

Probabilistische Szenenerkennung durch hierarchische Constellation Models über räumliche Relationen aus demonstrierten Objekttrajektorien

Masterarbeit
von

Joachim Gehrung

An der Fakultät für Informatik
Institut für Anthropomatik
Lehrstuhl Prof. Dr.-Ing. R. Dillmann

Erstgutachter:	Prof. Dr.-Ing. R. Dillmann
Zweitgutachter:	Prof. Dr.-Ing. J. Beyerer
Betreuernder Mitarbeiter:	Dipl.-Inform. Pascal Meißner

Hiermit erkläre ich an Eides statt, dass ich die von mir vorgelegte Arbeit selbstständig verfasst habe, dass ich die verwendeten Quellen, Internet-Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen – die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Karlsruhe, den 8. September 2014

Joachim Gehrung

Inhaltsverzeichnis

Abkürzungsverzeichnis	vii
Abbildungsverzeichnis	viii
1 Einführung	1
2 Motivation und Problemstellung	2
2.1 Motivation	2
2.2 Fokus der Arbeit	3
3 Grundlagen	5
3.1 Wahrscheinlichkeitsrechnung	5
3.1.1 Zufallsvariablen	5
3.1.2 Verbundverteilung	6
3.1.3 Marginalisierung	6
3.1.4 Gesetz der bedingten Wahrscheinlichkeiten	7
3.2 Wahrscheinlichkeitsverteilungen	7
3.2.1 Bernoulli-Verteilung	8
3.2.2 Multinomialverteilung	8
3.2.3 Gauss-Verteilung	8
3.2.4 Gauss-Mischverteilung	9
3.3 Probabilistische Modellbildung	9
3.3.1 Bayes-Theorem	9
3.3.2 Modell, Lernen, Inferenz	10
3.3.3 Diskriminative und generative Modelle	10
3.4 Lernen der Modellparameter	11
3.4.1 Maximum Likelihood	11
3.4.2 EM-Algorithmus	11
3.4.3 Lernen von Gauss-Mischverteilungen	13
3.4.4 Wahl der Modellkomplexität	15
4 Stand der Forschung	16
4.1 Objekterkennung	16
4.1.1 Modellbasierte Ansätze	16

4.1.2	Globale ansichtenbasierte Ansätze	17
4.1.3	Lokale ansichtenbasierte Ansätze	18
4.2	Szenenerkennung	19
4.2.1	Parallelen zur Objekterkennung	19
4.2.2	Inferenz von Szenen aus Bilddaten	20
4.2.3	Inferenz von Szenen aus Objekten	21
4.3	Teile- und Strukturmodelle	23
4.3.1	Implicit Shape Model	23
4.3.2	Constellation Model	23
4.4	Probabilistische Posenmodellierung	26
4.4.1	Repräsentation von Posen	26
4.4.2	Methoden zur Repräsentation von Unsicherheit	26
5	Konzeption des Szenenmodells	28
5.1	Motivation des Ansatzes	28
5.2	Abgrenzung zu anderen Arbeiten	31
5.3	Modellschnittstelle und -parameter	32
5.4	Szenenmodell	34
5.4.1	Herleitung des Szenenmodells	34
5.4.2	Rückweisungsklasse	37
5.5	Object Constellation Model	37
5.5.1	Unterschiede zum Constellation Model	37
5.5.2	Herleitung des OCM	39
5.5.3	Object Appearance	43
5.5.4	Scene Shape	45
5.5.5	Object Existence	48
5.5.6	Hintergrundmodell	49
5.6	Zusammenfassung der Modellstruktur	50
6	Lernen und Inferenz	52
6.1	Lernen der Modellparameter	52
6.1.1	Anforderungen	52
6.1.2	Bestimmung relevanter Relationen	54
6.1.3	Parameter der Szenenerkennung	57
6.1.4	Parameter des Object Constellation Model	58
6.1.5	Sample Relaxation	61
6.2	Inferenz der Szene	61
6.2.1	Inferenz im Szenenmodell	62
6.2.2	Inferenz im OCM	63
7	Implementierung	64
7.1	Das bestehende System	64
7.2	Das entwickelte System	65
7.2.1	Posenmodell	66

7.2.2	Probabilistisches Szenenmodell	67
7.2.2.1	Lerner	67
7.2.2.2	Modell	69
7.2.2.3	Inferenz	71
7.2.3	Relationsgraph-Generator	73
7.2.4	Szenengraph-Generator	75
7.2.5	Visualisierung	76
8	Evaluation	80
8.1	Verifikation der Hintergrundterme	80
8.2	Relative Position und Orientierung	81
8.3	Clutter und fehlende Objekte	85
8.4	Robustheit gegenüber Fehldetektionen	89
8.5	Auswirkungen weicher Entscheidungsgrenzen	90
8.6	Laufzeit und Modellwachstum	93
8.6.1	Einfache und komplexe Szene	95
8.6.2	Laufzeit des PSM	96
8.6.3	Modellwachstum	97
8.7	Fazit	98
9	Zusammenfassung und Ausblick	100
Literaturverzeichnis		102

Abkürzungsverzeichnis

BIC Bayesian Information Criterion

CM Constellation Model

CRF Conditional Random Fields

EM Expectation Maximization

ISM Implicit Shape Model

ML Maximum Likelihood

MRF Markov Random Fields

OCM Object Constellation Model

PdV Programmieren durch Vormachen

PSM Probabilistic Scene Model

Abbildungsverzeichnis

3.1	Darstellung diskreter Wahrscheinlichkeiten	6
3.2	Verbundverteilungen	6
3.3	Bernoulli- und Multinomialverteilung	7
3.4	Gaussverteilung und Gauss-Mischverteilung	9
3.5	Parameterlernern mittels Maximum Likelihood	11
3.6	Funktionsweise des EM-Algorithmus	12
3.7	Beispiel zum EM-Algorithmus	14
4.1	Objektmodell des RAPiD-Algorithmus	17
4.2	Gesichtserkennung mit dem Viola-Jones-Verfahren	18
4.3	Semantische Bedeutung von Relationen	20
4.4	Biologisch motivierte Szenenerkennung	21
4.5	Relationen in der Szenenerkennung	22
4.6	Fußgängererkennung mit dem ISM	24
5.1	Relevante Relationen	29
5.2	Beispiele zum Relationsgraphen	30
5.3	Konzept des Szenenobjekts	34
5.4	Grafische Darstellung des Szenenmodells	36
5.5	Beispiel eines Relationsgraphen	38
5.6	Grafische Darstellung des Object Constellation Model	39
5.7	Abbildung von Szenenobjekten auf Slots	40
5.8	Bewertung einzelner Hypothesen	42
5.9	Aussage der Object Appearance	44
5.10	Aussage der Scene Shape	45
5.11	Aussage der Object Existence	48
6.1	Warum wird das Objekt nicht mehr wahrgenommen?	53
6.2	Ursachen für das Verschwinden des Objekts	53
6.3	Relationen in einer Szene	55
6.4	Konstruktion des Relationsgraph	56
6.5	Anpassungs des Relationsbaums	57
6.6	Lernen einer relativen Position	60
6.7	Auswirkungen von Sample Relaxation	61

6.8	Visualisierung der Szenenerkennung	62
7.1	Abhangigkeiten zwischen den Paketen	66
7.2	Interaktionsdiagramm entwickelter und vorhandenen Softwarepakete	67
7.3	Klassendiagramm des Lerners	68
7.4	Szenenmodell einer Fruhstücksszene	70
7.5	Klassendiagramm der Inferenz	72
7.6	Klassendiagramm der OCM-Fusion	73
7.7	Klassendiagramm des Relationsgraph-Generators	74
7.8	Klassendiagramm der Visualisierung	77
7.9	Visualisierung der Inferenz	78
7.10	Visualisierung des gelernten Modells	79
8.1	Erkennungsergebnis nach Entfernung des Hintergrundterms	81
8.2	Beschreibung des Buro-Szenario	82
8.3	Anwendungsszenario Buro	83
8.4	Modelle zweier Haushaltsszenen	86
8.5	Ubergang zwischen zwei Haushaltsszenen	87
8.6	Fehlerkennung eines Bechers	89
8.7	Robustheit gegenuber Fehlerkennungen	90
8.8	Trajektorien zur Demonstration weicher Entscheidungsgrenzen	91
8.9	Bias auf Grund weicher Entscheidungsgrenzen	92
8.10	Modelle fur das Fehlerkennungsszenario	93
8.11	Auswertung des Fehlerkennungsszenario	94
8.12	Vergleich der Laufzeiten von PSM und ISM	95
8.13	Laufzeiten des PSM in Abhangigkeit von Objekten und Evidenzen	96
8.14	Skalierung der Laufzeiten des PSM	97
8.15	Beschreibung des Buro-Szenario	98

1. Einführung

Der Forschungsbereich der Servicerobotik beschäftigt sich mit der Entwicklung autonomer Agenten, die durch den bewussten Umgang mit ihrer Umwelt dazu in der Lage sind, den Menschen in Alltagssituationen zu unterstützen. Bereits heute existiert eine Fülle von Robotern mit ständig wachsenden Fähigkeiten, die im haushaltlichen Umfeld Aufgaben wie das Staubsaugen oder Fensterputzen übernehmen. Bei der Altenpflege werden sie zur Betreuung eingesetzt, in Krankenhäusern erledigen die Botengänge. Im industriellen Umfeld sind Roboter in Bereichen wie Fertigung und Montage nicht mehr wegzudenken.

Die besagten Aufgaben sind meist einfacher Natur, zumindest aus Sicht der Programmierung. Es existiert eine Vielzahl von Tätigkeiten, wie beispielsweise das Einschenken eines Glas Wassers in einer beliebig strukturierten Umwelt, die für den Menschen einfach zu lösen, für eine Maschine jedoch hochkomplex sind. Glas und Flasche müssen identifiziert und mit ausreichender Kraft gegriffen werden. Die Flasche muss geöffnet werden. Planung ist erforderlich um zu bestimmen, wie die Öffnung der Flasche über die Zeit hinweg platziert werden muss, um die richtige Menge Flüssigkeit in das Glas abzugeben. Beim Programmieren durch Vormachen (PdV) werden Methoden erforscht, deren Ansatzpunkt nicht darin besteht, jede der Aufgaben manuell zu programmieren. Vielmehr soll der Roboter diese anhand von Demonstrationen erlernen. Dies ist ein für den Menschen intuitiver Ansatz, der dem Demonstrator vor allem keine Kenntnisse der Informatik abverlangt.

Ein Roboter muss entscheiden können, unter welchen Umständen die ihm beigebrachte Aufgabe reproduzierbar ist. Dies umfasst sowohl die Identifikation der Umgebung wie auch der zur Verfügung stehenden Werkzeuge. Diese Aufgabe wird als *Szenenanalyse* mit dem Ziel der *Szenenerkennung* bezeichnet. Die hierfür benötigten Beobachtungen der Umwelt sind meist verrauscht und mit Unsicherheit behaftet. Im Rahmen dieser Arbeit wird daher ein System erforscht, welches die Identifikation bekannter Umweltzustände auf Basis unvollständiger und fehlerbehafteter Beobachtungen ermöglicht.

2. Motivation und Problemstellung

2.1 Motivation

Die steigende Komplexität von Robotern und deren Möglichkeiten zur Manipulation der Umwelt macht es notwendig, dass diese sich ihrer Umgebung bewusst sind. Im Bezug auf das Programmieren durch Vormachen besteht ein wichtiger Teil des Umweltverständnis in der Erkennung eines bekannten Umgebungszustands. Dies ist deswegen relevant, da tagtägliche Verrichtungen meist in einem spezifischen Kontext stattfinden, der als deren Voraussetzungen angesehen werden kann. Von einem theoretischen Standpunkt aus trägt die erfolgreiche Erkennung von Voraussetzungen dazu bei, den Raum der möglichen Aktionen einzuschränken. Dies vereinfacht die Planung und erlaubt es auf Begebenheiten zu reagieren, welche nicht mit der Erfüllung der gestellten Aufgabe im Einklang stehen. Die Zubereitung eines Kaffee setzt beispielsweise voraus, dass eine Tasse im Einflussbereich des Roboters vorhanden ist. Ist dies nicht der Fall, so muss zuerst eine beschafft werden.

Die alltägliche Umwelt mag auf den ersten Blick chaotisch erscheinen, wird jedoch interpretier- und berechenbar, sobald die darin innewohnenden Gesetzmäßigkeiten verstanden sind. Eine *Szene* übernimmt die Rolle eines semantischen Konstrukts, welches die Gesetzmäßigkeiten für einen bestimmten Umweltzustand beschreibt. Szenen in Indoor-Szenarien lassen sich am besten durch die darin enthaltenen Objekte beschreiben [QT09]. In Bezug auf das Programmieren durch Vormachen lassen sich Objekte als Entitäten betrachten, mit denen sich Aktionen durchführen lassen.

Die besagten Gesetzmäßigkeiten umfassen die für den Umweltzustand aussagekräftigen Objekte. Da beim gegenwärtigen Stand der Forschung deren Erkennung fehlerbehaftet ist, spielt nicht nur die Identität eines Objekts, sondern auch dessen Wahrnehmung eine Rolle. Beispielsweise kann eine Schale aus einem ungünstigen Blickwinkel als Teller erkannt werden, daher kann die Beobachtung eines Tellers auch auf eine Schale hindeuten. Weiterhin liefert die frequentistische Auftrittshäufigkeit einzelner Objekte wertvolle Zusatzinformationen, mit denen Mehrdeutigkeiten aufgelöst werden können.

Als Hauptinformationsträger werden räumliche Relationen in Form von relativen Objektlagen angesehen, wobei hier sowohl Position als auch Orientierung gemeint sind. Zum Beispiel liegt das Besteck vor dem Essen neben dem Teller, danach schräg auf dem Teller. Beide Szenen umfassen dieselben Objekte, die auch mit derselben Häufigkeit auftreten. Eine Unterscheidung wird jedoch erst durch die Betrachtung der räumlichen Anordnung möglich. Diese kann nicht in allen Fällen als statisch angesehen werden. So bewegt sich das zum Essen genutzte Besteck zwischen dem Teller und seinem Benutzer.

Bei alltäglichen Anwendungen ist mit Störungen zu rechnen. So kann es vorkommen, dass bestimmte Mahlzeiten ein komplettes Set an Besteck, andere dafür nur ein Messer oder einen Löffel erfordern, alle möglichen Szenarien jedoch nicht separat demonstriert werden. Es muss also davon ausgegangen werden, dass Objekte fehlen. Dafür könne zusätzliche Objekte auf dem Tisch stehen, die nicht Teil der ursprünglich eingelernten Szene sind.

2.2 Fokus der Arbeit

Die vorliegende Arbeit setzt es sich zum Ziel, ein System zur Erkennung von Szenen zu entwickeln. Hierbei sollen alle im vorherigen Abschnitt genannten Informationsquellen berücksichtigt werden. Dies umfasst die beteiligten Objekte bzw. deren Erscheinung, welche von den zur Erkennung eingesetzten Werkzeugen abhängig ist. Die Auftrittshäufigkeit der Objekte soll ebenfalls mit einfließen. Der Hauptinformationsträger sind die Relationen der Objekte untereinander, welche in Form von relativen Objektlagen berücksichtigt werden. Es soll berücksichtigt werden, dass Relationen nicht statisch, sondern dynamisch sind. Da eine Szene auch unabhängig von ihrem Ort der Demonstration erkannt werden soll, muss die Lagebeschreibung invariant gegenüber Rotation und Translation sein.

Weiterhin soll Robustheit gegenüber den genannten Störfaktoren bestehen. Fehlende Objekte sollen nach wie vor eine Erkennung der Szene erlauben, wenn auch mit entsprechend reduzierter Konfidenz. Da jedes Objekt fehlen kann darf es kein zentrales Referenzobjekt geben, von dem die Relationen zu den anderen Objekten der Szene ausgehen. Überzählige Objekte sollen sich nicht negativ auf das Erkennungsergebnis auswirken. Generell soll sich die Funktionsweise des entwickelten Systems im Rahmen dessen bewegen, was plausibel erscheint.

Da das System auf einer bereits existierenden Infrastruktur aufsetzt, sind einige zusätzliche Anforderungen und Einschränkungen gegeben. Es ist davon auszugehen, dass pro Objekttyp nur eine einzige Instanz vorkommt, also nicht mit zwei Objekten desselben Aussehens zu rechnen ist. Der Grund hierfür ist, dass keine Tracking-Komponente existiert und daher mehrere Instanzen desselben Objekts nicht voneinander unterschieden werden können. Weiterhin können die relevanten Relationen zwischen den Objekten einer Szene als gegeben angesehen werden, da eine entsprechende Komponente für deren Generierung bereits existiert. Das System sollte außerdem in ein probabilistisches Planungssystem integrierbar sein.

Der zur Umsetzung gewählte Ansatzpunkt basiert auf der *Bayesschen Statistik*, da diese das Ziehen von Schlussfolgerungen auf Basis von Unsicherheiten erlaubt. Weiterhin können die Erkennungsergebnisse leicht in die probabilistische Planung einbezogen werden. Zur Modellierung einer Szene wird das *Constellation Model* eingesetzt. Hierbei handelt es sich um ein Teile- und Strukturmodell, welches bereits zwei der drei oben genannten Informationsquellen berücksichtigt. Die modellierbaren Relationen entsprechen einer Sternform, können jedoch leicht auf beliebige Baumstrukturen erweitert werden. Weiterhin bietet es den nötigen Grad an Flexibilität, der für die Modellierung dynamischer Relationen vorausgesetzt wird.

Die vorliegende Arbeit ist wie folgend gegliedert. Zunächst werden in Kapitel 3 die benötigten Grundlagen der probabilistischen Modellierung sowie der wichtigsten Werkzeuge erläutert. Kapitel 4 gibt einen Überblick über den Stand der Forschung in den miteinander stark verwandten Bereichen der Objekt- und Szenenerkennung. Die für die Modellierung eingesetzten Teile- und Strukturmodelle werden erläutert und ein Überblick über die probabilistische Posenmodellierung gegeben. In Kapitel 5 wird das entwickelte Szenenmodell vorgestellt, die zugehörigen Algorithmen für Inferenz und Lernen werden in Kapitel 6 präsentiert. Kapitel 7 gibt einen Überblick über die bereits bestehende Infrastruktur sowie die hier entwickelten Komponenten und deren Architektur. Das System wird in Kapitel 8 einer Evaluation unterzogen und mit einem anderen Szenenerkennungssystem verglichen. Eine Zusammenfassung der Ergebnisse sowie Vorschläge für weiterführende Forschungen werden in Kapitel 9 gegeben.

3. Grundlagen

Für die Entwicklung des Wahrscheinlichkeitsmodells zur Szenenerkennung werden einige theoretische Grundlagen vorausgesetzt. Zunächst werden die wichtigsten Punkte der Wahrscheinlichkeitsrechnung erläutert. Im Anschluss werden relevante Wahrscheinlichkeitsverteilungen und ihre wichtigsten Eigenschaften genannt. Den Abschluss bildet eine Erläuterung des Konzepts der probabilistischen Modellierung, auf dem die vorliegende Arbeit aufbaut.

3.1 Wahrscheinlichkeitsrechnung

In diesem Abschnitt werden einige Grundbegriffe der Wahrscheinlichkeitsrechnung erläutert. Die Zusammenfassung beschränkt sich nur auf die wichtigsten Konzepte, elementare Themen wie die *Axiome von Kolmogorov* werden dabei als bekannt vorausgesetzt. Für eine ausführliche Einführung sei auf entsprechende Standardwerke wie [Pri12], [Bar12] und [TBF05] verwiesen.

3.1.1 Zufallsvariablen

Eine Zufallsvariable x beschreibt die Menge der möglichen Ausgänge eines Zufallsexperiments, wobei x sowohl diskret als auch kontinuierlich sein kann.

Eine Aussage über die Häufigkeit der von x beschriebenen Ereignisse liefert die Wahrscheinlichkeitsverteilung bei diskreten und die Wahrscheinlichkeitsdichtefunktion bei kontinuierlichen Zufallsvariablen. Diese wird in beiden Fällen durch $P(x)$ bezeichnet.

An eine gültige Verteilung sind bestimmte Randbedingungen geknüpft. So dürfen Wahrscheinlichkeiten nur größer als Null sein. Das Integral über eine Verteilung muss immer eins ergeben.

Diskrete Wahrscheinlichkeiten lassen sich wie in Abbildung 3.1 gezeigt als Histogramm oder Hinton-Diagramm darstellen, kontinuierliche durch den Graph der zugehörigen Wahrscheinlichkeitsdichtefunktion (vgl. hier auch beispielsweise [Pri12], [Bar12], [TBF05]).

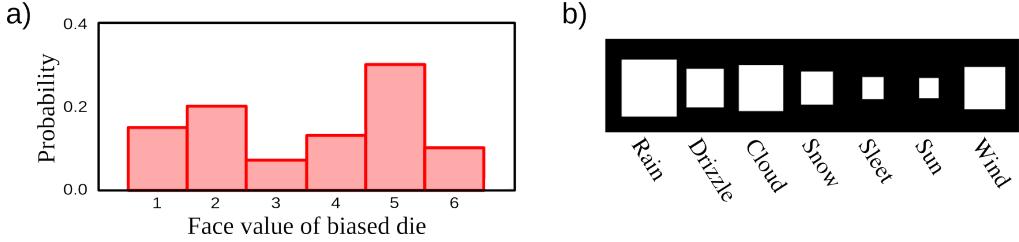


Abbildung 3.1: Zwei unterschiedliche Darstellungen diskreter Wahrscheinlichkeiten. a) Das Balkendiagramm eines gezinkten Würfels beschreibt, mit welcher Wahrscheinlichkeit eine Seite des Würfels oben landet. b) Das Hinton-Diagramm beschreibt die Auftrittswahrscheinlichkeit verschiedener Wettertypen in England. Quelle: [Pri12]

3.1.2 Verbundverteilung

Eine *gemeinsame Verteilung* oder auch *Verbundverteilung* (engl.: Joint Distribution) beschreibt eine Verteilung über mehrere Zufallsvariablen. Wie schon bei Verteilungen über einzelne Variablen muss das Integral den Wert eins ergeben.

Eine gemeinsame Verteilung kann diskrete und kontinuierliche Zufallsvariablen umfassen. Für die Zufallsvariablen x und y wird die gemeinsame Verteilung durch $P(x, y)$ dargestellt (vgl. hier auch beispielsweise [Pri12], [Bar12], [TBF05]). Abbildung 3.2 zeigt Verbundverteilungen über kontinuierliche, diskrete oder beide Arten von Variablen.

3.1.3 Marginalisierung

Die Wahrscheinlichkeit einer beliebigen Zufallsvariable kann aus einer gemeinsamen Verteilung zurückgewonnen werden. Hierzu wird über alle anderen Zufallsvariablen integriert

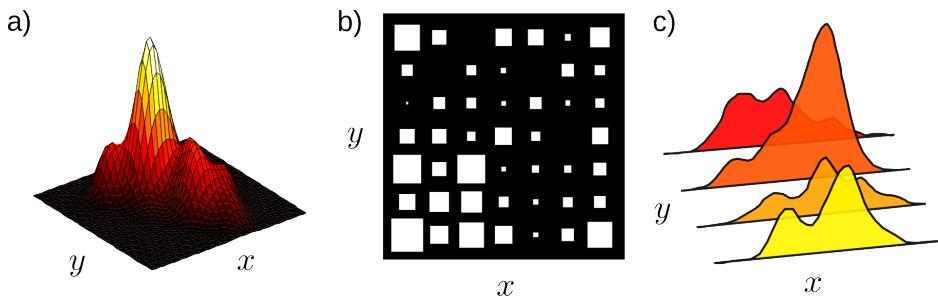


Abbildung 3.2: Verbundverteilungen über die Zufallsvariablen x und y . a) Eine Verteilung über zwei kontinuierliche Variablen. b) ein Hinton-Diagramm über zwei diskrete Variablen c) Verbundverteilung über eine kontinuierliche Variable x und eine diskrete Variable y . Quelle: [Pri12]

bzw. im diskreten Fall summiert.

$$P(x) = \int P(x, y) dy \quad (3.1)$$

Dieser Prozess wird als *Marginalisierung* bezeichnet, die resultierende Verteilung als *Marginal-Verteilung*. Gleichung 3.1 zeigt, wie $P(x)$ aus der gemeinsamen Verteilung $P(x, y)$ gewonnen werden kann (vgl. hier auch beispielsweise [Pri12], [Bar12], [TBF05]).

3.1.4 Gesetz der bedingten Wahrscheinlichkeiten

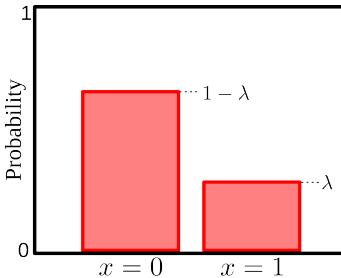
Das Gesetz der bedingten Wahrscheinlichkeiten beschreibt die relative Wahrscheinlichkeit von der Zufallsvariable x gegeben y . Der Wert von x ist also von y stochastisch abhängig.

$$P(x|y) = \frac{P(x, y)}{P(y)} \quad (3.2)$$

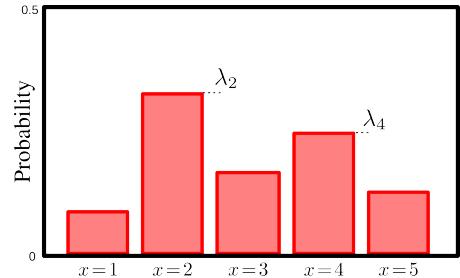
Dieses Gesetz eignet sich zum Zerlegen von Verbundverteilungen. Hierzu wird Gleichung 3.2 dementsprechend umgestellt und für die Substitution eingesetzt. Durch mehrmaliges Wiederholen kann so eine Verbundverteilung in eine Reihe von bedingten Verteilungen sowie eine unbedingte Verteilung zerlegt werden (vgl. hier auch beispielsweise [Pri12], [Bar12], [TBF05]).

3.2 Wahrscheinlichkeitsverteilungen

In der Arbeit werden im Rahmen des probabilistischen Modells einige Wahrscheinlichkeitsverteilungen einsetzt. Die dahinterstehenden Konzepte sowie die verwendete Schreibweise werden hier erläutert.



(a) Die Bernoulli-Verteilung verfügt über zwei mögliche Ergebnisse.



(b) Die Multinomialverteilung ist eine Generalisierung der Bernoulli-Verteilung mit K unterschiedlichen Ergebnissen.

Abbildung 3.3: Zwei gängige diskrete Wahrscheinlichkeitsverteilungen. Quelle: [Pri12]

3.2.1 Bernoulli-Verteilung

Die *Bernoulli-Verteilung* ist eine diskrete Wahrscheinlichkeitsverteilung. Hiermit lassen sich Situationen beschreiben, in denen nur zwei Ausgänge $x \in \{0, 1\}$ möglich sind.

$$Bern_x[\lambda] = \lambda^x(1 - \lambda)^{1-x} \quad (3.3)$$

Die in Gleichung 3.3 gezeigte Verteilung ist nur vom Parameter $\lambda \in [0, 1]$ abhängig. Dieser beschreibt die Wahrscheinlichkeit für den Fall $x = 1$ (vgl. hier auch beispielsweise [Pri12, Kapitel 3.1]).

3.2.2 Multinomialverteilung

Die *Multinomialverteilung* (engl.: Categorical Distribution) ist eine Verallgemeinerung der Bernoulli-Verteilung und damit ebenfalls diskret. Mit ihr lässt sich die Wahrscheinlichkeit für das Auftreten eines von K unterschiedlichen Ergebnissen beschreiben.

$$Cat_x[\boldsymbol{\lambda}] = \prod_{k=1}^K \lambda_k^{[x=k]} = \lambda_x \quad (3.4)$$

Die Wahrscheinlichkeit für die Beobachtung eines der Fälle ist in dem $K \times 1$ Vektor $\boldsymbol{\lambda}$ kodiert. Es gilt, dass $\lambda_k \in [0, 1]$ und $\sum_{k=1}^K \lambda_k = 1$. Gleichung 3.4 zeigt die Verteilung (vgl. hier auch beispielsweise [Pri12, Kapitel 3.3]).

3.2.3 Gauss-Verteilung

Die multivariate *Gauss-* oder auch *Normalverteilung* ist eine häufig verwendete, kontinuierliche Wahrscheinlichkeitsverteilung (siehe Abbildung 3.4). Sie verfügt über zwei Parameter, den Mittelwert $\boldsymbol{\mu}$ und die Kovarianz $\boldsymbol{\Sigma}$.

$$Norm_{\mathbf{x}}[\boldsymbol{\mu}, \boldsymbol{\Sigma}] = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp [-0.5(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})] \quad (3.5)$$

Bei ersterem handelt es sich um einen $D \times 1$ Vektor, von welchem die Position der Verteilung im D -dimensionalen Raum abhängt. Die Kovarianz $\boldsymbol{\Sigma}$ ist eine symmetrische $D \times D$ positiv definite Matrix und beschreibt die Form der Verteilung.

Gleichung 3.5 zeigt die Wahrscheinlichkeitsdichtefunktion der Gauss-Verteilung (vgl. hier auch beispielsweise [Pri12, Kapitel 3.7]).

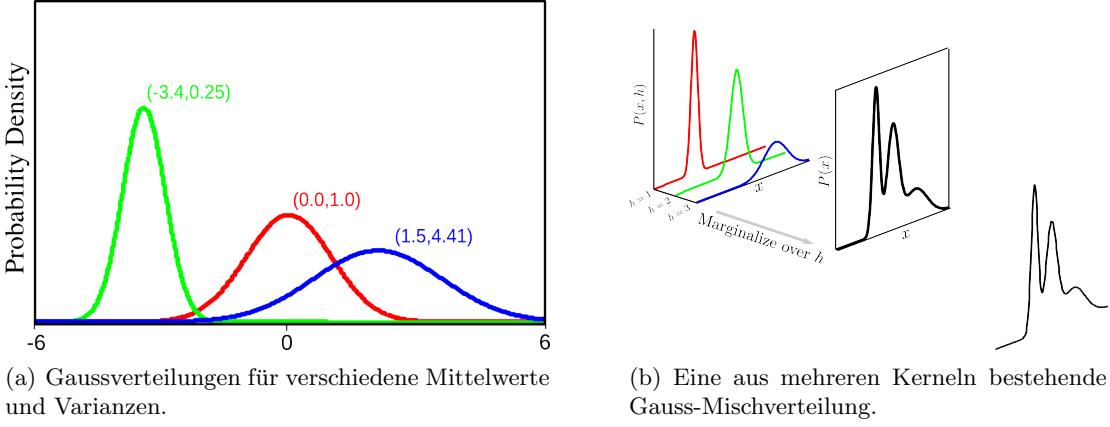


Abbildung 3.4: Gauss-Mischverteilung bestehen aus einzelnen Gauss-Kernen. Sie eignen sich zur Modellierung von Unsicherheiten über kontinuierlichen Zufallsvariablen. Quelle: [Pri12]

3.2.4 Gauss-Mischverteilung

Die *Gauss-Mischverteilung* (engl.: Gaussian Mixture Distribution) stellt eine aus der Gauss- und der Multinomialverteilung zusammengesetzte Verteilung dar.

$$GMM_{\mathbf{x}}[\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\Sigma}] = \sum_{i=1}^n \lambda_i Norm_{\mathbf{x}}[\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i] \quad (3.6)$$

Gleichung 3.6 zeigt die Wahrscheinlichkeitsdichtefunktion. Die n verschiedenen Gauss-Kernel sind durch den Parameter-Vektor $\boldsymbol{\lambda}$ der Multinomialverteilung gewichtet und tragen so unterschiedlich zur Gesamtwahrscheinlichkeit bei (vgl. hier auch beispielsweise [Pri12, Kapitel 7.4]).

3.3 Probabilistische Modellbildung

In diesem Abschnitt werden die für das Verständnis der Arbeit notwendigen Konzepte aus dem Bereich der *probabilistischen Modellbildung* zusammengefasst.

3.3.1 Bayes-Theorem

Das zentrale Element vieler probabilistischer Modelle bildet das *Bayes-Theorem*. Es beschreibt, wie sich der Zustand \mathbf{w} der Welt aus einer Beobachtung \mathbf{x} herleiten lässt.

$$P(\mathbf{w}|\mathbf{x}) = \frac{P(\mathbf{x}|\mathbf{w})P(\mathbf{w})}{P(\mathbf{x})} \quad (3.7)$$

Jeder Term aus Gleichung 3.7 trägt eine Bezeichnung. Das gesuchte Ergebnis $P(\mathbf{w}|\mathbf{x})$ wird auch als *posterior* bezeichnet. Dessen Herleitung basiert auf dem *prior* genannten Vorwissen $P(\mathbf{w})$ und der *likelihood* $P(\mathbf{x}|\mathbf{w})$. Letzteres ist die Wahrscheinlichkeit der Beobachtung auf Basis des aktuell bekannten Weltzustands. $P(\mathbf{x})$ wird als *evidence* bezeichnet und dient als Normalisierungsterm (vgl. hier auch beispielsweise [Pri12, Kapitel 2.5]).

3.3.2 Modell, Lernen, Inferenz

Zur Lösung des vorliegenden Problems sind drei Komponenten erforderlich:

- Das *Modell* spezifiziert eine Familie möglicher Relationen zwischen dem Weltzustand \mathbf{w} und der Beobachtung \mathbf{x} . Die Wahl der speziellen Relation erfolgt über die Modellparameter $\boldsymbol{\theta}$.
- Der *Lern-Algorithmus* bestimmt $\boldsymbol{\theta}$ auf Basis einer Menge von Trainingsdaten $\{\mathbf{w}_i, \mathbf{x}_i\}$ für die sowohl die Beobachtung als auch der Weltzustand bekannt sind.
- Der *Inferenz-Algorithmus* verwendet das Modell, um einen neuen Weltzustand $P(\mathbf{w}|\mathbf{x})$ aus einer Beobachtung \mathbf{x} herzuleiten.

Die Wahl des richtigen Modells ist essentiell, da hiervon die Mächtigkeit der Beschreibungsfähigkeit sowie die Komplexität des Lern- und Inferenz-Algorithmus abhängen (vgl. hier auch beispielsweise [Pri12, Kapitel 6.1]).

3.3.3 Diskriminative und generative Modelle

Es lassen sich zwei Kategorien von Modellen unterscheiden:

- *Diskriminative* Modelle beschreiben den Weltzustand auf Basis der Beobachtung $P(\mathbf{w}|\mathbf{x})$.
- *Generative* Modelle gehen umgekehrt vor und beschreiben die Beobachtungen auf Basis des Weltzustands $P(\mathbf{x}|\mathbf{w})$.

Wie sich unschwer erkennen lässt entspricht das diskriminative Modell dem *posterior* und das generative Modell dem *likelihood*-Term.

Beide Ansätze bieten unterschiedliche Vor- und Nachteile. So ist die Inferenz für diskriminative Modelle denkbar einfach, da in der Verteilung nur die gegebenen Daten eingesetzt werden müssen. Allerdings ist die Komplexität der Zusammenhänge, die sich damit modellieren lassen, begrenzt.

Generative Modelle sind in ihrer Beschreibungsfähigkeit weitaus mächtiger. Sie eignen sich besonders, wenn hinter dem zu lösenden Problem ein Prozess mit festen Gesetzmäßigkeiten steht, beispielsweise ein physikalischer Prozess. Hierdurch lässt sich Expertenwissen einbringen und auch über ungesehene Trainingsdaten interpolieren.

Die Inferenz ist jedoch aufwändiger, da die *posterior*-Verteilung über das Bayes-Theorem berechnet werden muss (vgl. hier auch beispielsweise [Pri12, Kapitel 6.2]).

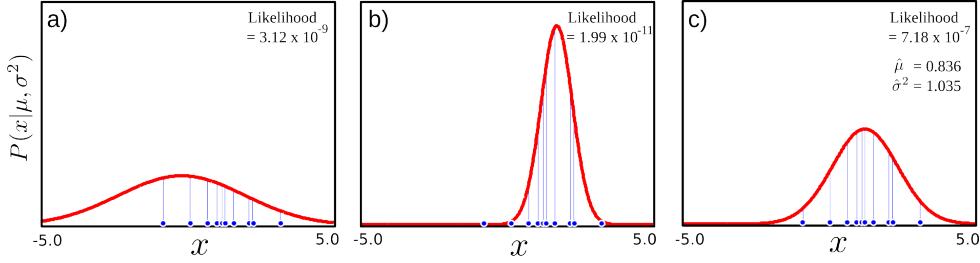


Abbildung 3.5: Parameterbestimmung nach Maximum Likelihood. Die Wahrscheinlichkeit für einen Datenpunkt entspricht der Höhe der Dichtefunktion an dieser Stelle. Die Gesamtwahrscheinlichkeit ist das Produkt aller einzelnen Wahrscheinlichkeiten a) Niedrige Wahrscheinlichkeit durch hohe Varianz. b) Wahrscheinlichkeit tendiert gegen Null, da ein Datenpunkt außerhalb der Dichtefunktion liegt. c) Die Maximum Likelihood Lösung ist die Menge der Parameter, für welche die Wahrscheinlichkeit maximal ist. Quelle: [Pri12]

3.4 Lernen der Modellparameter

In der Literatur sind die Lernmethoden *Maximum Likelihood*, *Maximum a Posteriori* und die *Bayessche Methode* weit verbreitet (vergleiche auch [Pri12, Kapitel 4]). Es wird nur der für die vorliegende Arbeit ausreichende erste Ansatz sowie eine darauf aufbauende Methode, der *EM-Algorithmus*, vorgestellt.

3.4.1 Maximum Likelihood

Die Maximum Likelihood (ML) Methode findet die Modellparameter $\hat{\theta}$, unter denen die Lerndaten am wahrscheinlichsten sind. Unter der Annahme, dass die Lerndaten unabhängig voneinander gezogen wurden, kann die gemeinsame Verteilung $P(\mathbf{x}_{1:n}|\boldsymbol{\theta})$ als Produkt der unabhängigen Verteilungen geschrieben werden.

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg \max_{\boldsymbol{\theta}} [P(\mathbf{x}_{1:n}|\boldsymbol{\theta})] \\ &= \arg \max_{\boldsymbol{\theta}} \left[\prod_{i=1}^I P(\mathbf{x}_i|\boldsymbol{\theta}) \right]\end{aligned}\tag{3.8}$$

Abbildung 3.5 veranschaulicht den Prozess. Diese Methode ist als abstraktes Framework zu betrachten. Eine konkrete Realisierung wird im nächsten Abschnitt beschrieben (vgl. hier auch beispielsweise [Pri12, Kapitel 4.1], [Bar12, Kapitel 8.7]).

3.4.2 EM-Algorithmus

Unter einem Expectation Maximization (EM)-Algorithmus versteht man eine iterative Methode, um die Maximum Likelihood bzw. Maximum a Posteriori-Schätzung der Modellparameter $\boldsymbol{\theta}$ zu ermitteln.

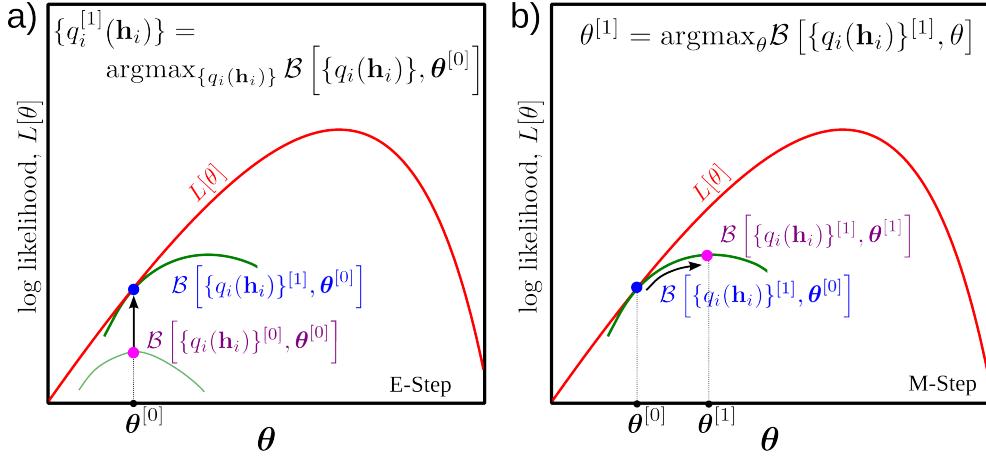


Abbildung 3.6: E-Step und M-Step. a) Beim E-Step wird die Verteilung $q_i(\mathbf{h}_i)$ manipuliert, um die beste neue Schranke für die gegebenen Parameter $\boldsymbol{\theta}$ zu ermitteln. b) Beim M-Step wird $q_i(\mathbf{h}_i)$ konstant gehalten und $\boldsymbol{\theta}$ optimiert.
Quelle: [Pri12]

In seiner heutigen Form wurde der Algorithmus 1977 von *Arthur Dempster, Nan Laird* und *Donald Rubin* [DLR77] entwickelt. Zuvor hatte *Herman O.Hartley* in den späten 1950er Jahren Pionierarbeit auf den Gebiet geleistet [Har58]. Im Jahr 1983 veröffentlichte *C. F. Jeff Wu* eine Korrektur der in der ursprünglichen Veröffentlichung enthaltenen Konvergenzanalyse, die sich als fehlerhaft herausgestellt hatte [Wu83]. Heute zählt der EM-Algorithmus zu den Standardverfahren der Parameterschätzung [DHS00].

$$P(\mathbf{x}|\boldsymbol{\theta}) = \int P(\mathbf{x}, \mathbf{h}|\boldsymbol{\theta}) d\mathbf{h} \quad (3.9)$$

Anwenden lässt sich der Algorithmus auf Modelle der in Gleichung 3.9 gezeigten Form, bei der das Modell von einer unbekannten, nicht beobachtbaren Variable abhängig ist. Kombiniert man den ML-Schätzer aus Gleichung 3.8 mit dem obigen Modell, so erhält man folgenden Term:

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \left[\sum_{i=1}^I \log \left[\int P(\mathbf{x}_i, \mathbf{h}_i | \boldsymbol{\theta}) d\mathbf{h}_i \right] \right] \quad (3.10)$$

Der EM-Algorithmus arbeitet, indem eine untere Schranke $\mathcal{B}[q_i(\mathbf{h}_i), \boldsymbol{\theta}]$ definiert und iterativ vergrößert wird. Hierbei handelt es sich um eine durch $\boldsymbol{\theta}$ parametrisierte Funktion, deren Rückgabe immer garantiert kleiner oder gleich der logarithmische Wahrscheinlichkeit $L[\boldsymbol{\theta}]$ ist. Dies gilt für alle möglichen Parameter $\boldsymbol{\theta}$.

Zusätzlich basiert die untere Schranke auf einer Reihe von Verteilungen $\{q_i(\mathbf{h}_i)\}_{i=1}^I$ über die versteckten Variablen $\{\mathbf{h}\}_{i=1}^I$. Auch hier gilt, dass Änderungen an der Verteilungen

zu Änderungen an der Schranke führen, jedoch das Ergebnis immer kleiner oder gleich der logarithmischen Wahrscheinlichkeit ist.

Der EM-Algorithmus manipuliert nun iterativ beide Parameter, um so die Schranke zu erhöhen (siehe Abbildung 3.6). Hierbei wechselt er zwischen den folgenden beiden Schritten:

- Update der Verteilungen $\{q_i(\mathbf{h}_i)\}_{i=1}^I$ zur Erhöhung der Schranke. Dies wird auch als *expectation*-Schritt oder *E-Step* bezeichnet.
- Update der versteckten Variablen $\{\mathbf{h}\}_{i=1}^I$ zur Erhöhung der Schranke. Dies wird auch als *maximization*-Schritt oder *M-Step* bezeichnet.

Im E-Step für die Iteration $t + 1$ werden die Verteilungen $q_i(\mathbf{h}_i)$ durch die posterior-Verteilungen $P(\mathbf{h}_i|\mathbf{x}_i, \boldsymbol{\theta}^{[t]})$ ersetzt. Diese werden mit Hilfe des Bayes-Theorem berechnet:

$$\hat{q}_i(\mathbf{h}_i) = P(\mathbf{h}_i|\mathbf{x}_i, \boldsymbol{\theta}^{[t]}) = \frac{P(\mathbf{x}_i|\mathbf{h}_i, \boldsymbol{\theta}^{[t]})P(\mathbf{h}_i|\boldsymbol{\theta}^{[t]})}{P(\mathbf{x}_i)} \quad (3.11)$$

Dieses Vorgehen maximiert die Schranke so viel wie möglich. Im M-Step werden die Parameter $\boldsymbol{\theta}$ und damit die Schranke an sich angepasst. Auch dieser Schritt garantiert eine Verbesserung der Schranke.

$$\hat{\boldsymbol{\theta}}^{[t+1]} = \arg \max_{\boldsymbol{\theta}} \left[\sum_{i=1}^I \int \hat{q}_i(\mathbf{h}_i) \log \left[\int P(\mathbf{x}_i, \mathbf{h}_i|\boldsymbol{\theta}) d\mathbf{h}_i \right] \right] \quad (3.12)$$

Die iterative Wiederholung beider Schritte führt zu einer kontinuierlichen Verbesserung der Schranke $\mathcal{B}[q_i(\mathbf{h}_i), \boldsymbol{\theta}]$. Hierdurch ist gewährleistet, dass zumindest ein lokales Maximum gefunden wird. Die Chancen auf ein besseres Ergebnis können erhöht werden, indem der EM-Algorithmus mehrmals mit unterschiedlichen Modellparametern $\boldsymbol{\theta}$ wiederholt und anschließend das beste Ergebnis ausgewählt wird (vgl. hier auch beispielsweise [Pri12, Kapitel 7.3] oder [DHS00, Kapitel 3.8]).

3.4.3 Lernen von Gauss-Mischverteilungen

Der EM-Algorithmus kann dafür eingesetzt werden, die in Abschnitt 3.2.4 vorgestellte Gauss-Mischverteilung zu erlernen. Dies ist notwendig, da die ML-Schätzung der Modellparameter $\boldsymbol{\theta} = \{\lambda_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ nicht in geschlossener Form berechnet werden kann.

Beide Schritte des Algorithmus sollen hier kurz angerissen werden. Im E-Step werden die Verteilungen $q_i(h_i)$ neu berechnet. Das Ergebnis wird durch den Faktor r_{ik} beschrieben, der die Wahrscheinlichkeit wiedergibt, mit welcher die k -te Normalverteilung für den i -ten Datenpunkt verantwortlich ist.

$$q_i(h_i) = \frac{\lambda_k \text{Norm}_{\mathbf{x}_i}[\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k]}{\sum_{j=1}^K \lambda_j \text{Norm}_{\mathbf{x}_i}[\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j]} = r_{ik} \quad (3.13)$$

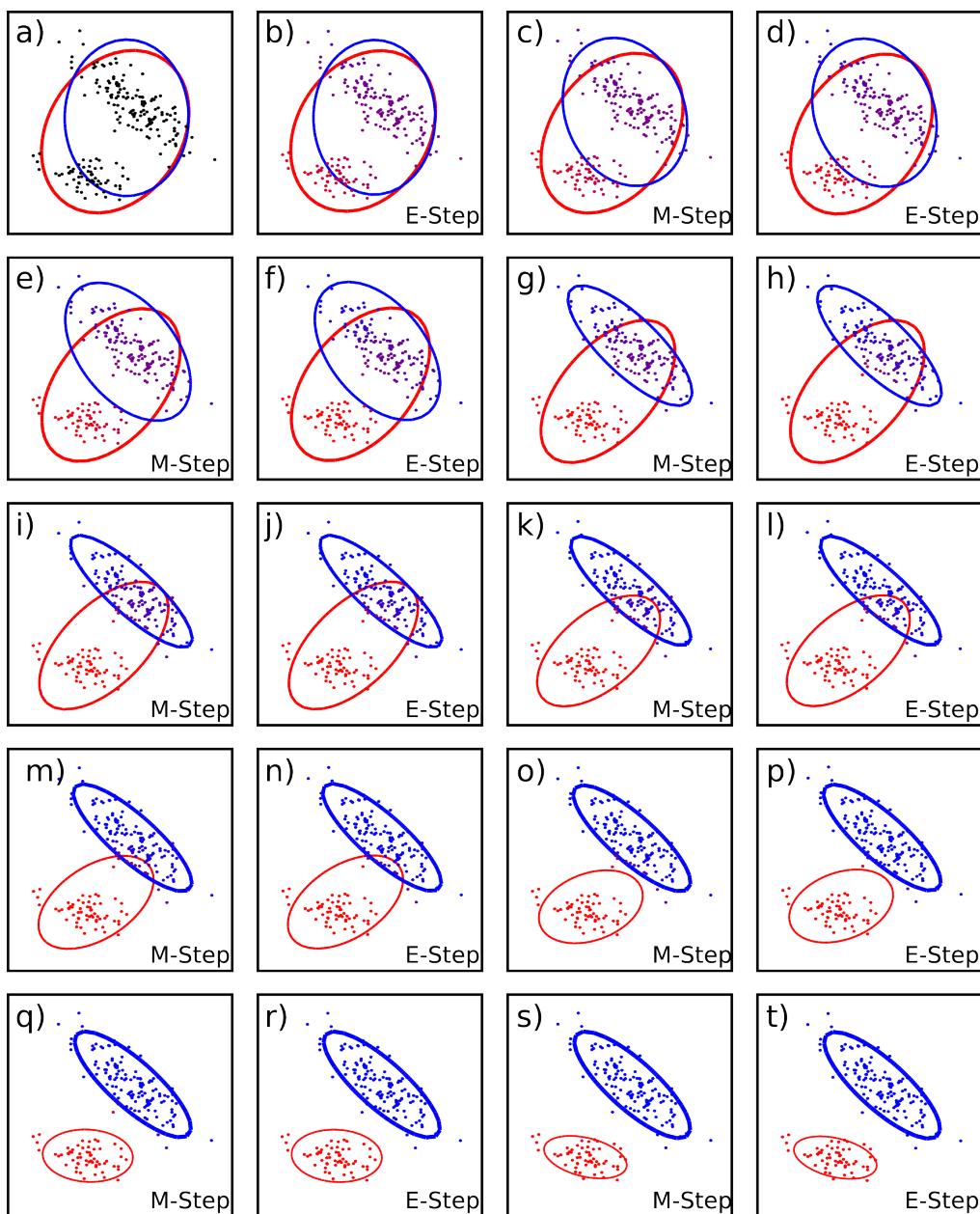


Abbildung 3.7: Durchlauf des EM-Algorithmus. Ausgehend vom initialen Modell werden iterativ E-Step und M-Step durchgeführt. Die Kovarianzellipsen kennzeichnen Mittelwert und Varianz der einzelnen Kernel, wobei die Gewichte in der Liniendicke kodiert sind. Quelle: [Pri12]

Im M-Step wird das Ergebnis aus dem E-Step eingesetzt, um die Modellparameter neu zu berechnen. Auch hier sollen nur die wichtigsten Gleichungen Erwähnung finden:

$$\begin{aligned}\lambda_k^{[t+1]} &= \frac{\sum_{i=1}^I r_{ik}}{\sum_{j=1}^K \sum_{i=1}^I r_{ij}} \\ \boldsymbol{\mu}_k^{[t+1]} &= \frac{\sum_{i=1}^I r_{ik} \mathbf{x}_i}{\sum_{i=1}^I r_{ik}} \\ \boldsymbol{\Sigma}_k^{[t+1]} &= \frac{\sum_{i=1}^I r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k^{[t+1]})(\mathbf{x}_i - \boldsymbol{\mu}_k^{[t+1]})^T}{\sum_{i=1}^I r_{ik}}\end{aligned}\tag{3.14}$$

Wie bereits im letzten Abschnitt erwähnt konvergiert der EM-Algorithmus nur in Richtung eines lokalen Maximums (vgl. hier auch beispielsweise [Pri12, Kapitel 7.4]). Abbildung 3.7 demonstriert das Konvergenzverhalten anhand eines Beispiels.

3.4.4 Wahl der Modellkomplexität

Bisher offen geblieben ist die Frage nach der *Modellkomplexität*, also der Anzahl der Gauss-Kernel, die mit dem im letzten Abschnitt beschriebenen EM-Algorithmus einge-lernt werden. Ein möglicher Ansatzpunkt ist das Erlernen von mehreren Modellen mit unterschiedlicher Kernelanzahl, wobei jedes i -te Modell mit i Gauss-Kerneln trainiert wird. Anschließend wird das Modell gewählt, das die höchste Wahrscheinlichkeit $L(\boldsymbol{\theta}|\mathbf{x})$ unter den Lerndaten erzielt.

Ein solcher Ansatz ist jedoch anfällig gegenüber Overfitting, da mit steigender Modellkomplexität auch die Lerndaten besser approximiert werden. Hier bietet sich das Bayesian Information Criterion (BIC) Kriterium an (siehe [BK10]). Es stellt ein Maß dar, um einen Kompromiss zwischen der Passgenauigkeit des Modells und der Modellkomplexität zu finden.

$$BIC = \log L(\boldsymbol{\theta}|\mathbf{x}) - \frac{|\boldsymbol{\theta}|}{2} \log n\tag{3.15}$$

In Gleichung 3.15 lässt sich erkennen, dass neben der logarithmischen Modellwahrscheinlichkeit $L(\boldsymbol{\theta}|\mathbf{x})$ ein zusätzlicher Term enthalten ist, der sich mit zunehmender Modellkomplexität negativ auf das Gesamtergebnis auswirkt und auf der Anzahl der Kernel $|\boldsymbol{\theta}|$ basiert. Die Anzahl der Lerndaten wird durch n bezeichnet. Es ist zu beachten, dass die o.g. Gleichung nur für entsprechend große n Gültigkeit besitzt.

4. Stand der Forschung

In diesem Kapitel wird der Stand der Forschung beschrieben. Zuerst wird ein Überblick über das Feld der Objekterkennung gegeben. Darauf aufbauend wird der Stand der Forschung im Bereich Szenenerkennung vorgestellt. Es folgt eine Erläuterung der für die Arbeit wichtigen *Teile- und Strukturmodelle*. Abschließend wird eine Zusammenfassung gängiger Methoden zur probabilistischen Modellierung von Posen gegeben.

4.1 Objekterkennung

In einem typischen PdV-Szenario führt der Mensch exemplarische Beispiele vor, die von der Maschine beobachtet und ausgewertet werden. Hierbei kann beliebig mit den Objekten hantiert werden. Um sinnvoll aus solchen Beispielen lernen zu können ist es notwendig, die 6D-Pose eines jeden Objekts - also Position und Orientierung - bestimmen zu können. Im Fokus dieses Abschnitts stehen daher Systeme, die sowohl zur *Objekterkennung*, als auch zur *Objektlokalisierung* fähig sind. *Azad et al.* unterscheiden hier zwischen ansichtenbasierten und modellbasierten Ansätzen [Aza08].

4.1.1 Modellbasierte Ansätze

Bei der modellbasierten Objekterkennung werden geometrische Modelle der zu erkennenden Objekte eingesetzt. Für einfache Formen bietet es sich an, Polygone zur Modellierung zu verwenden. Der von *DeMenthon et al.* vorgestellte Ansatz *POSIT* (Pose from Orthography and Scaling with Iterations) berechnet die 3D-Pose eines Objekts gegeben eine Menge von 2D-3D Punkt-Korrespondenzen, die aus dem Kamerabild und dem Modell extrahiert werden [DD92, DD95].

Dieser Ansatz ist für komplizierte Formen jedoch auf Grund der hohen Polygonanzahl sehr rechenintensiv und für Objekte mit gekrümmten Oberflächen nicht praktikabel. Der von *Harris et al.* entwickelte *RAPiD*-Algorithmus projiziert daher auf den Modelllinien liegende 3D-Kontrollpunkte ins Bild (siehe Abbildung 4.1) und nutzt deren Abstand



(a) Erste Aufnahme des Objekts. (b) Zweite Aufnahme des Objekts. (c) Das auf Basis der Aufnahmen erstellte Drahtgitter-Modell.

Abbildung 4.1: Objektmodell aus dem *RAPiD*-Algorithmus. Quelle: [AZ95]

zum nächstliegenden Kantenpunkten, um daraus in einem iterativen Prozess die Pose zu bestimmen [HS90]. Verschiedene Erweiterungen wurden entwickelt, so erweiterten *Armstrong et al.* den Ansatz um geometrische Primitive, mit denen das Objekt auf mehreren Ebenen beschrieben und so die Robustheit des Ansatzes erhöht wird [AZ95].

Neben reinen modellbasierten Ansätzen existieren auch Kombinationen mit ansichtenbasierten Methoden, beispielsweise auf Basis kompletter Objektansichten [AAD06, AAD09] oder auf einzelnen Bildmerkmalen [MB01, WH97]. Solche kombinierten Ansätze finden beispielsweise beim Greifen von Gegenständen durch Manipulatoren Anwendung [KMA01].

4.1.2 Globale ansichtenbasierte Ansätze

Die Gruppe der *ansichtenbasierten Ansätze* lässt sich in zwei Kategorien unterteilen. *Globale Ansätze* modellieren das Objekt als Ganzes, also durch komplettete Ansichten. Im Gegensatz dazu beschreiben *lokale Ansätze* ein Objekt auf Basis seiner einzelnen Teile. Die globalen Ansätze lassen sich weiter in histogrammbasierte und holistische Methoden unterteilen. Erstere ermöglichen die rotations- und skalierungsinvariante Erkennung von Objekten. Auch sind sie robust gegenüber partieller Verdeckung. Leider ermöglichen sie keine Objektlokalisierung und sind damit für die vorliegende Arbeit nicht relevant [CK99].

Holistische Ansätze nutzen eine komplette Ansicht des Objekts. Dies lässt sich mit einer Schablone vergleichen, die über das Bild gezogen wird. Für jeden Bildausschnitt wird die Korrelation mit der Schablone berechnet, das Ergebnis wird als Kriterium für eine mögliche Übereinstimmung verwendet. Dieses Vorgehen steht ganz im Kontext der klassischen Mustererkennung, bei der geeignete Merkmale mit Klassifikatoren erkannt werden (siehe auch [DHS00], [Bis07]).

Eines der ersten holistischen Systeme stammt von *Murase und Nayar*. Basierend auf einem Ansatz zur Erkennung von 3D-Objekten mit Hilfe der Hauptkomponentenanalyse [MN93] wurde ein System für die Erkennung von 100 farblichen Objekten vorgestellt. Allerdings ist die zu messende Pose nur eindimensional (umfasst lediglich die Orientierung

des Objekts um eine seiner Achsen) und es wird ein schwarzer Hintergrund vorausgesetzt [HM96]. *Zhang et al.* erweiterten die zu schätzende Pose auf drei Freiheitsgrade [ZSK99].

Der von *Viola und Jones* vorgestellte Ansatz erfordert keine Segmentierung. Es wird AdaBoost¹ eingesetzt, um einen mächtigen Klassifikator aus mehreren schwachen Klassifikatoren zu konstruieren. Hierzu werden immer komplexere Klassifikatoren zu einer Kaskade verschaltet. Dieses Vorgehen erlaubt es, Hintergrundregionen bereits in einem frühen Stadium zu verwerfen und damit der Verschwendungen von Rechenkraft vorzubeugen [VJ01]. Der Einsatz des Verfahrens für die Gesichtsdetektion ist in Abbildung 4.2 gezeigt.

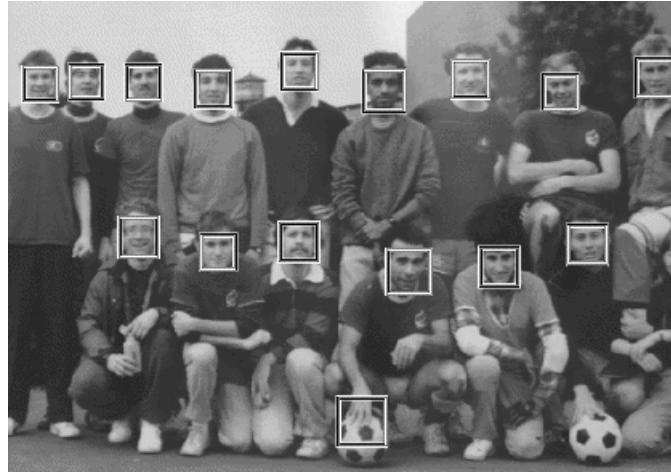


Abbildung 4.2: Gesichtserkennung mit dem von Viola und Jones entwickelten Klassifikator. Quelle: [VJ01]

Wie man erkennen kann eignen sich globale ansichtenbasierte Ansätze nur begrenzt zur Objekterkennung. Die zur Beschreibung eingesetzte Struktur ist unflexibel. Objektvariationen innerhalb einer Objektklasse können also nur dann modelliert werden, wenn für jede Variation einzelne Lerndaten vorliegen. Auch die Objektlokalisierung ist problematisch. Um die Orientierung eines Objekts in drei Dimensionen erfassen zu können ist (bereits ohne die Berücksichtigung von Variationen) eine unpraktikable Anzahl unterschiedlicher Ansichten erforderlich.

4.1.3 Lokale ansichtenbasierte Ansätze

Lokale ansichtenbasierte Ansätze zeichnen sich dadurch aus, dass das Objekt auf Basis seiner Teile modelliert wird. Dieses Vorgehen erlaubt ein höheres Maß an Flexibilität, da Variationen im Aussehen und der Lage der einzelnen Teile modellierbar sind, wodurch auch Variationen innerhalb der Objektklasse berücksichtigt werden können.

Unter den Teilen eines Objekts sind sogenannte lokale Merkmale (engl.: Features) zu verstehen. Hierbei handelt es sich um markante Bildausschnitte, die möglichst invariant

¹Siehe auch [FS97].

gegenüber Rotation und Skalierung sind und auch aus unterschiedlichen Blickwinkeln zuverlässig wiedergefunden werden können. In der Literatur sind eine ganze Reihe von Ansätzen zu finden; der *Harris Corner Detector* [HS88], die auch als *Good Features to Track* bekannten Shi-Tomasi Merkmale [ST94], *Scale-Invariant-Feature-Transform* [Low99], *Speeded-Up Robust Features* [BETVG08] und die *Maximally Stable Extremal Regions* [OM02] sind einige der populärsten.

Es existieren eine ganze Reihe von Systemen, die auf Basis von lokalen Merkmalen arbeiten [OM02] [MCT05]. *Azad et al.* ermitteln zuerste 2D-Korrespondenzen zwischen detektierten und bekannten Merkmalen und nutzen dann Tiefeninformationen eines Stereokamera-Systems, um daraus die Objektpose zu bestimmen [AAD07]. *Eidenberger et al.* triangulieren die Position von Merkmalen im dreidimensionalen Raum und nutzen einen probabilistischen Ansatz, um daraus die Objektpose zu inferieren [Eid10].

4.2 Szenenerkennung

In der aktuellen Forschungen finden sich zwei Richtungen, die das Problem der Szenenerkennung angehen. Die meisten Veröffentlichungen beschäftigen sich mit der Extraktion von Szenenwissen aus zweidimensionalen Bilddaten. Die andere Kategorie von Ansätzen führt die Szenenerkennung auf einer abstrakteren Ebene durch und arbeitet direkt mit Objektinstanzen [Mei13]. Beide Richtungen werden hier vorgestellt.

4.2.1 Parallelen zur Objekterkennung

Der Begriff *Szenenerkennung* beschreibt das Wiedererkennen einer Szene, also eines bestimmten Weltzustands. Im Englischen werden auch die Begriffe *Scene Recognition* oder *Scene Classification* verwendet. Ebenfalls in der Literatur ist *Scene Understanding* anzutreffen. Dieser Begriff ist jedoch irreführend, da er meist im Kontext der Objekterkennung eingesetzt wird.

Die beiden Forschungsrichtungen *Objekterkennung* und *Szenenerkennung* sind eng miteinander verzahnt, da die erfolgreiche Erkennung einer Szene auch die erfolgreiche Erkennung relevanter Umgebungsmerkmale voraussetzt. Die für das *Programmieren durch Vormachen* relevanten Indoor-Szenarien können am besten durch die Objekte beschrieben werden, die sie enthalten [QT09].

Aber nicht nur die Objekte, sondern auch räumliche Relationen zwischen den Objekten sind relevant und stellen eine zusätzliche Informationsquelle dar. Abbildung 4.3 soll dies mit dem Beispiel verdeutlichen, dass bereits in Kapitel 2 erwähnt wurde. Besteck liegt vor dem Essen neben dem Teller, danach auf dem Teller. Die Objekte in beiden Szenarien sind die gleichen, haben jedoch durch die relative Lage eine andere Aussage. Auch ist die Umwelt dynamisch, die Relationen dürfen also nicht als statisch angesehen werden [Mei13].

In den letzten fünfzig Jahren wurden unzählige Arbeiten auf dem Gebiet der Objekterkennung veröffentlicht [AT13]. Viele dieser Arbeiten wurden aufgegriffen, überarbeitet



(a) Das neben dem Teller liegende Besteck impliziert ein bevorstehendes Essen.
 (b) Messer und Gabel liegen auf dem Teller. Das Essen ist beendet.

Abbildung 4.3: Dieselben Objekte können je nach relativer Lage eine unterschiedliche Bedeutung haben.

und weiterentwickelt, so dass durchdachte und robuste Ansätze entstanden sind. Unter diesem Gesichtspunkt bietet es sich an, Methoden aus der Objekterkennung auch für die Erkennung von Szenen einzusetzen.

In Abschnitt 4.1 wurden drei Kategorien von Ansätzen vorgestellt. Wie sich der ausführlichen Beschreibung unterschiedlicher Ansätze entnehmen lässt sind modellbasierte Ansätze zu unflexibel, um dynamische Strukturen zu erlauben. Bei den globalen ansichtenbasierten Ansätzen stehen histogrammbasierte und holistische Methoden zur Verfügung, die jedoch ebenfalls zu unflexibel für die Repräsentation dynamischer Szenen sind.

Lokale ansichtenbasierte Ansätze eignen sich hingegen sehr gut zur Szenenerkennung. Ein Objekt besteht aus der Menge seiner Teile, eine Szene aus der Menge der zugehörigen Objekte. Diese Ähnlichkeit in der Struktur des Problems ist der Grund, der eine Anwendung der besagten Ansätze im Kontext der Szenenerkennung rechtfertigt. Das für diese Arbeit ausgewählte Modell wird im Abschnitt 4.3 beschrieben.

4.2.2 Inferenz von Szenen aus Bilddaten

Dieser Zweig der Forschung ist für die vorliegende Arbeit nur aus Gründen der Vollständigkeit von Interesse. Alle hier vorgestellten Ansätze leiten Szeneninformationen direkt aus Bilddaten ab, befassen sich dabei jedoch eher mit globalen Strukturen und Erscheinungen der Umgebung, nicht wie oben beschrieben anhand von Objekten und deren relativen Beziehungen.

In der Erkennung der Szene auf Basis von Bildmerkmalen gibt es drei Grundrichtungen [BI11]. Bei *low-level feature* Ansätzen findet die Szenenerkennung durch globale visuelle Informationen wie Farbe, Textur und Form statt. Der primäre Anwendungsbereich sind Outdoor-Szenen [BBL02]. *Local Feature* Ansätze nutzen *Interest Points* oder *Interest Regions* zur Beschreibung der Umgebung.



Abbildung 4.4: Eine Reihe von Beispielezenen aus dem von *Serre et al.* entwickelten Szenenerkennungssystem nach biologischem Vorbild. Oben: Rohdaten. Mitte: Handgelabelte Vorlage. Unten: Erkennungsergebnis des Systems. Quelle: [SWB⁺07]

Die letzte Kategorie bilden biologisch inspirierte Ansätze, die den Prozess des visuellen Kortex nachahmen. Ein hierauf basierendes Framework wurde in [SWB⁺07] vorgestellt. Ein auf dem Konzept des Kerninhalts (engl.: *gist*) basierender Ansatz bildet die Fähigkeit des Menschen nach, selbst nach nur kurzer Betrachtung eines Bildes die darauf gezeigte Szene beschreiben zu können [OT01, SI07]. Ein anderer Ansatz nutzt das Konzept der visuellen Aufmerksamkeit, bei der zunächst visuell auffällige Regionen einer Szene ausgewählt werden, so dass später komplexere Wahrnehmungsaufgaben darauf ausgeführt werden können [BI11].

4.2.3 Inferenz von Szenen aus Objekten

Der hier beschriebene Zweig der Forschung beschäftigt sich mit der Szenenerkennung auf Basis von Objekten. Statt direkt auf den Rohdaten zu arbeiten wird also eine weitere Abstraktionsebene eingezogen, auf der die Erkennung ausgeführt wird. Die vorliegende Arbeit ist in diesem Bereich einzuordnen.

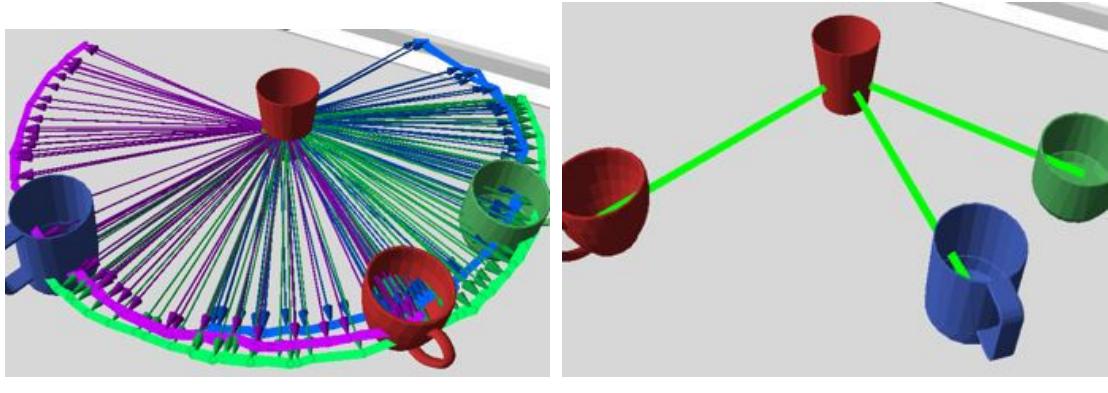
In *Kumar et al.* werden probabilistische graphische Modelle wie Markov Random Fields (MRF) und Conditional Random Fields (CRF) eingesetzt. In einem zweistufigen Framework werden zunächst Interaktionen kurzer Reichweite modelliert (d.h. Pixel mit Labels versehen). Auf der zweiten Stufe werden Interaktionen längerer Reichweite zwischen Regionen bzw. Objekten modelliert [KH05].

In *Souhey et al.* etwa werden relative Objektposen auf symbolische, qualitative Relationen (wie beispielsweise Objekt A *befindet sich über* Objekt B) abgebildet. Die Relationen werden eingesetzt, um Fehldetections von Objektdetectoren zu erkennen. Anhand von Trainingsdaten werden plausible Konstellationen erlernt, die dann für die Validierung der Detectionen eingesetzt werden [SL13].

Der von *Joho et al.* vorgestellte Ansatz ist ebenfalls erwähnenswert, auch wenn hier die Erkennung von Szenen nur implizit behandelt wird. Aus ungelabelten Daten werden wiederkehrende Objektkonfigurationen erlernt, die für die Vervollständigung einer Szenen bzw. der kompletten Generierung von Szeneninstanzen eingesetzt werden können [JTE⁺12].

Ranganathan et al. haben einen Ansatz zur semantischen Kartierung vorgestellt, der die Wiedererkennung von Orten in einer Büroumgebung anhand der darin enthaltenen Objekte erlaubt. Das probabilistische Modell baut auf der dreidimensionalen Erweiterung des Constellation Model auf, das Lernen von Objekten erfolgt überwacht und basiert auf grob vorsegmentierten Bildern. Der Schwerpunkt liegt dabei auf der Erkennung der Objekte, die Szenenerkennung wird nur am Rand behandelt. Auch werden weder Orientierungen von Objekten noch Relationen betrachtet. Positionen werden durch Gauss-Verteilungen modelliert [RD08].

Die Arbeiten von *Meißner et al.* befassen sich mit der Szenenerkennung in unstrukturierten, dynamischen Indoor-Szenarien (siehe Abbildung 4.5). Nach Kenntnis des Autors sind dies die einzigen Arbeiten, die relative Beziehungen zwischen Objekten behandeln und hierbei die komplette Pose der einzelnen Objekte berücksichtigen. Das Lernen von Szenen erfolgt überwacht anhand annotierter Beispiele.



(a) Tassen rotieren um ein stationäres Objekt. Beobachtete Trajektorien und relative Objektposen zum Zentrum.

(b) Durch das ISM komplett erkannte Szene.

Abbildung 4.5: Die von *Meißner et al.* entwickelte System nutzt relative Objektposen zur Identifikation der vorliegenden Szene. Quelle: [Mei14]

Als zu Grunde liegendes Modell wird ein Implicit Shape Model (ISM) eingesetzt, dessen Struktur - um Fehlklassifikationen vorzubeugen - auf einen Binärbaum erweitert wurde. Die im besagtem Baum modellierten Objektbeziehungen werden durch agglomeratives Clustering bestimmt, wobei Heuristiken als Distanzmaße verwendet werden [Mei13].

Eine Erweiterung kombiniert das passive Szenenverstehen mit aktiver Objektsuche. Hierzu werden die möglichen Positionen fehlender Objekte bestimmt und mittels einer Greedy-

Strategie der nächste Blickwinkel gewählt. Der Zweck hiervon ist es, Szenen auch in Fällen zu erkennen, wenn ein einzelner Blickpunkt nicht ausreicht [Mei14].

4.3 Teile- und Strukturmodelle

Die in diesem Abschnitt vorgestellten Modelle fallen in die Klasse der lokalen ansichtenbasierten Ansätze. Der Name ist vom englischen *Parts and Structure* angeleitet und spiegelt das zu Grunde liegende Paradigma der Gruppe von Ansätzen wieder, bei dem Objekte auf Basis ihrer Teile beschrieben werden. In den folgenden Abschnitten werden zwei populäre Modelle vorgestellt, die für die vorliegende Arbeit von Relevanz sind. Es handelt sich um das Implicit Shape Model und das Constellation Model (CM).

4.3.1 Implicit Shape Model

Das Implicit Shape Model basiert auf der generalisierten Hough-Transformation². Die einzelnen Teile des Objekts sind durch lokale Merkmale beschrieben, die in einem Codebuch abgelegt sind. Jeder Eintrag in besagtem Buch enthält zusätzlich die relative Position zum gemeinsamen Referenzpunkt des Objekts. Bei der Erkennung wird für jedes gefundene Merkmal die Position des Referenzpunkts prädiziert. Das Maximum im so gebildeten Abstimmungsraum wird als die Position des Referenzpunkts angenommen. Das so lokalisierte Objekt kann dann durch eine Rückprojektion segmentiert werden [LLS08]. Abbildung 4.6 zeigt den hier beschriebenen Prozess.

Das ISM baut auf einem probabilistischen Rahmenwerk auf, welches eine nicht-parametrische Repräsentation nutzt. Es wird auch als generatives Modell bezeichnet [LLG11]. Die Robustheit des Ansatzes basiert auf einer Marginalisierung, die im Gegensatz zu der multiplikativen Kombination von Evidenzen deutlich robuster ist [KHD98]. Die ursprüngliche Formulierung gab *Lehmann et al.* jedoch Anlass zur Kritik. So ist die besagte Marginalisierung aus probabilistische Sicht nicht formal korrekt, da die Evidenzen als Ausprägungen einer einzelnen Zufallsvariable betrachtet werden. Demzufolge kann pro Zeitschritt aus formaler Sicht nur jeweils eine Evidenz berücksichtigt werden. Lehmann schlägt vor, jede Evidenz als separate Zufallsvariable zu modellieren [LLG11].

Trotz mangelnder formaler Korrektheit wurden ISMs erfolgreich für die Erkennung von Fahrzeugen [LLS04] und später auch Fußgängern eingesetzt [LSS05]. *Gächter et al.* verwendeten ein ISM auf Basis von Range-Daten, um Möbelstücke anhand ihrer Einzelteile zu identifizieren [SGS08]. *Meißner et al.* stellten einen Ansatz zur Szenenerkennung vor [Mei13, Mei14].

4.3.2 Constellation Model

Das CM wurde erstmals für die Detektion und Lokalisierung von Gesichtern eingesetzt, jedoch auch erfolgreich auf andere Objektklassen wie Motorräder, Autos und gepunktete Raubkatzen angewendet. In diesem Abschnitt wird ein geschichtlicher Überblick

²Siehe auch [Bal81].

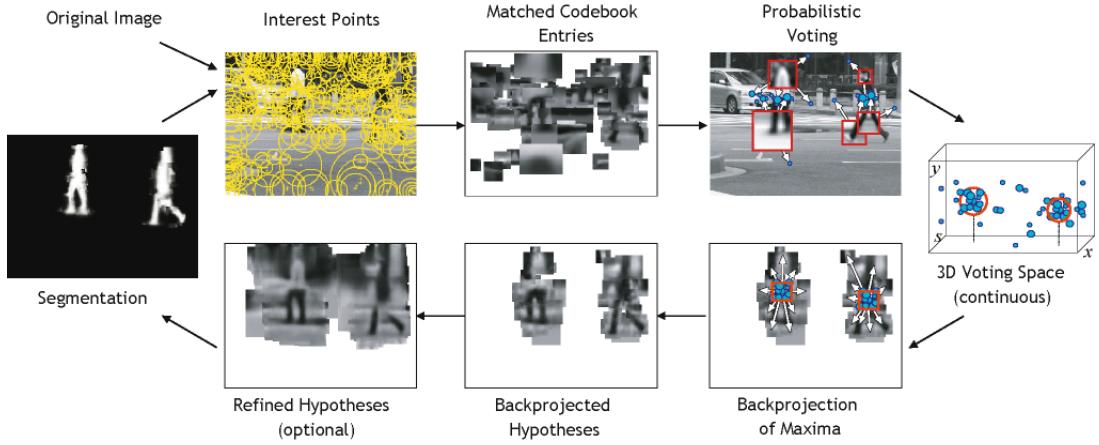


Abbildung 4.6: Der Erkennungsprozess des ISM am Beispiel einer Fußgängererkennung. Lokale Merkmale werden extrahiert und mit einem bestehenden Codebuch verglichen. Übereinstimmende Bereiche stimmen über die gemeinsame Referenz ab, die durch Maximasuche im Abstimmungsraum identifiziert wird. Durch Rückprojektion der ermittelten Hypothese wird eine kategoriespezifische Segmentierung erzielt. Quelle: [LLS08]

über die Entstehung des Constellation Model gegeben sowie die Grundzüge des Modells vorgestellt. Für weiterführende Informationen vergleiche auch [GL11, Sze10].

Beim Constellation Model handelt es sich um ein generatives probabilistisches Modell, das sich zur Modellierung von Objekten bzw. Objektklassen eignet. Je nach Variante werden entweder die relativen Lagen der einzelnen Objektteile zueinander oder zu einer gemeinsamen Referenz beschrieben. Die Verbindungen lassen sich durch Sprungfedern veranschaulichen, die sich bis zu einem gewissen Grad stauchen oder in die Länge ziehen lassen.

Ursprünglich wurde das Constellation Model im Jahr 1995 von *Burl et al.* zur Gesichtsdetektion und -lokalisierung entwickelt. Durch speziell auf einzelne Gesichtsmerkmale trainierte Klassifikatoren werden Kandidaten erzeugt. Auf Basis einer Heuristik werden Hypothesen aufgestellt und die zugehörigen Kandidaten in den sog. *Shape-Space* transformiert, um so Invarianz gegenüber Rotation, Translation und Skalierung zu erzeugen. Mittels einer zuvor erlernten räumlichen Verbundverteilung wird die relative Lage der einzelnen Objektteile zueinander evaluiert. Der Ansatz ist robust gegenüber Clutter, partieller Verdeckung sowie Fehldetections und nicht erkannten Merkmalen [BLP95, BP96].

Eine Reihe von Erweiterungen wurden vorgeschlagen. So konnte die Erkennungsgenauigkeit gesteigert werden, indem die Detektorausgabe im Modell berücksichtigt wurde [BWP98]. *Leung et al.* führten Invarianz gegenüber affinen Transformationen ein [LBP98]. *Weber et al.* stellten das unüberwachte Lernen aussagekräftiger Objektteile vor, wobei das Aussehen der Merkmale als auch die relative Lage automatisch erlernt werden

konnten, ohne dabei auf gelabelte oder segmentierte Lerndaten zurückgreifen zu müssen. Der Ansatz ist lediglich invariant gegenüber Translationen [WWP00b, WWP00a].

Im weiteren Verlauf werden eine Reihe von Ausdrücken verwendet, die hier definiert werden sollen. So ist unter **Shape** die räumliche Lage der einzelnen Teile zu verstehen. **Appearance** bezieht sich auf die Detektorausgabe und beschreibt das Aussehen der einzelnen Teile.

Der von *Weber et al.* vorgestellte Ansatz wurde im Jahr 2003 durch *Fergus et al.* auf Invarianz gegenüber Skalierung erweitert. Auch wurde variierende Appearance berücksichtigt. Die von *Burl et al.* und *Weber et al.* vorgestellten Modelle hatten sich nur auf Variationen in der relativen Lage beschränkt und waren von Teilen mit einer festen Appearance ausgegangen. Der neue Ansatz ermöglichte das gleichzeitige Erlernen von Shape und Appearance. Dies erlaubt sowohl die Modellierung von Objektkategorien mit fester Geometrie und variablem Aussehen, als auch Modelle mit einem charakteristischen Aussehen, aber einer flexiblen Form [FPZ03].

Das Constellation Model erlaubt bezüglich der Erkennung weiche Entscheidungen, d.h. es fordert keine Auswahl einer Teilmenge von Merkmalen. Vielmehr werden eine Reihe von Hypothesen aufgestellt und mit dem Modell bewertet. Die resultierenden Wahrscheinlichkeiten werden marginalisiert. In der 2003 von *Fergus et al.* vorgeschlagenen Form sind die Beziehungen zwischen den einzelnen Teilen als vollständig verbundener Graph modelliert. Dieser Sachverhalt beschränkt das Modell auf eine relativ geringe Anzahl von 5-6 Teilen [FPZ03].

Aus diesem Grund wurde 2005 das Sternmodell vorgeschlagen, bei dem der vollständig verbundene Graph durch eine Sterntopologie ersetzt wurde. Hierdurch konnte die Komplexität des Inferenzalgorithmus von $O(N^P)$ auf $O(N^2P)$ gesenkt werden, was eine größere Anzahl an Teilen erlaubte. Das Sternmodell hat jedoch einen entscheidenden Nachteil. Während beim vollständig verbundenen Modell beliebige Teile verdeckt sein können, ist dies bei der Sterntopologie nur eingeschränkt der Fall. Die Lage aller Teile ist relativ zu einem Referenzteil modelliert. Ist dieses Teil verdeckt, so ist keine Erkennung mehr möglich [FPZ05].

$$P(\mathbf{X}, \mathbf{D}, \mathbf{S}) = \sum_{\mathbf{h}} \underbrace{P(\mathbf{D}|\mathbf{h})}_{\text{Appearance}} \underbrace{P(\mathbf{X}|\mathbf{S}, \mathbf{h})}_{\text{Shape}} \underbrace{P(\mathbf{S}|\mathbf{h})}_{\text{Rel. Skalierung}} \underbrace{P(\mathbf{h})}_{\text{Sonstiges}} \quad (4.1)$$

Gleichung 4.1 beschreibt das Kernelement des Constellation Model. Die Vektoren \mathbf{X} , \mathbf{D} und \mathbf{S} beschreiben die Shape, Appearance und Skalierung aller extrahierten Merkmale. Die Hypothese $\mathbf{h} \in \mathbf{H}$ ist für die Zuordnung einer Teilmenge der gefundenen Merkmale zum Modell verantwortlich. Die Wahrscheinlichkeit $P(\mathbf{X}, \mathbf{D}, \mathbf{S})$ für das vorliegen eines Objektes faktorisiert sich in vier Terme, die (von links nach rechts) die Wahrscheinlichkeiten für Appearance, Shape, relative Skalierung und sonstige Faktoren liefern. Da hier ein Ansatz aus der bayesschen Entscheidungstheorie zur Entscheidung zwischen zwei Klassen herangezogen wird (vgl. hier auch beispielsweise [DHS00]), umfasst jeder Term ein Vorder- und Hintergrundmodell, die gegeneinander abgewogen werden [FPZ05].

4.4 Probabilistische Posenmodellierung

Die Modellierung von Unsicherheiten über Posen setzt eine geeignete Repräsentation für diese voraus. Für die Position an sich ist dies recht simpel, da der zu Grunde liegende Raum als realwertig, kontinuierlich und unendlich angenommen werden kann.

Problematisch ist jedoch die Orientierung, da hier ein zyklischer Raum vorliegt. Dies kann bei ungünstiger Modellierung zu Doppeldeutigkeiten, Einschränkungen in der Aussagekraft und auslaufender Wahrscheinlichkeitsmasse führen. In diesem Abschnitt wird der Stand der Forschung in diesem Bereich vorgestellt.

4.4.1 Repräsentation von Posen

Die Literatur bietet eine ganze Reihe von Ansätzen für die Repräsentation von Orientierungen, wobei zu den den populärsten Rotations-Matrizen, Euler-Winkel, Rodrigues Vektoren und Einheitsquaternonen zählen. Im Fall von Rotations-Matrizen wird die Orientierung zwischen zwei Koordinatensystemen in Form einer Matrix ausgedrückt, welche aus den Basisvektoren des zweiten Koordinatensystems aufgebaut ist. Neun Parameter müssen gesetzt werden (vgl. hier auch beispielsweise [SK08, Kapitel 1.2]).

Euler-Winkel drücken die Rotation um die Achse eines Koordinatensystems aus. Verkettet man mehrere Rotationen, so wird um ein sich bewegendes Koordinatensystem rotiert; also jeweils um das aus Koordinatensystem aus der vorherigen Rotation. Es wird eine Konvention für die Rotationsachsen benötigt, gängig sind **X-Y-Z**, **Z-Y-X** und **Z-Y-Z**. Zwar ist pro Rotation nur ein Parameter erforderlich, jedoch kann dieselbe Rotation durch verschiedene Parameter ausgedrückt werden. Außerdem besteht die Gefahr eines Gimbal-Lock (vgl. hier auch beispielsweise [SK08, Kapitel 1.2]).

Eine weitere Möglichkeit stellen Einheitsquaternonen dar. Sie umgehen das Problem des Gimbal-Lock, benötigt hierfür jedoch vier Parameter. *Stuelpnagel* legt nahe, dass Einheitsquaternonen eine ausreichende Repräsentation mit wenigen Parametern von Rotationen in 3D sind [Stu64]. In der Computergrafik werden meist Quaternonen verwendet, in der Robotik Rodrigues Vektoren [FLH13]. Letztere kodieren die Rotationsachse in Form eines Vektor, dessen Länge den Winkel kodiert. Auch diese Repräsentation ist mehrdeutig, zusätzlich kann eine Rotation um Null Grad nicht dargestellt werden (vgl. hier auch beispielsweise [SK08]).

4.4.2 Methoden zur Repräsentation von Unsicherheit

Eine Repräsentation der Pose (bzw. Orientierung) kann in dieser Arbeit nur dann eingesetzt werden, wenn sich Unsicherheiten darüber modellieren lassen. Die meisten Ansätze haben unter einer Reihe von Problemen zu leiden. So können beispielsweise Singularitäten auftreten oder Dualitäten dazu führen, dass ein Winkel durch mehrere Werte (z.B. θ und $2\pi - \theta$) beschrieben werden kann. Ebenfalls problematisch ist auslaufende Wahrscheinlichkeitsmasse, also Teile der Verteilung, die außerhalb des Intervalls $0 < r < 2\pi$ liegen [EGS⁺¹²].

Feiten et al. stellen die folgenden Anforderungen an eine Beschreibung: **Expressive Power, Efficiency, Information Fusion, Information Propagation** [FLH13]. Für die vorliegende Arbeit sind nur die ersten beiden Kriterien relevant. Auch werden Ansätze vorgezogen, die auf Gauss-Mischverteilungen arbeiten. Im folgenden wird die Literatur vorgestellt, die den hier genannten Kriterien entspricht.

Eidenberger et al. wählt - wohl der Einfachheit halber, eine Begründung wird nicht gegeben - Rodrigues Vektoren für die Beschreibung der Orientierung. Unsicherheiten über die Pose werden mit Gauss-Mischverteilungen modelliert. Zwar liegt eine gemeinsame Kovarianzmatrix vor, jedoch wird eine Korrelation zwischen Position und Orientierung bewusst unterbunden. Verteilungen mit großer Varianz werden nur mehrere spitze Verteilungen approximiert, um auslaufende Wahrscheinlichkeitsmasse zu verhindern [Eid10, EGS⁺12].

Glover et al. setzt die Bingham-Verteilung ein, um Unsicherheiten direkt auf der Hypersphäre des Einheitsquaternions zu repräsentieren. Der darauf basierende *Quaternion Bingham Filter* hat einen geringeren Tracking-Fehler als der normale Extended Kalman Filter, speziell wenn der Zustand hochdynamisch ist [GK13]. Der Filter wurde erfolgreich dazu eingesetzt, die Orientierung von Tischtennis-Bällen zu prädizieren [GK14]. Ein ähnlicher Ansatz wurde zeitgleich von *Kurz et al.* entwickelt [KGJH13]. *Marins et al.* nutzen einen Extended Kalman Filter auf Basis von Einheitsquaternonen zur Echtzeit-Bestimmung von Festkörper-Orientierungen. Dieser wird auf einen MARG (Magnetic, Angular Rate, and Gravity) Sensor angewendet [MYB⁺01].

Choe et al. stellt die Verteilungen durch auf einen Tangentenraum projizierte Gauss-Verteilungen dar, wobei zur Repräsentation Quaternonen eingesetzt werden [Cho06]. *Goddard et al.* nutzen Duale Quaternonen zur Objektverfolgung. Der Ansatz ist echtzeitfähig und erlaubt eine Korrelation zwischen Rotation und Translation, arbeitet aber leider nur mit unimodalen Gauss-Verteilungen [God97, GA98].

Der von *Feiten et al.* vorgestellte Ansatz nutzt sowohl duale Quaternonen als auch den Tangentenraum. Der Tangentenpunkt liegt auf der 3D-Sphäre, welche das Einheitsquaternion beschreibt. Die Bijektion zwischen Einheitssphäre und der Tangentenebene ist durch die Zentralprojektion gegeben. Die Beschreibung von Unsicherheiten basiert auf Gauss-Mischverteilungen, daher wird diese Klasse von Verteilungen von den Autoren auch als **Mixtures of Projected Gaussians** bezeichnet. Die Verwendung von Gauss-Mischverteilungen hat den Vorteil, dass viele dafür definierte Operationen auch auf diese neue Klasse von Verteilung angewendet werden können [FAEG09, LF12, FLH13].

5. Konzeption des Szenenmodells

Dieses Kapitel beschreibt das Konzept, dass zur Lösung der vorliegenden Problemstellung entwickelt wurde. In Abschnitt 5.1 werden zunächst die Gedankengänge vorgestellt, die den hier vorgestellten Lösungsweg motivieren. Das Grundprinzip der entwickelten Szenenmodellierung wird vorgestellt. Drauf aufbauend erfolgt in Abschnitt 5.2 die Abgrenzung zu anderen Arbeiten.

Im Anschluss werden das Szenenmodell und seine einzelnen Bestandteile ausführlich besprochen. Abschnitt 5.3 stellt die Schnittstellen des Modells vor. Außerdem werden die darin verwendeten Parameter definiert. Das eigentliche Modell besteht aus zwei Teilen und wird in den Abschnitten 5.4 und 5.5 beschrieben. Ausgehend von einer Verbundverteilung über die vorliegenden Beobachtungen wird schrittweise ein generatives Modell der Szene hergeleitet. Darauf aufbauend wird das Object Constellation Model (OCM) - der zentrale Baustein der Szenenerkennung - abgeleitet. Abschnitt 5.6 fasst beide Bestandteile zusammen und zeigt die komplette Gleichung des Modells.

5.1 Motivation des Ansatzes

In diesem Abschnitt werden die Überlegungen erläutert, die zur Konzeption des Szenenmodells in der hier beschriebenen Form geführt haben. Das Modell muss in Rahmen probabilistischer Planung einsetzbar sein, es muss also ebenfalls probabilistischer Natur sein. Wie im vorherigen Kapitel dargelegt lassen sich Szenen anhand von Objektrelationen beschreiben. Diese müssen bis zu einem (durch Lerndaten) definierbaren Grad verformbar sein, um dynamische Szenen beschreiben zu können. Weiterhin soll die Repräsentation möglichst kompakt sein, was für einen parametrischen Ansatz spricht, bei dem die Lerndaten durch ein parametrierbares Modell beschrieben werden. Die hier beschriebenen Anforderungen legen die Verwendung eines *Parts and Structure*-Modells nahe. Es wurde das Constellation Model gewählt, da es alle hier beschriebenen Anforderungen erfüllt und leicht angepasst werden kann.

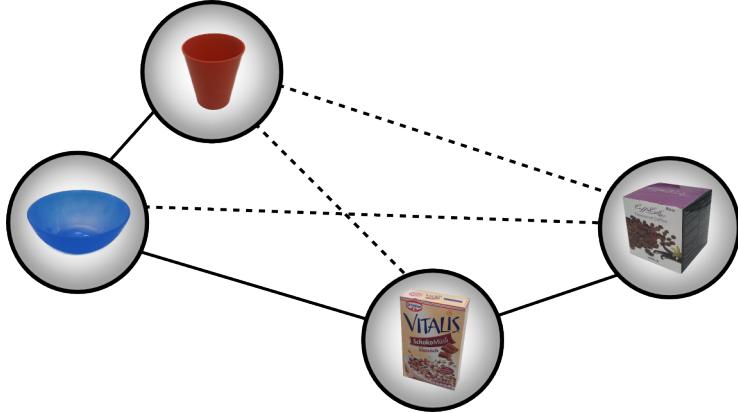


Abbildung 5.1: Die vorliegende Beispielszene kann durch wenige Relationen beschrieben werden, die als durchgezogene Linien dargestellt sind. Der Becher wird oft rechts oberhalb der Schale beobachtet. In einem festen Abstand dazu steht das Müsli, neben dem wiederum der Kaffee platziert ist.

Als nächstes soll die grundsätzliche Frage geklärt werden, welche Objektrelationen zu modellieren sind. Deren Anzahl ist näherungsweise quadratisch zu den in der Szene vorhandenen Objekten, dementsprechend nimmt auch der Aufwand für die Auswertung zu. Das vollverbundene Constellation Model nutzt einen solchen Ansatz und modelliert alle Relationen. Wie in Abschnitt 4.3 erläutert hat dies die Konsequenz, dass der hierfür notwendige Inferenz-Algorithmus über eine hohe Komplexität verfügt und so die maximale Anzahl an Teilen (und damit die Größe der zu modellierenden Szene) in der Praxis begrenzt ist.

Viele in einer Szene enthaltenen Objektrelationen sind nicht zwangsläufig notwendig, um die Szene eindeutig zu beschreiben und Fehlerkennungen zu vermeiden (vergleiche auch [Mei13]). Abbildung 5.1 illustriert diesen Sachverhalt. Ein Weg zur Auswahl der aussagekräftigen Relationen ist die Nutzung von Heuristiken. Ein solches Vorgehen wird von Meißner et al. praktiziert und im Rahmen dieser Arbeit aufgegriffen. Die Beschränkung auf relevante Relationen ist wichtig, um den Aufwand für die Auswertung so niedrig wie möglich zu halten.

Die Alternative zum vollverbundenen Constellation Model ist das sternförmige Constellation Model, bei dem nur die Relationen zwischen einem Referenzteil und den anderen Teilen berücksichtigt werden. Die Mächtigkeit einer Sternstruktur in Hinblick auf die Modellierung von Relationen ist jedoch begrenzt. Außerdem neigt ein hierauf basierendes Modell zu Fehlerkennungen der Szene (siehe [Mei13]). Aus diesem Grund wurde im Rahmen dieser Arbeit eine Erweiterung des sternförmigen Constellation Model auf eine allgemeinere Baumstruktur vorgenommen, welche in Abschnitt 5.5 ausführlich erläutert werden. Der Inferenz-Algorithmus bleibt im Ansatz unverändert, wodurch nach wie vor Invarianz gegenüber Translation, Clutter und fehlenden Objekten gegeben ist. Weiterhin erlaubt die Art und Weise der Umsetzung Invarianz gegenüber Rotation.

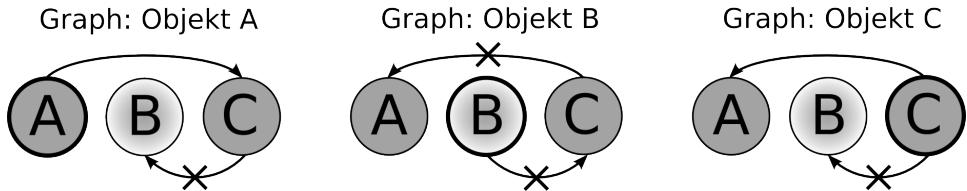


Abbildung 5.2: Relationsgraphen für die Objekte A-C. Das Objekt B wurde nicht beobachtet, daher sind die damit verbundenen Relationen und nachfolgenden Teilbäume nicht auswertbar. Dies wird durch ein Kreuz kenntlich gemacht.

Wie schon das sternförmige Constellation Model wird ein Referenzobjekt vorausgesetzt, an dem der Graph mit den Relationen festgemacht wird. Ein solches *Intermediate Object* schränkt jedoch stark ein. Es muss immer vorhanden sein, damit die Relationen zwischen den Objekten ausgewertet werden können. Ist die Lage des Referenzobjekts unbekannt, so kann sie meist nur mit einer ungezielte Suche bestimmt werden. Je größer der Suchraum, desto geringer ist auch die Chance, dass das Referenzobjekt mit vertretbarem Aufwand gefunden werden kann. Eine gängige Alternative - die Verwendung eines festen Referenzpunkts - scheidet aus, da dieses Vorgehen mit den geforderten Invarianzeigenschaften hinsichtlich Translation und Rotation in Konflikt steht.

Das Problem lässt sich lösen, indem pro Objekt in der Szene ein Constellation Model definiert wird. Ausgehend von dem besagten Objekt wird die globale Nachbarschaft - also die komplette Szene - modelliert. Hierzu werden die baumförmig organisierten Relationen herangezogen, die einmalig mit Hilfe der Heuristiken erzeugt und dann für das entsprechende Referenzobjekt angepasst werden. Es muss also kein Aufwand in die Suche nach einem Referenzobjekt investiert werden, da die Szene immer aus den bereits beobachteten Objekten abgeleitet werden kann.

Ausgehend von der Annahme, dass die Heuristiken alle relevanten Relationen erfasst haben, wird durch dieses Vorgehen die maximal mögliche Menge an räumlichen Informationen genutzt. Dies bedeutet, dass alle prüfbaren Relationen auch überprüft werden. Da diese baumförmig organisiert sind können nur die Teilbäume geprüft werden, für deren Elternknoten (bis hin zum Referenzobjekt) die entsprechenden Objekte beobachtet wurden. Dieser Sachverhalt soll anhand eines Beispiels demonstriert werden, das in Abbildung 5.2 gezeigt ist. Es sei eine Szene mit drei Objekten gegeben, die mit den Buchstaben A-C bezeichnet werden. Pro Objekt existiert ein Graph, der die komplette Szene beschreibt. Werden nun im Rahmen der Szenenerkennung die Objekte A und C beobachtet, so können nur die Relationen bewertet werden, die nicht mit Objekt B verbunden sind oder es in der Linie ihrer Vorfahren haben.

Aus den einzelnen Constellation Model wird das Modell ausgewählt, welches die vorliegenden Objektevidenzen am besten beschreibt. Dieses Vorgehen ergibt sich aus den limitierten Möglichkeiten, die der bayessche Formalismus mit sich bringt und wird in Abschnitt 5.4 ausführlich begründet.

5.2 Abgrenzung zu anderen Arbeiten

Nachdem nun der Ansatz in seinen Grundzügen vorgestellt wurde sollen nun die Unterschiede zu den anderen, im Abschnitt 4.2.3 vorgestellten Arbeiten des Forschungsbereichs herausgearbeitet werden. Für den direkten Vergleich eignen sich nur die Arbeiten von *Ranganathan et al.* und *Meißner et al.*, die anderen Arbeiten unterscheiden sich zu sehr von der vorliegenden Aufgabe. *Kumar et al.* beschäftigen sich nicht mit Relationen in 3D. *Southey et al.* setzten nur relativ grobe, von Hand erstellte Beschreibungen für relative Lagen ein, die für die Beschreibung komplexer Zusammenhänge und dynamischer Szenen nicht ausreichen. *Joho et al.* fokussieren sich nicht auf das Bestimmen eines Szenenlabels. Im weiteren Verlauf werden diejenigen Punkte bei *Ranganathan et al.* und *Meißner et al.* angesprochen, die im Rahmen dieser Arbeit aufgegriffen und weiterentwickelt werden.

Ranganathan et al. beschäftigt sich nur am Rand mit der Inferenz der Szene und ist eher auf die Bestimmung der vorliegenden Objekte fokussiert (siehe [RD08]). Das dort verwendete probabilistische Gerüst scheint nur eine unterstützende Rolle zur Beschreibung des Vorgehens zu spielen, unter anderem da mehrere Verteilungen dort eher Abstandsmaßen entsprechen.

So ist etwa die Objektwahrscheinlichkeit $P(O|L, A, S, Z)$ definiert als die Differenz zwischen zwei diskreten Verteilungen, bei denen es sich um die Typen der vorliegenden Objekte und die Vorhersage der Objekttypen für eine gegebene Szene handelt. Ähnliches gilt für die Berechnung der Orts wahrscheinlichkeit $P(T|L, O, A, S, Z)$, welche durch die Minimierung des Abstands zwischen den beobachteten Objekten und den korrespondierenden Objekten der Szene modelliert wird.

Weiterhin werden Szenen relativ zur Roboterplattform gelernt. Zur Erkennung der Szene muss also deren Position innerhalb der Szene bestimmt werden. Formal wird dies durch eine Marginalisierung dargestellt, in der Praxis kommt jedoch ein EM-Algorithmus zum Einsatz, der iterativ die Position der Plattform in der Szene bestimmt. Das Ergebnis fließt in die Orts wahrscheinlichkeit mit ein.

Die auf diesem Weg erzeugte Transformation gewährt nur Invarianz gegenüber Translation. Invarianz gegenüber Rotation ist aus theoretischer Sicht ebenfalls möglich, jedoch wird keine Aussage über die tatsächliche Umsetzung getroffen. Fakt ist jedoch, dass keine komplette Objektpose berücksichtigt wird, lediglich die Position der einzelnen Objekte fließt in die Szenenbestimmung mit ein.

Ranganathan et al. nutzen das sternförmige Constellation Model, um die Relationen zwischen Roboterplattform und Objekten zu modellieren. Relationen zwischen Objekten werden nicht modelliert, wodurch es bei komplexeren Szenen zu Fehlerkennungen kommen kann (vgl. hier [Mei13]). Auch werden nur Gauss-Verteilungen für die Modellierung der räumlichen Lage eingesetzt, wodurch dynamische Szenen nicht beschrieben werden können.

Die Veröffentlichungen von *Meißner et al.* bilden den Ansatzpunkt für die vorliegende Arbeit (siehe [Mei13, Mei14]). Es wird eine Hierarchie von Implicit Shape Models zur Szenenerkennung und Objektsuche eingesetzt. Die Erkennungsresultate der einzelnen ISMs

werden durch den Baum propagierte und zusammengefasst. Das Resultat ist jedoch ein Konfidenzmaß und kann daher nicht für die probabilistische Planung verwendet werden.

Zusatzinformationen wie die Auftrittswahrscheinlichkeit der Objekte und Szenen werden nicht berücksichtigt, auch wenn sie sich vom theoretischen Standpunkt aus nachrüsten lassen. Deren Wert ist nicht zu unterschätzen, da hierdurch nicht nur die Aussagekraft gesteigert wird. Auch können mehrdeutige Szenen konstruiert werden, die sich nur durch besagtes Wissen erkennen lassen.

Ein Problem des Implicit Shape Model ist es, dass nicht alle Szenenhypothesen verfolgt werden können. Dies lässt sich durch eine Design-Entscheidungen begründen. Da es sich beim ISM um eine generalisierte Variante der Hough-Transformation handelt, stimmen alle Teile über den Referenzpunkt des Modells ab. Hierzu wird der Abstimmungsraum diskretisiert, dann die Abstimmung und im Anschluss eine Maximasuche durchgeführt, welche die gesuchte Position liefert. Um den Abstimmungsraum beherrschbar zu halten wurde nur die Position des Referenzpunkts berücksichtigt, die Orientierung jedoch nicht. Liegen also mehrere Hypothesen mit gleicher Position, aber unterschiedlicher Orientierung des Referenzpunkts vor, so wird nur jeweils eine der Hypothesen verfolgt.

Als letzter Punkt soll die Modellkomplexität angesprochen werden. Diese steigt linear zur Anzahl der Beobachtungen, die im Training gemacht werden. Für komplexe Szenen, die mehrere Stunden andauern, sammeln sich daher große Datenmengen an. Eine Kompression oder Vereinfachung des Modells wird nicht vorgenommen, da das ISM ein nicht-parametrisches Modell ist.

Die vorliegende Arbeit unterscheidet sich gegenüber *Ranganathan et al.* und *Meißner et al.* dadurch, dass ein in sich geschlossenes, probabilistisches Modell eingesetzt wird. Dies kann auch zur probabilistischen Planung herangezogen werden. Im Gegensatz zu *Ranganathan et al.* erlaubt es die Modellierung dynamischer Szenen mit komplexen Beziehungen. Außerdem wird die komplette Pose der beteiligten Objekte berücksichtigt und auf diesem Weg Invarianz gegenüber Rotation erzeugt. Weitere Verbesserungen in Bezug auf *Meißner et al.* sind die Nutzung von Zusatzinformationen, im vorliegenden Fall die Auftrittswahrscheinlichkeit von Objekten. Weiterhin wird statt einem diskriminativen ein generatives Modell verwendet. So wird der im Implicit Shape Model vorliegende Fall ausgeschlossen, dass bestimmte Hypothesen nicht weiter verfolgt werden. Statt einem nicht-parametrischen wird ein parametrisches Modell gewählt, damit dieses nicht mehr linear, sondern nur noch mit einem von der Komplexität der Szene abhängigen Faktor skaliert.

5.3 Modellschnittstelle und -parameter

Bevor das Modell vorgestellt werden kann, müssen zunächst einige Vorarbeiten geleistet werden. In diese Abschnitt wird zunächst die Terminologie eingeführt, die sich im Rahmen der Arbeit entwickelt hat. Weiterhin werden die einzelnen Variablen des Modells sowie die Ein- und Ausgaben vorgestellt und erläutert.

Statt dem allgemeinen Begriff *Beobachtung* soll ein detektiertes Objekt von nun an als **Evidenz** oder **Objektevidenz** bezeichnet werden. Formal handelt es sich hierbei um Hypothesen, welche den Typ und die Pose eines Objektes beschreiben. Diese Hypothesen werden von den Objektdetektoren erzeugt, welche anhand von Beobachtungen der Umwelt auf die vorhandenen Objekte schließen. Ein Überblick über gängige Ansätze zur Objektdetektion wurde in Abschnitt 4.1 gegeben. Eine Objektevidenz wird durch das Tupel (a, \mathbf{x}) beschrieben¹. Auf Grund der probabilistischen Natur des Modells sind die Variablen als Zufallsvariablen zu interpretieren. Der Objekttyp a wird aus einer endlichen, diskreten Menge Ω_a gezogen, welche alle (aus Sicht des Systems) existierenden Objekttypen umfasst.

Die Pose wird durch $\mathbf{x} = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7)^T$ beschrieben, wobei die ersten drei Parameter die Position in kartesischen Koordinaten und die anderen vier die Orientierung in Form eines Einheitsquaternions darstellen. Wie in Abschnitt 4.4 besprochen existieren viele Ansätze zur probabilistischen Modellierung von Orientierungen bis hin zu kompletten Posen. Da dieser Bereich der Forschung noch relativ jung zu sein scheint mangelt es in der Literatur an Vergleichen zwischen den unterschiedlichen Verfahren. Daher ist schwer zu bestimmen, wie groß die qualitativen Unterschiede zwischen einfachen Standardansätzen und den komplexeren, neueren Verfahren sind.

Aus diesem Grund wurde beschlossen, mit einer möglichst einfache Repräsentation zu beginnen und diese bei Bedarf auszutauschen. Es wurde das Einheitsquaternion gewählt, da hier keine Singularitäten wie das Gimbal-Lock existieren und auch auslaufende Wahrscheinlichkeitsmasse kein Problem darstellt. Diese Repräsentation hat sich - wie die Literatur schon hat vermuten lassen - in verschiedenen Tests bewährt, weshalb sie beibehalten wurde.

Die Vektoren $\mathbf{A} = (a_i)_{i=1}^N$ und $\mathbf{X} = (x_i)_{i=1}^N$ beschreiben die Objekttypen und -posen aller N gefundenen Objektevidenzen. Der Zugriff auf die Vektoren erfolgt über die Funktionen $A(\cdot)$ und $X(\cdot)$, welche das Element für den übergebenen Index zurückgeben.

Wie oben erläutert handelt es sich beim Constellation Model um ein Teile- und Strukturmodell, wobei das Ganze durch eine Reihe von Teilen beschrieben wird. Deren Anzahl wird durch P beschrieben, wobei es sich hierbei nicht um eine Zufallsvariable, sondern eine Hilfsvariable handelt.

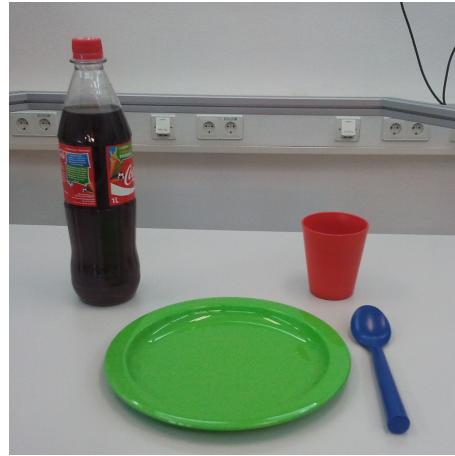
Die diskrete Zufallsvariable S bezeichnet die vorliegende Szene. Der zugehörige Ergebnisraum Ω_S umfasst die Menge aller erlernten Szenen. S kann als das Ergebnis betrachtet werden, das vom probabilistischen Modell inferiert wird.

Das Modell beinhaltet noch eine weitere Zufallsvariable. Bevor diese näher erläutert wird muss zunächst der Begriff des sogenannten *Szenenobjekts* definiert werden. Hierbei handelt es sich um ein Konzept, dass sowohl Informationen über ein Objekt als auch die Szene, zu der besagtes Objekt gehört, umfasst. Als Beispiel soll ein Teller herangezogen werden, wie er auf jedem Esstisch zu finden ist. Der Teller wird von seinem Besitzer

¹Die hier verwendete Notation ist an die Arbeiten von *Fergus et al.* angelehnt (vergleiche [FPZ03]).



(a) Der Teller im Kontext einer Frühstücksszene.



(b) Derselbe Teller im Rahmen eines Mittagessen-Szenario.

Abbildung 5.3: Ein Szenenobjekt bezeichnet ein Objekt im Kontext einer Szene. Dasselbe Objekt kann Teil mehrerer verschiedener Szenenobjekte sein.

sowohl zum Mittagessen, als auch Abendessen genutzt. Es liegen also zwei unterschiedliche Szenen vor, aber derselbe Teller. Um nun anzugeben, dass der Teller im Kontext des Mittagessens Anwendung findet, wird er als Szenenobjekt *Mittagessen-Teller* bezeichnet. Natürlich liegt hier ein Unterschied zum Szenenobjekt *Abendessen-Teller* vor, auch wenn beide Objekte identisch sind. Das Szenenobjekt wird durch die Zufallsvariable O bezeichnet, ein Vektor mit mehreren Szenenobjekten dementsprechend als \mathbf{O} . Der zugehörige Ergebnisraum ist definiert als $\Omega_O = \Omega_S \times \Omega_a$.

5.4 Szenenmodell

Wie bereits erwähnt besteht das Modell aus zwei Teilen, die zum besseren Verständnis in Reihenfolge ihrer Herleitung vorgestellt werden. In diesem Abschnitt wird der erste Teil - das Szenenmodell - vorgestellt. Ausgehend von der Verbundverteilung über die Beobachtungen, die Szene und die darin vorhandenen Szenenobjekte werden das Modell hergeleitet und die einzelnen Verteilungen erläutert. Da das System auch mit unbekannten Szenen konfrontiert werden kann wird abschließend eine für diesen Fall vorgesehene Rückweisungsklasse definiert.

5.4.1 Herleitung des Szenenmodells

Das Programmieren durch Vormachen setzt voraus, dass die Maschine aus möglichst wenig Beispielen ein aussagekräftiges Modell erlernen kann. Da die vorliegende Arbeit als Teil eines solchen Systems konzipiert ist, gilt dieselbe Anforderung. Daher bietet sich ein generatives Model. Wie in Abschnitt 3.3 erläutert lassen sich hierdurch die

Gesetzmäßigkeiten einer Szene beschreiben, wodurch über ungesehene Trainingsdaten interpoliert werden kann.

Das generative Modell beschreibt die zu erwartende Beobachtung für einen gegebenen Weltzustand. Hier von Interesse ist jedoch der gegenteilige Sachverhalt, also der Weltzustand gegeben die Beobachtungen. Unter letzterem sind die gefundenen Objektevidenzen zu verstehen. Wie im vorherigen Abschnitt beschrieben werden diese durch die Vektoren \mathbf{X}, \mathbf{A} dargestellt. Der Weltzustand ist die Szene S . Gleichung 5.1 zeigt, wie durch Einsatz des Bayes-Theorem die Szenenwahrscheinlichkeit $P(S|\mathbf{A}, \mathbf{X})$ aus dem generativen Teil $P(\mathbf{A}, \mathbf{X}|S)$ hergeleitet werden kann.

$$P(S|\mathbf{A}, \mathbf{X}) = \frac{P(\mathbf{A}, \mathbf{X}|S)P(S)}{P(\mathbf{A}, \mathbf{X})} \quad (5.1)$$

Die Verteilung über den diskreten Weltzustand ist nur von untergeordnetem Interesse, vielmehr interessiert die wahrscheinlichste Szene \hat{S} . Gleichung 5.2 bestimmt die in Hinblick auf die gegebenen Beobachtungen plausibelste Szene, wobei hier auf Gleichung 5.1 aufgebaut wird. Der Normalisierungsterm ist für die Bestimmung des Maximums irrelevant und entfällt daher. Die Auftrittswahrscheinlichkeit der Szene $P(S)$ ist durch eine Multinomialverteilung realisiert.

$$\hat{S} = \arg \max_S P(\mathbf{A}, \mathbf{X}|S)P(S) \quad (5.2)$$

In Abschnitt 5.1 wurde die mit der Auswahl des Referenzpunkts verbundene Problematik beschrieben. Es wurde entschieden, ausgehend von jedem Objekt in einer Szene ein globales Szenenmodell zu erlernen. Für die Fusion der einzelnen Modelle standen die Möglichkeiten *Faktorisierung*, *Marginalisierung* und *Maximum* zur Verfügung.

Bei der Faktorisierung wird angenommen, dass die Szenenobjekten voneinander unabhängig sind - eine in der Literatur oft getroffene vereinfachende Annahme, bei der der Nutzen den damit verbundenen Schaden überwiegt - und deren Verbundverteilung durch deren Multiplikation bestimmt werden kann. Hieraus ergibt sich jedoch ein ungewünschtes Systemverhalten. Wird ein Szenenobjekt z.B. auf Grund von Rauschen falsch erkannt, so senkt dies die Wahrscheinlichkeit der kompletten Szene.

Eine Alternative bietet die Marginalisierung, bei der die Szenenobjekte addiert werden. Dies ist jedoch aus formaler Sicht nicht machbar, da alle Szenenobjekte die Realisierungen einer einzelnen Zufallsvariable sind und daher nicht gleichzeitig nebeneinander existieren können. Genau dies ist jedoch der Fall. Die Wahl des besten Modells hat einen ähnlichen Effekt wie die Marginalisierung, jedoch ohne die damit verbundenen formalen Probleme und wurde daher zur Lösung des Problems herangezogen.

Dies wird in Gleichung 5.3 gezeigt. Die Verbundverteilung $P(\mathbf{A}, \mathbf{X}|S)$ wird um die für ein Szenenobjekt stehende Variable O erweitert und das Gesetz der bedingten Wahrscheinlichkeiten angewandt. Die daraus resultierende Verteilung $P(\mathbf{A}, \mathbf{X}|O)$ beschreibt

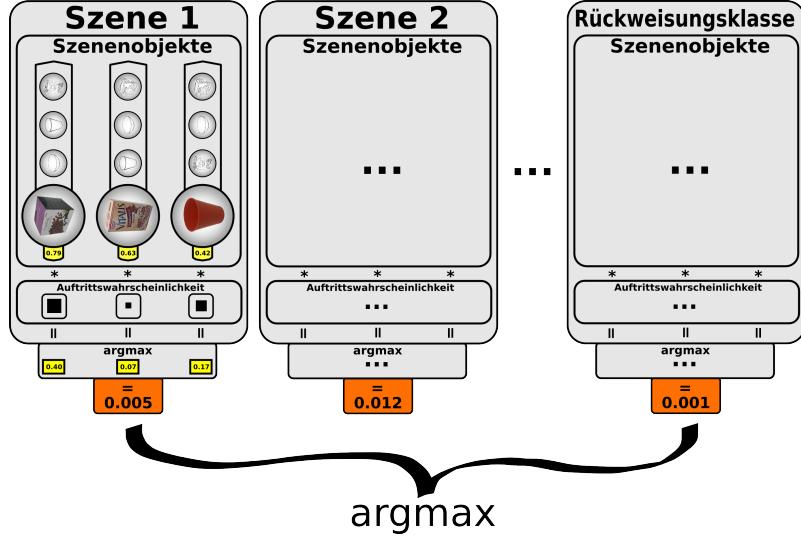


Abbildung 5.4: Eine grafische Darstellung des Szenenmodells. Es wird die Wahrscheinlichkeit einer Reihe von Szenen inklusive der Rückweisungsklasse berechnet. Jede Szene basiert auf einer Reihe von Szenenobjekten, wobei das mit der höchsten gewichteten Wahrscheinlichkeit die Wahrscheinlichkeit der Szene ergibt.

ein einzelnes Szenenobjekt und wird in Abschnitt 5.5 ausführlich erläutert. Die Variable S entfällt, da die Beobachtungen \mathbf{A} und \mathbf{X} unabhängig zur Szene S gegeben das Szenenobjekt O sind. Letzteres enthält also Szenenwissen und kann die Beobachtungen auch ohne zusätzliche Informationen über die Szene erklären.

$$\begin{aligned}
 P(\mathbf{A}, \mathbf{X}|S) &= \arg \max_O P(\mathbf{A}, \mathbf{X}, O|S) \\
 &= \arg \max_O P(\mathbf{A}, \mathbf{X}|O, S)P(O|S) \\
 &= \arg \max_O \underbrace{P(\mathbf{A}, \mathbf{X}|O)}_{\text{Szenenobjekt}} \underbrace{P(O|S)}_{\substack{\text{Auftrittswahrsch.} \\ \text{des Szenenobjekts}}}
 \end{aligned} \tag{5.3}$$

Die Verteilung $P(O|S)$ beschreibt die Auftrittswahrscheinlichkeit für das Szenenobjekt. Dies ist vergleichbar mit der Auftrittswahrscheinlichkeit für das Objekt, dass die Rolle der Referenz übernimmt. Da innerhalb eines Szenenobjekts die Wahrscheinlichkeit für das Auftreten der Referenz immer beim sicheren Ereignis liegen muss - sonst wäre es nicht da und könnte demzufolge auch nicht ausgewertet werden - bietet es sich an, das Auftreten an dieser Stelle zu behandeln. Umgesetzt wird $P(O|S)$ durch eine Bernoulli-Verteilung.

Abbildung 5.4 stellt das hier hergeleitete Szenenmodell grafisch dar. Die im folgenden erläuterte Rückweisungsklasse ist ebenfalls darin enthalten.

5.4.2 Rückweisungsklasse

Das System kann mit Evidenzen konfrontiert werden, die mit keiner erlernten Szene in Zusammenhang gebrachten werden können. Für diesen Fall wird eine Rückweisungsklasse vorausgesetzt, welche immer dann vorzuliegen hat, wenn keine der anderen Szenen vorhanden ist. Als Bezeichnung hierfür wird S_0 gewählt, trainierte Szenen werden mit einem Index größer als Null bezeichnet.

Die Rückweisungsklasse modelliert den Hintergrund. Alle Beobachtungen sind zufällig, alle ermittelten Objekte, deren relative Posen und Auftrittswahrscheinlichkeiten werden als keinem Muster entsprechend angenommen. Dieses Unwissen lässt sich durch eine Gleichverteilung modellieren. Die Berechnung erfolgt über Gleichung 5.3, jedoch mit einer speziellen Variante von $P(\mathbf{A}, \mathbf{X}|O)$, bei der alle darin enthaltenen Informationen gleichverteilt sind. Der Term wird ein einziges mal berechnet und dann als Ergebnis angenommen, aus offensichtlichen Gründen entfällt die Bestimmung des wahrscheinlichsten Szenenobjekts.

Die Rückweisungsklasse wird im Rahmen der Szenenerkennung wie eine herkömmliche Szene behandelt und mit den Modellen der anderen Szenen abgeglichen.

5.5 Object Constellation Model

Dieser Abschnitt ist der Modellierung eines einzelnen Szenenobjekts gewidmet, dass durch den zuvor eingeführten Term $P(\mathbf{A}, \mathbf{X}|O)$ beschrieben wird. Das Modell ist generativer Natur und beschreibt die zu erwartenden Beobachtungen für ein gegebenes Szenenobjekt. Da es auf dem in Abschnitt 4.3 beschriebenen *Constellation Model* aufgebaut wurde die Bezeichnung Object Constellation Model (OCM) gewählt.

5.5.1 Unterschiede zum Constellation Model

Das Object Constellation Model basiert auf dem sternförmigen Constellation Model, das bereits im letzten Kapitel vorgestellt wurde. In dieser Variante des Constellation Models werden die Positionen aller Teile relativ zu einem gemeinsamen Referenzteil modelliert. Wie in Abschnitt 5.1 dargelegt wurde ist Sternform nicht ausreichend, um alle benötigten Relationen modellieren zu können.

Die Beschreibungsfähigkeit des Modells lässt sich erhöhen, indem mehrere Sterne hierarchisch angeordnet werden. Die Blattknoten eines Sterns fungieren als Referenz für den Stern einer tieferen Hierarchieebene. Der daraus resultierende Graph hat eine baumförmige Struktur, die in ihrer Beschreibungsfähigkeit mächtiger ist als die reine Sternform, allerdings weniger mächtig als das vollverbundene Constellation Model. Der Graph kann in Fällen, in denen sich die relevanten Relationen nicht durch eine Sternform beschreiben lassen, nur eine Approximation des tatsächlichen Relationsgraphen liefern. Jedoch kann der einfache Inferenzalgorithmus der Sternform beibehalten werden, was zu einem klaren Geschwindigkeitsvorteil gegenüber dem vollverbundenen Constellation Model führt.

Die Eigenschaften der hier gewählten Graphenstruktur entsprechen der eines sogenannten gewurzelten Baums. Der Graph ist azyklisch, die Kanten sind gerichtet. Jeder Knoten ist durch genau einen gerichteten Pfad vom Wurzelknoten aus erreichbar. Jeder Knoten, bei dem es sich nicht um den Wurzelknoten handelt, muss also genau einen Elternknoten haben. Abbildung 5.5 zeigt einen Beispielgraphen.

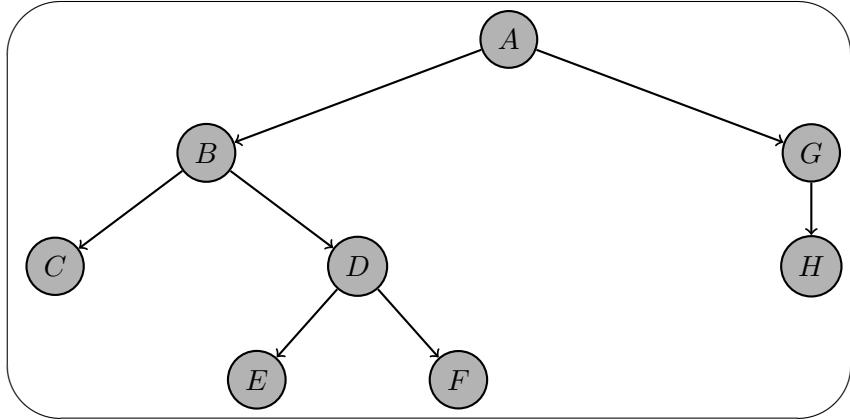


Abbildung 5.5: Beispiel für einen Relationsgraphen, der alle geforderten Eigenschaften erfüllt. Der Graph ist azyklisch, die Kanten sind gerichtet. Jeder Knoten ist durch genau einen gerichteten Pfad vom Wurzelknoten aus erreichbar.

Das Constellation Model wurde ursprünglich für den zweidimensionalen Bildraum entwickelt und wurde für die vorliegende Arbeit erweitert, um in dem vorliegenden 7-dimensionalen Posenraum arbeiten zu können. Relationen werden durch Kanten im Graph beschrieben. Die Abhängigkeit ist entgegen der Richtung, in der die Kante zeigt. Das Objekt, das mit dem Knoten am Ende der Kante verbunden ist, wird also relativ zu dem Objekt betrachtet, das zum Knoten am Beginn der Kante gehört. Anhand beider Objekte wird ein Raum aufgespannt, der die relative Pose zwischen beiden Objekten beschreibt. Eine über den Raum definierte Gauss-Mischverteilung wird für den entsprechenden Punkt im Raum evaluiert, um die Qualität einer relativen Pose zu bestimmen.

Die Gauss-Mischverteilung erlaubt eine verformbare Relation, wobei der Grad der Verformung durch die Lerndaten bestimmt wird. Zum besseren Verständnis soll eine Metapher aus der Robotik herangezogen werden. Der Graph wird je nach Szene als offene oder geschlossene kinematische Kette betrachtet, die Knoten als Kombination aus Kugelgelenk und translatorischem Gelenk. Die Lerndaten schränken den Gelenkkraum eines jeden Gelenks ein. Verändert man die Gelenkstellungen, so bewegen sich die Objekte in der Szene. Auf Grund der Einschränkung des Gelenkkraums liegt für jede Gelenkstellung eine gültige Szene vor. Die Menge aller gültigen Szenen entspricht also der Menge aller Punkte im Arbeitsraum.

5.5.2 Herleitung des OCM

Bei der Definition des Szenenmodells in Abschnitt 5.4 wurde bereits die Verteilung $P(\mathbf{A}, \mathbf{X}|\mathcal{O})$ vorgestellt. Dahinter verbirgt sich das Object Constellation Model, hierunter wird das probabilistische Modell eines einzelnen Szenenobjekts, also eines Objekts im Kontext einer Szene, verstanden. Das Modell beschreibt die relevanten Beziehungen zwischen dem zugrunde liegenden Szenenobjekt und den anderen Objekten der Szene, modelliert also die komplette Szene. Abbildung 5.6 zeigt eine grafische Darstellung. In diesem Abschnitt wird die Herleitung des Modells gezeigt.

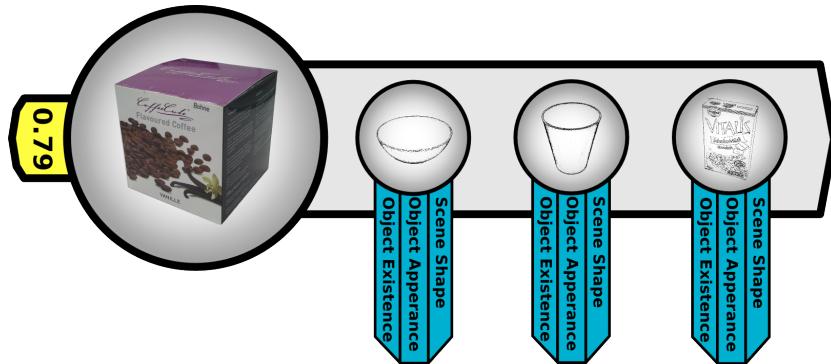


Abbildung 5.6: Das Object Constellation Model beschreibt ein einzelnes Szenenobjekt, also ein Objekt (hier die Kaffeebox) im Kontext einer Szene. Die Slots werden durch die kleinen Kreise rechts symbolisiert, jeder Slot beschreibt ein Objekt der Szene mit den in blau dargestellten Parametern. Der gelbe Kasten gibt an, mit welcher Wahrscheinlichkeit das Szenenobjekt vorliegt.

Der Modellentwurf setzt voraus, dass zuerst eine Reihe von Problemen gelöst werden. In einer perfekten Welt würde sich das Modell durch einfaches Einsetzen der beobachteten Objektevidenzen auswerten lassen. In der Realität verfügt jeder Objektdetektor auch immer über eine Fehldetectionsrate hinsichtlich sogenannter *false-positives* und *false-negatives*. Dies bedeutet, dass der Detektor gelegentlich nicht existierende Objekte erkennt oder existierende Objekte nicht wahrgenommen werden. Auch eine Kombination beider Fehler ist denkbar, bei der ein Objekt als ein anderes erkannt wird. Dies erschwert auch den Umgang mit Clutter, also irrelevanten, nicht zur Szene gehörenden Objekten. Diese können nicht einfach aussortiert werden, schließlich könnte es sich um fehlerhaft erkannte Objekte der Szene handeln, die noch durch ihre Pose zur Erkennung beitragen könnten.

Eine Möglichkeit zur Lösung des Problems besteht darin, wie bei *Eidenberger et al.* jedem Detektor ein Fehlermaß zuzuordnen [Eid10]. Dieses muss entsprechend der jeweiligen Funktionsweise modelliert und somit für jeden Detektor einzeln zugeschnitten werden. Dies ist aufwändig und nicht immer möglich, daher wird hier eine andere, generischere Lösung bevorzugt. *Fergus et al.* nutzen eine Marginalisierung über den Raum aller

möglichen Zuordnungen (vergleiche [FPZ05]). Alle möglichen Kombinationen der vorliegenden Beobachtungen werden gebildet, jede Einzelne wird mit dem Modell überprüft. Die Ergebnisse werden addiert, was erheblich zur Robustheit des Ansatzes beiträgt. Viele der so aufgestellten Hypothesen erzielen niedrige Bewertungen, einige wenige jedoch haben eine hohe Aussagekraft und liefern hohe Ergebnisse.

$$P(\mathbf{A}, \mathbf{X}|O) = \sum_{\mathbf{h} \in H} P(\mathbf{A}, \mathbf{X}, \mathbf{h}|O) \quad (5.4)$$

Dasselbe Vorgehen wird auch im Rahmen dieser Arbeit eingesetzt². Gleichung 5.4 führt die sogenannte *Zuordnungshypothese* ein, die eine Untermenge der Objektevidenzen dem Modell zuordnet. An dieser Stelle soll der Begriff **Slot** eingeführt werden. Eine Szene besteht aus mehreren Szenenobjekten. Lernt man aus diesen das Modell, so wird aus jedem Szenenobjekt ein Slot, der über bestimmte Eigenschaften wie Objekttyp, relative Pose zu anderen Slots und Auftrittswahrscheinlichkeit verfügt. Die Eigenschaften spiegeln das Szenenobjekt wieder, aus dem der Slot erstellt wurde. Abbildung 5.7 illustriert die Abbildung von Szenenobjekten auf Slots. Wird das Modell ausgewertet, so wird jedem Slot maximal eine Objektevidenz zugeordnet. Deren Objekttyp und Pose werden mit den Slotparametern abgeglichen, aus dem Grad der Übereinstimmung ergibt sich die Bewertung bzw. die Wahrscheinlichkeit, dass das entsprechende Szenenobjekt vorliegt.

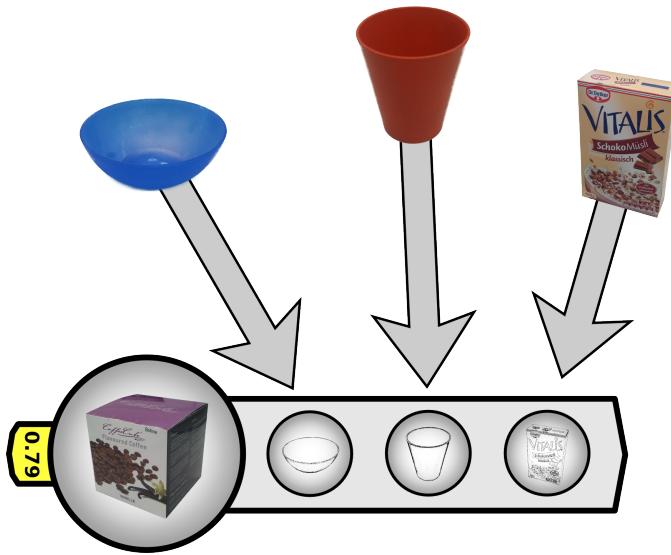


Abbildung 5.7: Für jedes Objekt der Szene wird ein Slot im OCM erstellt, der die Erscheinungsform des Objekts sowie die Auftrittswahrscheinlichkeit wieder gibt. Die Relationen zwischen den Slots geben die Relationen zwischen den Szenenobjekten wieder.

²Um diesen Zusammenhang hervorzuheben ist die Terminologie möglichst analog zur Originalquelle gewählt. Sie wurde jedoch so angepasst, dass ein klarer Bezug zur Szenenerkennung vorliegt.

Die Zuordnungshypothese \mathbf{h} ist ein Vektor, dessen Länge der Anzahl aller Slots im Modell entspricht. h_p bezeichnet das p -te Element des Vektors \mathbf{h} . Das p -te Element des Vektors enthält einen Index, der in Zusammenhang mit den Vektoren \mathbf{A} und \mathbf{X} Typ und Pose derjenigen Objektevidenz zurückgibt, die dem p -ten Slot zugeordnet ist. Der Raum aller Hypothesen H umfasst alle möglichen Kombinationen der Zuordnungen von Objektevidenzen zu Slots und hat eine Größe von $|H| = (N + 1)^P$, wobei N der Anzahl der Objektevidenzen und P der Anzahl der Slots entspricht. Das Ziehen von Hypothesen aus dem Raum entspricht *Ziehen mit Zurücklegen*. Das Zurücklegen geschieht, damit einer Hypothese mehrmals das sogenannte *Nullobject* zugeordnet werden kann. Dessen Aussage ist es, dass einem Slot keine Objektevidenzen zugeordnet ist.

Im Raum über die Zuordnungshypothese sind auch Hypothesen vorhanden, deren Evaluation gezielt durch die Nutzung von Ausschlusswissen unterbunden werden muss. So kann eine Hypothese nicht evaluiert werden, wenn dem ersten Slot keine Evidenz zugewiesen wurde (was wiederum durch Zuweisung des Nullobjekts repräsentiert wird). Dieser entspricht nämlich dem Referenzobjekt, relativ zu dem alle Relationen definiert sind. Nicht erwünscht sind ebenfalls Hypothesen, bei denen eine Objektevidenz mehreren Slots zugewiesen wird. Eine Ausnahme bilden hier die Hypothesen, bei denen das Nullobject mehrmals vorhanden ist. Es ist eine durchaus valide Annahme, dass mehrere Slots nicht mit Evidenzen belegt sind, z.B. da die damit assoziierten Objekte nicht vorhanden sind. Abbildung 5.8 zeigt alle Hypothesen für ein einzelnes Object Constellation Model.

$$\begin{aligned} P(\mathbf{A}, \mathbf{X}|O) &= \sum_{\mathbf{h} \in H} P(\mathbf{A}, \mathbf{X}, \mathbf{h}|O) \\ &= \sum_{\mathbf{h} \in H} P(\mathbf{A}, \mathbf{X}|\mathbf{h}, O)P(\mathbf{h}|O) \\ &= \sum_{\mathbf{h} \in H} P(\mathbf{A}|\mathbf{X}, \mathbf{h}, O)P(\mathbf{X}|\mathbf{h}, O)P(\mathbf{h}|O) \end{aligned} \quad (5.5)$$

Im übrigen Abschnitt wird der Teil des Modells hergeleitet, der sich mit der Bewertung einer einzelnen Hypothese befasst. Durch die mehrmalige Anwendung des Gesetzes der bedingten Wahrscheinlichkeiten wird Gleichung 5.4 in Gleichung 5.5 überführt.

$$P(\mathbf{A}, \mathbf{X}|O) = \sum_{\mathbf{h} \in H} \underbrace{P(\mathbf{A}|\mathbf{h}, O)}_{\substack{\text{Object} \\ \text{Appearance}}} \underbrace{P(\mathbf{X}|\mathbf{h}, O)}_{\substack{\text{Scene} \\ \text{Shape}}} \underbrace{P(\mathbf{h}|O)}_{\substack{\text{Object} \\ \text{Existence}}} \quad (5.6)$$

Es wird die vereinfachende Annahme getroffen, dass der Typ eines Objekts nicht von dessen Pose abhängt. Daher wird \mathbf{X} aus $P(\mathbf{A}|\mathbf{X}, \mathbf{h}, O)$ entfernt. Das Resultat - Gleichung 5.6 - ist die zentrale Gleichung des Object Constellation Model. Die Gleichung setzt sich aus drei Verteilungen zusammen, welche (in Reihenfolge ihres Auftretens) folgende Funktionen erfüllen:

Evidenzen:



Zuweisung	OK?	Anmerkung
		Referenzobjekt verdeckt
		Referenzobjekt verdeckt
		Referenzobjekt verdeckt
		Unvollständige Hypothese
		Doppelte Zuweisung
		Vollständige Hypothese
		Unvollständige Hypothese
		Vollständige Hypothese
		Doppelte Zuweisung

Abbildung 5.8: Die Menge aller Hypothesen für das Szenenobjekt "Kaffeebox". Links ist die Zuweisung der Evidenzen zu Slots zu sehen. Die mittlere Spalte gibt Auskunft darüber, ob die Hypothese ausgewertet oder verworfen wird. Die Anmerkungen rechts geben eine kurze Beschreibung der Hypothese.

- Die **Objekt Appearance** bewertet die Objekttypen der Evidenzen.
- Die **Scene Shape** bewertet die relative Lagen der Evidenzen, wobei der zuvor beschriebene Relationsgraph Anwendung findet.
- Unter dem Term **Object Existence** werden die Auftrittswahrscheinlichkeiten sowie die Länge und Wahrscheinlichkeit der gegebenen Hypothese behandelt.

Eine nähere Erklärung der einzelnen Verteilungen sowie deren Ausformulierung wird im weiteren Verlauf gegeben.

5.5.3 Object Appearance

Der *Object Appearance*-Term ist Teil der zentralen Gleichung des Object Constellation Models und für die Auswertung der Typinformationen zuständig (siehe Abbildung 5.9). Das Produkt in Gleichung 5.7 beschreibt den Vordergrund, der Term links davon den Hintergrund. Unter dem Vordergrund werden all diejenigen Evidenzen bewertet, die Slots zugewiesen sind. Alle nicht berücksichtigten Evidenzen werden als Teil des Hintergrunds angenommen und entsprechend bewertet.

$$P(\mathbf{A}|\mathbf{h}, \mathcal{O}) = \underbrace{\left(\frac{1}{K+1} \right)^{N-\psi(\mathbf{h})}}_{\text{Hintergrund}} \underbrace{\prod_{p=1}^P \text{Cat}_{A(h_p)}[\boldsymbol{\lambda}_p^O]}_{\text{Vordergrund}} \quad (5.7)$$

Aus Sicht des Modells besteht die Welt nur aus den Objekten, die in der Szene vorkommen. Die Menge der erwarteten Typen ist daher endlich und diskret. Die Verteilung der Wahrscheinlichkeiten lässt sich daher am besten durch eine Multinomialverteilung beschreiben, also in Form einer Wahrscheinlichkeitstabelle. Für K in der Szene vorkommende Objektklassen werden $K+1$ Einträge benötigt. Dies ist notwendig, falls ein Objekt beobachtet wird, dass nicht Teil der Szene ist. Da nicht immer alle vom Detektor aufspürbaren Objekte auch in der Szene vorhanden sein müssen, ist dies ein realistisches Szenario. Der zusätzliche Eintrag in der Verteilung ist also für unbekannte Objekttypen reserviert. Pro Slot im Model wird eine Multinomialverteilung benötigt, deren Ergebnisse miteinander multipliziert werden.

Die rechte Seite von Gleichung 5.7 zeigt die Multinomialverteilung für alle P Slots. Es wird von der vereinfachenden Annahme ausgegangen, dass alle Slots voneinander unabhängig sind. $\boldsymbol{\lambda}_p^O$ bezeichnet die Verteilungsparameter für den Slot p des zum Szenenobjekt O gehörenden Modells. Die Einträge der Tabelle entsprechen $\boldsymbol{\lambda} = [\lambda_0, \lambda_1, \dots, \lambda_K]$, wobei λ_0 die Wahrscheinlichkeit für die Klasse der unbekannten Objekttypen bezeichnet. Die restlichen Einträge geben die Wahrscheinlichkeiten für die in der Szene enthaltenen Objektklassen an.

Die Verteilung wird mittels $A(h_p)$ adressiert. Dies soll zum besseren Verständnis näher erläutert werden. Zunächst wird das p -te Element aus dem Vektor der Zuordnungshypothese entnommen. Es bezeichnet die Nummer der Objektevidenz, welche dem p -ten

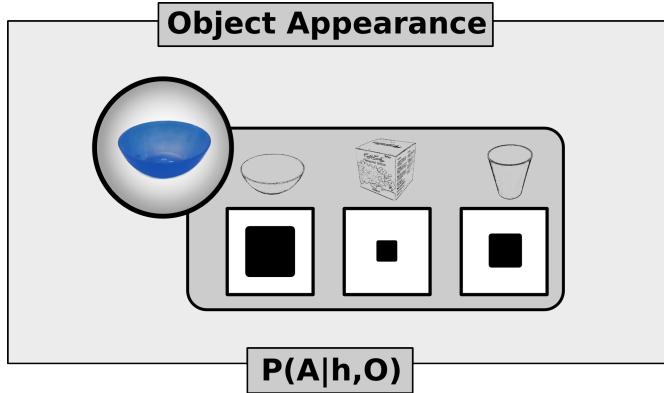


Abbildung 5.9: Für jeden Slot im Modell hält der Object Appearance Term eine Multinomialverteilung bereit, hier dargestellt durch ein Hinton-Diagramm. Diese bewertet die Objekttypen der in den Slot eingesetzten Objektevidenz.

Slot zugeordnet ist. $A(\cdot)$ liefert den zu der besagten Evidenz gehörigen Typ. Durch dessen Einsetzen in die Multinomialverteilung wird die damit assoziierte Wahrscheinlichkeit abgerufen.

Die hier beschriebene Vorgehensweise kann dazu eingesetzt werden, das Fehldetektionsverhalten von Objektdetektoren zu kompensieren. Dies setzt jedoch die (durchaus relativistische) Annahme voraus, dass das Fehlverhalten systematisch ist. Bei Versuchen mit den Detektoren wurde häufig festgestellt, dass statt einer Tasse ein Becher gleicher Farbe beobachtet wurde. Findet in zwei von zehn Fällen eine solche Fehldetektion statt, dann liegt die Wahrscheinlichkeit für den Typ *Tasse* bei 0,8 und für *Becher* bei 0,2. Wird im Rahmen der Szenenerkennung nun die Tasse falsch erkannt, so ist die resultierende Wahrscheinlichkeit zwar geringer, aber immer noch größer als Null.

$$\psi(\mathbf{h}) = \sum_{i=1}^{\text{len}(\mathbf{h})} 1 - \delta[h_i] \quad (5.8)$$

Die linke Seite von Gleichung 5.7 stellt das Hintergrundmodell dar. Alle Objektevidenzen, die keinem Slot zugewiesen wurden, werden unter einer Gleichverteilung evaluiert. Da die Wahrscheinlichkeitstabelle $K + 1$ Einträge hat, ist auch die Hintergrundverteilung dementsprechend ausgelegt. Der Bruch wird mit der Anzahl der nicht zugewiesenen Evidenzen potenziert, die mit der Hilfsfunktion $\psi(\mathbf{h})$ bestimmt wird.

Die in Gleichung 5.8 gezeigte Hilfsfunktion berechnet die Anzahl der Slots, die mit einer Evidenz belegt wurden. Übergeben wird der Hypothesenvektor, aus dem neben der Zuweisung auch die Anzahl aller Slots abgeleitet werden kann. Das Kronecker-Delta $\delta[\cdot]$ ist über die ganzen Zahlen definiert. Es nimmt nur dann den Wert Eins an, wenn die Eingabe Null ist. Im vorliegenden Fall also, wenn einem Slot keine Evidenz zugewiesen wurde. Die Subtraktion von Eins kehrt den Effekt des Kronecker-Delta um.

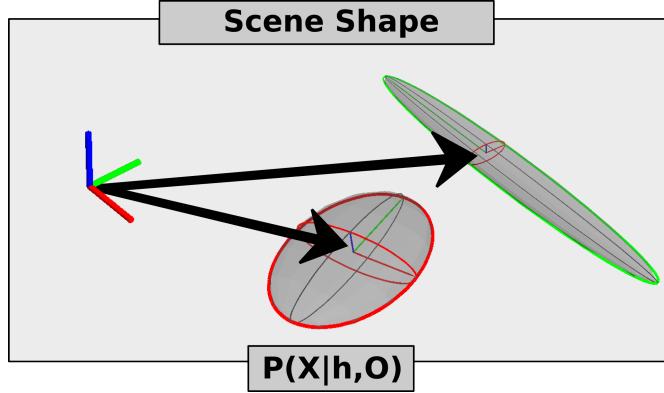


Abbildung 5.10: Der Scene Shape Term beschreibt die Relationen zwischen den einzelnen Objekten einer Szene. Im hier gezeigten Beispiel werden die Lagen zweier Objekte relativ zu einem dritten beschrieben.

In der Literatur ist es nicht unüblich, auch jenseits der Hintergrundklasse einen Hintergrundterm in die Gleichungen einzubauen. In diesem Fall muss der Nutzen jedoch hinterfragt werden. So verhindert die Berücksichtigung der nicht zugewiesenen Evidenzen, dass gegebene Informationen nicht berücksichtigt werden. Andererseits kann sich eine große Anzahl von Objektevidenzen auf Grund des Hintergrundterms negativ auf die Erkennungsleistung auswirken. Der Nutzen des Hintergrundterms wird im Rahmen von Kapitel 8 überprüft.

5.5.4 Scene Shape

Der *Scene Shape*-Term ist Teil der zentralen Gleichung des Object Constellation Model und für die Bewertung der relativen Posen verantwortlich (siehe Abbildung 5.10). An dieser Stelle wird der Relationsgraph wieder aufgegriffen, der im Rahmen des modifizierten Constellation Model in Unterabschnitt 5.5.1 beschriebenen wurde. Seine Struktur wird durch die Wahl entsprechender Verteilungen auf das Modell übertragen, die Slots übernehmen die Rolle der Knoten. Wie schon bei der Object Appearance ist auch hier die Gleichung in Vorder- und Hintergrund unterteilt, wobei letzterer wieder die von der Hypothese unberücksichtigten Evidenzen bewertet.

Bevor der eigentliche Term vorgestellt wird soll zunächst der Gedankengang dahinter erläutert werden. Gegeben ist eine Verbundverteilung über die absoluten Posen aller P in der Szene vorhandenen Objekte. Diese muss entsprechend faktorisiert werden. Da die Relationen durch einen baumförmigen Graphen beschrieben werden bietet sich hierfür ein bayessches Netz an. Dieses ist in der Lage, die relativen Posen durch bedingte Wahrscheinlichkeiten darzustellen.

$$P(x_1, \dots, x_P) = \prod_{p=1}^P P(x_p | x_{pa[p]}) \quad (5.9)$$

Ein beliebiges bayessches Netz wird durch Gleichung 5.9 beschreiben (vergleiche [Pri12, Kapitel 10.2]). Die Funktion $pa[p]$ beschreibt die Identität der Elternknoten von p . Da das Referenzobjekt von keinem anderen Objekt abhängig ist kann es wie in Gleichung 5.10 gezeigt aus dem Produkt herausgezogen werden.

$$P(x_1, \dots, x_P) = p(x_1) \prod_{p=2}^P P(x_p | x_{pa[p]}) \quad (5.10)$$

Die *Scene Shape*-Gleichung 5.11 ist nach dem gleichen Schema aufgebaut. Der Verteilung ohne Abhängigkeiten fällt die Rolle des Referenzobjektes zu. Dieses ist keinem anderen Szenenobjekt untergeordnet, da es ja für alle anderen als *Intermediate Object* fungiert - also als Objekt, relativ zu dem die Lagen aller anderen Objekte beschrieben werden. Folglich lässt sich also auch keine relative Pose bewerten, weshalb an dieser Stelle mit der absoluten Pose gearbeitet werden muss. Da hierüber jedoch keine Aussage getroffen werden kann ist von einer Gleichverteilung auszugehen, welche durch $Uniform[V]$ repräsentiert wird. Alle anderen Annahmen würden die geforderten Invarianzeigenschaften hinsichtlich Translation und Rotation verletzen. Die angesprochene Gleichverteilung wurde in der Gleichung mit dem Hintergrundterm verrechnet.

$$P(\mathbf{X}|\mathbf{h}, O) = \underbrace{Uniform[V]}^{Hintergrund \text{ und } Pose \text{ des Referenzobjekts}}^{N-\psi(\mathbf{h})+1} \underbrace{\prod_{p=2}^P P(x_p | x_{pa[p]}, h)}_{Vordergrund} \quad (5.11)$$

Die Multiplikation im rechten Teil der Gleichung bewertet die relativen Posen. Für jeden der verbliebenen $P - 1$ Slots wird die Verteilung über alle zugehörigen Relationen, also alle eingehenden Kanten im Graphen, evaluiert. Der Verteilung werden die absoluten Posen des gegenwärtig betrachteten Slots sowie dessen Elternslots übergeben.

$$P(x_i | x_j, h) = \sum_{k=1}^n \lambda_k \exp \left(-Mah_{rel(X(h_i), X(h_j))}[\boldsymbol{\mu}_{ij}^O, \boldsymbol{\Sigma}_{ij}^O] \right) \quad (5.12)$$

Die relative Pose zwischen zwei Objekten wird durch eine Gauss-Mischverteilung dargestellt. Deren Parameter beschreiben die Lage des Objekts j im Koordinatensystem des Objekts i , natürlich in Abhängigkeit vom Szenenobjekt O . Übergeben wird die relative Pose zwischen beiden Objekten, die mittels der Funktion $rel(., .)$ bestimmt wird. Die absolute Pose der zugewiesenen Evidenzen wird über die Funktion $X(.)$ abgerufen. Es wird daran erinnert, dass die Bewertung einer Relation nur dann stattfinden kann, wenn der Pfad zwischen dem aktuell betrachteten Knoten im Relationsgraphen und dem

Wurzelknoten komplett mit Evidenzen besetzt ist. Ist dies nicht der Fall, so werden die Teilbäume mit unbesetzten Elternknoten durch Marginalisierung entfernt.

$$Mah_{\mathbf{x}}[\boldsymbol{\mu}, \boldsymbol{\Sigma}] = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})} \quad (5.13)$$

Wie Gleichung 5.12 zeigt erfolgt die Auswertung der Gauss-Mischverteilung jedoch auf einem Umweg. Ziel der Formulierung ist es, für das vorliegende Modell die Wahrscheinlichkeit einer gegebenen Evidenz zu berechnen. Nun handelt es sich bei der besagten Verteilung um eine Dichtefunktion und das Einsetzen der Evidenz liefert einen numerischen Wert, keine gültige Wahrscheinlichkeit. Auch ist der Maximalwert abhängig von den Einträgen der Kovarianzmatrix. Liegen zwei Gauss-Kernel mit unterschiedlichen Kovarianzen vor, so erzeugt das Einsetzen der jeweiligen Mittelwerte unterschiedlich hohe Werte, was aus Sicht der Problemstellung ein ungewolltes Verhalten ist.

Aus diesem Grund wird die in Gleichung 5.13 Mahalanobis-Distanz zwischen der Pose der Evidenz und dem jeweiligen Kernel der Gauss-Mischverteilung berechnet (vergleiche auch [Mah36]). Das Ergebnis wird auf das Intervall $[-\infty, 0]$ der e -Funktion abgebildet. Das Resultat kann als Wahrscheinlichkeit betrachtet werden, welche mit den Axiomen von Kolmogorov konform ist. Die einzelnen Kernel werden wie von der Gauss-Mischverteilung miteinander verrechnet.

Zwar wird die Pose als Ganzes evaluiert, modelliert wurden Position und Orientierung jedoch getrennt. Versuche mit dem Lernalgorithmus ergaben, dass komplexe Trajektorien nur dann mit ausreichender Präzision erlernt werden konnten, wenn Pose und Orientierung unabhängig voneinander betrachtet wurden. Über die Ursache kann nur spekuliert werden. An dieser Stelle soll keine Verallgemeinerung getroffen und die Erlernbarkeit der kompletten Pose generell in Frage gestellt werden. Vielmehr sollte diese Aussage als Ansatzpunkt für weiterführende Forschungen betrachtet werden.

Das Hintergrundmodell umfasst die linke Hälfte von Gleichung 5.11. Wie schon bei der Object Appearance werden alle Objektevidenzen, die keinem Slot zugewiesen wurden, unter einer Gleichverteilung evaluiert. Die hierfür benötigte Verteilung $Uniform[V]$ ist dank der Sonderrolle des Referenzobjekts schon vorhanden und wird mit der Anzahl der nicht zugewiesenen Evidenzen potenziert. Diese wird wieder mit der Hilfsfunktion berechnet. Zusätzlich wird in der Potenz der Wert eins addiert, da hier wie oben erwähnt die Gleichverteilung über die Pose des Referenzobjekt mit eingeflossen ist. Auch hier ist der Nutzen des Hintergrundterms umstritten. Auf der einen Seite sollen keine Informationen verschenkt werden, auf der anderen Seite kann eine große Anzahl von Objektevidenzen die Erkennungsleistung senken. Im Rahmen von Kapitel 8 werden die Auswirkungen näher überprüft.

$$Uniform[V] = \frac{1}{V2\pi^2} \quad (5.14)$$

Es bleibt die Beschreibung der Gleichverteilung über die Pose in Gleichung 5.14. Die Variable V im Nenner des Bruchs beschreibt eine Gleichverteilung über die Position.

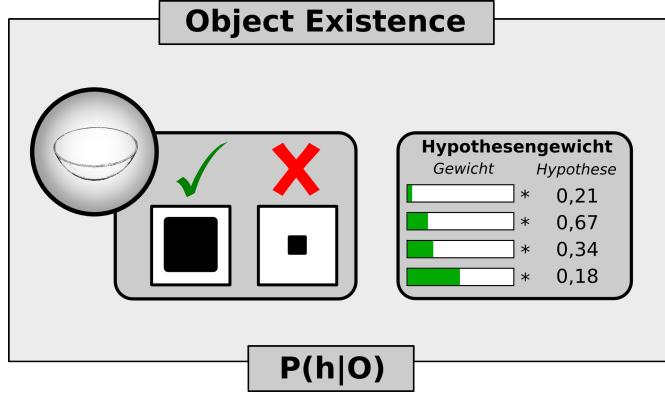


Abbildung 5.11: Der Object Existence Term bringt zwei Faktoren in das Modell mit ein. Links: Für jeden Slot liegt eine Multinomialverteilung vor, die das Auftreten des Objekts in der Szene beschreibt. Rechts: Eine Poisson-Verteilung über die Länge der Hypothese bewertet vollständige Hypothesen höher.

Da der Term für eine Gleichverteilung über einen unendlich großen Raum einen Wert um die Null annehmen würde, wurde ein Arbeitsraum mit Volumen V definiert, innerhalb dem die Szenenerkennung stattfindet. Die rechte Hälfte des Bruchs stellt eine Gleichverteilung über die vierdimensionale Orientierung dar. Diese wird durch ein Einheitsquaternion beschrieben. Die Menge aller Einheitsquaternionen lässt sich als Sphäre \mathbb{S}^3 im vierdimensionalen Raum interpretieren, deren Volumen $2\pi^2$ entspricht (vergleiche auch [Haz02]).

5.5.5 Object Existence

Der *Object Existence*-Term ist der letzte Teil der zentralen Gleichung des Object Constellation Model. Er besteht aus zwei Bestandteilen (in Reihenfolge ihres Auftretens in Gleichung 5.15): der Bewertung für die Vollständigkeit der vorliegenden Hypothese und die Wahrscheinlichkeiten für das Auftauchen der mit den Slots assoziierten Objekte.

$$P(\mathbf{h}|O) = \underbrace{\frac{\psi(\mathbf{h})}{\sum_{\mathbf{h} \in H} \psi(\mathbf{h})}}_{\text{Vollständigkeit der Hypothese}} \prod_{p=1}^P \underbrace{\text{Bern}_{\delta[h_p]}[\lambda_p^O]}_{\text{Auftrittswahrsch. der Objekte}} \underbrace{\frac{1}{2}^{N-\psi(\mathbf{h})}}_{\text{Hintergrund}} \quad (5.15)$$

Der Bruch bewertet die Vollständigkeit der vorliegenden Hypothese. Eine Hypothese gilt dann als vollständig, wenn jedem Slot eine Objektevidenz zugewiesen ist. Eine vollständige Hypothese wird mit einem höheren Faktor gewichtet als eine unvollständige. Aus diesem Weg soll gewährleistet werden, dass die aussagekräftigsten Hypothesen am meisten zur Gesamtwahrscheinlichkeit beitragen.

Die Gewichtung erfolgt anhand der belegten Slots. Eine Normalisierung durch die Anzahl aller belegten Slots gewährleistet, dass die Summe aller Gewichte Eins ergibt. Berücksichtigt werden nur die Slots der Hypothesen, deren Wahrscheinlichkeit ungleich Null ist. Dies lässt sich zugleich auch hypothesengewicht interpretieren.

Der mittlere Teil der Gleichung beschreibt die Auftrittswahrscheinlichkeiten für die mit den Slots assoziierten Objekte. Über die Bernoulli-Verteilung wird definiert, mit welcher Wahrscheinlichkeit das Objekt entweder vorhanden ist oder fehlt, wobei λ_p^O den Parameter für Slot p von Szenenobjekt O beschreibt. Auch hier kommt wieder das Kronecker-Delta zum Einsatz. Auf Grund dessen Eigenschaft gibt λ_p^O die Wahrscheinlichkeit wieder, dass das Objekt *nicht* existiert. Der hintere Term beschreibt den Hintergrund. Alle keinem Slot zugewiesenen Objekte, deren Anzahl $N - \psi(\mathbf{h})$ entspricht, werden unter einer Gleichverteilung evaluiert.

Der Nutzen dieser Zusatzinformation erschließt sich nicht gleich, kann jedoch über ein Beispiel verdeutlicht werden. Gegeben seien zwei Szenen, die sich in genau einem Objekt überschneiden (siehe Teller in Abbildung 5.3). In der einen Szenen taucht das Objekt häufig auf, in der anderen seltener. In der ersten Szene liegt also eine höhere Auftrittswahrscheinlichkeit vor als in der zweiten. Wird nun das Objekt beobachtet, so kann man davon ausgehen, dass die erste Szenen vorliegt, da sie auf Grund der höheren Auftrittswahrscheinlichkeit des Objekts wahrscheinlicher ist.

5.5.6 Hintergrundmodell

In Abschnitt 5.4 wird die Notwendigkeit einer Rückweisungsklasse motiviert, die immer dann vorliegt, wenn die vorliegende Szene keiner erlernten Konfiguration entspricht. Das hierfür nötige Hintergrundmodell muss aus Gründen der Vergleichbarkeit wie das Object Constellation Model beschaffen sein. Es werden zwei Annahmen getroffen. Zum einen wird pro Objektevidenz von einem Szenenobjekt ausgegangen. Für das Hintergrundmodell bedeutet dies, dass für jede Evidenz ein Object Constellation Model vorhanden ist, das wiederum ebenso viele Slot hat, wie Evidenzen vorhanden sind. Zum anderen wird davon ausgegangen, dass keinerlei Annahmen über die Struktur des Hintergrunds getroffen werden können, also absolutes Unwissen über die Typen der beteiligten Objekte, deren relative Beziehungen und die Auftrittswahrscheinlichkeiten vorliegt.

Das Hintergrundmodell wird durch Anpassung der zentralen OCM-Gleichung 5.6 gebildet. Die Grundstruktur der Gleichung wird beibehalten, nur die Verteilungen für *Object Appearance*, *Scene Shape* und *Object Existence* werden ausgetauscht. Die stattdessen verwendeten Verteilungen werden im weiteren Verlauf erläutert.

Die angepasste *Object Appearance* wird durch Gleichung 5.16 beschrieben. Es wird wieder in einen Vorder- und Hintergrundterm unterschieden. Ersterer bewertet die den Slots zugewiesenen Evidenzen, letzterer die keinen Slots zugewiesenen. Da jedoch die Objekttypen im Vorder- und Hintergrund als gleichverteilt angenommen werden, lassen sich beide Terme zusammenfassen. Eine unbekannte Objektklasse kann nicht existieren, da

für jede Evidenz ein Szenenobjekt angenommen wird. Daher wird hier nur von K statt $K + 1$ unterschiedlichen Typen ausgegangen.

$$P(\mathbf{A}|\mathbf{h}, O) = \frac{1}{K^{\psi(\mathbf{h})} K^{N-\psi(\mathbf{h})}} = K^{-N} \quad (5.16)$$

Gleichung 5.17 beschreibt die *Scene Shape* des Hintergrundmodells. Da auf Grund der angenommenen Unwissenheit keine Relationen angenommen werden können, wird die absolute Pose berücksichtigt, über die allerdings auch keine Annahme getroffen werden kann. Daher wird hier eine Gleichverteilung über den Posenraum gewählt. Auf Grund der Gleichheit von Vorder- und Hintergrundterm können beide wie schon oben zusammengefasst werden.

$$P(\mathbf{X}|\mathbf{h}, SO) = \text{Uniform}[V]^{\psi(\mathbf{h})} \text{Uniform}[V]^{N-\psi(\mathbf{h})} = \text{Uniform}[V]^N \quad (5.17)$$

Gleichung 5.18 zeigt, dass die *Object Existence* nach wie vor in drei Teile unterteilt ist. Der Normalisierungsterm bleibt unverändert, die Poisson-Verteilung ist über die Menge aller gefundenen Evidenzen definiert. Dies ist durch die oben getroffene Annahme begründet, dass für jede Objektevidenz ein Slot im Modell vorhanden ist. Auch hier im Hintergrundmodell sollen Hypothesen, bei denen alle Slots belegt sind, höher bewertet werden als unvollständige Hypothesen. Dies ist aus Gründen der Vergleichbarkeit mit dem herkömmlichen Object Constellation Model notwendig. Als Argument der Verteilung wird wie schon beim Originalterm der Object Existence die Anzahl der belegten Slots übergeben. Das Schema der Auftrittswahrscheinlichkeit ist bereits bekannt, deren Vorder- und Hintergrundterm können zusammengefasst und die Gleichung damit vereinfacht werden.

$$\begin{aligned} P(\mathbf{h}|O) &= \frac{\psi(\mathbf{h})}{\sum_{\mathbf{h} \in H} \psi(\mathbf{h})} \frac{1^{\psi(\mathbf{h})}}{2} \frac{1^{N-\psi(\mathbf{h})}}{2} \\ &= \frac{\psi(\mathbf{h})}{\sum_{\mathbf{h} \in H} \psi(\mathbf{h})} \frac{1^N}{2} \end{aligned} \quad (5.18)$$

In diesem Abschnitt wurde Wert darauf gelegt, zwischen den Vorder- und Hintergrundteilen der einzelnen Terme zu unterscheiden. Letztere werden im Rahmen von Kapitel 8 auf ihre Plausibilität hin untersucht und wurden daher hier entsprechend hervorgehoben.

5.6 Zusammenfassung der Modellstruktur

In den Abschnitten 5.4 und 5.5 wurden die beiden Teile des Szenenmodells vorgestellt. Dieser Abschnitt dient der Zusammenführung zu einem gemeinsamen Modell. Gleichung 5.19 zeigt die Hauptgleichung des Szenenmodells, erstellt aus den Gleichungen 5.2 und 5.3. An dieser Stelle lässt sich klar erkennen, dass das System neben der vorliegenden

Szene zugleich auch das beste Szenenobjekt - also das beste Modell, dass die vorliegenden Evidenzen beschreibt - bestimmt.

$$\hat{S} = \arg \max_{S,O} P(S)P(\mathbf{A}, \mathbf{X}|O)P(O|S) \quad (5.19)$$

Die Gleichung wird für jede erlernte Szene sowie die Rückweisungsklasse evaluiert. Hierzu werden die darin enthaltenen Object Constellation Model ausgewertet, die durch die Verteilung $P(\mathbf{A}, \mathbf{X}|O)$ dargestellt werden. Die Ergebnisse werden mit $P(O|S)$ gewichtet und aufmultipliziert. Letztendlich wird die apriori-Wahrscheinlichkeit der Szene eingearbeitet. Die Szene, welche Gleichung 5.19 maximiert, wird als vorliegende Szene angenommen.

$$\begin{aligned} P(\mathbf{A}, \mathbf{X}|O) = & \frac{\text{Uniform}[V]}{\sum_{\mathbf{h} \in H} \psi(\mathbf{h})} \sum_{\mathbf{h} \in H} \left(\psi(\mathbf{h}) \left(\frac{1}{2K+2}^{N-\psi(\mathbf{h})} \right)^P \right. \\ & \prod_{p=2}^P \text{Uniform}[V]^{N-\psi(\mathbf{h})} \sum_{k=1}^{n_{p,pa[p]}} \lambda_k \exp(-\text{Mah}_{\text{rel}(X(h_p), X(h_{p,a[p]}))}[\boldsymbol{\mu}_k^O, \boldsymbol{\Sigma}_k^O]) \\ & \left. \prod_{p=1}^P \text{Cat}_{A(h_p)}[\boldsymbol{\lambda}_p^O] \text{Bern}_{\delta[h_p]}[\lambda_p^O] \right) \end{aligned} \quad (5.20)$$

Für das Object Constellation Model existieren zwei verschiedene Varianten für die Szenen und die Rückweisungsklasse. Gleichung 5.20 zeigt die komplette Gleichung für erstere. Diese wurde durch zusammenfassen, umstellen und vereinfachen der in Abschnitt 5.5 vorgestellten Terme gebildet. Für das Object Constellation Model der Rückweisungsklasse wurde ähnlich verfahren. Das Resultat ist in Gleichung 5.21 zusammengefasst.

$$P(\mathbf{A}, \mathbf{X}|O) = \frac{\psi(\mathbf{h})((2K)^{-1}\text{Uniform}[V])^N}{\sum_{\mathbf{h} \in H} \psi(\mathbf{h})} \quad (5.21)$$

In diesem Kapitel wurde das Modell vorgestellt, das für die Szenenerkennung eingesetzt wird. Damit ist der wichtigste Teil der probabilistischen Modellbildung besprochen. Bis-her noch nicht erwähnt wurden die beiden verbliebenen Teile, das Lernen und die Inferenz. Beide Themen werden im folgenden Kapitel behandelt.

6. Lernen und Inferenz

Im vorherigen Kapitel wurde das Modell zur Szenenerkennung vorgestellt. Bisher offen ist noch die Frage, wie die Parameter des Modells bestimmt werden. Beim Programmieren durch Vormachen wird eine möglichst geringe Menge an Lernbeispielen zur Parameterschätzung herangezogen. Da sich die hier vorliegende Arbeit in diesem Kontext bewegt, wird auch hier automatisches Lernen bevorzugt.

Das Kapitel ist in zwei Abschnitte unterteilt. Zunächst wird der Algorithmus zum Lernen der Modellparameter aus Beispielen vorgestellt. Im Anschluss wird besprochen, wie die Inferenz, also die Bestimmung der Szene, durchgeführt wird.

6.1 Lernen der Modellparameter

Der Abschnitt ist in vier Teile untergliedert. Zunächst wird auf die Anforderungen eingegangen, die der Lernalgorithmus an das System stellt, und die Probleme, die bei deren Nichterfüllung auftreten. Für das Parameterlernen werden Informationen benötigt, die nur der Relationsgraph zur Verfügung stellen kann. Daher wird als nächstes darauf eingegangen, auf welchem Weg der Graph erstellt wird. In den darauffolgenden beiden Unterabschnitten wird besprochen, wie die Parameter für die beiden Teile des Modells erlernt werden.

6.1.1 Anforderungen

Die elementare Frage beim Parameterlernen ist, warum ein bisher beobachtetes Objekt nicht mehr wahrgenommen wird. Abbildung 6.1 zeigt eine Reihe von Messungen für einen Becher. Zum Zeitpunkt $t = 3$ konnte der Becher nicht wahrgenommen werden. Für den Lernalgorithmus ist es wichtig zu wissen, ob das Objekt durch den Benutzer entfernt wurde und es nicht mehr Teil der Szene ist. Dies würde eine Anpassung der Parameter erfordern, wie später noch verdeutlicht wird.

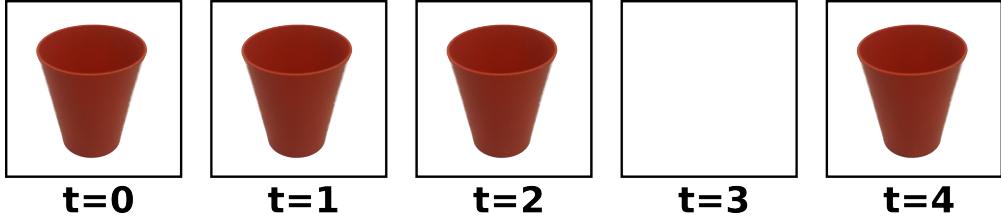


Abbildung 6.1: Ein beobachteter Becher kann nicht mehr durch die Objektdetektoren wahrgenommen werden. Was ist passiert? Wurde der Becher aus der Szene entfernt? Ist er außer Sicht oder verdeckt? Oder fand eine Fehlerkennung statt?

Um diese Frage zu klären müssen alle möglichen anderen Ursachen geprüft und ausgeschlossen werden. Abbildung 6.2 zeigt die möglichen Ursachen. Ein Grund wäre, dass sich das Objekt außerhalb des Kamerabilds befindet und daher nicht mehr von den Objektdetektoren beobachtet werden kann. Um einen solchen Fall identifizieren zu können wird ein sogenanntes *Beobachtungsmodel* benötigt. Dies gleicht das Sichtfeld der Kamera und die Position des Objekts ab und kann so eine Aussage über die Beobachtbarkeit des Objekts treffen.

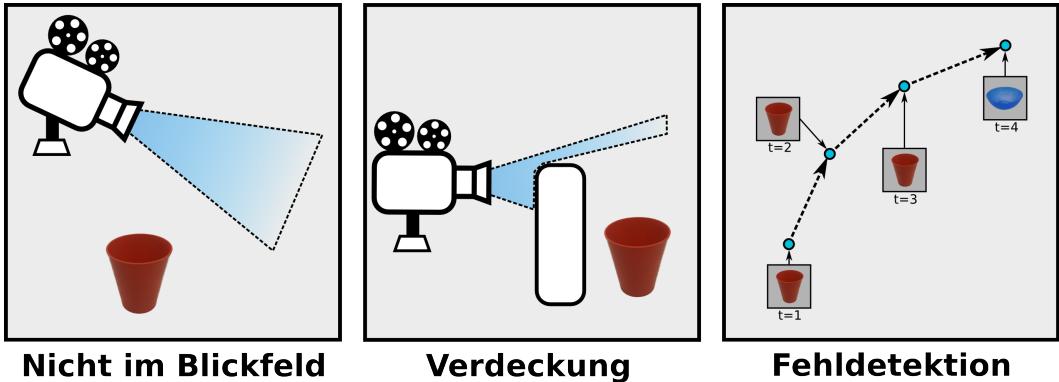


Abbildung 6.2: Wird ein Objekt nicht aus der Szenene entfernt, so kann sein Verschwinden drei Ursachen haben. Es kann sich außerhalb des Blickfelds befinden, durch die Umwelt bzw. ein anderes Objekt verdeckt oder fehlerhaft erkannt sein.

Eine andere Möglichkeit besteht darin, dass ein anderes Objekt oder ein Teil der Umgebung den Becher verdeckt. Ein *Verdeckungsmodell* schafft hier Abhilfe. Der Aufbau eines solchen Modells ist allerdings alles andere als trivial. *Eidenberger et al.* nutzen Wissen über die Form der Modelle, um daraus Verdeckungen und Teilverdeckungen zu berechnen [EGZ09, EZS09]. Dieser Ansatz schließt jedoch kein Wissen über die Struktur der Umgebung mit ein und ist speziell auf den verwendeten Detektor zugeschnitten. *Potthast et al.* nutzen Tiefendaten, um daraus ein Voxel-Gitter über die räumliche Belegung des

Arbeitsraums zu erzeugen. Aus diesem kann mittels eines Ray Traversal Algorithmus die Verdeckung bestimmt werden [PS11]. Nicht berücksichtigt wird allerdings, ob Voxel zu Objekten gehören, dementsprechend müsste nach jeder Objektbewegung eine neue Messung vorgenommen werden.

Eine letzte Ursache bleibt noch zu erläutern. Wird das Objekt durch den Detektor falsch erkannt, so taucht statt dem erwarteten Objekt ein anderes Objekt auf. Um solche Fälle zuverlässig ausschließen zu können muss das *Tracking-Problem* gelöst sein. Ein Kalman-Filter kann dazu eingesetzt werden, die Pose eines Objekts zum Zeitpunkt der nächsten Messung vorherzusagen. Taucht an dieser Stelle ein anderes Objekt auf, so kann davon ausgegangen werden, dass es sich um das ursprüngliche Objekt handelt, das lediglich falsch erkannt wurde. Generell kann so auf zuverlässigerem Weg eine Objekttrajektorie erzeugt werden, da dies dank dem Tracking nicht über den Objekttyp geschehen muss.

Die hier vorliegende Arbeit baut auf einem System auf, das weder über Beobachtungs- noch ein Verdeckungsmodell verfügt. Auch ist das Tracking-Problem nicht gelöst, so dass Objekte nur anhand ihres Typs identifiziert werden können. Dies schränkt die Leistungsfähigkeit des Lerners ein. Im weiteren Verlauf werden die einzelnen Lernalgorithmen vorgestellt und die jeweiligen Einschränkungen erläutert.

6.1.2 Bestimmung relevanter Relationen

Wie im letzten Kapitel mehrmals erwähnt wird eine Szene durch die räumlichen Beziehungen der darin enthaltenen Objekte modelliert. Nicht alle Relationen sind notwendig oder geeignet, um die Szene zu beschreiben. Weiterhin skaliert die Anzahl der Relationen nahezu quadratisch zu den in den Szene enthaltenen Objekten, so dass der nötige Rechenaufwand mit zunehmender Szenengröße immer weiter ansteigt.

Es muss also eine Teilmenge der Relationen ausgewählt werden. Diese muss ausreichend sein, um die Szene eindeutig zu beschreiben, so dass es nicht zu Fehlerkennungen im Sinne von *false positives* oder *false negatives* kommen kann. Die besagte Teilmenge wird dann durch das baumförmige Constellation Model dargestellt, auf dem das Object Constellation Model aufbaut.

Für die Auswahl der Relationen wurde derselbe Ansatz wie bei *Meissner et al.* ausgewählt (vergleiche auch [Mei13, Mei14]). Der optimale Relationsgraph wird durch die Anwendung von Heuristiken approximiert. Eine Heuristik lässt sich als eine Methode zur Bestimmung der Übereinstimmung zweier Trajektorien definieren. Im Rahmen dieser Arbeit werden Trajektorien auf Parallelität hin überprüft. Die Bewertung eines Trajektorienpaares hängt - vereinfacht gesprochen - davon ab, zu welchem Prozentsatz beide zueinander (in Position und Orientierung) parallel sind. Details zu dieser Heuristik sind den o.g. Quellen zu entnehmen. Abbildung 6.3 zeigt zwei Beispieltrajektorien, auf welche die Heuristik angewandt wurde. Weitere Relationen sind denkbar, beispielsweise könnte die Stabilität der Rotation eines Objekts um ein anderes bewertet werden. Die Menge der zu verwendenden Heuristiken wird dem System vorgegeben.

Eine Heuristik untersucht Lerndaten in Form von Trajektorien, welche den räumlichen und zeitlichen Verlauf der Objekte in der Szene beschreiben. Alle Trajektorien werden untereinander verglichen, jedem Paar wird ein numerischer Wert zugewiesen, der die Güte der Relation wiedergibt. Die Generierung der Trajektorien aus Einzelmessungen ist problematisch, da das Tracking-Problem nicht gelöst ist. Objekte werden lediglich anhand ihres Typs identifiziert, Fehldetections mit anderem Typ werden also nicht als zur Trajektorie zugehörig erkannt.

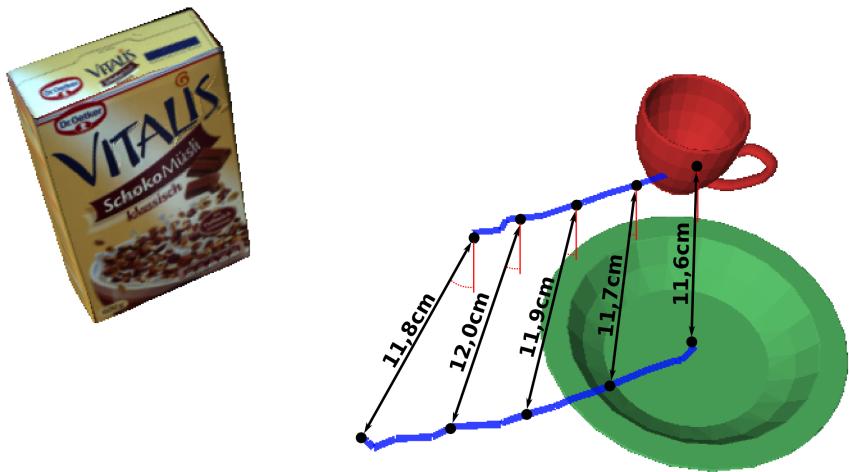


Abbildung 6.3: Tasse und Teller bewegen sich parallel zueinander, wie die Abstände zeigen. Die Winkel (hier in rot) zwischen den einzelnen Messungen variieren ebenfalls nur minimal. Das Paar erhält daher von der im Rahmen dieser Arbeit verwendeten Heuristik eine hohe Bewertung.

Die besten Relationen werden ausgewählt und zur Generierung des Relationsgraphen herangezogen. In den Arbeiten von *Meissner et al.* ist dieser als eine Art Binärbaum realisiert, der durch hierarchisches agglomeratives Clustering gebildet wird. Objekte bzw. deren Trajektorien werden sukzessive zu Clustern zusammengefasst, diese wiederum zu größeren Clustern. Für jeden Cluster wird eine der Trajektorien als Referenz ausgewählt, die dann für den Vergleich mit anderen Clustern genutzt werden kann. Hierzu wird das aus räumlicher Sicht stabilere Objekt gewählt. Die Cluster werden anhand der Bewertungen durch die Heuristiken zusammengefasst. Das Resultat ist ein Binärbaum. An dessen Wurzelknoten werden alle verbliebenen Trajektorien angehängt, deren Relationen derart bewertet wurden, dass sie einen vorgegebenen Schwellwert unterschreiten. Ein Beispiel für den hier vorgestellten Algorithmus ist in Abbildung 6.4 gezeigt.

Der Ansatz wird in unveränderter Form für die vorliegende Arbeit eingesetzt, um einmalig den Relationsbaum zu generieren. Es wird von der Annahme ausgegangen, dass der von den Heuristiken generierte Baum die optimalen Relationen umfasst. Daher ist es nur sinnvoll, dass jedes Object Constellation Model denselben Baum verwendet. Jedes Ob-

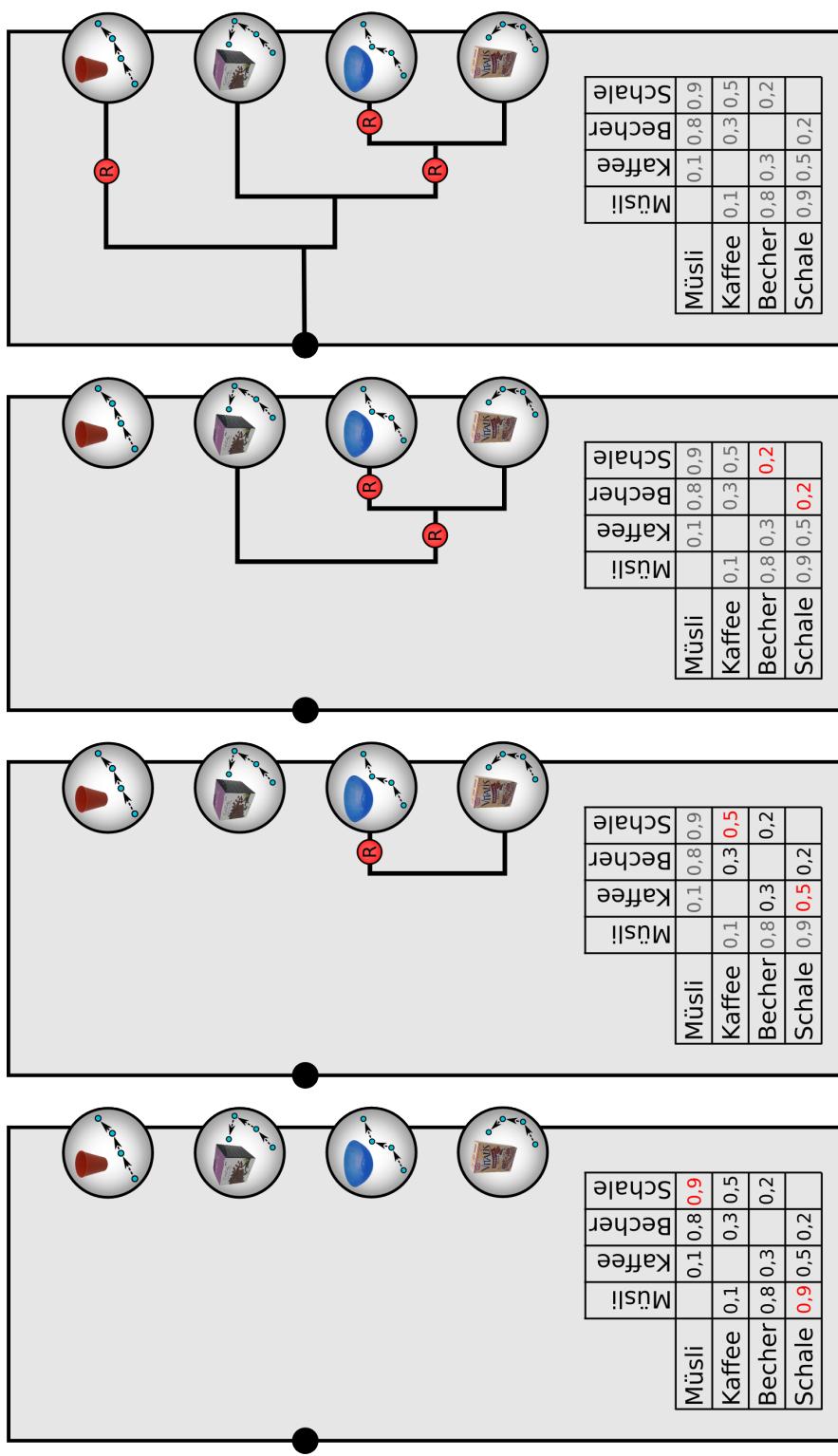


Abbildung 6.4: Demonstration des agglomerativen Clustering, mit dem der Relationsgraph aufgebaut wird. Es sind die Trajektorien von vier Objekten vorgegeben. Die Matrix zeigt die Bewertungen aller Relationen durch eine einzelne Heuristik. Ausgegraute Zeilen stehen für Objekte, die als Kindsknoten an einen Teilbaum angehängt wurden. Aus den roten Einträgen lässt sich ableiten, welche Teilbäume als nächstes zusammengefassst werden. Das "R" kennzeichnet das Referenzobjekt des jeweiligen Clusters.

ject Constellation Model modelliert die Szene von einem bestimmten Szenenobjekt aus. Der Baum wird daher so angepasst, dass der Wurzelknoten dem besagten Referenzobjekt entspricht. Ein Beispiel wird in Abbildung 6.5 gezeigt.

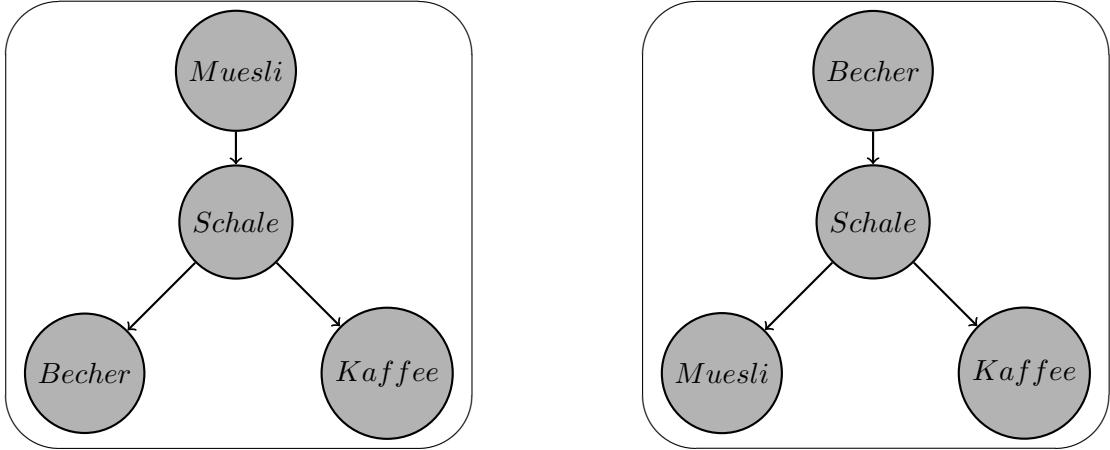


Abbildung 6.5: Der Relationsbaum für das Objekt "Becher" wird so angepasst, dass das Objekt "Muesli" den Wurzelknoten bildet.

6.1.3 Parameter der Szenenerkennung

Im Szenenerkennungsteil des Modells sind zwei Parameter zu bestimmen. Hierbei handelt es sich um die Wahrscheinlichkeit der Szene $P(S)$ und die Auftrittswahrscheinlichkeit des Szenenobjekts $P(O|S)$. Erstere kann nicht sinnvoll bestimmt werden. Es wäre denkbar, die Verteilung über die Anzahl der Lernbeispiele pro Szene zu schätzen. Diese trifft jedoch keine Aussage über die tatsächliche Auftrittshäufigkeit, da einige wenige Beispiele bereits zum Lernen ausreichen. Daher wurde beschlossen, die Verteilung mit einer Gleichverteilung zu initialisieren.

Die Auftrittswahrscheinlichkeit $P(O|S)$ wird durch eine Bernoulli-Verteilung dargestellt. Deren Parameter λ steht dafür, dass das Objekt nicht auftritt und wird wie in Algorithmus 1 beschrieben über die frequentistische Häufigkeit bestimmt. Für jede Demonstration einer Trajektorie wird ermittelt, wie oft das Szenenobjekt gemessen wird. Der ermittelte Wert wird durch die Länge der entsprechenden Trajektorie dividiert. Über alle Demonstrationen wird ein Mittelwert berechnet.

Der Lernalgorithmus arbeitet pro Lernbeispiel eine Reihe von Trajektorien ab und berechnet daraus den Parameter. Keineswegs trivial ist jedoch das Aufstellen der Trajektorie, da hierfür bestimmt werden muss, wann ein Szenenobjekt nicht vorliegt. Die Problematik wurde bereits ausführlich besprochen. Kann ein Szenenobjekt nicht beobachtet werden, so kann dies daran liegen, dass es außerhalb des Blickfelds bzw. verdeckt ist oder vom Detektor als anderes Objekt erkannt wurde. Um diese Fälle ausschließen zu können werden ein Beobachtungs- und Verdeckungsmodell benötigt, zusätzlich muss das

Algorithmus 1 Bestimmen der negierten Auftrittswahrscheinlichkeit

```

1: procedure BESTIMMENEGIERTEAUFTRITTSWAHRSCHEINLICHKEIT(beispiele)
2:   wahrscheinlichkeit  $\leftarrow 0$ 
3:   for all trajektorie  $\in$  beispiele do            $\triangleright$  Iteriere über alle Demonstrationen
4:     z  $\leftarrow 0$ 
5:     for all messung  $\in$  trajektorie do            $\triangleright$  Iteriere über alle Messungen
6:       if messung = Null then z  $\leftarrow z + 1$ 
7:       wahrscheinlichkeit  $\leftarrow$  wahrscheinlichkeit + (z : len(trajektorie))
8:   return wahrscheinlichkeit : len(beispiele)

```

Tracking-Problem gelöst sein. Nur dann kann mit Sicherheit davon ausgegangen werden, dass das Szenenobjekt nicht vorliegt.

Wie bereits erwähnt ist im gegenwärtigen System keines der drei Probleme gelöst. Daher muss mit der Annahme gearbeitet werden, dass ein Objekt nicht aus der Szene entfernt werden kann. Auch wenn es nicht beobachtet wird ist es nach wie vor da, mit der zuletzt ermittelten Pose. Beim Aufbau der Trajektorie wird diese Annahme umgesetzt, indem die letzte Messung kopiert wird. Dies umgeht die ersten beiden Probleme. Fehlerkennungen werden durch das Aussortieren verhältnismäßig kleiner Trajektorien umgangen.

Für den Lernalgorithmus hat dies die folgenden Auswirkungen. Da die Trajektorie das Objekt bei jeder Messung als vorhanden anzeigt, ergeben der Bruch und die anschließende Berechnung des Mittelwerts über alle Beispiele immer den Wert eins. Nach der Normalisierung beschreibt λ das Auftreten des Szeneobjekts als sicheres Ereignis. Auf Grund der auferlegten Einschränkungen kann der Lernalgorithmus an dieser Stelle also kein zusätzliches Wissen zur Verfügung stellen.

6.1.4 Parameter des Object Constellation Model

Die zentrale Gleichung des Object Constellation Model ist in die Terme *Object Appearance*, *Scene Shape* und *Object Existence* gegliedert. Zum Erlernen der Parameter werden zwei Informationsquellen herangezogen, die beide aus den Lernbeispielen extrahiert wurden. Hierbei handelt es sich um den Relationsbaum und die Trajektorien. Der Baum wird dazu eingesetzt, das Modell zu initialisieren und die Trajektorien bereitzustellen. Für jeden Knoten im Relationsgraphen wird ein Slot im Modell angelegt. Jedem Knoten ist noch von der Generierung des Baums eine Trajektorie zugewiesen, diese wird mit dem entsprechenden Slot assoziiert. Anhand der Trajektorien werden die Parameter der oben genannten Terme erlernt. Für *Scene Shape* wird der Relationsbaum als zusätzliche Informationsquelle herangezogen. Nach Abschluss des Lernvorgangs werden die Trajektorien nicht mehr benötigt und verworfen.

Die *Object Appearance* bewertet die Typinformationen der Evidenzen. Über jeden Slot des Modells ist eine Multinomialverteilung definiert, deren Parameter λ wie in Algorithmus 2 beschrieben aus der zugeordneten Trajektorie bestimmt werden muss. Der Vektor umfasst $K + 1$ Elemente, wobei K der Anzahl der in der Szene vorkommenden

Objekte entspricht und durch Abzählen ermittelt werden kann. Aus jeder Demonstration wird die zum Szenenobjekt zugehörige Trajektorie entnommen. Alle Beobachtungen der Trajektorie werden durchlaufen und für jeden Objekttyp das zugehörige Element in λ inkrementiert. Dies geschieht für alle Trajektorien, danach wird der Vektor durch die Summe aller Messungen normalisiert.

Algorithmus 2 Bestimme Appearance-Verteilung

```

1: procedure BESTIMMEAPPEARANCE(beispiele, objekte)
2:   z  $\leftarrow$  0
3:   for i  $\leftarrow$  0 to objekte do                                 $\triangleright$  Initialisiere Tabelle
4:     tabelle[i]  $\leftarrow$  0
5:   for all trajektorie  $\in$  beispiele do                 $\triangleright$  Berechne Tabelle
6:     for all messung  $\in$  trajektorie do
7:       t  $\leftarrow$  typ(messung)
8:       tabelle[t]  $\leftarrow$  tabelle[t] + 1
9:       z  $\leftarrow$  z + 1
10:    for i  $\leftarrow$  0 to objekte do                          $\triangleright$  Normalisiere Tabelle
11:      tabelle[i]  $\leftarrow$  tabelle[i] : z
12:    return tabelle

```

Da das Tracking-Problem nicht gelöst ist werden die Trajektorien anhand des Objekttyps gebildet. Dies hat zur Folge, dass sich die komplette Wahrscheinlichkeitsmasse auf den zugehörigen Eintrag in λ zentriert. Dies verhindert, dass das systematische Fehlerkennungsverhalten der Objektdetektoren modelliert werden kann, wirkt sich darüber hinaus jedoch nicht negativ auf die Erkennung der Szene aus.

Die durch den Baum beschriebenen Relationen fließen in die *Scene Shape* ein. Jeder Slot entspricht einem Knoten im Baum, bis auf den Wurzelknoten sind alle Knoten von mindestens einem anderen Knoten abhängig. Für jede Relation wird wie folgend verfahren. Für jeden Elternknoten wird die zugehörige Trajektorie durchlaufen. Für jede darin enthaltene Beobachtung wird die zeitgleiche Beobachtung aus dem Kindsknoten entnommen. Es liegen zwei Objektposen vor, anhand derer die relative Pose berechnet werden kann. Dieses Vorgehen setzt voraus, dass beide Trajektorien dieselbe Anzahl an Messungen haben, andernfalls muss eine Interpolation vorgenommen werden. Auf Grund begünstigender technischer Umstände wurde das erste Vorgehen vorgezogen.

Wurde dies für alle Beobachtungen ausgeführt, so erhält man eine Menge von relativen Posen, über die eine Gauss-Mischverteilung erlernt wird. Abbildung 6.6 illustriert die relativen Posen und einen darüber gelernten Gauss-Kernel. Hierfür werden die in Abschnitt 3.4 vorgestellten Werkzeuge eingesetzt. Die Parameter der Verteilung werden mit dem EM-Algorithmus ermittelt. Zur Bestimmung der Modellkomplexität, also der Menge der Gauss-Kernel, wird das *Bayesian Information Criterion (BIC)* herangezogen. Insgesamt werden mehrere Gauss-Mischverteilungen mit unterschiedlicher Kernelanzahl erlernt und der Kernel mit der besten Bewertung ausgewählt.

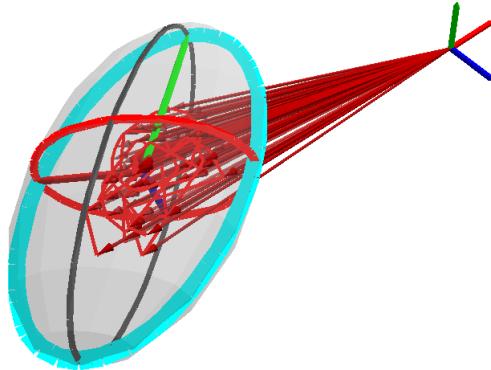


Abbildung 6.6: Lernen einer Relation anhand eines Beispiels. Pfeile kennzeichnen die relativen Positionen, ausgehen vom Koordinatensystem des Elternknotens. Ein einzelner Gauss-Kernel wurde erlernt.

Die Mittelwerte jeden Kernels werden mit der Position eines zufällig gewählten Lerndatums initialisiert. Die Kovarianzmatrix wird als Einheitsmatrix vorgegeben, die mit einem vom Benutzer vorgegebenen Skalar multipliziert wird. Hierdurch kann der Lerner optimal an größere und kleinere Strukturen angepasst werden. Für jede Gauss-Mischverteilung werden mehrere Versionen erlernt, die in im vorherigen Absatz beschriebenen Auswahlprozess einfließen.

Wie bereits erwähnt werden für Position und Orientierung voneinander unabhängige Gauss-Mischverteilungen erlernt. Ursache hierfür sind eine Reihe von Experimenten mit dem Lernverfahren, im Zuge derer das Erlernen einer L-förmige Trajektorie angestrebt wurde. Vom Lerner wurde eine unzureichende Verteilung in Form eines einzigen Gauss-Kernel vorgeschlagen. Nach der Trennung der Pose in Position und Orientierung wurde die Trajektorie zufriedenstellend mit zwei Kernen erlernt. Da mit einem Closed-Source-Lerner gearbeitet wurde konnte über die Gründe nur spekuliert werden. Der Autor sieht hier Potential für weiterführende Forschungen.

Im Rahmen der *Object Existence* müssen die Parameter für eine Verteilung erlernt werden. Die Bernoulli-Verteilungen, welche die Auftrittswahrscheinlichkeiten der assoziier-ten Objekte beschreiben, verfügen je über einen Parameter λ_p für jeden Slot p . Der Pa-rameter gibt wieder, mit welcher Wahrscheinlichkeit ein Objekt nicht detektiert wurde. Die relative Häufigkeit hierfür wird mit Algorithmus 1 aus den vorliegenden Demonstra-tionen ermittelt. Da kein Beobachtungs- und Verdeckungsmodell existieren und daher das Objekt immer als vorhanden angenommen wird, konzentriert sich die komplette Wahrscheinlichkeitsmasse auf das Vorhandensein des Objekts.

Das Fehlen eines Beobachtungs- und Verdeckungsmodells sowie das ungelöste Tracking-Problem erlauben es nicht, dass akkurate Trajektorien zur Verfügung stehen. Dies verhin-dert, dass die hier geschilderten Lernverfahren ihr volles Potential entfalten und erschwe-ren dadurch das Lernen der Parameter. Dies hat jedoch keine negativen Auswirkungen auf die eigentliche Erkennungsleistung der Szene. Vielmehr wird hierdurch verhindert,

dass der Ansatz zusätzliche Robustheit gegenüber Detektorfehlern gewinnt.

6.1.5 Sample Relaxation

Die menschliche Wahrnehmung erlaubt bei der Interpretation von Szenen große Toleranzen. Steht eine Tasse wenige Zentimeter von dem Punkt entfernt, an dem sie ursprünglich beobachtet wurde oder ist um wenige Grad verdreht, so kann der Mensch die Szene nach wie vor erkennen. Das PSM hingegen toleriert nur exakt eingehaltene Posen. Detektorrauschen und geringfügige Fehlpositionierungen können daher ein Problem sein. Zwar kann der Demonstrator die Objektposen durch Verschieben und Drehen aufweichen, dies ist jedoch umständlich und auch nicht immer möglich.

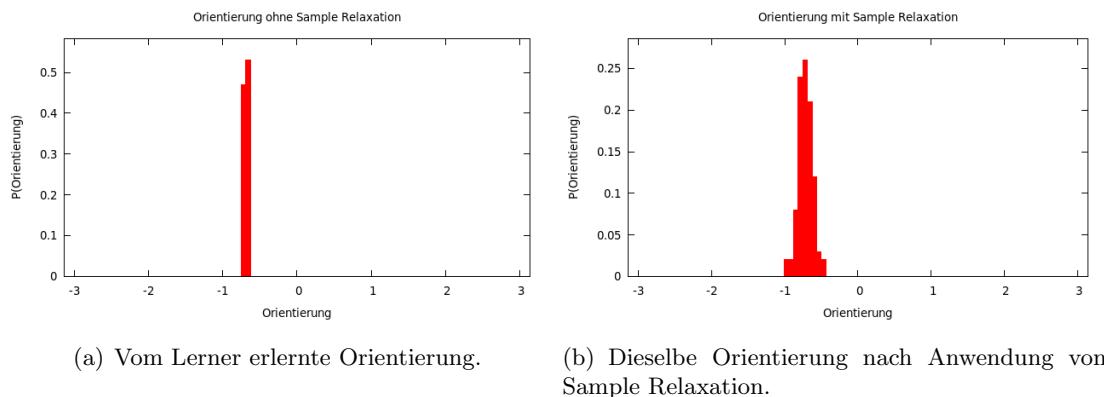


Abbildung 6.7: Histogramm über die Orientierung in Radian vor und nach Anwendung der Sample Relaxation. Es wurden 20 zusätzliche Stichproben im Intervall von plus/minus 5 Grad generiert.

Aus diesem Grund wurde die *Sample Relaxation*-Technik entwickelt. Für jedes Lerndatum werden mehrere zusätzliche Datenpunkte generiert. Dies geschieht relativ zur Pose des Datums und erfolgt mittels einer Gleichverteilung, um dem Anwender Kontrolle über die genauen Ausmaße der Aufweichung zu geben. Für Position und Orientierung lassen sich Werte in Metern und Grad sowie die gewünschte Anzahl der Stichproben übergeben. Hiermit kann der Prozess exakt eingestellt werden.

Abbildung 6.7 zeigt die Orientierung eines Objekts um eine seiner Achsen. Ohne Verrauschen der Lerndaten ist der vom Lerner erzeugte Gauß-Kernel sehr spitz. Nach Anwendung der Sample Relaxation ist eine deutliche Erhöhung der Varianz zu erkennen. Versuche haben gezeigt, dass eine Aufweichung der Orientierung schon um wenige Grad die Robustheit der Szenenerkennung stark steigern kann.

6.2 Inferenz der Szene

Unter Inferenz versteht man die Auswertung des Szenenmodells für eine Menge von Objektevidenzen, um die Wahrscheinlichkeit der Szene zu bestimmen. An dieser Stelle

wird der komplette Inferenzprozess zusammengefasst. Der Abschnitt ist in zwei Teile untergliedert. Zunächst wird der Prozess von den Objektevidenzen ausgehend bis hin zu den OCMs besprochen. Die Inferenz innerhalb des Object Constellation Model ist komplexer und wird daher in einem separaten Teil behandelt.

6.2.1 Inferenz im Szenenmodell

Dem Szenenmodell werden die beobachteten Objektevidenzen übergeben. Diese werden durch die beiden Vektoren \mathbf{A} und \mathbf{X} beschrieben, welche Typen und Posen aller Evidenzen umfassen. Die einzelnen Object Constellation Model sind anhand der zugehörigen Szenen zusammengefasst. Innerhalb einer Szene werden die Evidenzen an die entsprechenden OCMs weitergereicht. Das von diesen gelieferte Ergebnis wird mit der Auftrittswahrscheinlichkeit des Referenzobjekts gewichtet. Dasjenige Object Constellation Model mit der höchsten Wahrscheinlichkeit ist das Modell, welches die vorliegenden Evidenzen am besten beschreibt. Dessen Wahrscheinlichkeit entspricht auch der Gesamtwahrscheinlichkeit der Szene. Abbildung 6.8 zeigt ein Beispiel für die Erkennung einer Szene mit zwei Objekten.

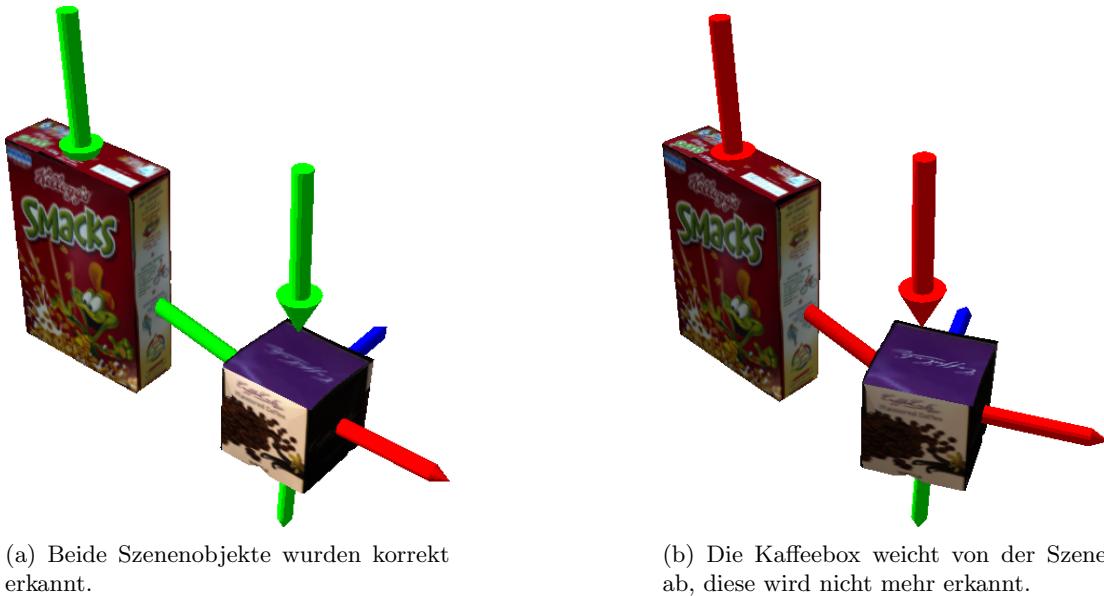


Abbildung 6.8: Visualisierung der Szenenerkennung für eine Szene mit zwei Szenenobjekten. Das Erkennungsergebnis des jeweiligen Szenenobjekts wird als senkrechter Pfeil visualisiert, Relationen durch Linien. Die Farben der Pfeile und Relationen spiegeln die damit assoziierten Wahrscheinlichkeiten wieder, wobei Grün hohe und Rot niedrige Werte bezeichnet.

Dasselbe Vorgehen wird in der Rückweisungsklasse durchgeführt. Ein gleichverteiltes Object Constellation Model wird ausgewertet, wobei für jede Evidenz ein Slot im Modell

erzeugt wird. Abschließend werden die von den Szenen und der Hintergrundklasse gelieferten Ergebnisse miteinander verglichen. Das Maximum entschiedet über die aktuell vorliegende Szene.

6.2.2 Inferenz im OCM

An dieser Stelle wird die Auswertung des Object Constellation Model für gegebene Objektevidenzen beschrieben. Ziel ist die Berechnung der Wahrscheinlichkeit, mit der das hierdurch modellierte Szenenobjekt vorhanden ist. Die Vektoren \mathbf{A} und \mathbf{X} werden dem Modell übergeben. Es wird eine Liste von Hypothesen generiert, welche die Zuordnung von Evidenzen zu Slots beschreiben. Der zu Grunde liegende Mechanismus entspricht dem Ziehen mit Zurücklegen, um die mehrfache Belegung von Slots mit dem Nullobjekt gewährleisten zu können, das wiederum keiner Zuordnung einer Evidenz zu einem Slot entspricht.

Ausschlusswissen wird angewandt, um ungültige Hypothesen auszusortieren. Hierunter fallen diejenigen, denen kein Referenzobjekt zugewiesen wird und auch solche, bei denen eine Evidenz mehreren Slots zugewiesen wird. Jeder Eintrag der verbliebenen Liste wird mit der zentralen Gleichung des Object Constellation Model bewertet. Es ist damit zu rechnen, dass vielen Hypothesen nur eine relativ geringe Wahrscheinlichkeit zugeordnet wird, da sie Clutter oder Falschzuordnungen beinhalten. Einige wenige jedoch enthalten korrekte Zuordnungen ohne Clutter und erzielen dementsprechend hohe Bewertungen. Alle Teilergebnisse werden addiert und bilden so die Wahrscheinlichkeit des Object Constellation Model.

7. Implementierung

Bisher wurde das im Rahmen dieser Arbeit entwickelte Konzept nur aus theoretischer Sicht besprochen. An dieser Stelle soll daher nun auf den praktischen Teil eingegangen werden - die Implementierung. Das Kapitel unterteilt sich in zwei Abschnitte. Zuerst werden kurz auf das Robot Operating System und das bereits bestehende System vorgestellt. Letzteres bietet den Ansatzpunkt für das hier entwickelte probabilistische Szenenmodell, welches im zweiten Abschnitt erläutert wird.

Die wichtigsten Pakete werden aufgelistet und ein Überblick über deren Interaktionen und Abhängigkeiten gegeben. Anschließend werden die einzelnen Pakete nacheinander abgehandelt. Von den wichtigsten Architekturen werden Klassendiagramme präsentiert und besprochen.

7.1 Das bestehende System

In diesem Abschnitt wird das bereits bestehende, von *Meissner et al.* entwickelte Active-Vision-System vorgestellt und eine Einordnung der vorliegenden Arbeit gegeben. Das System baut auf dem *Robot Operating System (ROS)* auf [QCG⁺09]. Herbei handelt es sich um ein Open-Source Meta-Betriebssystem, welches auf einem traditionellen Betriebssystem aufsetzt. Die Entwicklung von ROS begann zuerst an der Stanford Universität und wird heute hauptsächlich durch die Firma Willow Garage fortgesetzt.

Durch ROS soll der Aufwand der Softwareentwicklung im Bereich der Robotik reduziert werden. Die Hauptaufgaben sind Hardwareabstraktion und Visualisierung, sowie die Bereitstellung einer Kommunikationsinfrastruktur und Paketverwaltung¹. Auch stehen für viele der üblichen Sensoren (wie z.B. die Kinect) und die üblichen Robotikprobleme (wie z.B. Navigation) Softwarelösungen zur Verfügung. Ein auf ROS basierendes

¹Ein ausführlicher Überblick ist im deutschen und englischen Wikipedia-Eintrag zu finden, denen auch einige der hier genannten Informationen entnommen wurden.

System ist in sogenannte Knoten gegliedert. Hierbei handelt es sich eigenständige Programme, die über die zur Verfügung gestellte Kommunikationsinfrastruktur miteinander kommunizieren. Die Knoten sind zusammen mit anderen Ressourcen wie Bibliotheken und Konfigurationsdateien in Paketen gegliedert.

Die Infrastruktur des Active-Vision-System wurde von *Valerij Wittenbeck* aufgebaut [Wit13]. Es wurden ein Flock-of-Birds Magnetfeldtracker sowie eine Pan-Tilt-Unit mit aufgesetztem Kamerasystem eingebunden. Darauf aufbauend wurden drei zentrale Probleme gelöst, namentlich das Erlernen der kinematischen Kette zwischen den einzelnen Komponenten, die Objektsuche und die Objektverfolgung. Das eingesetzte Objekterkennungssystem stammt von *Pedram Azad* [Aza08].

Darauf aufbauend wurde von *Reno Reckling* eine Szenenerkennung entwickelt [Rec13]. Diese verfolgt einen nicht-parametrischen Ansatz auf Basis des Implicit Shape Model. Das aus beiden Arbeiten resultierende System ist der Ausgangspunkt für mehrere Veröffentlichungen [Mei14, Mei14].

7.2 Das entwickelte System

Die hier vorliegende Arbeit baut auf der von Wittenbeck geschaffene Infrastruktur auf. Die stellt einen parallelen Ansatz zu der bestehenden, von Reckling entwickelten Szenerkennung dar. Es wurden ca. 17000 Zeilen Quellcode geschrieben. Die Arbeit wurde wie für ROS üblich in unterschiedliche Pakete eingeteilt. Die folgende Liste nennt den Paketnamen sowie die darin enthaltene Komponente:

- Posenmodell in *resources_for_psm*
- Probabilistisches Szenenmodell in *psm*
- Relationsgraph-Generator in *relation_graph_generator*
- Visualisierung in *visualisation_server*

Das bereits bestehende Paket *scene_graph_generator* wurde erweitert und angepasst. Einige der hier genannten Pakete beinhalten gemeinsam genutzten Programmcode, andere aktiv mit dem Rest des Systems kommunizierende Software. Weiterhin wurde eine Fremdbibliothek namens *ProBT* eingebunden, die ebenfalls in Paketform vorliegt. Laut Website des Herstellers *ProbaYes* handelt es sich hierbei um "a formalism, a methodology, an API and an inference engine to solve problems with incomplete and uncertain information"². ProBT wurde im Rahmen dieser Arbeit eingesetzt, um einen Teil der probabilistischen Berechnungen durchzuführen, aber auch für andere Operationen wie das Lernen von Gauss-Mischverteilungen.

Abbildung 7.1 zeigt die Abhängigkeiten zwischen den einzelnen Softwarepaketen. Aus Gründen der Erweiterbarkeit wurden die Visualisierung in einem separaten Paket untergebracht, wodurch auch das Posenmodell ausgelagert werden musste. Letzteres wird

²<https://team.inria.fr/>

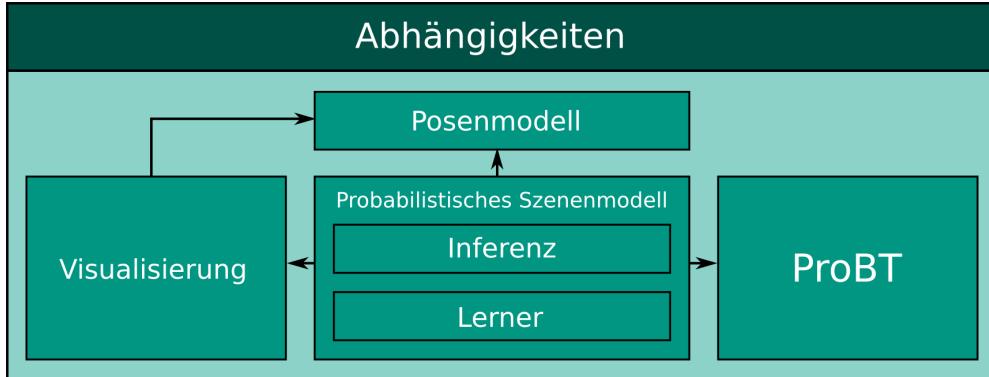


Abbildung 7.1: Abhängigkeiten zwischen den Paketen. Das probabilistische Szenenmodell ist der zentrale Teil der Arbeit. Das Posenmodell beschreibt Objektposen. Die Visualisierung stellt die Ergebnisse von Lerner und Inferenz dar, die beiden letzteren nutzen von ProBT zur Verfügung gestellte Werkzeuge.

hauptsächlich durch das probabilistischen Szenenmodell genutzt. Das Szenenmodell ist in den Lerner und die Inferenz unterteilt. Beide setzen ProBT ein. Der Lerner ist für das Erlernen des Modells aus Beispielen zuständig, die Inferenz nutzt das Modell und gegebene Beobachtungen, um darauf basierend die Szenen zu erkennen.

Abbildung 7.2 verdeutlicht die Interaktionen zwischen den einzelnen Knoten. Es beschreibt, wie die hier implementierten Pakete untereinander und mit dem bereits bestehenden System kommunizieren. Die kinematische Kette stellt die Transformationen zwischen den einzelnen Koordinatensystemen zur Verfügung. Der Objekterkennung liefert die beobachteten Objekte an die Inferenz im Szenenmodell. Der Szenengraph-Generator konkateniert die Beobachtungen und erzeugt für jede Objektinstanz eine Trajektorie. Diese wird durch den Lerner zum Erlernen der Modellparameter genutzt. Der Relationsgraph-Generator liefert die Relationen, die die globale Nachbarschaft eines jeden Szenenmodells beschreiben. Sowohl Lerner als auch Inferenz nutzen die Visualisierung, um ihre Ergebnisse zu präsentieren.

7.2.1 Posenmodell

Das Posenmodell ist im Paket *resources_for_psm* gekapselt. Es stellt das gemeinsame Element zwischen dem probabilistischen Szenenmodell und der Visualisierung dar. Die Auslagerung erlaubt es, das Posenmodell auch in zukünftigen Projekten zu nutzen. Außerdem wird hierdurch die Unterbringung der Visualisierung in einem separaten Paket möglich.

Implementiert ist das Posenmodell als einzelne Klasse, welche die Pose eines einzelnen Objekts kapselt. Die Position wird durch einen dreidimensionalen Vektor beschrieben, die Orientierung über ein Quaternion. Es werden eine Reihe von Funktionalitäten zur

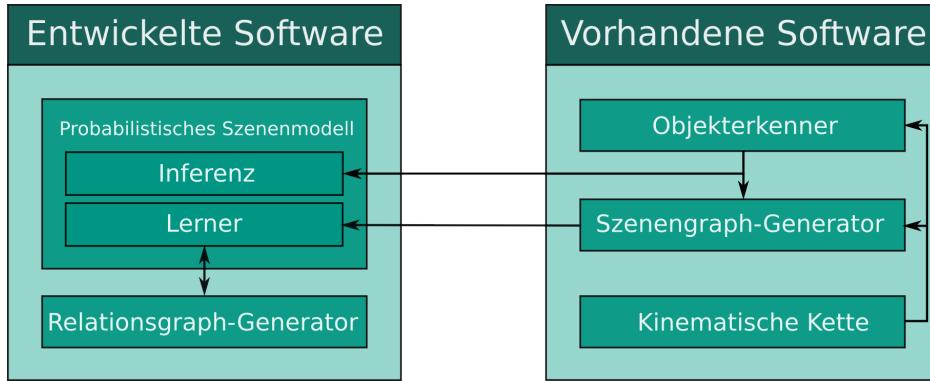


Abbildung 7.2: Interaktion zwischen den entwickelten und bereits vorhandenen Software-paketen. Der Objekterkennner liefert Objektevidenzen. Der Szenengraph-Generator fasst diese zu Trajektorien zusammen, welche vom Lernen benötigt werden. Der Relationsgraph-Generator ermittelt aussagekräftige Relationen innerhalb einer Szene. Die Inferenz bestimmt anhand der gelieferten Beobachtungen die Szene. Die kinematische Kette transformiert zwischen den unterschiedlichen Koordinatensystemen.

Verfügung gestellt. So etwa die Konvertierung der Pose auf drei anderen Datentypen, die an anderen Stellen der Implementierung durch eingebundene Bibliotheken bedingt sind. Auch die Berechnung der relativen Pose, wie sie für Lernen und Inferenz nötig ist, ist hier enthalten.

Sowohl für die Repräsentation als auch die Berechnungen wird die von ROS mitgelieferte mathematische Bibliothek *Eigen* verwendet, welche Datenstrukturen und Algorithmen aus der linearen Algebra zur Verfügung stellt.

7.2.2 Probabilistisches Szenenmodell

Das auch als PSM bezeichnete probabilistische Szenenmodell bildet das Herzstück der Arbeit. Das zugehörige Paket *psm* beinhaltet sowohl den Lerner, der auf Basis von Beispielen neue Modelle erzeugt, als auch die Inferenz, die basierend auf einem erlernten Modell die vorliegende Szenen bestimmt. Die Implementierungen beider Programme werden hier vorgestellt. Zusätzlich wird als Bindeglied zwischen beiden auf das verwendete Modell eingegangen.

7.2.2.1 Lerner

Der Lerner nutzt die vom Szenengraph-Generator erzeugten Trajektorien der in der Szenen enthaltenen Objekte, um daraus die Parameter des Szenenmodells zu erlernen. Die gewählte Architektur spiegelt die Struktur des Problems wieder und wir din Abbildung 7.3 gezeigt. Eine zentrale Klasse kapselt die Lerner für die einzelnen Szenen, diese wiederum kapseln die Lerner für die einzelnen *OCM*. Ein ausführlicherer Einblick in die Implementierung wird im weiteren Verlauf gegeben.

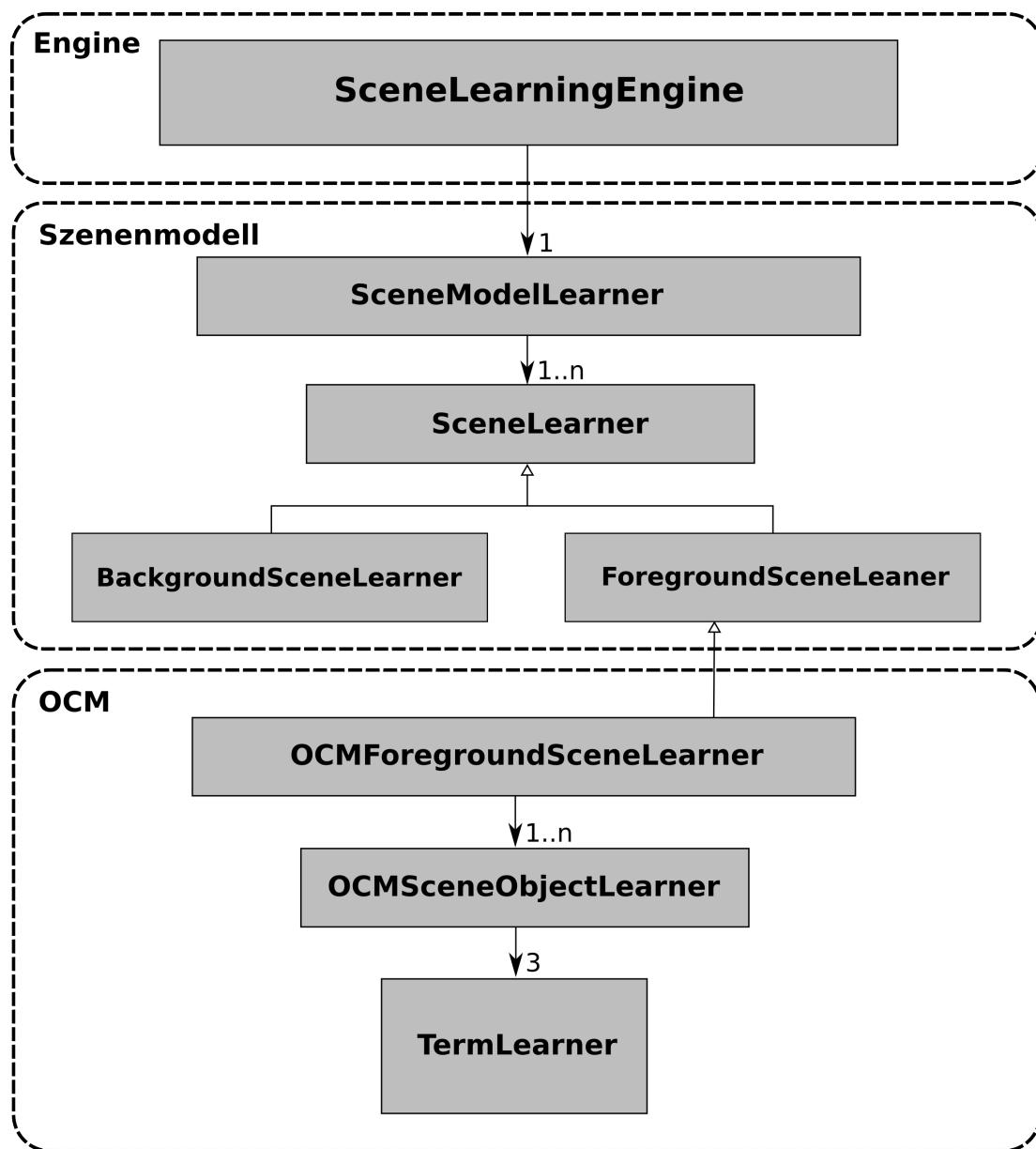


Abbildung 7.3: Klassendiagramm des Lerners. Die Architektur ist in die drei Teile Engine, Szenenmodell und OCM untergliedert. Ersteres bildet die Schnittstelle und kapselt das eigentliche Modell, welches durch die letzteren beiden Abschnitte gebildet wird.

Die Klasse *SceneLearningEngine* dient als Schnittstelle zum Rest des Systems. Sie konfiguriert das System anhand der Übergabeparameter und verwaltet außerdem die Ressourcen für das Speichern und die Visualisierung des gelernten Modells. Darüber hinaus nimmt sie die Trajektorien entgegen, die entweder über die Kommunikationsinfrastruktur von ROS eintreffen oder aus einer Datei geladen werden. Diese werden an den *SceneModelLearner* weitergegeben, der die Lerner für die einzelnen Szenen verwaltet. Die Trajektorien werden auf die zugehörigen Lerner verteilt, wobei bei Bedarf ein neuer Lerner angelegt wird. Für die Hintergrundklasse ist ein separater Lerner vorhanden, dem alle Trajektorien übergeben werden.

Die Lerner für Vorder- als auch Hintergrundklasse erben von der abstrakten Basisklasse *SceneLearner*, welche eine gemeinsame Schnittstelle für das Lernen, Speichern und Visualisieren zur Verfügung stellt. Der Lerner für die Hintergrundklasse *BackgroundSceneLearner* erfasst und speichert lediglich die Anzahl der unterschiedlichen Objekttypen. Der *ForegroundSceneLearner* ist eine abstrakte Klasse, um unterschiedliche Repräsentationen einer Szene bzw. der darin enthaltenen Szenenobjekte zu erlauben. Im Rahmen dieser Arbeit wurde nur ein Ansatz verfolgt, namentlich die Repräsentation der Szenenobjekte durch das *OCM*. Die Unterklasse *OCMForegroundSceneLearner* kapselt die Lerner für das *OCM*, welche durch die Klasse *OCMSceneObjectLearner* implementiert werden.

Wie aus Kapitel 6 ersichtlich findet ein Großteil des eigentlichen Lernens im Object Constellation Model statt. Hier wird zunächst der Relationsgraph-Generator aus dem Paket *relation_graph_generator* dazu eingesetzt, den Relationsgraphen zu generieren. Hieraus und aus den Trajektorien wird eine gemeinsame Datenstruktur erzeugt. Für jeden der einzelnen Terme *Scene Shape*, *Object Appearance* und *ObjectExistence* der zentralen OCM-Gleichung ist ein Lerner vorhanden, der von der abstrakten Klasse *TermLearner* erbt. Die zuvor erwähnte gemeinsame Datenstruktur wird diesen Lernern übergeben, die daraus die Parameter des jeweiligen Terms bestimmt.

7.2.2.2 Modell

Das Modell wird in Form einer XML-Datei abgelegt. Dieses für den Menschen einfach zu entziffernde und editierbare Datenformat wurde vorgezogen, damit das Modell gegebenenfalls von Hand verändert werden kann. So können Modellparameter getestet werden, die auf Grund des ungelösten Tracking-Problems nicht erlernt werden können. Auch kann so einfach ein Modell aus den Szenen verschiedener Modelle zusammengefasst werden.

Abbildung 7.4 zeigt das Szenenmodell einer Frühstücks- sowie der obligatorischen Hintergrundszene. Der Vollständigkeit ist für jede Szenen eine apriori-Wahrscheinlichkeit gegeben, diese ist jedoch wie im Theorieteil beschrieben gleichverteilt. Die Hintergrundszene benötigt nur Informationen über die Anzahl der unterschiedlichen Objekttypen und das Volumen des Arbeitsraums. Es ist zu beachten, dass letzteres nochmals im Object Constellation Model genannt wird. Diese Redundanz vereinfacht den Ladeprozess.

Die Szene mit der Bezeichnung *breakfast* enthält die beiden Szenenobjekte *CoffeeBox* und *Cup*. Aus Gründen der Übersichtlichkeit ist nur ersteres komplett ausgeklappt. Jedes

```

-<psm>
-<scenes>
  -<scene type="background" name="background" priori="0.5">
    <description objects="2" volume="27"/>
  </scene>
  -<scene type="ocm" name="breakfast" priori="0.5">
    -<object name="CoffeeBox" type="ocm" priori="0.5">
      <slots number="2"/>
    -<shape>
      -<root volume="27">
        -<child name="Cup">
          +<pose></pose>
        </child>
      </root>
    </shape>
    -<appearance>
      -<mapping>
        <map id="1" name="CoffeeBox"/>
        <map id="2" name="Cup"/>
      </mapping>
    -<table>
      <entry values="0 1 0"/>
      <entry values="0 0 1"/>
    </table>
  </appearance>
  -<occlusion>
    -<table>
      <entry values="0 1"/>
      <entry values="0.5 0.5"/>
    </table>
  </occlusion>
  </object>
  +<object name="Cup" type="ocm" priori="0.5"></object>
</scene>
-<scenes>
</psm>

```

Abbildung 7.4: Das Szenenmodell einer Frühstücksszene mit Kaffeekanne und Tasse. Das Modell umfasst eine Vorder- und Hintergrundszene. Erstere ist aus den beiden besagten Objekten aufgebaut, welche die Parameter für die drei Terme des OCM enthalten.

Szenenobjekt wird mit seiner relative Auftrittswahrscheinlichkeit gewichtet, die hier mit "apriori" bezeichnet wird. Das Szenenobjekt enthält Informationen über die Anzahl der Slots. Diese Information kann aus dem Rest des Modells hergeleitet werden, vereinfacht jedoch den Ladevorgang und wird zur Modellverifikation herangezogen. Weiterhin sind für jeden Term des OCM die entsprechenden Parameter abgelegt.

Der *Scene Shape*-Term enthält einen Baum, der die Relationen zwischen den Objekten beschreibt. Da hier nur ein weiteres Objekt vorhanden ist liegt besteht der Baum nur aus einem Wurzel- und einem Kindsknoten. Ersterer enthält erneut das Volumen des Arbeitsraums, letzterer die Informationen zur Gauss-Mischverteilung, welche die relative Pose im Koordinatensystem des Elternknoten beschreibt. Der entsprechende Tag ist aus Gründen der Übersichtlichkeit eingeklappt, enthält jedoch die Gewichte, Mittelwerte und Kovarianzmatrizen der einzelnen Gauss-Kernel.

Die *Object Appearance* ist in zwei Abschnitte gegliedert. Zunächst folgt eine Abbildung von Objekttypen auf Indices einer Wahrscheinlichkeitstabelle. Die besagte Tabelle ist für jeden Slot des Models definiert und beschreibt die Auftrittswahrscheinlichkeit des assoziierten Typs. Dem aufmerksamen Betrachter mag auffallen, dass die Abbildungen keinen Index mit dem Wert Null umfassen. Dieser ist für unbekannte Objekttypen vorbehalten, der erste Wert jeder Wahrscheinlichkeitstabelle ist hierfür definiert.

Im Fall der *Object Appearance* ist keine Abbildung notwendig. Für jeden Slot eine Wahrscheinlichkeitstabelle definiert, welche die Auftrittswahrscheinlichkeit des assoziierten Objekts beschreibt. Es ist zu beachten, dass die Beobachtung des Referenzobjekts als sicheres Ereignis gilt. Ist es nicht vorhanden, so kann das zugehörige Szenenobjekt auch nicht evaluiert werden. Umgekehrt formuliert wird das Szenenobjekt immer genau dann evaluiert, wenn das Referenzobjekts vorhanden ist, das Auftreten von letzterem muss also als sicheres Ereignis angesehen werden.

7.2.2.3 Inferenz

Die Architektur der Inferenz zeigt starke Ähnlichkeiten zu der des Lerners. Abbildung 7.5 zeigt das komplette Klassendiagramm. Als Schnittstelle zur Außenwelt fungiert die Klasse *SceneInferenceEngine*. Sie parametrisiert das System und kapselt die Ressourcen für den Ladevorgang des Modells und die Visualisierung. Weiterhin nimmt sie die von den Objektdetektoren über die Kommunikationsinfrastruktur gesendeten Objekt-Evidenzen entgegen und übergibt sie der Klasse *ObjectEvidence*, welche sie bis zur Durchführung des nächsten Updates sammelt.

Die gesammelten Objekt-Evidenzen werden dann der Klasse *SceneModelDescription* übergeben. Diese hat zuvor bereits das übergebene Modell geladen. Für jede Szene im Modell wird eine Instanz der Klasse *SceneDescription* erstellt. Ob hierdurch eine Szenen oder die Hintergrundklasse beschrieben wird hängt von der Unterkategorie der darin gekapselten abstrakten Klasse *SceneContent* ab. Von deren Unterklassen *ForegroundSceneContent* und *BackgroundSceneContent* werden je nach Bedarf entsprechende Instanzen erzeugt.

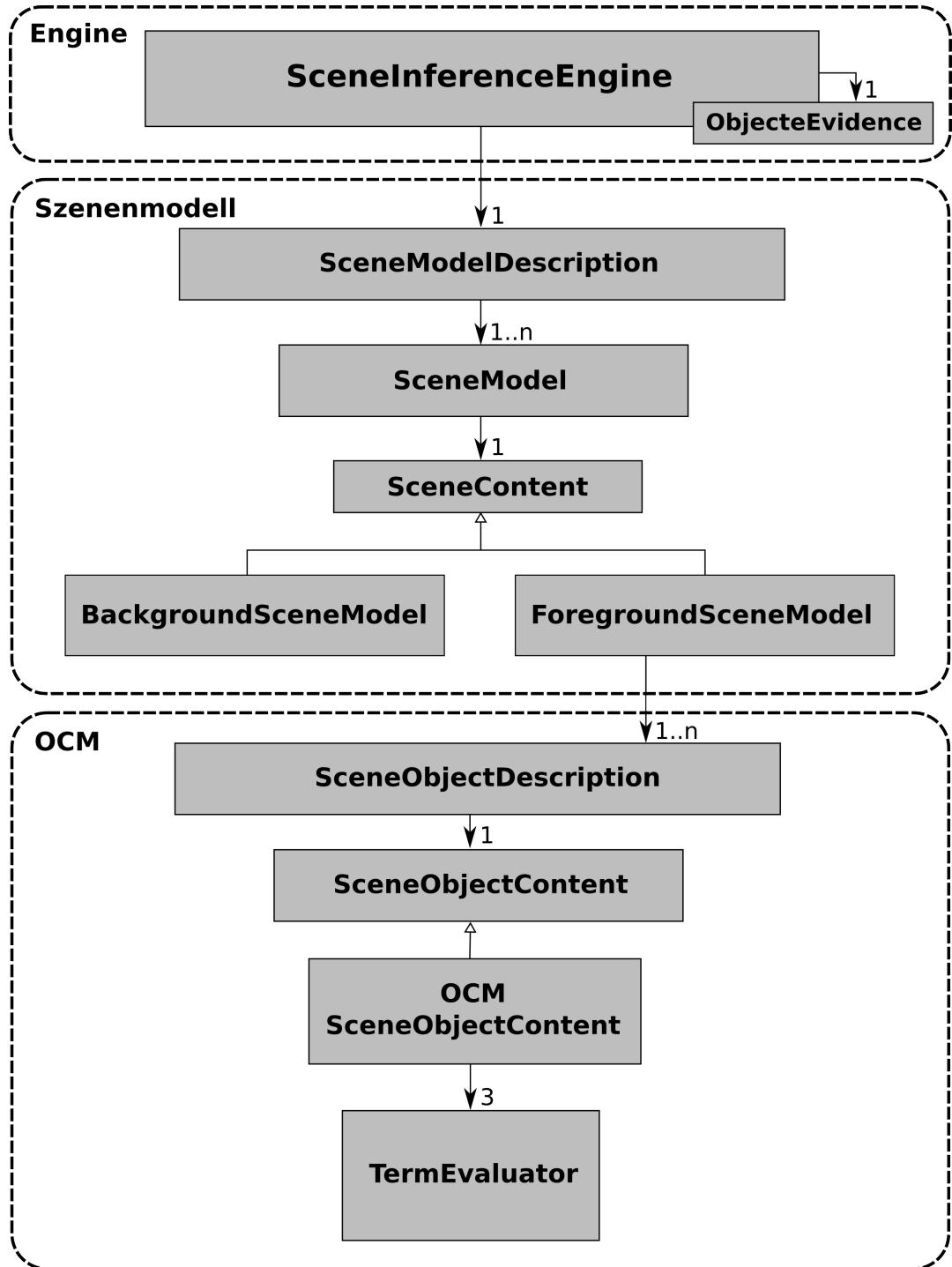


Abbildung 7.5: Klassendiagramm der Inferenz. Die Architektur ähnelt der des Lerners. Auch hier sind wieder die drei Teile Engine, Szenenmodell und OCM vorhanden, die Schnittstelle und Modell realisieren.

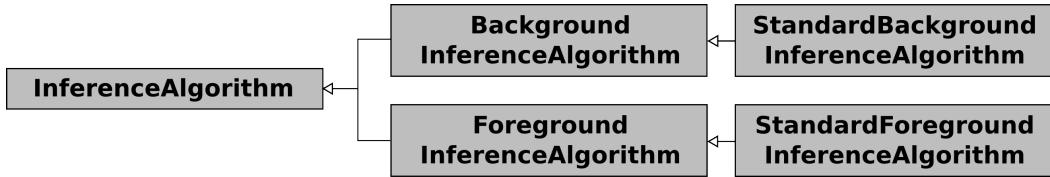


Abbildung 7.6: Klassendiagramm zur OCM-Fusion. Im Verlauf der Arbeit wurde mit mehreren Inferenzalgorithmen experimentiert, welche für die Fusion der einzelnen OCM verantwortlich sind. In der finalen Implementierung ist nur noch der Standard-Algorithmus vorhanden.

In der Klasse *ForegroundSceneContent* sind Instanzen von *SceneObjectDescription* gekapselt. Wie schon beim Lerner ist hier der Fall berücksichtigt, dass eine Szene durch mehrere Ansätze repräsentiert werden kann. Hierzu enthält die angesprochene Klasse die abstrakte Klasse *SceneObjectContent*. Da im Rahmen dieser Arbeit nur das Object Constellation Model behandelt wurde, existiert nur die Unterkategorie *OCMSceneObjectContent*.

Diese Klasse implementiert den Erkenner für ein einzelnes Object Constellation Model. Die Architektur ist analog zu der des Lerners. Es existiert eine Basisklasse *TermEvaluator*, von der für jeden der drei Terme *Scene Shape*, *Object Appearance* und *Object Existence* eine Unterkategorie existiert. Die Hypothesen werden in der Klasse *OCMSceneObjectContent* erzeugt und dann nacheinander durch die *TermEvaluator*-Unterklassen bewertet.

Im Zuge der Arbeit wurden mit mehreren Algorithmen zur Fusion der einzelnen Szenenobjekte experimentiert. Letztendlich hat sich hieraus das in Kapitel 5 beschriebene Verfahren entwickelt. Zur Vereinfachung der Entwicklung wurde die Architektur so ausgelegt, dass per Konfiguration ein beliebiger Fusionsalgorithmus vorgegeben werden kann. Dieser Mechanismus ist nicht mehr unbedingt notwendig, wurde jedoch für den Fall zukünftiger Anpassungen des Systems beibehalten. Es wurde eine Basisklasse *InferenceAlgorithm* sowie zwei Unterklassen *ForegroundInferenceAlgorithm* und *BackgroundInferenceAlgorithm* für Szene und Hintergrundklasse implementiert. Der finale Fusionsalgorithmus mit der Bezeichnung *Standard* wurde für beide Klassen separat implementiert, um einen Performance-Vorteil bei der Berechnung der Hintergrundwahrscheinlichkeit zu erzielen. Abbildung 7.6 zeigt das zugehörige Klassendiagramm.

7.2.3 Relationsgraph-Generator

Der im Paket *relation_graph_generator* enthaltene Relationsgraph-Generator liefert den Graphen bzw. Baum mit den Relationen, der die globale Nachbarschaft eines jeden Szenenobjekts beschreibt. Ein Teil der Implementierung stammt aus der Arbeit von *Reckling*. Sowohl das ISM, als auch das probabilistische Szenenmodell nutzen die Heuristiken, mit denen die Relationen zwischen den Objekten einer Szene auf Relevanz für die Szenenerkennung hin untersucht werden. Der dies betreffende Code wurde übernommen. Darauf

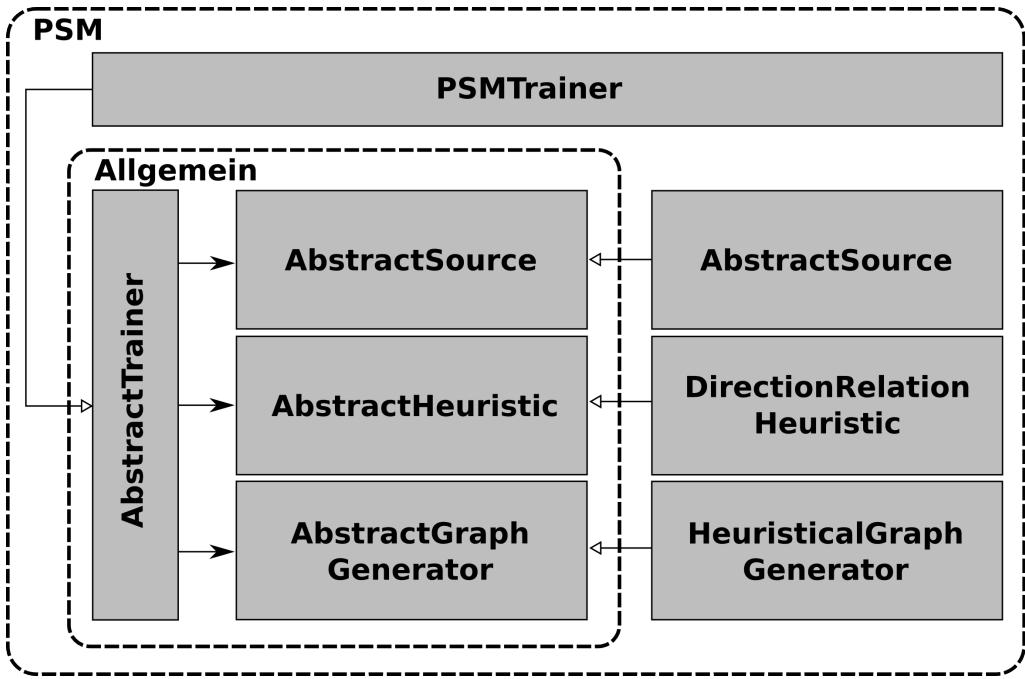


Abbildung 7.7: Klassendiagramm des Relationsgraph-Generators. Der allgemeine Teil wird von beiden Ansätzen zur Szenenerkennung genutzt. Das PSM implementiert eine eigene Schnittstelle und Unterklassen.

aufbauend wurde der in Kapitel 6 beschriebene Cluster-Algorithmus implementiert, mit dem der Relationsgraph aufgebaut wird.

Da bei dieser Arbeit der reine Proof-of-Concept im Vordergrund steht wurde nur das Lernen von Baumstrukturen umgesetzt. Der Algorithmus kann jedoch leicht auf das Lernen des im Theorieteil beschriebenen kompletten Graphen erweitert werden. Die Heuristik für die Suche zeitlich und räumlich paralleler Trajektorien wurde übernommen. Neue Heuristiken wurden nicht implementiert, da zusätzliche Heuristiken den Vergleich zwischen dem ISM und dem hier umgesetzten probabilistischen Szenenmodell erschweren würde.

Um eine gemeinsame Schnittstelle zu schaffen, die von Szenenerkennungssystemen genutzt werden kann, wurde eine entsprechende Softwarearchitektur gewählt und umgesetzt. Die abstrakte Klasse *AbstractTrainer* kapselt eine Reihe ebenfalls abstrakter Komponente, die für die Generierung des Graphen benötigt werden. Da wäre die Klasse *AbstractSource*, welche die übergebenen Trajektorien in ein internes Format umwandelt. Im Fall des ISM stammen die Daten aus einer Datenbank, hier werden sie direkt aus einer ROS-Nachricht extrahiert. Je nach verwendeten System wird also eine andere Quelle benötigt.

Bei *AbstractHeuristic* handelt es sich um die Basisklasse, von der alle Heuristiken abgeleitet werden. Ihre Hauptaufgabe besteht darin, einen neuen Knoten für den Graphen

zu ermitteln. Dies geschieht durch den Vergleich von Trajektorien, wofür zwei Methoden zur Verfügung stehen. Eine führt den nächsten Schritt des agglomerativen Clusterings durch und ist für das ISM bestimmt. Die andere nimmt die Menge der bereits zugewiesenen Knoten sowie die noch nicht zugewiesenen Objekte (bzw. deren Trajektorien) entgegen und bestimmt, welche Objekte wo in den Graphen als neue Knoten angehängt werden sollen. Eine weitere Methode liefert die anzuhängenden Knoten sowie den Elternknoten, an den angehängt werden soll. Jede Heuristik speichert die beste Bewertung für den aktuellen Durchlauf, so dass die beste Heuristik durch das Sortieren einer Liste ermittelt werden kann. Gegenwärtig ist nur die Heuristik *DirectionRelationHeuristic* implementiert.

Die Basisklasse *AbstractGraphGenerator* kapselt den Algorithmus zum Bau der Graphenstruktur. Die von den Heuristiken gelieferten Relationen werden hierzu eingesetzt. Der *HeuristicalGraphGenerator* implementiert den in Kapitel 6 beschriebenen Algorithmus zum Bau einer baumförmigen Graphenstruktur. Die hier genannten Komponenten werden durch eine Unterkategorie von *AbstractTrainer* zusammengefasst. Für das probabilistische Szenenmodell übernimmt diese Aufgabe die Klasse *PSMTrainer*. Sie kombiniert die vom *AbstractTrainer* zur Verfügung gestellte Ablaufsteuerung mit einer passenden Datenquelle, Heuristiken und dem Algorithmus für den Bau des Graphen und definiert darüber hinaus eine passende Schnittstelle nach außen hin. Abbildung 7.7 zeigt das Klassendiagramm des Relationsgraph-Generators.

7.2.4 Szenengraph-Generator

Paket *scene_graph_generator* beinhaltet den Szenengraph-Generator. Dieser akkumuliert die von den Objektdetektoren gelieferten Einzelmessungen und bereitet diese auf. Hierzu werden die zu einer Objektinstanz gehörenden Einzelmessungen zu einer Trajektorie zusammengefasst. Um das fehlende Beobachtungs- und Verdeckungsmodells sowie das ungelöste Tracking-Problem zu kompensieren wurden an dieser Stelle die in Kapitel 6 beschriebenen Workarounds umgesetzt. Die Trajektorien sind für das Lernen des Modells entscheidend, allerdings nicht für die eigentliche Szenenerkennung.

Da den Beobachtungen wegen des fehlenden Trackings keine Instanzbezeichnungen angehängt sind, werden die Objekte anhand ihres Typs identifiziert. Dies wird durch die Anforderung ermöglicht, dass pro Objekttyp nur eine einzige Instanz vorhanden sein darf. Damit alle Trajektorien die gleiche Anzahl an Beobachtungen enthalten werden für jede neu an eine Trajektorie angefügten Beobachtung die letzte Beobachtung aller anderen Trajektorien dupliziert. Wird ein neues Objekt gefunden, so wird davon ausgegangen, dass es schon immer an der besagten Stelle war. Die Beobachtung wird mehrmals dupliziert, bis die neue Trajektorie die selbe Länge wie alle anderen Trajektorien hat. Da kein Beobachtungs- und Verdeckungsmodell vorliegen verschwinden Objekte nicht, wenn sie nicht mehr beobachtet werden. Vielmehr werden sie als verdeckt oder außerhalb des Blickfelds angenommen, es wird also an dieser Stelle keine Sonderbehandlung durchgeführt.

7.2.5 Visualisierung

Eine der Anforderungen hinsichtlich der praktischen Umsetzung dieser Arbeit war es, die Visualisierung in ein separates Paket auszulagern, so dass andere Teile des Systems bei Bedarf darauf zugreifen können. Das Resultat ist das Paket *visualisation_server*. Es beinhaltet alle Visualisierungen, die von Lerner und Inferenz benötigt werden. Hierbei handelt es sich um die Darstellung von Balkendiagrammen, sowie die Anzeigen von Lerndaten und Inferenzergebnissen. Die Visualisierung setzt auf der von ROS zur Verfügung gestellten Visualisierungslösung *RVIZ* auf. Über die Kommunikationsinfrastruktur werden Nachrichten an das eigenständiges Programm gesendet. Diese enthalten Anweisungen zum Zeichnen von geometrischen Primitiven, Trajektorien oder 3D-Modellen.

Beim Entwurf der Architektur wurde darauf geachtet, den vorliegenden Begebenheiten Rechnung zu tragen. Einige Aktionen wie beispielsweise das Zeichnen von Kovarianzellipsen werden sowohl für die Visualisierung der Lerndaten, als auch der Inferenzergebnissen benötigt. Aus diesem Grund wurde beschlossen, die entsprechende Logik in eigenständigen Modulen abzulegen. Weiterhin sind die Architekturen von Lerner und Inferenz sehr ähnlich. Die Visualisierung wurde daher so strukturiert, dass sie von beiden genutzt werden kann. Der gewählte Ansatz sieht vor, dass der Visualisierung alle Informationen übergeben werden. Die visuelle Darstellung dieser Informationen erfolgt dann beim nächsten Update des Systems. Für die einzelnen Visualisierungsaufgaben stehen unterschiedliche Visualisierungsmodi zur Verfügung.

Das Klassendiagramm im unteren Teil von Abbildung 7.8 gibt einen Überblick über die Visualisierungsmodule und die dahinterstehende Vererbungsstruktur. Die Basisklasse *AbstractVisualizer* stellt die grundlegenden Methoden zur Verfügung. Die Klasse *AbstractExtendedVisualizer* erweitert den Funktionsumfang um die Vorgabe einer Farbe und eines Skalierungsfaktors. Der *CoordinateFrameVisualizer* zeichnet die Basisvektoren eines orthonormalen Koordinatensystems, das anhand der übergebenen Pose ausgerichtet ist. Für die Visualisierung von Lerndaten sorgt der *SampleVisualizer*. Aus einer Menge von Positionen erzeugt dieser eine Punktwolke, wobei jeder Punkt durch eine Sphere dargestellt wird. Dies erlaubt es, das gelernte Modell mit den Lerndaten zu vergleichen.

Ein einzelner Gauss-Kernel kann mit dem *GaussianKernelVisualizer* gezeichnet werden. Die Darstellung erfolgt über eine Kovarianzellipse, die von drei Ringen umgeben ist. Diese können beliebig eingefärbt werden, um die Zugehörigkeit zu einer übergeordneten Struktur, z.B. einer Gauss-Mischverteilung zu verdeutlichen. Die Implementierung hat sich als nicht-trivial erwiesen. Die Eigenvektoren der Kovarianzmatrix bilden das Basiskoordinatensystem der Sphere und werden in Form einer Rotationsmatrix eingesetzt, um ein Sphären-Primitiv entsprechend auszurichten. Die Skalierung erfolgt über die Eigenwerte. Bei diesem Vorgehen ist zu beachten, dass das Koordinatensystem der Rechte-Hand-Regel folgt. Da die meisten Eigenwert-Löser die Eigenvektoren nach der Größe der Eigenwerte geordnet ausgeben ist dies nicht immer der Fall. Um dies zu prüfen wird das Kreuzprodukt der ersten beiden Eigenvektoren berechnet. Zeigen der resultierende Vektor und der dritte Eigenvektor nicht in dieselbe Richtung, so müssen der zweite und dritte Eigenvektor vertauscht werden.

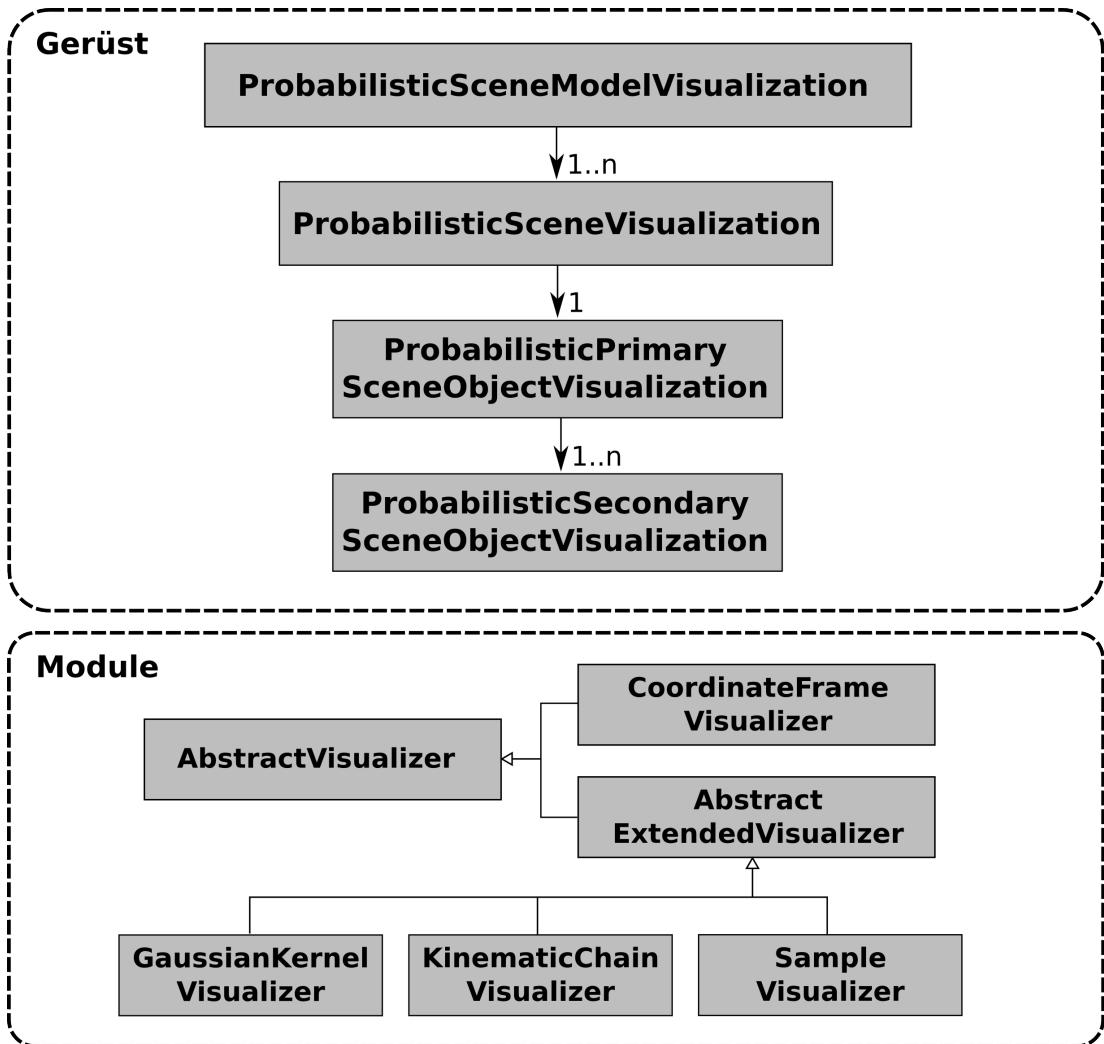


Abbildung 7.8: Klassendiagramm der Visualisierung. Der obere Teil zeigt das Gerüst, welches die Visualisierung logik kapselt und von Lerner und Inferenz gleichermaßen genutzt wird. Unten wird ein Überblick über die Visualisierungsmodule gegeben, welche unterschiedliche Aufgaben erfüllen.

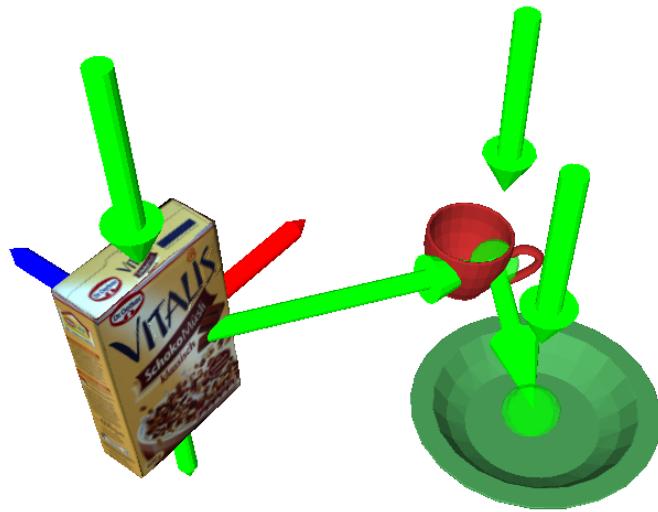


Abbildung 7.9: Visualisierung der Inferenz für eine Szene mit drei Objekten. Die Pfeile geben das Erkennungsergebnis für das darunterliegende Szenenobjekt wieder. Die Verbindungen symbolisieren die erfolgreiche Erkennung für alle Relationen des Szenenobjekts "Müsli".

Mit dem *KinematicChainVisualizer* werden die Relationen zwischen den Objekten eines Object Constellation Model dargestellt. Pfeile geben an, welche Objekte zueinander in Relation stehen, wobei der Pfeil immer vom übergeordneten Objekt ausgeht. Die Farbe der Linie gibt die Wahrscheinlichkeit der Relation wieder, wobei Grün und Rot das Vorliegen bzw. Fehlen einer Relation widerspiegeln. Über dem Objekt, das der Referenz des Object Constellation Model zugeordnet ist, wird zusätzlich ein senkrechter Pfeil visualisiert, dessen Farbe den Grad der Erkennung des Szenenobjekts wiedergibt. Die farbliche Kodierung entspricht denen der Relation. Es wird pro OCM immer nur die beste Zuordnungshypothese zur Visualisierung herangezogen.

Die oben beschriebenen Module kapseln die eigentliche Visualisierung. Darauf aufbauend ist die Logik implementiert, die es erlaubt, sowohl die Resultate des Lernprozesses als auch der Inferenz darzustellen. Wie im oberen Teil von Abbildung 7.8 gezeigt ist eine baumförmiges Gerüst aus vier unterschiedlichen Klassen geschaffen worden, die während dem Lernen bzw. Laden des Modells parallel dazu errichtet wird. *ProbabilisticSceneModelVisualization* bildet den Ausgangspunkt. Über einen entsprechenden Methodenaufruf kann hier der Visualisierungsmodus gewählt und der Zeichenvorgang eingeleitet werden. Daran angehängt wird pro Szene eine Instanz der Klasse *ProbabilisticSceneVisualization*. Hier werden die Farbkodierungen der Ringe festgelegt, die jeden Gauss-Kernel umgeben. Pro Szene wird eine eindeutige Farbe gewählt. Für jedes OCM wird eine Instanz der Klasse *ProbabilisticPrimarySceneObjectVisualization* angehängt, welche eine ähnliche Farbverteilung für die einzelnen Szenenobjekte festlegt.

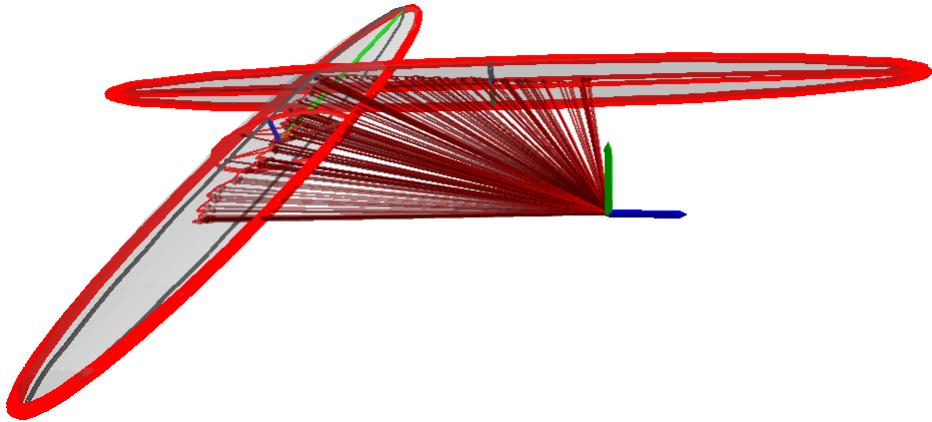


Abbildung 7.10: Visualisierung der gelernten Modells. Die Kovarianzellipsen beschreiben die relative Lage eines Objekts im Koordinatensystem eines übergeordneten Objekts. Die relative Trajektorie zu letzterem wird durch die Pfeile gekennzeichnet.

Den Abschluss bildet die Klasse *ProbabilisticSecondarySceneObjectVisualization*, die alle Objekte im OCM visualisiert und der zuvor erwähnten Klasse untergeordnet ist. Hier findet das eigentliche Zeichnen statt. Die Ringe der einzelnen Gauss-Kernel werden anhand der Szene und des Szenenobjekts eingefärbt. Im Visualisierungsprogramm RVIZ kann aus einer Liste ausgewählt werden, welches der visualisierten Szenenobjekte angezeigt werden soll, so dass es nicht zu Überlagerungen der einzelnen Visualisierungen kommt. Die Abbildungen 7.9 und 7.10 zeigen die Visualisierung in Aktion.

8. Evaluation

In diesem Kapitel wird das im Rahmen dieser Arbeit entwickelte probabilistische Szenenmodell - kurz PSM - evaluiert. Es werden grundlegende Funktionen wie die Fähigkeit zur Überprüfung relativer Posen und der Umgang mit Clutter und fehlenden Objekten geprüft. Weitere Aspekte des Ansatzes werden untersucht. Im Kern der Evaluation steht ein Vergleich mit dem auch als ISM bezeichneten Implicit Shape Model.

Zu dieser Arbeit existiert eine bislang noch nicht publizierte Veröffentlichung (siehe auch [PMD], in die einige der hier vorgestellten Bilder und Experimente eingeflossen sind. Zur Vermeidung eines Selbstplagiats wird daher an dieser Stelle darauf hingewiesen.

8.1 Verifikation der Hintergrundterme

In den Termen *Scene Shape*, *Object Appearance* und *Object Existence* des OCM werden Evidenzen, welche nicht durch die Hypothese berücksichtigt werden, als Teil des Hintergrunds angenommen und unter einem entsprechenden Hintergrundterm evaluiert. Dieser Term hat aus theoretischer Sicht eine klare Existenzberechtigung, soll jedoch im Rahmen dieses Abschnitts daraufhin überprüft werden, ob die daraus resultierenden Erkennungsergebnisse mit dem gewünschten Verhalten konsistent sind.

Hierzu wurden Szenen mit zwei, drei und vier Objekten eingelernt, wobei jede Szene eine Teilmenge der Objekte der nächstgrößeren Szene ist. Die Erkennungsergebnisse aller drei Szenen wurden zueinander in Relation gesetzt. Damit alle Objekte mit korrekter Pose und akkurater Erkennung vorliegen, wurde das Experiment mit simulierten Eingabedaten durchgeführt.

Jede Szene für sich wurde mit hoher Wahrscheinlichkeit erkannt. Setzt man die Erkennungsergebnisse jedoch zueinander in Relation, so lässt sich beobachten, dass jede Szene weitaus wahrscheinlicher als die jeweils kleinere ist. Die Ursache hierfür sind die überzähligen Objekte. Je kleiner die Szene, desto mehr Clutter ist vorhanden. Deren Bewertungen werden durch den Hintergrundterm aufmultipliziert. Da die entsprechenden

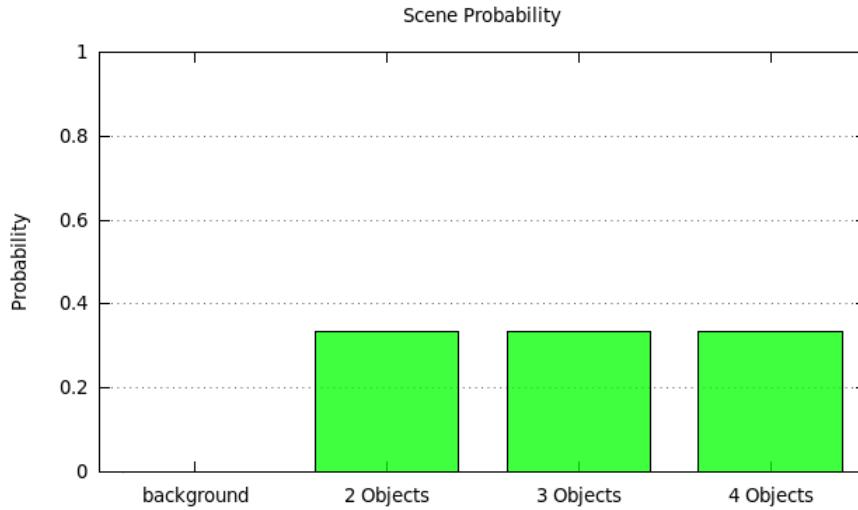


Abbildung 8.1: Erkennungsergebnis der verschachtelten Szenen nach Entfernung des Hintergrundterms.

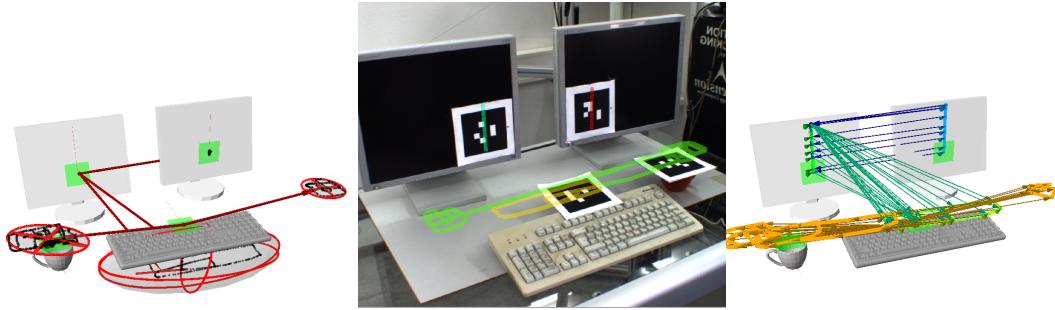
Werte kleiner als eins sind so für jedes zusätzliche irrelevante Objekt die Gesamtwahrscheinlichkeit gesenkt.

Ein weiteres Experiment soll diesen Sachverhalt verdeutlichen. Eine Szene mit zwei Objekten wird mit einer zunehmenden Anzahl von Clutter konfrontiert. Das Hinzufügen jedes weiteres Objekts führt zu einem Abfall der Wahrscheinlichkeit.

Wie demonstriert beeinflusst der Hintergrundterm das Erkennungsergebnis negativ, da für die Szene irrelevante Objekte in das Ergebnis mit einfließen. Aus diesem Grund wurde die Hintergrundterme aus dem OCM gestrichen, alle weiteren Experimente wurden mit dem so angepassten Modell durchgeführt. Abbildung 8.1 zeigt das Erkennungsergebnis der verschachtelten Szenen nach Entfernung der Terme. Wie zu erwarten wurden alle Szenen unabhängig von ihrer Größe mit identischen Wahrscheinlichkeiten bewertet.

8.2 Relative Position und Orientierung

In diesem Abschnitt wird anhand einiger ausgewählter Objektkonstellationen demonstriert, wie sich die relative Lage auf die Erkennung der Szene auswirkt. Das hier vorgestellte Experiment wurde sowohl für das PSM als auch das ISM durchgeführt. Die Ergebnisse für beide Systeme werden hier verglichen. Um den Bezug zur Praxis zu verdeutlichen wurde ein Büro-Szenario gewählt. Zwei höhenverstellbare Bildschirme stehen auf einem Tisch, davor befindet sich eine Tastatur. Bedient wird diese von zwei fiktiven Benutzern, einem Links- und einem Rechtshänder, die ihre Kaffeetasse jeweils an der für sie günstigeren Seite der Tastatur platzieren. Der Henkel der Tasse zeigt immer von der Tastatur weg. Die einzelnen Objekte werden über die daran angebrachten Marker erkannt, die in Abbildung 8.2 zu erkennen sind.



(a) Das vom PSM verwendete Modell der relativen Objektlagen.
(b) Die Trajektorien aller Objekte, projiziert auf das Kamerabild.
(c) Objekttrajektorien, eingesetzt zum Lernen des ISM.

Abbildung 8.2: Das Büro-Szenario umfasst zwei höhenverstellbare Bildschirme und eine Tastatur. Relativ dazu ist eine Tasse platziert.

Abbildung 8.3 zeigt zehn verschiedene Situationen, die alle einen ausgewählten Sachverhalt in Bezug auf relative Objektlagen verdeutlichen sollen. Pro Situation existiert ein Kasten, der die Visualisierungen beider Systeme enthält. In der linken Hälfte ist die Visualisierung des PSM untergebracht, in der rechten Hälfte die des ISM. Die vom PSM berechneten Szenenwahrscheinlichkeiten für Hintergrund- und Büroszene befinden sich in dem kleinen Kasten darunter.

In den *Situationen 1 und 2* wurde der rechte Bildschirm entlang unterschiedlicher Achsen verschoben. Im Training wurden beide Bildschirme immer parallel zueinander beobachtet, es ist also eine klare Abweichung von der erlernten Szene gegeben. Beide Systeme interpretieren die Situation korrekt. Die nicht eingehaltene Relation wird vom PSM rot eingefärbt, da die Daten hier schlecht zu dem erlernten Modell der Szene passen. Da die relativen Lageinformationen verunreinigt werden, sinkt hierdurch die Wahrscheinlichkeit für die komplette Szene, wie sich am roten senkrechten Pfeil über dem jeweiligen Referenzobjekt ablesen lässt. Als Folge hiervon ist die Büroszene um mehrere Größenordnungen unwahrscheinlicher als die Hintergrundszene.

Dieses abrupte Abfallen der Wahrscheinlichkeit lässt sich darauf zurückführen, dass das Auftreten des Bildschirms als sicheres Ereignis vorgegeben ist. Eine geringere Auftrittswahrscheinlichkeit würde die Auswahl einer anderen, unvollständigen Hypothese ohne den rechten Bildschirm erlauben und damit auch eine feinere Abstufung der Szenenwahrscheinlichkeit. Aus Gründen der Übersichtlichkeit wird dieser Faktor allerdings erst im nächsten Experiment genauer betrachtet.

Das zugehörige ISM besteht aus zwei Clustern, wobei der eine beide Bildschirme und der andere Tastatur und Tasse beinhaltet. Der erste Cluster wird auf Grund der Verschiebung nicht mehr erkannt, der andere hingegen nach wie vor. Da das ISM aus drei verschachtelten Modellen besteht und nur eines dieser Modelle nicht zu den Daten passt, führt dies zu einer Bewertung im mittleren Bereich.

In *Situation 3* wurde der rechte Bildschirm gegen den Uhrzeigersinn verdreht. Das Re-

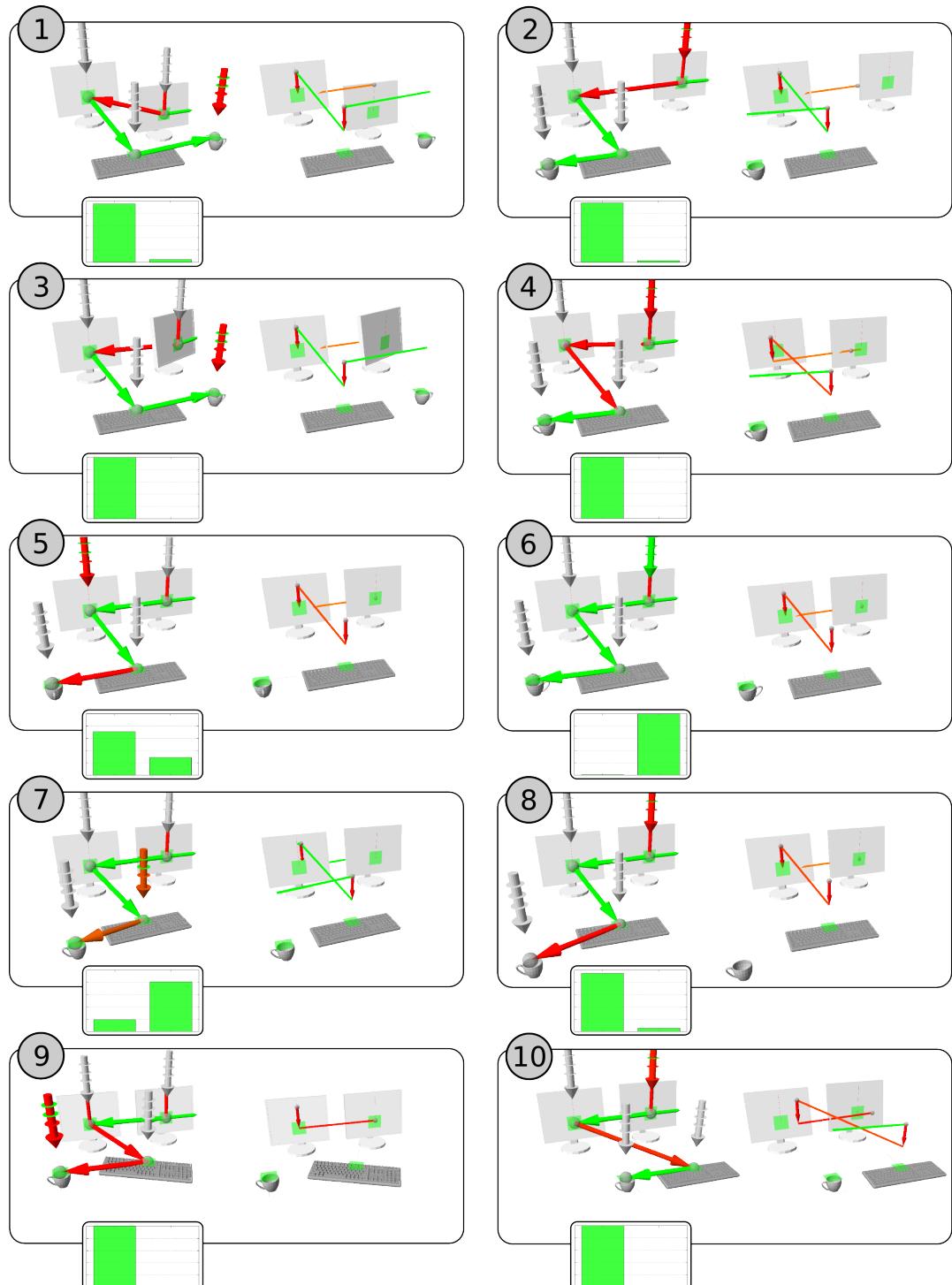


Abbildung 8.3: Anhand eines alltäglichen Büro-Szenario wird die Auswirkungen relativer Objektlagen auf die Erkennung der Szene demonstriert.

sultat ist analog zu den beiden vorangegangenen Situationen. Die entsprechende Relation gilt als nicht eingehalten, was von beiden Systemen mit einer Herabstufung der Szenenbewertung bestraft wird. Dies demonstriert, dass die relative Orientierung als gleichberechtigt zur relativen Position betrachtet wird.

Bisher wurde nur ein Objekt manipuliert, dass sich an einem Ende des Relationsbaums befunden hat. Daher wurde in *Situation 4* das linke Display - ein Objekt mitten im Baum - nach hinten verschoben. Beide Systeme erkennen die verletzten Relationen korrekt und reagieren entsprechend mit einer Herabstufung der Bewertung. Die Hintergründe hinsichtlich der Quantität der Herabstufung für PSM und ISM wurden bereits im Rahmen der ersten beiden Situationen erläutert, daher wird hier nicht mehr darauf eingegangen.

Die *Situationen 5 und 6* verdeutlichen die Probleme, welche die Entkopplung von Position und Orientierung in einem parametrischen Modell mit sich bringt. Wie einführend erwähnt befindet sich die Tasse entweder links oder rechts der Tastatur. Außerdem zeigt der Henkel immer von der Tastatur weg. *Situation 5* demonstriert, dass eine Abweichung von den beiden erlernten Orientierungen von beiden Systemen korrekt als Abweichung vom Modell erkannt wird. Zeigt die Tasse wie in *Situation 6* allerdings mit dem Henkel zur Tastatur, so führt dies beim PSM zu einer Fehlerkennung.

Die Ursache hierfür ist folgende. Position und Orientierung der Tasse relativ zur Tastatur sind durch voneinander unabhängige Gauss-Mischverteilungen modelliert. Die Tasse befindet sich an einer der beiden eingelernten relativen Positionen und hat auch eine der beiden eingelernten relativen Orientierungen. Die Ergebnisse beider Verteilungen werden verundet und führen daher zu einer hohen Wahrscheinlichkeit. Es wird jedoch nicht berücksichtigt, dass die eine Orientierung nur in Korrelation mit der einen Position und die andere Orientierung nur mit der anderen Position erscheint. Das modellbedingte Verwerfen dieser relevanten Information ist verantwortlich für die Fehlerkennung.

Da das ISM als nicht-parametrisches Modell lediglich die Trainingsdaten abspeichert, ist die Kopplung zwischen Position und Orientierung automatisch gegeben. Daher bestätigt das besagte System die für die gegebene Situation vorliegende Abweichung von der Szene.

Situation 7 und 8 illustrieren die Unterschiede beider Systeme hinsichtlich ihrer Entscheidungsgrenzen. Hierzu wird die Tasse schrittweise von ihrer erlernten Position aus in eine beliebige Richtung verschoben. Bei einer geringen Abweichung bewertet das PSM die Relation mit einer Wahrscheinlichkeit im mittleren Bereich, was durch den orangefarbenen Pfeil verdeutlicht wird. Die Bewertung führt zu einem ebenfalls mittleren Erkennungsergebnis. Beim ISM hingegen gilt die betroffene Relation und damit auch die Szene als komplett erkannt. Verschiebt man die Tasse weiter, so registrieren beide Systeme eine Abweichung vom Modell.

Dies demonstriert, dass das PSM über weiche Entscheidungsgrenzen verfügt, das ISM jedoch über harte. Dies kann als Vor- aber auch als Nachteil angesehen werden. So erlaubt eine weiche Entscheidungsgrenze einen fließenden Übergang, führt jedoch auch einen Bias ein, der bestimmte Posen anderen vorzieht. Letzteres wird in Abschnitt 8.5 untersucht.

In *Situation 9* wurde die Tastatur im Uhrzeigersinn gedreht. Das PSM erkennt die beiden Relationen zwischen Tasse und Tastatur sowie Tastatur und dem linken Bildschirm als nicht eingehalten. Dem ISM hingegen ist es nicht mehr möglich, die Relation zwischen Tasse und Tastatur zu erkennen. Die Relation zum Bildschirm kann als Folge dessen ebenfalls nicht mehr erkannt werden, so dass der verbliebenen Relation zwischen beiden Bildschirmen eine niedrige Bewertung zugewiesen wird.

Für *Situation 10* wurde die relative Pose zwischen Tasse und Tastatur beibehalten, beide sind jedoch gegenüber den zwei Bildschirmen verschoben und verdreht. Das PSM bewertet die einzelnen Relationen und nimmt dabei die Abweichung zwischen den beiden Clustern wahr. Das ISM erkennt diese Relation zwar auch als fehlerhaft, dies führt jedoch zu einer Fehlerkennung der untergeordneten Relation. Das korrekt erkannte Teilmodell an der Spitze der Hierarchie generiert Hypothesen für die Position des linken Bildschirms, die jedoch abseits dessen tatsächlicher Position liegen. Als Folgefehler wird auch die Relation zwischen den beiden Bildschirmen als fehlerhaft bewertet.

Die Evaluation relativer Posen gilt mit diesem Experiment als abgeschlossen. Alle weiteren Experimente befassen sich mit anderen Faktoren des PSM.

8.3 Clutter und fehlende Objekte

Dieses Experiment dient dazu, den Umgang mit Clutter und fehlenden Objekten zu demonstrieren. Außerdem wird die Bedeutung der im *Object Existence*-Term modellierten Auftrittswahrscheinlichkeiten der einzelnen Objekte verdeutlicht. Es wurden zwei Haushaltsszenen A und B mit jeweils vier Objekten erlernt, die sich in zwei Objekten - Teller und Becher - überschneiden. Die Auftrittswahrscheinlichkeit für alle Objekte liegt bei 70%. Eine Ausnahme bildet der in beiden Szenen enthaltene Teller, dieser hat in Szene B eine Auftrittswahrscheinlichkeit von 90%. Abbildung 8.4 zeigt die Modelle beider Szenen für PSM und ISM.

Die oben genannten Faktoren werden erläutert, indem Szene A schrittweise in Szene B überführt wird. Für jede der sieben Situation des Experiments wird - wie schon im vorherigen Abschnitt - für beide Systeme die jeweilige Visualisierung gezeigt. Weiterhin werden darunter die vom PSM gelieferten Szenenwahrscheinlichkeiten dargestellt. Bei diesen handelt es sich (von links nach rechts) um Hintergrund, Szene A und Szene B. Zur Erkennung wurden textur- und segmentbasierte Detektoren eingesetzt, Detektoren auf Markerbasis wurden nicht verwendet. Im diesem Experiment wurde darauf geachtet, dass alle relativen Posen eingehalten wurden. Hierzu wurden Objektbewegungen soweit möglich vermieden. Da es sich jedoch um reale Daten handelt, ist mit geringfügigen Positionierungsfehlern und Detektorrauschen zu rechnen. Deren Einfluss ist minimal, jedoch nicht zu vernachlässigen, wie später erläutert wird.

In *Situation 1* ist Szene A komplett aufgebaut, für Szene B sind nur die beiden gemeinsamen Objekte vorhanden. Die Visualisierung für das PSM wurde so konfiguriert, dass sie die jeweils beste Hypothese für beide Szenen zeichnet. Da auf die Einhaltung der relativen Posen geachtet wurde, werden diese immer mit einer hohen Wahrscheinlichkeit



Abbildung 8.4: Erlernte Modelle der beiden Haushaltsszenen A (links) und B (rechts) für die beiden Systeme PSM (oben) und ISM (unten).

erkannt, was durch die grüne Färbung symbolisiert wird. Die senkrechten grünen Indikatorpfeile deuten eine gute Erkennung beider Szenen an, wie das Balkendiagramm über die Szenenwahrscheinlichkeiten zeigt, dominiert jedoch klar Szene A. Das ISM hat die erste Szene ebenfalls vollständig erkannt, die zweite Szene erhält nur eine geringfügige Bewertung, da die Daten nicht zu zwei der drei verschachtelten ISMs passen.

In den *Situationen 2 und 3* werden nacheinander zuerst die linke, dann die rechte Müsli-Packung entfernt, wodurch Szene A schrittweise auf die gemeinsame Objektmenge beider Szenen reduziert wird. Durch das Entfernen der linken Packung steigt die Wahrscheinlichkeit für Szene B sichtbar an. Nach dem Entfernen der zweiten Müsli-Packung wäre eigentlich damit zu rechnen, dass beide Szenen gleich wahrscheinlich sind. Wie oben erläutert hat der Teller in Szene B jedoch eine weitaus höhere Auftrittswahrscheinlichkeit als alle anderen Objekte, daher ist Szene B wahrscheinlicher als Szene A. Ebenfalls nennenswert ist das Ansteigen der Hintergrundwahrscheinlichkeit. Da nur zwei Objekte vorhanden sind, ist die Wahrscheinlichkeit, dass es sich dabei um einen Zufall handelt, im Vergleich zu den Wahrscheinlichkeiten der beiden anderen Szenen relativ hoch.

Das ISM zeigt ein ähnliches Verhalten. Nach dem Entfernen der ersten Packung erhalten die beiden mit Szene A assoziierten Relationen eine mittlere Bewertung, die beiden Relationen für Szene B eine geringe. Hinzu kommt für jede Szene eine Relation mit hoher Bewertung. In der Summe liegt eine höhere Gesamtbewertung für Szene A vor. Nach dem Entfernen der zweiten Packung ist - neben der gemeinsamen gut bewerteten Relation -

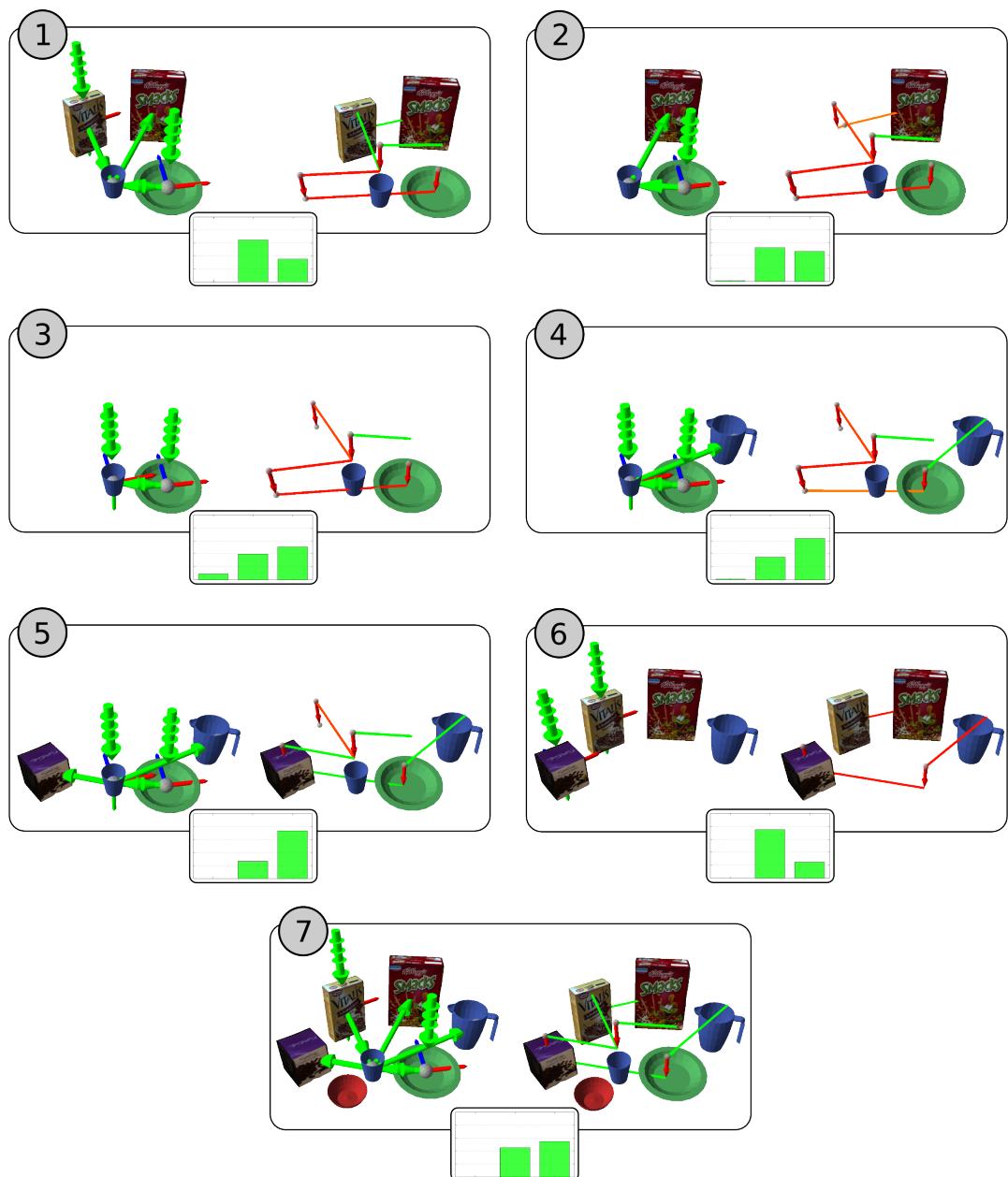


Abbildung 8.5: Der Übergang zwischen zwei Haushaltsszenen wird dazu herangezogen, um den Umgang mit Clutter, fehlenden Objekten und der Auftrittswahrscheinlichkeit zu demonstrieren.

für Szene A nur eine als mittelmäßig beurteilte Relation vorhanden, die dasselbe Gewicht hat wie die Summe der beiden niedrig bewerteten Relationen in Szene B. Hieraus ergeben sich identische Bewertungen für beide Szenen.

Das eben behandelte Szenario verdeutlicht den Mehrwert, den Auftrittswahrscheinlichkeiten mit sich bringen. Während das ISM beide Szenen in *Situationen 3* gleich bewertet, nutzt das PSM eine aus den Lernbeispielen extrahierte Zusatzinformation und schließt daraus, dass Szene B im vorliegenden Fall als geringfügig wahrscheinlicher anzunehmen ist.

Die schrittweise Überführung nach Szene B erfolgt in den *Situationen 4 und 5*. Das Hinzufügen von Messbecher und Kaffeebox sorgen für einen sprunghaften Anstieg der Wahrscheinlichkeit für Szene B. Ähnlich verhält es sich beim ISM. Szene B überwiegt zunächst gegenüber Szene A durch eine zusätzliche gute sowie eine mittelmäßige Relation. Nach dem Auftauchen der Kaffeebox wird Szene B im Gegensatz zu Szene A komplett erkannt.

Situation 6 behandelt den Sonderfall, in dem beide Szenen vorhanden sind, jedoch die beiden gemeinsamen Objekte fehlen. Im Fall des PSM ist der Sachverhalt ähnlich zu *Situation 3*. Der Teller fehlt, was in den Lerndaten nur in 10% aller Fälle vorkommt. Hieraus schließt das System, dass Szene A mit relativ hoher Wahrscheinlichkeit vorliegen muss. Das ISM bewertet beide Szenen wiederum gleich, da es keine Auftrittswahrscheinlichkeiten berücksichtigt.

In *Situation 7* sind beide Szenen komplett vorhanden, hinzu kommt Clutter in Form einer roten Schale. Die Rolle von Clutter lässt sich zwar bereits implizit aus den vorherigen Beispielen ableiten, wird an dieser Stelle jedoch zum besseren Verständnis explizit erwähnt. Beide Systeme zeigen das gewohnte Verhalten, da das überschüssige Objekt für sie keine Rolle spielt. Im Fall des PSM ist der Teller wieder dafür verantwortlich, dass Szene B geringfügig wahrscheinlicher ist. Das ISM stellt identische Bewertungen für beide Szenen fest.

Wie zu Beginn erwähnt spielt die Auftrittswahrscheinlichkeit eine weitere wichtige Rolle. Auf Grund von Positionierungsfehlern und Detektorrauschen kommt es selbst bei sorgfältigem Platzieren der Objekte zu Abweichungen. Diese Fehler akkumulieren sich auf und fallen besonders bei einer hohen Anzahl an Objekten stark ins Gewicht. Dem entgegen wirkt die Auftrittswahrscheinlichkeit. Diese erlaubt die Berücksichtigung von Hypothesen, in denen die Szene nur teilweise vorhanden ist. In der Gesamtsumme sorgt dies für eine höhere Szenenwahrscheinlichkeit. Ist die Auftrittswahrscheinlichkeit zusätzlich noch asymmetrisch - d.h. wurde das Objekt im Training häufiger beobachtet als nicht beobachtet - so werden diejenigen Hypothesen höher bewertet, bei denen die Objekte vorliegen - eine weitere Verbesserung der Szenenwahrscheinlichkeit.

In diesem Experiment wurde demonstriert, dass sowohl PSM als auch ISM mit Clutter und fehlenden Objekten umgehen können. Zusätzlich wurde der Nutzen von Zusatzwissen in Form von Auftrittswahrscheinlichkeiten hervorgehoben.

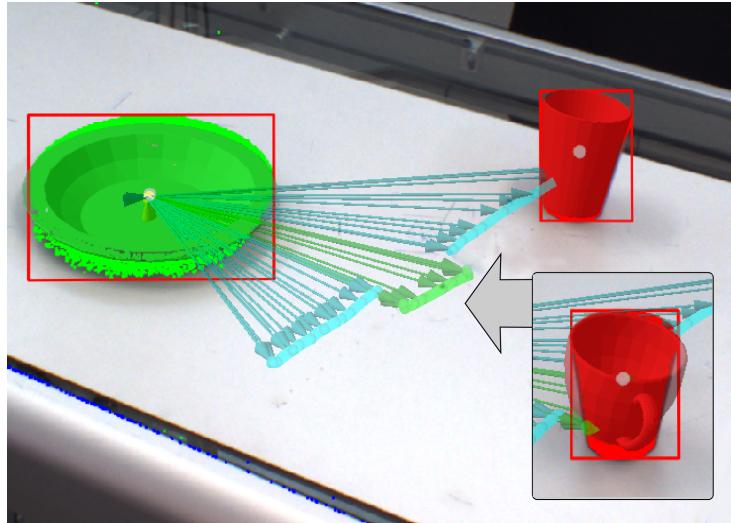


Abbildung 8.6: Die grüne Teiltrajektorie veranschaulicht den Bereich der Fehlerkennung. Statt einem Becher wird eine Tasse erkannt.

8.4 Robustheit gegenüber Fehldetektionen

Ein häufiges Problem bei der Objekterkennung besteht darin, dass Objekte mit einem falschen Objekttyp erkannt werden. Gerade bei holistischen Ansätzen kann es vorkommen, dass bei ungünstigen Umweltbedingungen ein Teller als Schale, ein Messbecher als Becher oder ein Becher als Tasse erkannt wird. Kommt es während des Lernens zu einer solche Fehlerkennung, so muss bei der Erkennung der Szene dieselbe Fehlerkennung auftreten, damit die Szene erkannt werden kann.

Um dieses Problem zu lösen wurde der *Object Appearance*-Term eingeführt, der für jedes Objekt der Szene eine Wahrscheinlichkeitstabelle über die Objekttypen verwaltet. Hierdurch wird über den konkreten Ort der Fehlerkennung abstrahiert und lediglich die Information behalten, in welchem Maß eine spezifische Fehlerkennung aufgetreten ist.

Zur Evaluation des Terms wurde gezielt die in Abbildung 8.6 gezeigte Fehlerkennung herbeigeführt. Der Becher wurde entlang der blauen Trajektorie verschoben. Ungefähr in der Mitte wurde für einen gewissen Zeitraum auf die Detektion einer ebenfalls roten Tasse umgeschalten. Dies lässt sich an der grünen Trajektorie ablesen, welche den Weg der Tasse bis zum Umschalten auf den ursprünglichen Detektor beschreibt.

Mit diesen Daten wurden Modelle für PSM und ISM trainiert. Anschließend wurde der Becher im Bereich der grünen Trajektorie platziert. Die Erkennungsergebnisse sind in Abbildung 8.7 zu sehen. Wie erwartet ist es dem PSM möglich, die Szenen korrekt zu erkennen. Dies liegt daran, dass die Form der Szene im *Scene Shape*-Term modelliert wird, der Objekttyp im davon abhängigen *Object Appearance*-Term. Das Ergebnis wird verundet. Diese gezielte Trennung beider TeilmODELLE erlaubt es, über den Ort der Fehlerkennung zu abstrahieren.

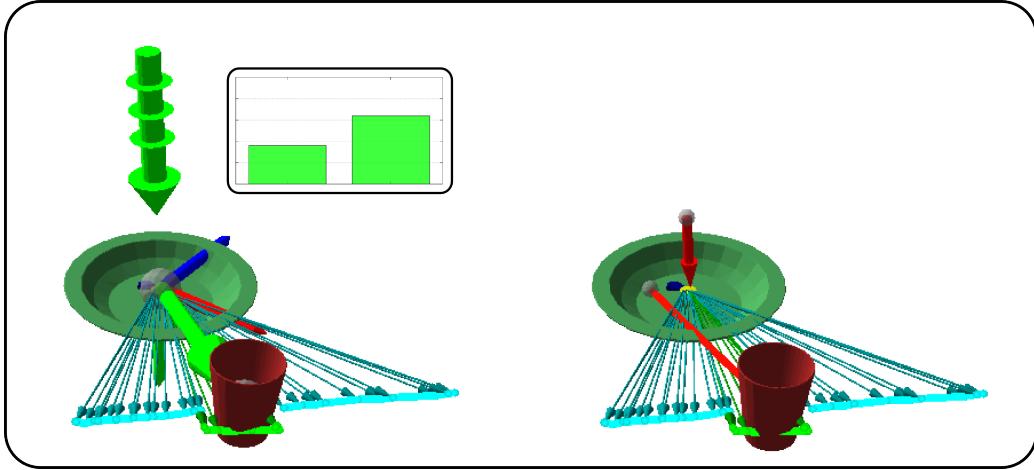


Abbildung 8.7: Das ISM hat das Objekt an der gegebenen Position nie beobachtet und kann die Szene daher nicht erkennen. Das PSM hingegen bewertet Lage und Objekttypen getrennt, so dass eine korrekte Erkennung vorliegt.

Da das ISM seine Entscheidungen strikt auf Basis der abgespeicherten Trainingsdaten trifft, sucht es den Teller an einem Punkt im Raum, an dem dieser nicht vorhanden ist. Die rote Verbindungsline verdeutlicht, dass der Teller nicht gefunden und damit die Relation nicht erfüllt wurde. Das ISM verfügt im Bereich der grünen Trajektorie nicht über relative Posen zwischen Teller und Becher, nur über Teller und Tasse. Daher kann auch keine erfolgreiche Erkennung stattfinden.

Als Fazit soll ein kurzer Exkurs in das Feld der Modellierung unternommen werden. Dem aufmerksamen Leser mag nicht entgangen sein, dass in einem der vorangegangenen Experimente die Trennung zweier Teilmodelle - Position und Orientierung - zu Fehlerkennungen geführt hat. In diesem Experiment wird eine Entkopplung zweier Teilmodelle befürwortet, um einen Sachverhalt zu erzeugen, der als gezielte Fehlerkennung angesehen werden kann. Der Autor vertritt die Meinung, dass dies kein Widerspruch in sich ist. Vielmehr erlaubt die (Ent-)Kopplung von Teilmodellen, das Gesamtmodell gezielt an die Anforderungen der zu lösenden Aufgabe anzupassen. In diesem Fall ist es ein klarer Vorteil, den beobachteten Typ eines Objekts von der konkreten Position im Raum zu trennen, um so das Problem von Fehldetections zu lösen.

8.5 Auswirkungen weicher Entscheidungsgrenzen

Wie bereits angesprochen verfügt das PSM im Gegensatz zum ISM über weiche Entscheidungsgrenzen. Die daraus folgenden Implikationen für die Szenenerkennung werden hier näher untersucht. Zu diesem Zweck wurden zwei Experimente konzipiert. Der erste Versuch befasst sich mit der Bevorzugung derjenigen Positionen bzw. Orientierungen, deren Mahalanobis-Distanz näher am Mittelwert der Gauss-Verteilung liegt. Der zweite Versuch untersucht, ob Hotspots und überhängende Wahrscheinlichkeitsmasse zu

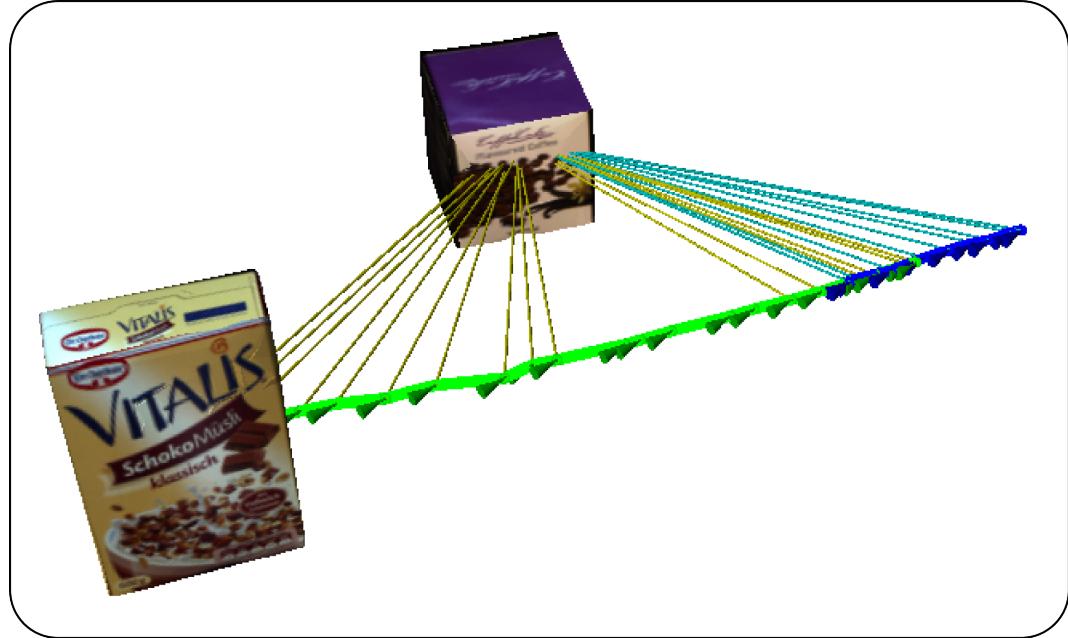


Abbildung 8.8: Szene A besteht aus einer langen, Szene B aus einer kurzen Trajektorie. Beide Trajektorien überlagern sich.

Fehlerkennungen einer Szene führen können. Beide Experimente wurden mit realen Objekten durchgeführt, deren Pose und Objekttyp durch einen entsprechenden Detektor ermittelt wurde.

Für das erste Experiment wurden zwei Szenen erlernt. In Szene A wird eine Müslipackung relativ zu einem anderen Objekt verschoben. In Szene B ebenfalls - allerdings so, dass die resultierende Trajektorie um ein Vielfaches kürzer ist und sich mit der Trajektorie aus Szene A überlagert. Abbildung 8.8 zeigt die beiden Trajektorien, wobei grün für Szene A und blau für Szene B steht.

Der Lerner des PSM erzeugt hieraus zwei Gauss-Kernel, deren 3σ -Visualisierungen dieselben Breiten und Höhen, allerdings unterschiedliche Längen haben. Beide Kernel überlappen sich. Die Müslipackung wird nun so platziert, dass sie sich näher am Mittelwert der des zu Szene B gehörenden Kernels befindet. Als geeignetes Distanzmaß wird - wie schon im Konzept-Kapitel - die Mahalanobis-Distanz angesehen, da diese die Kovarianzmatrix nutzt, um den unterschiedlichen Ausprägungen der einzelnen Achsen auszugleichen.

Der Ort der Platzierung wurde so gewählt, dass sich das Objekt im Abstand von ca. $1,5\sigma$ von Kernel A und ca. 1σ von Kernel B befindet. Abbildung 8.9 veranschaulicht dies. Die ebenfalls dort gezeigten Wahrscheinlichkeiten stehen (von links nach rechts) für Hintergrund, Szene A und Szene B. Es ist deutlich zu erkennen, dass Szene B auf Grund des geringeren Objekt-Abstands wahrscheinlicher ist, obwohl die Anforderungen beider Szenen erfüllt sind.

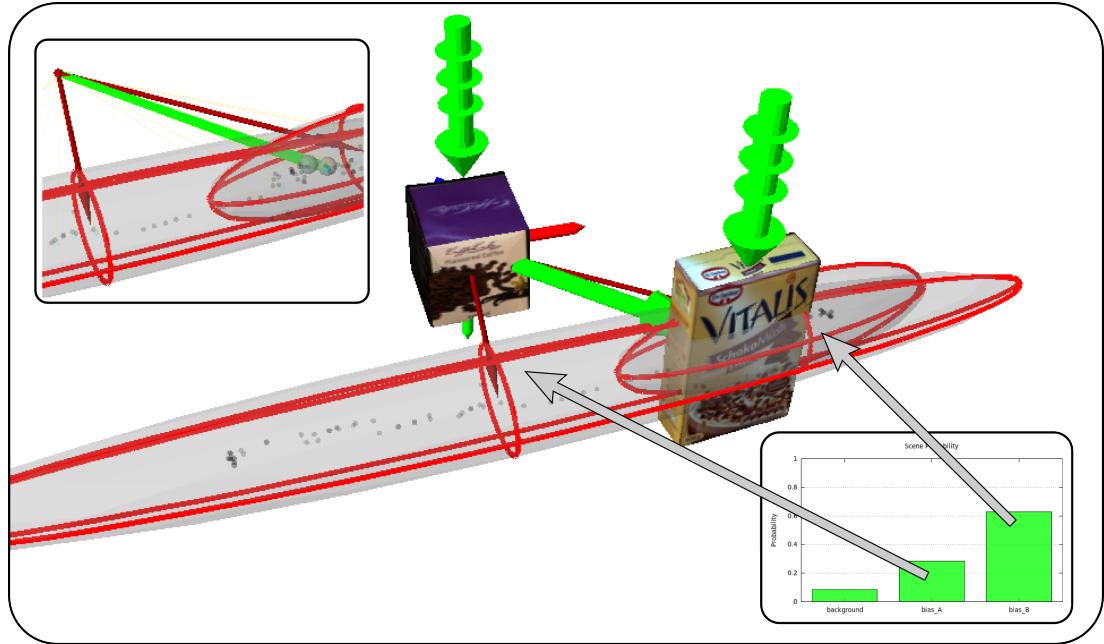


Abbildung 8.9: Die weichen Entscheidungsgrenzen des PSM führen dazu, dass bestimmte Posen bevorzugt werden. Die Müslipackung erfüllt die Anforderungen beider Szenen, dennoch ist Szene B wahrscheinlicher.

Das ISM (oben links im Bild) erkennt beide Szenen mit der bestmöglichen Bewertung, wie sich an den beiden grün eingefärbten Relationen erkennen lässt. Das Implicit Shape Model ist eine generalisierte Variante der Hough-Transform, der Hough-Raum führt auf Grund seiner diskreten Natur zu harten Entscheidungsgrenzen.

Im zweiten Experiment wurde mit der Müslipackung eine rechteckige Trajektorie abgefahren. Abbildung 8.10 zeigt die aus vier Kerneln bestehende Gauss-Mischverteilung, mit der die Kanten des Rechtecks modelliert werden. Wie deutlich zu erkennen ist stehen die Kernel an den Ecken über und erzeugen so überhängende Wahrscheinlichkeitsmasse, also Wahrscheinlichkeitsmasse, die sich abseits der beobachteten Lerndaten befindet. Deren Auswirkung auf die Erkennung der Szene wird hier untersucht. Weiterhin wird experimentell ermitteln, inwiefern innerhalb des Rechtecks ein Hotspot entsteht, also ein Bereich, indem die akkumulierte Wahrscheinlichkeitsmasse verschiedener Kernel einen kritischen Punkt erreicht und so zu Fehlerkennungen führt.

Abbildung 8.10 zeigt die untersuchten Fälle und die daraus resultierenden Wahrscheinlichkeiten. *Fall 1* zeigt die Müslipackung, die im Bereich der überhängenden Wahrscheinlichkeitsmasse platziert wurde. Das Diagramm darunter gibt (von links nach rechts) die Wahrscheinlichkeiten für Hintergrund und Szene wieder. Wie daraus abzulesen ist besteht keine Fehlerkennung der Szene, vielmehr ist der Hintergrund deutlich wahrscheinlicher.



(a) Die mit der Packung abgefahrene, rechteckige Trajektorie.
(b) Die Trajektorie wird mit vier Gauss-Kerneln abgedeckt, zeigt jedoch überhängende Wahrscheinlichkeitsmasse.

Abbildung 8.10: Mittels der Müslipackung wurde eine rechteckige Trajektorie erzeugt. Anhand dieser wird die Anfälligkeit für Fehlerkennungen auf Grund von Hotspots oder überhängender Wahrscheinlichkeitsmasse untersucht.

Der *Fall 2* dient dazu, eine korrekte Erkennung zu demonstrieren. Die Packung wird nahe dem Mittelwert eines der Kernel platziert. Wie erwartet wird die Szene erkannt. In *Fall 3* wird die Packung so verschoben, dass sie sich im Zentrum des von der Trajektorie gebildeten Rechtecks befindet. Es zeigt sich, dass die Hintergrundklasse dominiert, sich also kein Hotspot ausgebildet hat, der zu einer Fehlererkennung der Szene führt. Für *Fall 4* wurden mittels Sample Relaxation die Kernel künstlich aufgebläht. Die Position der Packung wurde nicht verändert, sondern eine erneute Erkennung mit dem modifizierten Modell durchgeführt. Nach einigen Anläufen konnte ein Hotspot erzeugt werden, auf Grund dessen die Szene geringfügig wahrscheinlicher ist als der Hintergrund.

Lerndaten müssen für die Erzeugung eines Hotspots in einem Maß aufgebläht werden, das im Rahmen einer praktischen Anwendung keine Vorteile bringt. Die Gefahr von Hotspots in der Praxis besteht also eher nicht.

Die beiden durchgeführten Experimente haben gezeigt, dass Anomalien auf Grund der gewählten Lagerepräsentation keine negativen Auswirkungen auf die Erkennung einer Szene haben. Es wurde demonstriert, dass der ebenfalls hierdurch eingeführte Bias eine messbare Auswirkung auf die Szene hat. Diese liegt allerdings im akzeptablen Bereich, da dies nach Erfahrungen des Autors zu einem Teil die menschliche Denkweise widerspiegelt. Letztendlich ist die Erkennung von Szenen eine objektive Sache, über deren psychologische Gesetzmäßigkeiten bisher nur wenig bekannt ist.

8.6 Laufzeit und Modellwachstum

Der letzte Abschnitt der Evaluation ist die Messung der Laufzeit. Als Testsystem wurde ein Intel Core i5-2500K Prozessor mit 3.30 GHz und 8 GiB Hauptspeicher eingesetzt. Es werden zunächst die Laufzeiten von PSM und ISM miteinander verglichen. Das Experiment wurde so aufgebaut, dass die Vor- und Nachteile beider Systeme ersichtlich werden. Darauf folgt eine detaillierte Betrachtung der Laufzeit des PSM in Abhängigkeit aller dafür relevanten Faktoren. Abschließend wird die Skalierung der Laufzeit des

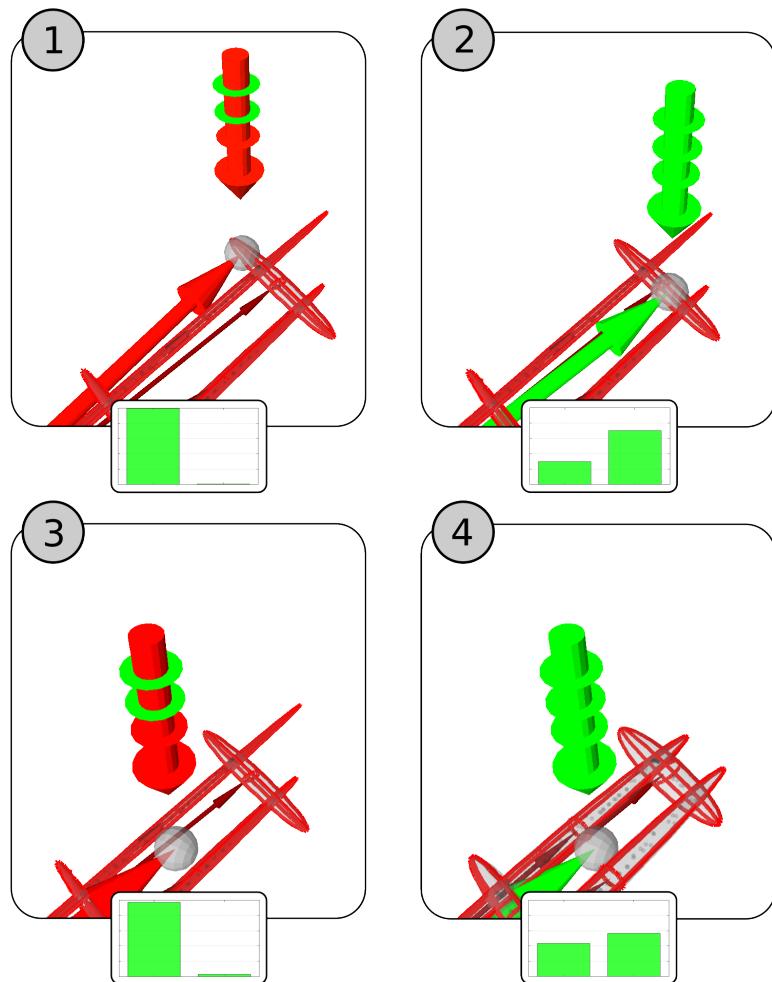


Abbildung 8.11: (1) Überhängende Wahrscheinlichkeitsmasse wirkt sich nicht nennenswert auf die Szenenwahrscheinlichkeit aus. (2) Korrekte Erkennung der Szene. (3) Daten und Modell passen nicht zueinander, die Szene wird korrekterweise nicht erkannt. (4) Durch Sample Relaxation aufgeblähte Kernel erzeugen einen Hotspot, der zu einer Fehlererkennung der Szene führt.

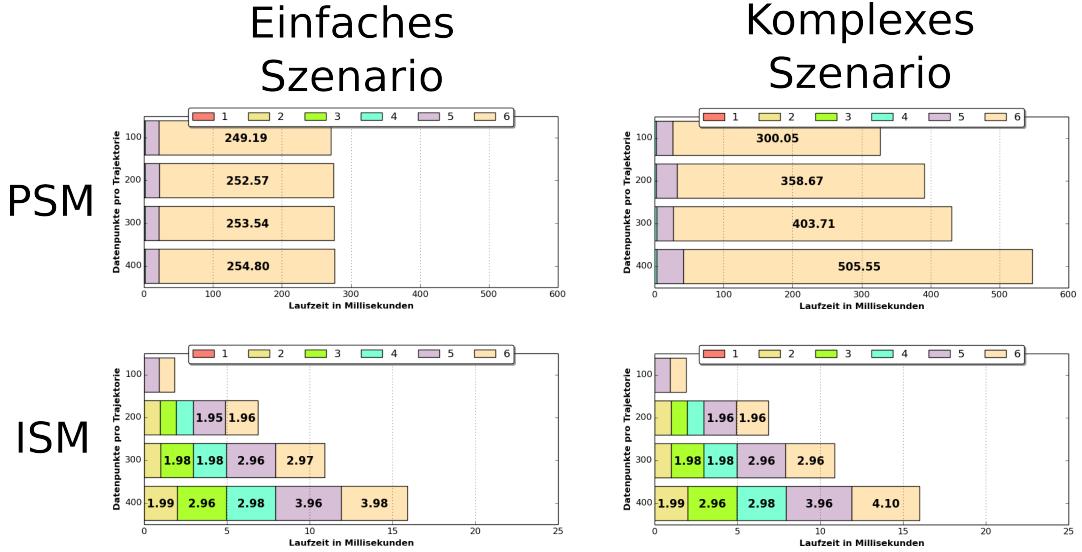


Abbildung 8.12: Vergleich der Laufzeiten von PSM und ISM für ein einfaches und ein komplexes Szenario. Das ISM skaliert mit der Anzahl der Lerndaten, das PSM mit der Komplexität der Szene.

PSM diskutiert. Zusätzlich wird das Wachstum der Modelle beider Systeme verglichen. Dies geschieht anhand des Szenarios, dass für das erste Experiment erstellt wurde. Aus Gründen der Übersichtlichkeit ist dieser Abschnitt entsprechend in Unterabschnitte unterteilt.

8.6.1 Einfache und komplexe Szene

Für den Vergleich der Laufzeiten von PSM und ISM wurden zwei verschiedene Szenen herangezogen. Die einfache Szene besteht aus einer linienförmigen Trajektorie, deren Länge zwischen 100 – 400 Datenpunkten variiert. Die andere, komplexe Szene besteht aus einer treppenförmigen Trajektorie, bei der die Anzahl der Stufen variiert. Alle 100 Datenpunkte ändert die Trajektorie ihre Richtung. Anhand beider Szenen sollen die Vorteile und Nachteile beider Systeme demonstriert werden. Das Experiment sieht vor, dass für jedes erwartete Objekt in der Szene genau eine Evidenz vorhanden ist.

Abbildung 8.12 zeigt die Ergebnisse der Laufzeitmessungen, die nun in Hinblick auf ihre Skalierung miteinander verglichen werden. Das PSM zeigt für die einfache Szene identische Laufzeiten, die von der Länge der Trajektorie unabhängig sind. Dies liegt daran, dass die Trajektorie auf Grund ihrer Linienform jeweils immer mit nur einem einzigen Gauss-Kernel erlernt wird. Der Aufwand für dessen Auswertung ist immer identisch.

Anders verhält es sich für die komplexe Szene, bei der die Laufzeit mit der Komplexität der Szene steigt. Für die treppenförmige Trajektorie wird pro Teilsegment ein generer

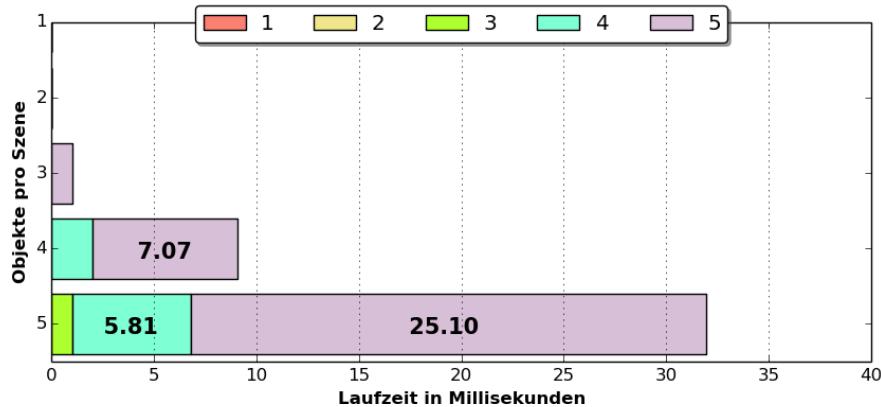


Abbildung 8.13: Laufzeiten des PSM in Abhängigkeit von Objekten und Evidenzen.

Gauss-Kernel benötigt. Mit jedem Richtungswechsel steigt also die Anzahl der Kernel und damit der Aufwand für die Auswertung.

Die Szenenkomplexität spielt für das ISM keine Rolle, wie sich aus einem Vergleich der Diagramme für die einfache und komplexe Szene erkennen lässt. Klar ersichtlich ist jedoch eine Abhängigkeit der Laufzeit zur Menge der Datenpunkte.

Das Verhalten von PSM und ISM ist nicht überraschend, da es der in Form beider Systeme vorliegende Spezialfall aus dem Allgemeinen ableiten lässt. Ein nicht-parametrisches Modell berücksichtigt lediglich Daten, ein parametrisches Modell abstrahiert über die Daten, extrahiert also daraus Informationen, die durch die entsprechenden Modellparameter beschrieben werden.

8.6.2 Laufzeit des PSM

Bisher wurden nicht alle Parameter betrachtet, die sich auf die Laufzeit des PSM auswirken. Im obigen Experiment wurde die Anzahl der Evidenzen als fix betrachtet und entsprach der Anzahl der Objekte in der Szene. Hier werden nun die Auswirkungen beider Parameter betrachtet.

Abbildung 8.13 zeigt die Laufzeitbetrachtung für beide Parameter. Der Balken kodiert die Anzahl der Objekte pro Szene, die einzelnen Abschnitte die Laufzeit für die in der Legende beschriebene Menge an Evidenzen. Der ausschlaggebende Faktor ist die Anzahl der Objekte. Ebenfalls markant sind die Evidenzen, wenn auch deren Auswirkung auf die Laufzeit weniger stark ausgeprägt ist.

Der Grund für das Laufzeitverhalten lässt sich mit der vollständigen Durchsuchung des Hypothesenraums begründen. Der hiermit verbundene Aufwand ist exponentiell, wobei die Evidenzen als Basis und die Objekte als Exponent fungieren. Eine Verbesserung der Laufzeit würde sich also durch eine effizientere Durchsuchung des Hypothesenraums erzielen lassen, beispielsweise mit einem auf Stichproben basierten Verfahren.

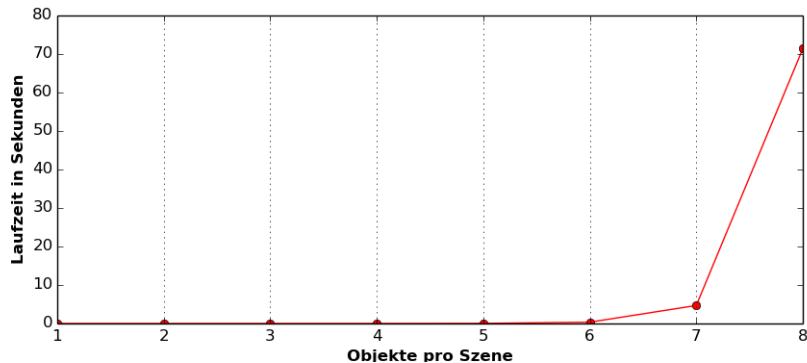


Abbildung 8.14: Das PSM skaliert exponentiell zur Anzahl der Objekte pro Szene.

Abbildung 8.14 illustriert das exponentielle Wachstum der Laufzeit. Für die Aufnahme wurde wieder davon ausgegangen, dass pro Objekt eine Evidenz vorliegt. Für einschließlich sechs Objekte liegt die Laufzeit bei unter einer halben Sekunde, das System kann also als echtzeitfähig betrachtet werden. Für eine höhere Objektzahl steigt die Dauer eines Durchlaufs bereits auf knapp fünf Sekunden bzw. über eine Minute.

Das ISM skaliert deutlich günstiger. Sein Laufzeitverhalten ist linear, im ungünstigsten Fall einer Vollvermaschung quadratisch (vergleiche auch [Rec13]). Die Laufzeit ist im Vergleich zum PSM im Schnitt um zwei Größenordnungen geringer und liegt im Bereich von wenigen Millisekunden. Hierbei muss jedoch berücksichtigt werden, dass das ISM in diesem Experiment nur mit wenigen hundert Datenpunkten zu arbeiten hat. Für eine Szenengröße von bis zu sechs Objekten zeigt sich für den menschlichen Betrachter kein wahrnehmbarer Unterschied, daher können beide Systeme für diesen Bereich als vergleichbar angesehen werden.

8.6.3 Modellwachstum

Wie schon die Laufzeit spielt die Komplexität der Szene eine große Rolle für die Modellgröße. Es bietet sich daher an, die im ersten Unterabschnitt definierten Szenen für die Betrachtung der Modellskalierung heranzuziehen. Der Einfachheit halber wurde die Modellgröße in KiB ermittelt. Die Modelldatei besteht beim PSM aus einer XML-Datei, beim ISM aus einer sqlite-Datenbank. Zwar legt letztere die Daten effizienter ab, da hier jedoch nur die Skalierung betrachtet wird spielt dies keine Rolle.

Die Modellskalierung wird zuerst anhand der einfachen Szene betrachtet. Die linke Grafik in Abbildung 8.15 illustriert, dass das PSM eine konstante Modellgröße beibehält. Dies lässt sich wieder dadurch begründen, dass die linienförmige Trajektorie durch nur einen einzelnen Kernel modelliert wird. Das ISM skaliert linear zur Anzahl der Lerndaten, die es alle in seiner Datenbank ablegt.

Die rechte Grafik zeigt die Skalierung in Abhängigkeit von der Anzahl der Stufen bzw. Cluster in der treppenförmigen Trajektorie. Für das PSM skaliert die Größe des Modells

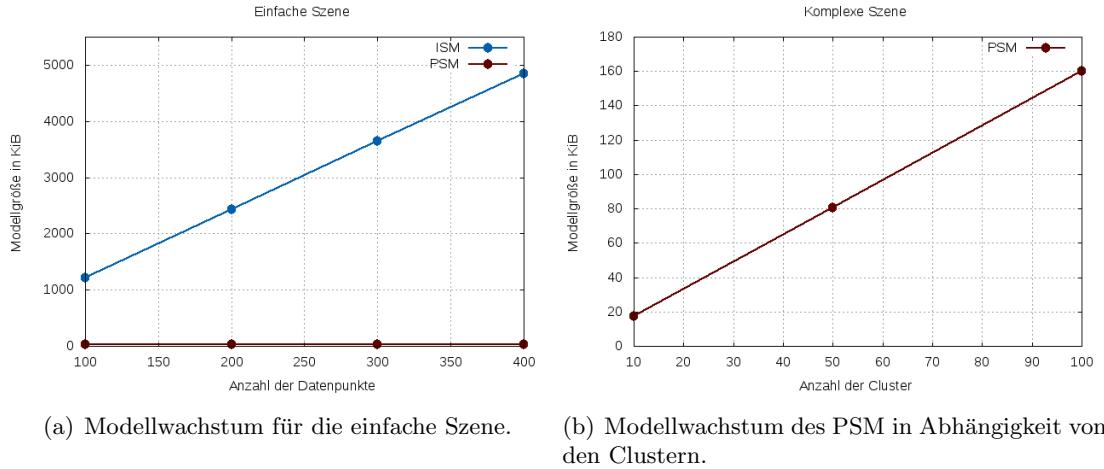


Abbildung 8.15: Das Modellwachstum beider Systeme für das einfache und komplexe Szenario.

linear. Das ISM wird in dieser Grafik bewusst nicht gezeigt. Wie bereits bekannt wächst dessen Modell linear in Abhängigkeit zu den darin enthaltenen Daten. Unter der Annahme, dass jeder Cluster dieselbe Menge an Datenpunkten enthält, würde die Steigung der Geraden hier von der Anzahl der Punkte pro Cluster abhängen. Da diese Anzahl jedoch beliebig wählbar ist, würde das Resultat neben dem bereits bekannten linearen Wachstum keine nennenswerte Aussage treffen.

8.7 Fazit

Die Evaluation hat die Anforderungen an das im Rahmen dieser Arbeit entwickelte Szenenerkennungssystem bestätigt. Es ist in der Lage, Relationen auf Basis relativer Posen zu bewerten. Der Umgang mit Clutter und fehlenden Objekten stellt ebenfalls kein Problem dar. Durch Verzicht auf harte Objektidentitäten ist die Robustheit gegenüber Fehldetections der Objektdetektoren gegeben.

Die modellbedingten weichen Entscheidungsgrenzen bringen einen Bias gegenüber bestimmten relativen Posen mit sich, dies ist jedoch im akzeptablen Bereich. Eine Reihe von Experimenten legt den Schluss nahe, dass überhängende Wahrscheinlichkeitsmasse und Hotspots auf Grund der Überlagerung von Gauss-Kerneln keine Rolle spielen.

Als problematisch hat sich die Laufzeit herausgestellt. Das PSM liefert nur in einem relativ eingeschränkten Arbeitsbereich akzeptable Laufzeiten. Auf Grund des exponentiellen Wachstums ist die Echtzeiterkennungen maximal bis zu einer Szenengröße von sechs Objekten möglich. Eine solche Einschränkung existiert beim ISM nicht.

Das Modell hingegen skaliert linear zur Komplexität der Szene. Dies ist ein großer Vorteil gegenüber dem ISM, dessen Modell zur Anzahl der Trainingsdaten steigt und daher

Probleme mit langen bzw. vielen Trainingssequenzen hat. Weiterhin kann über eine entsprechende Konfiguration des Lerners ein Kompromiss zwischen Szenenkomplexität und Genauigkeit geschlossen werden, mit dem die Modellgröße nach Bedarf beeinflusst werden kann. Ein solcher Mechanismus existiert beim ISM nicht.

9. Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein System zur Szenenerkennung entwickelt, das beim *Programmieren durch Vormachen* eingesetzt werden kann, um anhand von Beobachtungen der Umwelt die Realisierbarkeit von Aktionen abzuleiten. Das System wurde basierend auf der bayesschen Statistik entwickelt, um den Umgang mit Unsicherheiten zu ermöglichen, wie sie bei der Objektdetektion und -lokalisierung auftreten können. Zur Modellierung der Szene wurde ein *Constellation Model* eingesetzt, das um Auftrittswahrscheinlichkeiten erweitert wurde. Weiterhin wurde das sternförmige Relationsmodell auf eine Baumstruktur erweitert.

Es wurden drei Informationsquellen in das Modell mit einbezogen. Die Erscheinungsformen der an der Szene beteiligten Objekte, deren Auftrittshäufigkeiten sowie Relationen zwischen den Objekten in Form von relativen Posen. Letztere erlauben einen gewissen Grad an Dynamik, der von den Lerndaten vorgegeben wird. Das darunterliegende probabilistische Posenmodell umfasst sowohl Position als auch Orientierung und wurde auf Basis von Quaternionen realisiert. Invarianzeigenschaften bezüglich Translation und Rotation sind gegeben, wodurch eine Wiedererkennung der Szenen auch abseits vom ursprünglichen Ort der Demonstration möglich ist.

Robustheit gegenüber fehlenden Objekten wurde eingearbeitet. Die Szene wird bis zu einem bestimmten Grad der Degeneration erkannt, die Unvollständigkeit spiegelt sich in einem Absinken der zugehörigen Wahrscheinlichkeit wieder. Das Problem des Referenzobjekts wurde dadurch umgangen, dass für jedes Objekt ein Modell erlernt wurde, in welchem es den Referenzpunkt bildet. Überzählige Objekte werden im Inferenzprozess berücksichtigt, wirken sich aber nicht negativ auf die Erkennungsleistung aus. Detektorfehlerkennungen betreffend die Erscheinung eines Objekts werden vom System kompensiert, insofern ein ähnlicher Vorfall in den Lerndaten aufgetreten ist.

Das System wurde aus Gründen der Wiederverwendbarkeit einzelner Komponente in mehrere Teile untergliedert und in die bereits bestehende Infrastruktur integriert. Im

Rahmen der Evaluation wurde die korrekte Funktionsweise des Systems und die Einhaltung aller Anforderungen bestätigt. Unterschiede und Gemeinsamkeiten zum Implicit Shape Model wurden herausgearbeitet. Aus problematisch stellte sich die Laufzeit des Systems heraus. Diese ist exponentiell, weswegen Echtzeitfähigkeit nur bis zu maximal sechs Objekten pro Szene vorliegt.

An dieser Stelle befindet sich auch das größte Potential für weiterführende Forschungen. Der Aufwand kann gesenkt werden, indem der Hypothesenraum nur partiell durchsucht wird, beispielsweise mit einem auf Stichproben basierten Verfahren. Ebenfalls möglich wäre eine organisatorische Lösung, bei der einzelne Szenenmodelle, die einen Teil der Szene modellieren, hierarchisch gestapelt werden. Echtzeitfähigkeit für beliebig viele Objekte wird dann erreicht, indem die Anzahl auf maximal sechs Objekte pro Modell beschränkt wird. Generell ist es auch möglich, den Prozess der Inferenz zu parallelisieren, entweder auf Ebene der Szenenmodelle oder der darin untersuchten Hypothesen.

Erweiterungen am Szenenmodell bieten sich ebenfalls an. So können durch eine Korrelation von Position und Orientierung Fehlerkennungen der Szene vermieden werden. Dies ließe sich z.B. realisieren, indem pro Gauss-Kernel der Position eine eigene Gauss-Mischverteilung über die Orientierung gelernt wird. Die Robustheit des Systems könnte durch die Integration eines rekursiven Bayes-Filters erhöht werden, der die Erkenntnisse über die Szene als Vorwissen für den nächsten Zeitschritt übernimmt. Eine weitere Steigerung ließe sich durch die Betrachtung ganzer Objektklassen erreichen. Hierfür müsste lediglich eine Komponente vorgeschaltet werden, welche die Objektklassen bestimmt.

Das wohl größte Forschungspotential besteht in der Integration des Szenenerkenners in ein *Active Vision System* (siehe [YT96, Eid10]), das zur Exploration der Szene eingesetzt wird. Hierzu müsste das System lediglich um die Prädiktion noch unentdeckter Objekte erweitert werden.

Literaturverzeichnis

- [AAD06] Pedram Azad, Tamim Asfour und Rüdiger Dillmann: *Combining Appearance-based and Model-based Methods for Real-Time Object Recognition and 6D Localization*. In: *IROS*, Seiten 5339–5344. IEEE, 2006.
- [AAD07] P. Azad, T. Asfour und R. Dillmann: *Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems*. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Seiten 919–924, 2007.
- [AAD09] Pedram Azad, Tamim Asfour und Rüdiger Dillmann: *Accurate shape-based 6-DoF pose estimation of single-colored objects*. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.
- [AT13] Alexander Andreopoulos und John K. Tsotsos: *50 Years of object recognition: Directions forward*. Computer Vision and Image Understanding, 117(8):827–891, 2013, ISSN 1077-3142. <http://www.sciencedirect.com/science/article/pii/S107731421300091X>.
- [AZ95] M. Armstrong und A. Zisserman: *Robust Object Tracking*. In: *Asian Conference on Computer Vision*, Band I, Seiten 58–61, 1995.
- [Aza08] Pedram Azad: *Visual Perception for Manipulation and Imitation in Humanoid Robots*. Dissertation, Karlsruhe Institute of Technology, 2008.
- [Bal81] Dana H. Ballard: *Generalizing the Hough transform to detect arbitrary shapes*. Pattern Recognition, 13(2):111–122, 1981.
- [Bar12] David Barber: *Bayesian Reasoning and Machine Learning*. Cambridge University Press, New York, NY, USA, 2012, ISBN 0521518148, 9780521518147.
- [BBL02] M. R. Boutell, C. B. Brown und J. Luo: *Review of the State of the Art in Semantic Scene Classification*. Technischer Bericht, Rochester, NY, USA, 2002.
- [BETVG08] Herbert Bay, Andreas Ess, Tinne Tuytelaars und Luc Van Gool: *Speeded-Up Robust Features (SURF)*. Comput. Vis. Underst., 110(3):346–359, Juni 2008, ISSN 1077-3142. <http://dx.doi.org/10.1016/j.cviu.2007.09.014>.

- [BI11] Ali Borji und Laurent Itti: *Scene classification with a sparse set of salient regions*. In: *International Conference on Robotics and Automation (ICRA)*, Seiten 1902–1908. IEEE, 2011. <http://dblp.uni-trier.de/db/conf/icra/icra2011.html#BorjiI11>.
- [Bis07] Christopher M. Bishop: *Pattern Recognition and Machine Learning*. Springer, 1st ed. 2006. corr. 2nd printing 2011 Auflage, 2007, ISBN 0387310738.
- [BK10] H. S. Bhat und N. Kumar: *On the derivation of the Bayesian Information Criterion*. Technischer Bericht, University of California, Merced, 2010.
- [BLP95] M.C. Burl, T. K. Leung und P. Perona: *Face Localization via Shape Statistics*. In: *International Workshop on Automatic Face and Gesture Recognition*, 1995.
- [BP96] M.C. Burl und P. Perona: *Recognition of Planar Object Classes*. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seiten 223–230, 1996.
- [BWP98] Michael C. Burl, Markus Weber und Pietro Perona: *A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry*. In: *Proceedings of the 5th European Conference on Computer Vision*, Band 2 der Reihe *ECCV '98*, Seiten 628–641, London, UK, UK, 1998. Springer-Verlag, ISBN 3-540-64613-2. <http://dl.acm.org/citation.cfm?id=645312.648915>.
- [Cho06] S. Choe: *Statistical Analysis of Orientation Trajectories via Quaternions with Applications to Human Motion*. Dissertation, University of Michigan, 2006.
- [CK99] Peng Chang und John Krumm: *Object Recognition with Color Cooccurrence Histogram*. In: *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 1999.
- [DD92] Daniel DeMenthon und Larry S. Davis: *Model-Based Object Pose in 25 Lines of Code*. In: Giulio Sandini (Herausgeber): *ECCV*, Band 588 der Reihe *Lecture Notes in Computer Science*, Seiten 335–343. Springer, 1992, ISBN 3-540-55426-2.
- [DD95] Daniel F. DeMenthon und Larry S. Davis: *Model-Based Object Pose in 25 Lines of Code*. International Journal of Computer Vision, 15:123–141, 1995.
- [DHS00] Richard O. Duda, Peter E. Hart und David G. Stork: *Pattern Classification*. Wiley-Interscience, 2. Auflage, 2000.
- [DLR77] A. P. Dempster, N. M. Laird und D. B. Rubin: *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the Royal Statistical Society, Series B, 39:1—38, 1977.

- [EGS⁺12] Robert Eidenberger, Thilo Grundmann, Martin Schneider, Wendelin Feiten, Michael Fiegert, Georg v. Wichert und Gisbert Lawitzky: *Towards Service Robots for Everyday Environments*, Kapitel Scene Analysis for Service Robots, Seiten 181–213. Springer-Verlag Berlin Heidelberg, 2012, ISBN 978-3-642-25116-0.
- [EGZ09] Robert Eidenberger, Thilo Grundmann und Raoul Zoellner: *Probabilistic action planning for active scene modeling in continuous high-dimensional domains*. In: *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, ICRA'09, Seiten 2639–2644, Piscataway, NJ, USA, 2009. IEEE Press, ISBN 978-1-4244-2788-8. <http://dl.acm.org/citation.cfm?id=1703775.1703877>.
- [Eid10] Robert Eidenberger: *Probabilistic Active Perception Planning for Autonomous Robots in Everyday Environments*. Dissertation, Johannes Kepler University Linz, 2010.
- [EZS09] Robert Eidenberger, Raoul Zoellner und Josef Scharinger: *Probabilistic Occlusion Estimation in Cluttered Environments for Active Perception Planning*. In: *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Seiten 1248–1253, 2009, ISBN 978-1-4244-2852-6.
- [FAEG09] Wendelin Feiten, Pradeep Atwal, Robert Eidenberger und Thilo Grundmann: *6D Pose Uncertainty in Robotic Perception*. In: Torsten Kröger und Friedrich M. Wahl (Herausgeber): *Advances in Robotics Research*, Kapitel 9, Seiten 89–98. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, ISBN 978-3-642-01212-9. http://dx.doi.org/10.1007/978-3-642-01213-6_9.
- [FLH13] W. Feiten, M. Lang und S. Hirche: *Rigid Motion Estimation using Mixtures of Projected Gaussians*. In: *16th International Conference on Information Fusion (FUSION)*, Seiten 1465–1472, Jul 2013.
- [FPZ03] Robert Fergus, Pietro Perona und Andrew Zisserman: *Object class recognition by unsupervised scale-invariant learning*. In: *In CVPR*, Seiten 264–271, 2003.
- [FPZ05] Robert Fergus, Pietro Perona und Andrew Zisserman: *A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition*. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Band Volume 1 der Reihe *CVPR '05*, Seiten 380–387, Washington, DC, USA, 2005. IEEE Computer Society, ISBN 0-7695-2372-2. <http://dx.doi.org/10.1109/CVPR.2005.47>.
- [FS97] Yoav Freund und Robert E. Schapire: *A Decision-theoretic Generalization of On-line Learning and an Application to Boosting*. Journal of Computer and System Sciences, 55(1):119–139, August 1997, ISSN 0022-0000.

- [GA98] James S. Goddard und Mongi A. Abidi: *Pose and Motion Estimation Using Dual quaternion-based Extended Kalman Filtering*. In: Richard N. Ellson und Joseph H. Nurrr (Herausgeber): *Three-Dimensional Image Capture and Applications*, Band 3313 der Reihe *SPIE Proceedings*, Seiten 189–200. SPIE, 1998.
- [GK13] Jared Glover und Leslie Pack Kaelbling: *Tracking 3-D Rotations with the Quaternion Bingham Filter*. Technischer Bericht, MIT, 2013.
- [GK14] Jared Glover und Leslie Pack Kaelbling: *Tracking the Spin on a Ping Pong Ball with the Quaternion Bingham Filter*. In: *IEEE Conference on Robotics and Automation (ICRA)*, 2014. <http://lis.csail.mit.edu/pubs/glover-icra14.pdf>.
- [GL11] Kristen Grauman und Bastian Leibe: *Visual Object Recognition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.
- [God97] James S. Goddard: *Pose and Motion Estimation from Vision Using Dual Quaternion-based Extended Kalman Filtering*. Dissertation, The University of Tennessee, 1997.
- [Har58] H. O. Hartley: *Maximum likelihood estimation from incomplete data*. *Biometrika*, 14(2):174–194, 1958.
- [Haz02] Michiel Hazewinkel (Herausgeber): *Encyclopaedia of Mathematics*. Springer-Verlag, 2002.
- [HM96] S. K. Nayar S. A. Nene H. Murase: *Real-time 100 object recognition system*. *Robotics and Automation 1996 IEEE International Conference*, 3:2321–2325, April 1996.
- [HS88] Chris Harris und Mike Stephens: *A combined corner and edge detector*. In: *In Proc. of Fourth Alvey Vision Conference*, Seiten 147–151, 1988.
- [HS90] C. Harris und C. Stennett: *RAPID - a video rate object tracker*. In: *Proceedings of the British Machine Vision Conference*, Seiten 15.1–15.6. BMVA Press, 1990.
- [JTE⁺12] Dominik Joho, Gian Diego Tipaldi, Nikolas Engelhard, Cyrill Stachniss und Wolfram Burgard: *Nonparametric Bayesian Models for Unsupervised Scene Analysis and Reconstruction*. In: *Proceedings of Robotics: Science and Systems (RSS)*, Sydney, Australia, 2012.
- [KGJH13] Gerhard Kurz, Igor Gilitschenski, Simon J. Julier und Uwe D. Hanebeck: *Recursive Estimation of Orientation Based on the Bingham Distribution*. In: *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, 2013.

- [KH05] Sanjiv Kumar und Martial Hebert: *A Hierarchical Field Framework for Unified Context-Based Classification*. In: IEEE (Herausgeber): *Tenth IEEE International Conference on Computer Vision (ICCV '05)*, Band 2, Seiten 1284 – 1291, October 2005.
- [KHDM98] Josef Kittler, Mohamad Hatef, Robert P. W. Duin und Jiri Matas: *On combining classifiers*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20:226–239, 1998.
- [KMA01] D. Kragic, A. T. Miller und P. K. Allen: *Real-time tracking meets online grasp planning*. In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, Seiten 2460–2465. IEEE, 2001, ISBN 0-7803-6576-3.
- [LBP98] Thomas K. Leung, Michael C. Burl und Pietro Perona: *Probabilistic Affine Invariants for Recognition*. In: *In Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, Seiten 678–684, 1998.
- [LF12] Muriel Lang und Wendelin Feiten: *MPG - Fast Forward Reasoning on 6 DOF Pose Uncertainty*. In: *ROBOTIK 2012 - 7th German Conference on Robotics*. VDE-Verlag, 2012, ISBN 978-3-8007-3418-4. <http://dblp.uni-trier.de/db/conf/robotik/robotik2012.html#LangF12>.
- [LLG11] Alain Lehmann, Bastian Leibe und Luc Van Gool: *Fast PRISM: Branch and Bound Hough Transform for Object Class Detection*. International Journal of Computer Vision, 94(2):175–197, June 2011.
- [LLS04] Bastian Leibe, Ales Leonardis und Bernt Schiele: *Combined Object Categorization and Segmentation With An Implicit Shape Model*. In: *In ECCV workshop on statistical learning in computer vision*, Seiten 17–32, 2004.
- [LLS08] Bastian Leibe, Aleš Leonardis und Bernt Schiele: *Robust Object Detection with Interleaved Categorization and Segmentation*. Int. J. Comput. Vision, 77(1-3):259–289, May 2008, ISSN 0920-5691. <http://dx.doi.org/10.1007/s11263-007-0095-3>.
- [Low99] David G. Lowe: *Object Recognition from Local Scale-Invariant Features*. In: *Proceedings of the International Conference on Computer Vision*, Seiten 1150–1157. IEEE Computer Society, 1999, ISBN 0-7695-0164-8. <http://dl.acm.org/citation.cfm?id=850924.851523>.
- [LSS05] Bastian Leibe, Edgar Seemann und Bernt Schiele: *Pedestrian Detection in Crowded Scenes*. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Band 1 der Reihe *CVPR '05*, Seiten 878–885, Washington, DC, USA, 2005. IEEE Computer Society, ISBN 0-7695-2372-2. <http://dx.doi.org/10.1109/CVPR.2005.272>.
- [Mah36] P. C. Mahalanobis: *On the generalised distance in statistics*. In: *Proceedings National Institute of Science, India*, Band 2, Seiten 49–55, April 1936.

- [MB01] Eric Marchand und Patrick Bouthemy: *A 2D-3D Model-based Approach to Real-time Visual Tracking*. IVC, 19:941–955, 2001.
- [MCT05] Erik Murphy-Chutorian und Jochen Triesch: *Shared Features for Scalable Appearance-Based Object Recognition*. In: *7th IEEE Workshop on Applications of Computer Vision / IEEE Workshop on Motion and Video Computing (WACV/MOTION 2005)*, Seiten 16–21. IEEE Computer Society, 2005, ISBN 0-7695-2271-8.
- [Mei13] Pascal Meißner: *Recognizing Scenes with Hierarchical Implicit Shape Models based on Spatial Object Relations for Programming by Demonstration*. In: *16th International Conference on Advanced Robotics (ICAR)*, 2013.
- [Mei14] Pascal Meißner: *Active Scene Recognition for Programming by Demonstration using Next-Best-View Estimates from Hierarchical Implicit Shape Models*. In: *Proc. of The International Conference in Robotics and Automation (ICRA)*, 2014.
- [MN93] H. Murase und S.K. Nayar: *Learning and Recognition of 3D Objects from Appearance*. In: *IEEE Workshop on Qualitative Vision*, Seiten 39–50, Jun 1993.
- [MYB⁺01] Joao Luis Marins, Xiaoping Yun, Eric R. Bachmann, Robert B. McGhee und Michael Zyda: *An extended Kalman filter for quaternion-based orientation estimation using MARG sensors*. In: *International Conference on Intelligent Robots and Systems (IROS)*, Seiten 2003–2011, 2001.
- [OM02] Stepán Obdrzálek und Jiri Matas: *Object Recognition using Local Affine Frames on Distinguished Regions*. In: Paul L. Rosin und A. David Marshall (Herausgeber): *British Machine Vision Conference (BMVC)*. British Machine Vision Association, 2002, ISBN 1-901725-19-7.
- [OT01] Aude Oliva und Antonio Torralba: *Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope*. International Journal of Computer Vision, 42:145–175, 2001.
- [PMD] Sven R. Schmidt Rohr Pascal Meißner, Joachim Gehrung und Rüdiger Dillmann: *Probabilistic Scene Recognition using Hierarchical Constellation Models over Spatial Relations extracted from demonstrated Object Trajectories*.
- [Pri12] Simon J. D. Prince: *Computer vision: models, learning, and inference*. Cambridge University Press, New York, 2012, ISBN 978-1-107-01179-3.
<http://www.computervisionmodels.com/>.
- [PS11] Christian Potthast und Gaurav S. Sukhatme: *A Probabilistic Framework for Next Best View Estimation in a Cluttered Environment*. 2011.
- [QCG⁺09] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler und Andrew Y. Ng: *ROS: an open-source Robot Operating System*. In: *ICRA Workshop on Open Source Software*, 2009.

- [QT09] Ariadna Quattoni und Antonio Torralba: *Recognizing indoor scenes*. In: *Computer Vision and Pattern Recognition*, Seiten 413–420, 2009.
- [RD08] Ananth Ranganathan und Frank Dellaert: *Semantic modeling of places using objects*. 2008.
- [Rec13] Reno Reckling: *Szenenerkennung mittels hierarchischer Implicit Shape Models basierend auf räumlichen Objektrelationen für das Programmieren durch Vormachen*. Diplomarbeit, Karlsruhe Institute of Technology, 2013.
- [SGS08] A. Harati S. Gächter und R. Siegwart: *Structure verification toward object classification using a range camera*. In: *International Conference on Intelligent Autonomous Systems (IAS)*, 2008.
- [SI07] Christian Siagian und Laurent Itti: *Rapid Biologically-Inspired Scene Classification Using Features Shared with Visual Attention*. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 29(2):300–312, 2007, ISSN 0162-8828. <http://dx.doi.org/10.1109/TPAMI.2007.40>.
- [SK08] Bruno Siciliano und Oussama Khatib: *Springer Handbook of Robotics*. Springer, 2008. <http://www.libreka.de/9783540239574/>.
- [SL13] Tristram Southey und James J. Little: *3D spatial relationships for improving object detection*. In: *ICRA*, Seiten 140–147. IEEE, 2013, ISBN 978-1-4673-5641-1. <http://dblp.uni-trier.de/db/conf/icra/icra2013.html#SoutheyL13>.
- [ST94] Jianbo Shi und Carlo Tomasi: *Good Features to Track*. In: *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seiten 593–600, 1994.
- [Stu64] J. Stuelpnagel: *On the Parametrization of the Three-Dimensional Rotation Group*. SIAM Review, 6:422–430, 1964.
- [SWB⁺07] Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber und Tomaso Poggio: *Robust object recognition with cortex-like mechanisms*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29:411–426, 2007.
- [Sze10] Richard Szeliski: *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st Auflage, 2010, ISBN 1848829345, 9781848829343.
- [TBF05] Sebastian Thrun, Wolfram Burgard und Dieter Fox: *Probabilistic Robotics*. Intelligent robotics and autonomous agents. The MIT Press, August 2005, ISBN 0262201623.
- [VJ01] Paul Viola und Michael Jones: *Robust real-time object detection*. In: *Workshop on Statistical and Computational Theories of Vision*, 2001.

- [WH97] Patrick Wunsch und Gerd Hirzinger: *Real-time visual tracking of 3D objects with dynamic handling of occlusion*. In: *ICRA*, Seiten 2868–2873. IEEE, 1997, ISBN 0-7803-3612-7.
- [Wit13] Valerij Wittenbeck: *Reactive Object Search for Scene Recognition in Programming by Demonstration*. Diplomarbeit, Karlsruher Institut für Technologie (KIT), 2013.
- [Wu83] C. F. Jeff Wu: *On the convergence properties of the EM algorithm*. The Annals of Statistics, 11(1):95–103, 1983.
- [WWP00a] M. Weber, M. Welling und P. Perona: *Towards Automatic Discovery of Object Categories*. In: *Proc. IEEE Comp. Soc. Conf. Comp. Vision and Pattern Recog. (CVPR)*, 2000.
- [WWP00b] M. Weber, M. Welling und P. Perona: *Unsupervised learning of models for recognition*. In: *6th European Conference on Computer Vision, ECCV2000*, Seiten 18–32, 2000.
- [YT96] Yiming Ye und John K. Tsotsos: *Sensor Planning in 3D Object Search*. CVIU, 73:145–168, 1996.
- [ZSK99] Jianwei Zhang, Ralf Schmidt und Alois Knoll: *Appearance-Based Visual Learning in a Neuro-Fuzzy Model for Fine-Positioning of Manipulators*. In: *ICRA*, Seiten 1164–1169. IEEE Robotics and Automation Society, 1999.