite 156)

J. IJsselmuiden, A.-K. Grosselfinger, D. Münch, M. Arens, and R. Stiefelhagen. Automatic Behavior Understanding in Crisis Response Control Rooms. In *Proceedings of the Third International Joint Conference on Ambient Intelligence (Aml 2012)*, volume 7683 of *LNCS*, pages 97–112. Springer, 2012. Doi: 10.1007/978-3-642-34898-3_7. (Zitiert auf Seite 54)

J. IJsselmuiden, D. Münch, A.-K. Grosselfinger, M. Arens, and R. Stiefelhagen. Automatic Understanding of Group Behavior Using Fuzzy Temporal Logic. *Journal of Ambient Intelligence and Smart Environments*, 6(6): 623–649, 2014. Doi: 10.3233/AIS-140290. (Zitiert auf Seiten ii und vi).

Y.A. Ivanov and A.F. Bobick. Recognition of Visual Activities and Interactions by Stochastic Parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000. Doi: 10.1109/34.868686. (Zitiert auf Seite 19)

A. Jain, D. Kopell, K. Kakligian, and Y.-F. Wang. Using Stationary-Dynamic Camera Assemblies for Wide–area Video Surveillance and Selective Attention. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, volume 1, pages 537–544. IEEE, 2006. Doi: 10.1109/CVPR.2006.327. (Zitiert auf Seite 157)

D. Jain. *Probabilistic Cognition for Technical Systems: Statistical Relational Models for High-Level Knowledge Representation, Learning and Reasoning*. Dissertation, Technische Universität München, Fakultät für Informatik, 2012. (Zitiert auf Seite 26)

S.-W. Joo. Attribute Grammar-Based Event Recognition and Anomaly Detection. In *Proceedings of the Conference on Computer Vision and Pat-*

tern Recognition Workshop (CVPRW 2006). IEEE, 2006. Doi: 10.1109/CVPRW.2006.32. (Zitiert auf Seite 19)

R. Kadouche, H. Pigot, B. Abdulrazaka, and S. Giroux. Support Vector Machines for Inhabitant Identification in Smart Houses. In *Proceedings of the7th International Conference on Ubiquitous Intelligence and Computing (UIC 2010)*, volume 6406 of *LNCS*, pages 83–95. Springer, 2010. Doi: 10.1007/978-3-642-16355-5_9. (Zitiert auf Seiten 117, 118 und 119).

A. Katumba. Rule Based Situation Recognition in Video Streams. Diplomarbeit, Fakultät für Maschinenbau. Institut für Mess- und Regelungstechnik (MRT). KIT, 2013. (Zitiert auf Seiten 99 und 144).

A. Kembhavi, T. Yeh, and L. S. Davis. Why Did the Person Cross the Road (There)? Scene Understanding Using Probabilistic Logic Models and Common Sense Reasoning. In *Proceedings of the 11th European Conference on Computer Vision (ECCV 2010)*, volume 6312 of *LNCS*, pages 693–706. Springer, 2010. Doi: 10.1007/978-3-642-15552-9_50. (Zitiert auf Seite 26)

H. Kieritz, S. Becker, W. Hübner, and M. Arens. Online Multi-Person Tracking using Integral Channel Features. In *IEEE Conference on Advanced Video and Signal-based Surveillance (AVSS 2016)*, 2016. (Zitiert auf Seite 91)

K.M. Kitani, Y. Sato, and A. Sugimoto. Recovering the Basic Structure of Human Activities from Noisy Video-Based Symbol Strings. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(8):1621–1646, 2008. Doi: 10.1142/S0218001408006776. (Zitiert auf Seite 19)

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009. ISBN 0262013193. (Zitiert auf Seiten 16 und 27).

D.I. Kosmopoulos. Bayesian filter based behavior recognition in work ows allowing for user feedback. *CVIU*, 2012. (Zitiert auf Seite 17)

W. Krüger. *Situationsmodellierung in der Bildfolgenauswertung*. Dissertation, Fakultät fiir Informatik der Universität Karlsruhe (TH), 1991. (Zitiert auf Seiten 23, 38, 40 und 42).

T. Lan, Y. Wang, W. Yang, S.N. Robinovitch, and G. Mori. Discriminative Latent Models for Recognizing Contextual Group Activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1549–1562, 2011. Doi: 10.1109/TPAMI.2011.228. (Zitiert auf Seite 72)

G. Lavee, E. Rivlin, and M. Rudzsky. Understanding Video Events: A Survey of Methods for Automatic Interpretation of Semantic Occurrences in Video. *IEEE Transactions on System, Man, and Cybernetics, Part C: Applications and Reviews*, 39(5):489 –504, 2009. Doi: 10.1109/TSMCC.2009.2023380. (Zitiert auf Seite 13)

S. Laxman. *Discovering frequent episodes: fast algorithms, connections with HMMs and generalizations*. phdthesis, Indian Institute of Science. Bangalore, 2006. (Zitiert auf Seiten 125 und 127).

S. Laxman and P.S. Sastry. A survey of temporal data mining. *Sadhana. Academy Proceedings in Engineering Sciences*, 31(2):173–198, 2006. Doi: 10.1007/BF02719780. (Zitiert auf Seite 125)

J. Li and N.M. Allinson. A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10-12):1771–1787, 2008. Doi: 10.1016/j.neucom.2007.11.032. (Zitiert auf Seite 158)

H.-C. Liao, M.-Ho. Pan, H.-W. Hwang, M.-C. Chang, and Chen P.-C. An automatic calibration method based on feature point matching for the cooperation of wide-angle and pan-tilt-zoom cameras. *Information Technology And Control*, 40(1):41–47, 2011. Doi: 10.5755/j01.itc.40.1.191. (Zitiert auf Seite 157)

S. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. Doi: 10.1109/TIT.1982.1056489. (Zitiert auf Seite 73)

H. Mannila, H. Toivonen, and A.I. Verkamo. Discovery of Frequent Episodes in Event Sequences. *Data Mining and Knowledge Discovery*, 1(3): 259–289, 1997. Doi: 10.1023/A:1009748302351. (Zitiert auf Seite 125)

E. Michaelsen and J. Meidow. Stochastic reasoning for structural pattern recognition: An example from image-based UAV navigation. *Pattern Recognition*, 47(8):2732–2744, 2014. Doi: 10.1016/j.patcog.2014.02.009. (Zitiert auf Seite 56)

K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 (10):1615–1630, 2005. Doi: 10.1109/TPAMI.2005.188. (Zitiert auf Seiten 158 und 159).

M. Minas and G. Viehstaedt. DiaGen: A Generator for Diagram Editors Providing Direct Manipulation and Execution of Diagrams. In *Proceedings of the 11th IEEE International Symposium on Visual Languages*, pages 203–210. IEEE, 1995. Doi: 10.1109/VL.1995.520810. (Zitiert auf Seite 42)

D. Minnen, I. Essa, and T. Starner. Expectation Grammars: Leveraging High-Level Expectations for Activity Recognition. In *Proceeedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, volume 2, pages II–626. IEEE, 2003. Doi: 10.1109/CVPR.2003.1211525. (Zitiert auf Seite 22)

K.K. Mohanty and M.K. Gellaboina. A semi-automatic relative calibration of a fixed and PTZ camera pair for master-slave control. In *3rd European Workshop on Visual Information Processing (EUVIP 2011)*, pages 229 –234. IEEE, 2011. Doi: 10.1109/EuVIP.2011.6045523. (Zitiert auf Seite 157)

V.I. Morariu and L.S. Davis. Multi-agent event recognition in structured scenarios. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pages 3289–3296. IEEE, 2011. Doi: 10.1109/CVPR.2011.5995386. (Zitiert auf Seite 26)

D. Münch. *SGTyEditor User Manual v1.1*, 2015. (Zitiert auf Seite 42)

D. Münch, J. IJsselmuiden, M. Arens, and R. Stiefelhagen. High-level Situation Recognition Using Fuzzy Metric Temporal Logic, Case Studies in Surveillance and Smart Environments. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops 2011)*, pages 882–889. IEEE, 2011a. Doi: 10.1109/ICCVW.2011.6130345. (Zitiert auf Seiten ii, v und 71).

D. Münch, K. Jüngling, and M. Arens. Towards a Multi-purpose Monocular Vision-based High-Level Situation Awareness System. In *International Workshop on Behaviour Analysis and Video Understanding (ICVS 2011)*, pages 1–10, 2011b. (Zitiert auf Seite 104)

D. Münch, J. IJsselmuiden, A.-K. Grosselfinger, M. Arens, and R. Stiefelhagen. Rule-Based High-Level Situation Recognition from Incomplete Tracking Data. In *Proceedings of the 6th International Symposium (RuleML 2012)*, volume 7438 of *Programming and Software Engineering*, pages 317–324. Springer, 2012a. Doi: 10.1007/978-3-642-32689-9. (Zitiert auf Seiten ii, vi, 108, 109 und 110).

D. Münch, E. Michaelsen, and M. Arens. Supporting Fuzzy Metric Temporal Logic Based Situation Recognition by Mean Shift Clustering. In *Proceedings of the KI 2012: Advances in Artificial Intelligence*, volume 7526 of *LNCS*, pages 233–236. Springer, 2012b. Doi: 10.1007/978-3-642-33347-7_21. (Zitiert auf Seiten ii und vi).

D. Münch, S. Becker, A.-K. Grosselfinger, W. Hübner, and M. Arens. Towards Situational Awareness Systems based on Semi-Stationary Multi-Camera Components. In *Proceedings of the 8th Future Security Research Conference (FI 2013)*, pages 38–44. Fraunhofer Verlag, 2013a. (Zitiert auf Seite 158)

D. Münch, A.-K. Grosselfinger, W. Hübner, and M. Arens. Automatic Unconstrained Online Configuration of a Master-Slave Camera System. In

Proceeding of the 9th International Conference on Computer Vision Systems (ICVS 2013), volume 7963 of *LNCS*, pages 1–10. Springer, 2013b. Doi: 10.1007/978-3-642-39402-7_1. (Zitiert auf Seite iii)

D. Münch, W. Hübner, and M. Arens. Generalized Hough Transform-based time invariant action recognition with 3D pose information. In *Proceedings of SPIE Optics and Photonics for Counterterrorism, Crime Fighting and Defence*, volume 9253, pages 1–11. SPIE, 2014. Doi: 10.1117/12.2065805. (Zitiert auf Seite 91)

D. Münch, B. Hilsenbeck, H. Kieritz, S. Becker, A.-K. Grosselfinger, W. Hübner, and M. Arens. Detection of Infrastructure Manipulation with Knowledge-based Video Surveillance. In *Proceedings of SPIE Optics and Photonics for Counterterrorism, Crime Fighting and Defence 2016*, 2016. (Zitiert auf Seite 2)

H.-H. Nagel. Repräsentation von Wissen zur Auswertung von Bildern. In *Proceedings Angewandte Szenenanalyse 1. DAGM Symposium*, pages 3–21. Springer, 1979. ISBN:3-540-09665-5. (Zitiert auf Seiten 23 und 31).

H.-H. Nagel. Analyse und Interpretation von Bildfolgen. *Informatik-Spektrum*, 1985. (Zitiert auf Seiten 23 und 31).

H.-H. Nagel. From Image Sequences Towards Conceptual Descriptions. *Image and Vision Computing*, 6(2):59–74, 1988. Doi: 10.1016/0262-8856(88)90001-7. (Zitiert auf Seiten 4, 23, 31 und 37).

H.-H. Nagel. La representation de situations et leur reconnaissance a partir de sequences d'images. In *8. Congres Reconnaissance des Formes et Intelligence Artificielle. Vol.3 1991*, pages 1221–1229. Association Francaise pour la Cybernetique Economique et Technique (AFCET), 1991. (Zitiert auf Seiten 23 und 31).

H.-H. Nagel. Zur strukturierung eines bildfolgen-auswertungssystems. *Informatik - Forschung und Entwicklung*, 11:3–11, 1996. ISSN 0178-3564. (Zitiert auf Seite 24)

H.-H. Nagel. Image Sequence Evaluation: 30 Years and Still Going Strong. In *Proceedings of the 15th International Conference on Pattern Recognition 2000*, pages 149–158, 2000. Doi: 10.1109/ICPR.2000.905294. (Zitiert auf Seiten 6, 7, 23 und 31).

H.-H. Nagel. Steps toward a Cognitive Vision System. *AI Magazine*, 25(2): 31–50, 2004. Doi: 10.1609/aimag.v25i2.1759. (Zitiert auf Seiten 6, 7, 23 und 31).

H.-H. Nagel. *Cognitive Vision Systems. On Sampling the Spectrum of Approaches Toward Cognitive Vision Systems*, volume 3948 of *LNCS*, pages 315–319. Springer, 2006. Doi: 10.1007/11414353_18. (Zitiert auf Seiten 23 und 31).

H.-H. Nagel. Vision, Logic, and Language - Toward Analyzable Encompassing Systems. In *Proceedings of the KI 2010: Advances in Artificial Intelligence*, volume 6359 of *LNCS*, pages 1–22. Springer, 2010. Doi: 10.1007/978-3-642-16111-7_1. (Zitiert auf Seiten 6, 23 und 31).

H.-H. Nagel, T. Beth, and J. Calmet. *Forschung und Lehre am IAKS: ein Tätigkeitsbericht anläßlich des zehnjährigen Bestehens 1985-1995*. Institut für Algorithmen und Kognitive Systeme, KIT, 1995. (Zitiert auf Seiten 23 und 31).

L. Naish. Higher-order logic programming in Prolog. Technical report, Workshop on Multi-Paradigm Logic Programming (JICSLP 1996), 1996. (Zitiert auf Seite 82)

P. Natarajan and R. Nevatia. Coupled Hidden Semi Markov Models for Activity Recognition. In *Proceedings of the IEEE Workshop on Motion and Video Computing (WMVC 2007)*, page 10. IEEE, 2007. Doi: 10.1109/WMVC.2007.12. (Zitiert auf Seite 15)

B. Neumann. Bayesian compositional hierarchies – a probabilistic structure for scene interpretation. Technical Report FBI-HH-B-282/08, Department Informatik, Universität Hamburg, 2008. (Zitiert auf Seite 26)

R. Nevatia, T. Zhao, and S. Hongeng. Hierarchical Language-based Representation of Events in Video Streams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPR 2003)*, page 39. IEEE, 2003. Doi: 10.1109/CVPRW.2003.10038. (Zitiert auf Seite 22)

N.T. Nguyen, D.Q. Phung, S. Venkatesh, and H. Bui. Learning and Detecting Activities from Movement Trajectories Using the Hierarchical Hidden Markov Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05)*, pages 955–960. IEEE, 2005. Doi: 10.1109/CVPR.2005.203. (Zitiert auf Seite 17)

S. Oh, A. Hoogs, A. Perera, and N. Cuntoor. A Large-Scale Benchmark Dataset for Event Recognition in Surveillance Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pages 3153–3160. IEEE, 2011. Doi: 10.1109/CVPR.2011.5995586. (Zitiert auf Seiten 95, 99, 100, 102, 103 und 104).

N. Oliver. Layered Representations for Human Activity Recognition. In *Proccedings of the Fourth IEEE International Conference on Multimodal Interfaces 2002*. IEEE, 2002. Doi: 10.1109/ICMI.2002.1166960. (Zitiert auf Seite 16)

D.C. Oppen. A $2^{2^{2^{pn}}}$ upper bound on the complexity of Presburger Arithmetic. *Journal of Computer and System Sciences*, 16(3):323–332, 1978. Doi: 10.1016/0022-0000(78)90021-1. (Zitiert auf Seite 35)

L. Patino, H. Benhadda, E. Corvee, F. Brémond, and M. Thonnat. Extraction of activity patterns on large video recordings. *IET Computer Vision*, 2(2): 108–128, 2008. Doi: 10.1049/iet-cvi:20070062. (Zitiert auf Seite 22)

D. Patnaik, P.S. Sastry, and K.P. Unnikrishnan. Inferring Neuronal Network Connectivity from Spike Data: A Temporal Data Mining Approach. *Scientific Programming*, 16(1):49–77, 2008. Doi: 10.3233/SPR-2008-0242. (Zitiert auf Seite 127)

J. Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3: 96–146, 2009. Doi: 10.1214/09-SS057. (Zitiert auf Seite 139)

J. Pearl, M. Glymour, and N.P. Jewell. *Causal Inference in Statistics: A Primer*. Wiley, 2016. ISBN: 978-1-119-18684-7. (Zitiert auf Seite 139)

C. Pinhanez and A. Bobick. Human Action Detection Using PNF Propagation of Temporal Constraints. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1998)*, page 898, 1998. ISSN 1063-6919. Doi: 10.1109/CVPR.1998.698711. (Zitiert auf Seiten 21 und 22).

H. Possegger, M. Rüther, S. Sternig, T. Mauthner, M. Klopschitz, P. M. Roth, and H. Bischof. Unsupervised Calibration of Camera Networks and Virtual PTZ Cameras. In *Proceedings of the 17th Computer Vision Winter Workshop (CVWW 2012)*, pages 1–8, 2012. (Zitiert auf Seiten 141 und 143).

C.A. Ramirez Fandiño. Time Invariant Action Recognition with 3D Pose Information Based on the Generalized Hough Transformation. Diplomarbeit, Institute for Robotics and Process Control (IRP). TU Braunschweig, 2013. (Zitiert auf Seite 92)

C.A. Ramirez Fandiño and D. Münch. Time Invariant Action Recognition with 3D Pose Information Based on Hough Forests. Technical report, Fraunhofer IOSB, 2014. (Zitiert auf Seite 92)

A. Resch, R. Pfeil, M. Wegener, and A. Stelzer. Review of the LPM local positioning measurement system. In *International Conference on Localization and GNSS (ICL-GNSS 2012)*, pages 1–5. IEEE, 2012. Doi: 10.1109/ICL-GNSS.2012.6253104. (Zitiert auf Seite 151)

M. Richardson and P. Domingos. Markov Logic Networks. *Machine Learning*, 62(1-2):107–136, 2006. ISSN 0885-6125. Doi: 10.1007/s10994-006-5833-1. (Zitiert auf Seite 25)

B. Rosario, N. Oliver, and A. Pentland. A Synthetic Agent System for Bayesian Modeling of Human Interactions. In *Proceedings of the third annual*

conference on Autonomous Agents (AGENTS 1999), pages 342–343. ACM, 1999. Doi: 10.1145/301136.301225. (Zitiert auf Seiten 16 und 17).

M. Ryoo and J. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, volume 2, 2006. Doi: 10.1109/CVPR.2006.242. (Zitiert auf Seite 22)

M. Ryoo and J. Aggarwal. Semantic Representation and Recognition of Continued and Recursive Human Activities. *International Journal of Computer Vision*, 82(1):1–24, 2009. Doi: 10.1007/s11263-008-0181-1. (Zitiert auf Seiten 22 und 72).

M. Ryoo and J. Aggarwal. Stochastic Representation and Recognition of High-Level Group Activities. *International Journal of Computer Vision*, 93(2):183–200, 2011. Doi: 10.1007/s11263-010-0355-. (Zitiert auf Seite 72)

K. Schäfer. *Unscharfe zeitlogische Modellierung von Situationen und Handlungen in Bildfolgenauswertung und Robotik*. Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Dissertationen zur Künstlichen Intelligenz 135, Akademische Verlagsgesellschaft Aka GmbH, 1996. (Zitiert auf Seiten 33, 36, 47, 49 und 53).

K. Schäfer and C. Brzoska. "F-Limette" fuzzy logic programming integrating metric temporal extensions. *Journal of Symbolic Computation. Special Issue: executable temporal logics*, 22(5-6):725–727, 1996. (Zitiert auf Seite 23)

S.N. Sinha and M. Pollefeys. Pan-Tilt-Zoom Camera Calibration and High-Resolution Mosaic Generation. *Computer Vision and Image Understanding*, 103(3):17 –183, 2006. Doi: 10.1016/j.cviu.2006.06.002. (Zitiert auf Seite 157)

J. M. Siskind. Grounding the Lexical Semantics of Verbs in Visual Perception using Force Dynamics and Event Logic. *Journal Of Artificial Intelligence*, 15(1):31–90, 2001. Doi: 10.1613/jair.790. (Zitiert auf Seite 21)

A. Skarlatidis, G. Paliouras, G.A. Vouros, and A. Artikis. Probabilistic Event Calculus based on Markov Logic Networks. In *Proceedings of the 5th International Symposium (RuleML 2011)*, volume 7018 of *LNCS*, pages 155–170. Springer, 2011. Doi: 10.1007/978-3-642-24908-2_19. (Zitiert auf Seite 26)

A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000. Doi: 10.1109/34.895972. (Zitiert auf Seite 8)

G. Szwoch, P. Dalka, A. Ciarkowski, P. Szczuko, and A. Czyzewski. Visual object tracking system employing fixed and PTZ cameras. *Intelligent Decision Technologies*, 5(2):177–188, 2011. Doi: 10.3233/IDT-2011-0105. (Zitiert auf Seiten 156 und 157).

Y. Tian, L. Brown, A. Hampapur, M. Lu, A. Senior, and C. Shu. IBM smart surveillance system (S3): event based video surveillance system with an open and extensible framework. *Machine Vision and Applications*, 19(5-6):315–327, 2008. Doi: 10.1007/s00138-008-0153-z. (Zitiert auf Seite 156)

S. Tran and L. Davis. Event Modeling and Recognition Using Markov Logic Networks. In *Proceedings of the 10th European Conference on Computer Vision (ECCV 2008) Part II*, volume 5303 of *LNCS*, pages 610–623. Springer, 2008. Doi: 10.1007/978-3-540-88688-4_45. (Zitiert auf Seite 26)

R. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987. Doi: 10.1109/JRA.1987.1087109. (Zitiert auf Seite 157)

E. Tulving and F.I.M. Craik. *The Oxford Handbook of Memory*. Oxford University Press, 2005. (Zitiert auf Seite 8)

P. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea. Machine Recognition of Human Activities: A Survey. *IEEE Transactions on Circuits and*

Systems for Video Technology, 18(11):1473–1488, 2008. Doi: 10.1109/TCSVT.2008.2005594. (Zitiert auf Seite 13)

T.L.M. van Kasteren, G. Englebienne, and B.J.A. Kröse. *Human Activity Recognition from Wireless Sensor Network Data: Benchmark and Software*, pages 165–186. Atlantis Press, 2011. Doi: 10.2991/978-94-91216-05-3_8. (Zitiert auf Seiten 118, 119 und 120).

S. Venugopalan, M. Rohrbach, J. Donahue, R.J. Mooney, T. Darrell, and K. Saenko. Sequence to Sequence - Video to Text. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2015)*, pages 4534–4542. IEEE, 2015a. Doi: 10.1109/ICCV.2015.515. (Zitiert auf Seite 28)

S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R.J. Mooney, and K. Saenko. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2015)*, pages 1494–1504. NAACL, 2015b. (Zitiert auf Seiten 27 und 28).

S. Vishwakarma and A. Agrawal. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29 (10):983–1009, 2013. Doi: 10.1007/s00371-012-0752-6. (Zitiert auf Seite 13)

A. Voulodimos, D. Kosmopoulos, G. Vasileiou, E. Sardis, A. Doulamis, V. Anagnostopoulos, C. Lalos, and T. Varvarigou. A dataset for workflow recognition in industrial scenes. In *18th IEEE International Conference on Image Processing (ICIP 2011)*, pages 3249–3252. IEEE, 2011. Doi: 10.1109/ICIP.2011.6116362. (Zitiert auf Seite 2)

V.-T. Vu, F. Bremond, and M. Thonnat. Automatic Video Interpretation: A Novel Algorithm for Temporal Scenario Recognition. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 1295–1300. Morgan Kaufmann, 2003. (Zitiert auf Seiten 21, 22 und 23).

J. Weisbrod. *Unscharfes Schließen.* Dissertation, Karlsruher Institut für Technologie (KIT), Fakultät für Informatik, Institut für Programmstrukturen und Datenorganisation (IPD), 1996. Dissertationen zur Künstlichen Intelligenz (DISKI). (Zitiert auf Seiten 50 und 51).

L. Wittgenstein. *Philosophische Untersuchungen.* Suhrkamp, 2003. ISBN: 978-3-518-22372-7. (Zitiert auf Seite 50)

C. Wojek, K. Nickel, and R. Stiefelhagen. Activity recognition and room-level tracking in an office environment. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 25–30. IEEE, 2006. Doi: 10.1109/MFI.2006.265608. (Zitiert auf Seite 16)

Z. Wu and R. Radke. Keeping a Pan-Tilt-Zoom Camera Calibrated. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1994–2007, 2012. Doi: 10.1109/TPAMI.2012.250. (Zitiert auf Seite 157)

K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R.S. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, pages 2048–2057. CoRR, 2015. (Zitiert auf Seite 28)

L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing Videos by Exploiting Temporal Structure. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2015)*, pages 4507–4515. IEEE, 2015. Doi: 10.1109/iccv.2015.512. (Zitiert auf Seite 28)

J. Ye, S. Dobson, and S. McKeever. Situation Identification Techniques in Pervasive Computing: A Review. *Pervasive and Mobile Computing*, 8(1): 36–66, 2012. ISSN 1574-1192. Doi: 10.1016/j.pmcj.2011.01.004. (Zitiert auf Seite 13)

R. D. X. Yi, J. Gao, and M. Antolovich. Novel methods for high–resolution facial image capture using calibrated PTZ and static cameras. In *IEEE In-*

ternational Conference on Multimedia and Expo 2008, pages 45–48. IEEE, 2008. Doi: 10.1109/ICME.2008.4607367. (Zitiert auf Seite 156)

S. Zaidenberg, B. Boulay, C. Garate, D. P. Chau, E. Corvee, and F. Bremond. Group interaction and group tracking for video-surveillance in underground railway stations. In *International Workshop on Behaviour Analysis and Video Understanding (ICVS 2011)*, pages 1–10, 2011. (Zitiert auf Seite 72)

T. Zepf. From the Articulated 3D Pose of the Human Body to Basic Action Recognition. Diplomarbeit, Fakultät für Informatik. Institut für Anthropomatik (IFA). KIT, 2012. (Zitiert auf Seite 91)

D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Modeling Individual and Group Actions in Meetings with Layered HMMs. *IEEE Transactions on Multimedia*, 8(3):509–520, 2006. Doi: 10.1109/TMM.2006.870735. (Zitiert auf Seite 17)

S. Zhang, R. Benenson, and B. Schiele. Filtered Channel Features for Pedestrian Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, pages 1751–1760. IEEE, 2015. (Zitiert auf Seite 90)

X. Zhou, R. T. Collins, T. Kanade, and P. Metes. A Master–Slave System to Acquire Biometric Imagery of Humans at Distance. In *First ACM SIGMM international workshop on Video surveillance (IWVS 2003)*, pages 113–120. ACM, 2003. Doi: 10.1145/982452.982467. (Zitiert auf Seite 157)

H.J. Zimmermann. *Fuzzy Set Theory - and Its Applications*. Springer, 4 edition, 2001. ISBN 9780792374350. Doi: 10.1007/978-94-010-0646-0. (Zitiert auf Seiten 50 und 51).

Eigene und betreute Arbeiten

Eigene Arbeiten - Situationsanalyse

[1] D. Münch, S. Becker, H. Kieritz, W. Hübner, M. Arens. **Video-based log generation for security systems in indoor surveillance scenarios.** Proc. of the 11th Future Security Research Conference, Berlin, Germany, 2016.

[2] D. Münch, B. Hilsenbeck, H. Kieritz, S. Becker, A.-K. Grosselfinger, W. Hübner, M. Arens. **Detection of infrastructure manipulation with knowledge-based video surveillance.** Proc. of the Security and Defence Conference SPIE 2016, Edinburgh, UK, 2016.

[3] D. Münch, S. Becker, H. Kieritz, W. Hübner, M. Arens. **Knowledge-based Situational Analysis of Unusial Events in Public Places.** Proc. of the 10th Future Security Research Conference, Berlin, Germany, 2015.

[4] D. Münch, A.-K. Grosselfinger, H. Kieritz, W. Hübner, M. Arens. **Architecture for and Evaluation of Situational Analysis in the Real World.** Proc. of the 9th Future Security Research Conference, Berlin, Germany, 2014.

[5] J. IJsselmuiden, D. Münch, A.-K. Grosselfinger, M. Arens, R. Stiefelhagen. **Automatic understanding of group behavior using fuzzy temporal logic.** Journal of Ambient Intelligence and Smart Environments (JAISE). 2014.

[6] D. Münch, S. Becker, A.-K. Grosselfinger, W. Hübner, M. Arens. **Towards Situational Awareness Systems based on Semi-Stationary Multi-Camera Components.** Proc. of the 8th Future Security Research Conference, Berlin, Germany, 2013.

[7] D. Münch, A.-K. Grosselfinger, W. Hübner, M. Arens. **Automatic Unconstrained Online Configuration of a Master-Slave Camera System.** Proc. of the 9th International Conference on Computer Vision Systems, St. Petersburg, Russia, 2013. **(Best Paper Award)**

[8] J. IJsselmuiden, A.-K. Grosselfinger, D. Münch, M. Arens, R. Stiefelhagen. **Automatic Behavior Understanding in Crisis Response Control Rooms.** Proc. of International Joint Conference on Ambient Intelligence, Pisa, Italy, 2012.

[9] D. Münch, E. Michaelsen, M. Arens. **Supporting Fuzzy Metric Temporal Logic Based Situation Recognition by Mean Shift Clustering.** Proc. of the 35rd Annual German Conference on Advances in Artificial Intelligence, Saarbrücken, Germany, 2012.

[10] D. Münch, J. IJsselmuiden, A.-K. Grosselfinger, M. Arens, R. Stiefelhagen. **Rule-Based High-Level Situation Recognition from Incomplete**
Tracking Data. Proc. of the 6th International Symposium on Rules: Research Based and Industry Focused, Montpellier, France, 2012.

[11] D. Münch, S. Becker, W. Hübner, M. Arens. **Towards a Real-time Situational Awareness System for Surveillance Applications in Unconstrained Environments.** Proc. of the 7th Future Security Research Conference, Bonn, Germany, 2012.

[12] D. Münch, J. IJsselmuiden, M. Arens, R. Stiefelhagen. **High-Level Situation Recognition Using Fuzzy Metric Temporal Logic, Case Studies in Surveillance and Smart Environments.** Proc. of the

International Conference on Computer Vision Workshops, IEEE Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams, Barcelona, Spain, 2011.

[13] D. Münch, K. Jüngling, M. Arens. **Towards a Multi-purpose Monocular Vision-based High-Level Situation Awareness System.** Proc. of the International Conference on Computer Vision Systems Workshops, International Workshop on Behaviour Analysis and Video Understanding, Sophia Antipolis, France, 2011.

[14] D. Münch, K. Jüngling, M. Arens. **Video Analysis for Situation and Threat Recognition.** Proc. of the 6th Future Security Research Conference, Berlin, Germany, 2011.

Eigene Arbeiten - Aktionserkennung

[15] B. Hilsenbeck, D. Münch, A.-K. Grosselfinger, W. Hübner, M. Arens. **Action Recognition in the Longwave Infrared and the Visible Spectrum using Hough Forests.** Proc. of The IEEE International Symposium on Multimedia, San Jose, California, USA, 2016.

[16] B. Hilsenbeck, D. Münch, H. Kieritz, W. Hübner, M. Arens. **Hierarchical Hough Forests for View-Independent Action Recognition.** Proc. of 23rd International Conference on Pattern Recognition, Cancún, México, 2016.

[17] C. Ramirez, D. Münch. **Time invariant action recognition with 3D pose information based on Hough forests.** Technical Report, Fraunhofer IOSB, 2014.

[18] D. Münch, W. Hübner, M. Arens. **Generalized Hough Transform based Time Invariant Action Recognition with 3D Pose Information.** Proc. of the Security and Defence Conference SPIE 2014, Amsterdam, Netherlands, 2014.

Eigene Arbeiten - Weitere

[19] E. Michaelsen D. Münch, M. Arens. **Searching Remotly Sensed Images for Meaningful Nested Gestalten.** ISPRS Archives, Prague,Czech Republic, 2016.

[20] S. Becker, D. Münch, H. Kieritz, W. Hübner, M. Arens. **Detecting abandoned objects using interacting multiple models.** Proc. of the Security and Defence Conference SPIE 2015, Toulouse, France, 2015.

[21] E. Michaelsen, D. Münch, M. Arens. **Recognition of Symmetry Structure by Use of Gestalt Algebra.** Proc. of Conference on Computer Vision and Pattern Recognition Workshops 2013, Portland, USA, 2013.

[22] A.-K. Grosselfinger, D. Münch, W. Hübner, M. Arens. **Feature-based Automatic Configuration of Semi-Stationary Multi-Camera Components.** Proc. of the Security and Defence Conference SPIE 2013, Dresden, Germany, 2013.

[23] P. Azad, D. Münch, T. Asfour, R. Dillmann. **6-DoF Model-based Tracking of Arbitrarily Shaped 3D Objects.** Proc. of the IEEE International Conference on Robotics and Automation, Shanghai, China, 2011.

[24] D. Münch. **6-DoF-particle filter-based tracking of arbitrarily shaped objects.** Diplomarbeit Fakultät für Informatik. Institut für Anthropomatik (IFA). Karlsruhe 2010.

[25] D. Münch. **Parallel programming with CUDA – Architecture, Analysis, Application.** Studienarbeit Fakultät für Informatik. Institut für Programmstrukturen und Datenorganisation (IPD). Karlsruhe 2009.

Betreute Studentische Arbeiten

[1] A. Katumba. **Rule-based Situation Recognition in Video Streams.** Masterarbeit. Fakultät für Machinenbau. Institut für Mess- und Regelungstechnik (MRT). Karlsruhe 2013.

[2] E. Dursun. **Gaze Control Strategies for the Observation of Dynamic Scenes Using a Master-Slave Camera System.** Studienarbeit. Fakultät für Informatik. Institut für Anthropomatik (IFA). Karlsruhe 2013.

[3] C. Ramirez. **Time Invariant Action Recognition with 3D Pose Information based on the Generalized Hough Transformation.** Masterarbeit. Department Informatik. Institut für Robotik und Prozessinformatik (IRP). Braunschweig 2013.

[4] A.-K. Grosselfinger. **Automatische Konfiguration eines Mehrkamerasystems zur Verfolgung von Objekten in unstrukturierten Umgebungen.** [in German] Diplomarbeit. Fakultät für Informatik. Institut für Anthropomatik (IFA). Karlsruhe 2012.

[5] S. Bauer. **Comparing Ontologies and Situatin Graph Trees in Cognitive Vision Systems.** Diplomarbeit. Fakultät für Informatik. Institut für Anthropomatik (IFA). Karlsruhe 2012.

[6] T. Zepf. **From the Articulated 3D Pose of the Human Body to Basic Action Recognition.** Diplomarbeit. Fakultät für Informatik. Institut für Anthropomatik (IFA). Karlsruhe 2012.

[7] A.-K. Grosselfinger. **Zur Behandlung von Unsicherheit und unvollständiger Information bei der videobasierten Situationserkennung.** [in German] Studienarbeit. Fakultät für Informatik. Institut für Anthropomatik (IFA). Karlsruhe 2011.

Karlsruher Schriftenreihe zur Anthropomatik (ISSN 1863-6489)

Herausgeber: Prof. Dr.-Ing. Jürgen Beyerer

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar oder als Druckausgabe bestellbar.

Band 1 Jürgen Geisler
Leistung des Menschen am Bildschirmarbeitsplatz. 2006
ISBN 3-86644-070-7

Band 2 Elisabeth Peinsipp-Byma
Leistungserhöhung durch Assistenz in interaktiven Systemen zur Szenenanalyse. 2007
ISBN 978-3-86644-149-1

Band 3 Jürgen Geisler, Jürgen Beyerer (Hrsg.)
Mensch-Maschine-Systeme. 2010
ISBN 978-3-86644-457-7

Band 4 Jürgen Beyerer, Marco Huber (Hrsg.)
Proceedings of the 2009 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2010
ISBN 978-3-86644-469-0

Band 5 Thomas Usländer
Service-oriented design of environmental information systems. 2010
ISBN 978-3-86644-499-7

Band 6 Giulio Milighetti
Multisensorielle diskret-kontinuierliche Überwachung und Regelung humanoider Roboter. 2010
ISBN 978-3-86644-568-0

Band 7 Jürgen Beyerer, Marco Huber (Hrsg.)
Proceedings of the 2010 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2011
ISBN 978-3-86644-609-0

Band 8 Eduardo Monari
Dynamische Sensorselektion zur auftragsorientierten Objektverfolgung in Kameranetzwerken. 2011
ISBN 978-3-86644-729-5

Band 9 Thomas Bader
Multimodale Interaktion in Multi-Display-Umgebungen. 2011
ISBN 3-86644-760-8

Band 10 Christian Frese
Planung kooperativer Fahrmanöver für kognitive Automobile. 2012
ISBN 978-3-86644-798-1

Band 11 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2011 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2012
ISBN 978-3-86644-855-1

Band 12 Miriam Schleipen
Adaptivität und Interoperabilität von Manufacturing Execution Systemen (MES). 2013
ISBN 978-3-86644-955-8

Band 13 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2012 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2013
ISBN 978-3-86644-988-6

Band 14 Hauke-Hendrik Vagts
Privatheit und Datenschutz in der intelligenten Überwachung: Ein datenschutzgewährendes System, entworfen nach dem „Privacy by Design" Prinzip. 2013
ISBN 978-3-7315-0041-4

Band 15 Christian Kühnert
Data-driven Methods for Fault Localization in Process Technology. 2013
ISBN 978-3-7315-0098-8

Band 16 Alexander Bauer
Probabilistische Szenenmodelle für die Luftbildauswertung. 2014
ISBN 978-3-7315-0167-1

Band 17 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2013 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2014
ISBN 978-3-7315-0212-8

Band 18 Michael Teutsch
Moving Object Detection and Segmentation for Remote Aerial Video Surveillance. 2015
ISBN 978-3-7315-0320-0

Band 19 Marco Huber
Nonlinear Gaussian Filtering:
Theory, Algorithms, and Applications. 2015
ISBN 978-3-7315-0338-5

Band 20 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2014
ISBN 978-3-7315-0401-6

Band 21 Todor Dimitrov
Permanente Optimierung dynamischer Probleme der Fertigungssteuerung unter Einbeziehung von Benutzerinteraktionen. 2015
ISBN 978-3-7315-0426-9

Band 22 Benjamin Kühn
Interessengetriebene audiovisuelle Szenenexploration. 2016
ISBN 978-3-7315-0457-3

Band 23 Yvonne Fischer
Wissensbasierte probabilistische Modellierung für die Situationsanalyse am Beispiel der maritimen Überwachung. 2016
ISBN 978-3-7315-0460-3

Band 24 Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2015 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2016
ISBN 978-3-7315-0519-8

Band 25 Pascal Birnstill
Privacy-Respecting Smart Video Surveillance Based on Usage Control Enforcement. 2016
ISBN 978-3-7315-0538-9

Band 26 Philipp Woock
Umgebungskartenschätzung aus Sidescan-Sonardaten für ein autonomes Unterwasserfahrzeug. 2016
ISBN 978-3-7315-0541-9

Band 27 Janko Petereit
Adaptive State × Time Lattices: A Contribution to Mobile Robot Motion Planning in Unstructured Dynamic Environments. 2017
ISBN 978-3-7315-0580-8

Band 28 Erik Ludwig Krempel
Steigerung der Akzeptanz von intelligenter Videoüberwachung in öffentlichen Räumen. 2017
ISBN 978-3-7315-0598-3

Band 29 Jürgen Moßgraber
Ein Rahmenwerk für die Architektur von Frühwarnsystemen. 2017
ISBN 978-3-7315-0638-6

Band 30 Andrey Belkin
World Modeling for Intelligent Autonomous Systems. 2017
ISBN 978-3-7315-0641-6

Band 31 Chettapong Janya-Anurak
Framework for Analysis and Identification of Nonlinear Distributed Parameter Systems using Bayesian Uncertainty Quantification based on Generalized Polynomial Chaos. 2017
ISBN 978-3-7315-0642-3

Band 32 David Münch
Begriffliche Situationsanalyse aus Videodaten bei unvollständiger und fehlerhafter Information. 2017
ISBN 978-3-7315-0644-7

Lehrstuhl für Interaktive Echtzeitsysteme
Karlsruher Institut für Technologie

Fraunhofer-Institut für Optronik, Systemtechnik
und Bildauswertung IOSB Karlsruhe

Aktuelle Systeme der Videoüberwachung generieren eine große Menge an Bildfolgen, die eine manuelle inhaltliche Auswertung annähernd unmöglich und unwirtschaftlich macht. Diese Arbeit beschäftigt sich mit der automatischen Erkennung komplexer Situationen in Bildfolgen im Videoüberwachungskontext. Konzeptionell liegt dem umgesetzten Ansatz ein kognitives System zugrunde. Dies transformiert u.a. Bildsensordaten in semantische Beschreibungen, die dann zur Schlussfolgerung über das Auftreten von erwarteten Situationen der durch die Bildsensoren beobachteten Szene verwendet werden können. Bei der Behandlung von Daten aus natürlichen Umgebungen ergeben sich verschiedene Schwierigkeiten, wie z.B. Mess-, Quantisierungs- und Verfahrensfehler. Diese sind unvermeidbar. Daher muss die Situationserkennung explizit mit diesen Effekten umgehen können. Diese Arbeit erweitert dazu den verwendeten Formalismus um die Behandlung von Unschärfe, fehlender Information und Komplexität, zeigt die Robustheit der Situationserkennung bei natürlichen Szenarien und stellt die generische Anwendbarkeit auch über Diskursbereichsgrenzen hinaus heraus.

ISSN 1863-6489
ISBN 978-3-7315-0644-7

Gedruckt auf FSC-zertifiziertem Papier