

Sven Hellbach

**Entwicklung von Methoden zur Unterscheidung
und Interpretation von Bewegungsmustern in
dynamischen Szenen**

**Entwicklung von Methoden zur
Unterscheidung und Interpretation
von Bewegungsmustern in
dynamischen Szenen**

Von Sven Hellbach



Universitätsverlag Ilmenau
2010

Impressum

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über <http://dnb.d-nb.de> abrufbar.

Diese Arbeit hat der Fakultät für Informatik und Automatisierung der Technischen Universität Ilmenau als Dissertation vorgelegen.

Tag der Einreichung: 25. September 2010

1. Gutachter: Prof. Dr.-Ing. Horst-Michael Groß
(Technische Universität Ilmenau)

2. Gutachter: Dr. Julian P. Eggert
(Honda Research Institute Europe GmbH)

3. Gutachter: Prof. Dr. Martin Giese
(Universität Tübingen)

Tag der Verteidigung: 24. Februar 2010

Technische Universität Ilmenau/Universitätsbibliothek

Universitätsverlag Ilmenau

Postfach 10 05 65

98684 Ilmenau

www.tu-ilmenau.de/universitaetsverlag

Herstellung und Auslieferung

Verlagshaus Monsenstein und Vannerdat OHG

Am Hawerkamp 31

48155 Münster

www.mv-verlag.de

ISBN 978-3-939473-79-4 (Druckausgabe)

urn:nbn:de:gbv:ilm1-2010000073

Titelfoto: photocase.com

世界上本没有路。走的人多了，也就成了路。

Ein Weg entsteht, wenn man ihn geht.

Chinesisches Sprichwort

Kurzbeschreibung

Im Forschungsfeld der mobilen Assistenzroboter spielen Bewegungsabläufe eine zunehmend wichtige Rolle. Gerade in den Bewegungen der mit dem Assistenzroboter handelnden Person verstecken sich eine ganze Reihe Informationen, die zur Verbesserung der Interaktion herangezogen werden können.

Auch in der Biologie finden sich einige Hinweise, dass neben Formrepräsentationen auch Bewegungsabläufe zur Entscheidungsfindung verwendet werden. Zu Beginn dieser Arbeit werden daher auch eine Reihe an Publikationen aus der biologischen Forschung aufgearbeitet. Sie dienen als Motivation, sich mit Bewegungen zu beschäftigen und sollen Ideen liefern, die in den Bereich der mobilen Assistenzrobotik übertragen werden können.

Eine wichtige Fragestellung bezüglich der Analyse von Bewegungen stellt die Repräsentation der Bewegungstrajektorien dar. Außerdem muss geklärt werden, welche Ähnlichkeitsmaße in den komplexeren Verfahren zum Einsatz kommen können bzw. welche speziellen Anforderungen sie erfüllen müssen. Daher werden in dieser Arbeit in erster Linie eine Reihe von Repräsentationsformen aber auch Ähnlichkeitsmaßen systematisch vorgestellt und stellenweise erstmals bezüglich der Anwendung auf realen Bewegungen experimentell evaluiert. Um aus der Vielzahl an Repräsentationsformen die für die entsprechende Problemstellung richtige Auswahl zu treffen, wird in Zusammenhang mit den Untersuchungen eine Möglichkeit vorgestellt, den Auswahlprozess automatisiert durchführen zu können.

Den eigentlichen Kern der Arbeit stellen drei Verfahren dar, die im Wesentlichen den weiteren Verlauf einer beobachteten Bewegung über einen längeren Zeitraum vorhersagen können. Dabei handelt es sich um Echo State Netzwerke, Local Models und die spatio-temporale nicht-negative Matrixfaktorisierung (NMF). Für die aus der Literatur bekannten Echo State Netzwerke und Local Models werden einige Verbesserungen und Erweiterungen vorgestellt. Diese ermöglichen eine gezielte Anwendung zur Prädiktion von realen Bewegungsabläufen. Damit werden diese aus theoretischer Sicht gut erforschten Verfahren hinsichtlich eines praxistauglichen Einsatzes untersucht und angepasst.

Während Echo State Netzwerke und Local Models prinzipiell bereits für den Einsatz von Zeitreihenvorhersagen konzipiert sind, stellt die spatio-temporale NMF eine neue Möglichkeit der Trajektorien Dekomposition in eine Art Basisalphabet dar. Diese Dekomposition ermöglicht eine Beschreibungsform einer beobachteten Bewegung, die beispielsweise für semantische Repräsentation der Bewegung eingesetzt werden kann. Darüber hinaus besitzt sie die Fähigkeit, die Bewegung zu präzisieren. Zusätzlich kann gezeigt werden, dass Local Models und die spatio-temporale NMF auch zur Klassifikation von Bewegungsabläufen einsetzbar sind.

Die Arbeit als Ganzes versteht sich als einer der ersten Schritte zur systematischen Untersuchung von Bewegungsabläufen. Es wird daher auch nicht versucht, eine Lösung für ein konkretes Anwendungsszenario zu finden. Mit dieser Arbeit soll ein Entwickler in der Lage sein, aus einer breiten Palette an Werkzeugen sich für das Richtige für seinen speziellen Anwendungsfall zu entscheiden.

Abstract

In the field of mobile service robotics the analysis of motion sequences plays an increasing role. In particular, the movement of the person which is interacting with the robot is conveying a huge amount of information. This information can be exploited to improve the human-robot-interaction.

In the field of biology a lot of hints can be found that, beside the representation of shapes, also sequences of motion sequences are used to take decisions. Hence, at the beginning of this thesis a number of publications from biological research are revisited. They are meant as a motivation to concentrate on motions and will help to find ideas, which can be adapted to be used with mobile service robots.

An important problem concerning the analysis of motion poses the representation of the motion trajectories. Furthermore, it needs to be clarified, which similarity measures can be used within the more complex approaches and which demands they need to fulfill. Hence, a number of representations and similarity measures are presented systematically and are evaluated, partially for the first time, with respect to the application on real motion trajectories. To be able to select from the huge amount of representation forms, a method is proposed to automatically select the right form depending on the given task.

The essential part of this thesis is represented by three approaches, which are primarily meant to result in a long-term prediction of the observed motion. These approaches are Echo State Networks, Local Models and spatio-temporal non-negative matrix factorization (NMF). For the Echo State Networks and Local Models, which are known from literature, some improvements and extensions are presented. Those allow a dedicated application on real motion sequences. Hence, the theoretically well known approaches are evaluated and adapted for practical application.

While Echo State Networks and Local Models are designed to be applicable for the prediction of time series, the spatio-temporal NMF approach offers a new possibility to decompose motion into a special kind of basis alphabet. This decomposition allows a description of an observed motion, which can be used for a semantic labeling of the motion.

Furthermore it offers the ability to predict the further motion. Additionally, it can be shown, that Local Models and the spatio-temporal NMF are able to perform a motion trajectory classification task.

The thesis as a whole is meant to take first steps towards a systematic research of motion trajectories. Hence, it is not attempted to find a tailored solution for a certain application scenario. With this thesis a developer should be able to choose for his particular application from the wide range of tools.

Danksagung

Ich möchte an dieser Stelle die Gelegenheit nutzen mich bei all denjenigen zu bedanken, die mich auf dem Weg zu dieser Arbeit begleitet und unterstützt haben. Auf Grund der Vielzahl der Leute, hoffe ich, dass ich niemanden vergesse. Sollte sich dennoch jemand übergangen fühlen, sei ihm besonders gedankt.

In den letzten 4 Jahren hatte ich die Gelegenheit – vielmehr die Chance – an zwei wirklich guten Forschungseinrichtungen tätig sein zu können. Dabei war nicht nur das wissenschaftliche Niveau äußerst ergiebig, auch die menschliche Seite hat es ermöglicht, dass ich mich immer wohlgefühlt habe.

Als erstes sei meinem Doktorvater Prof. Horst-Michael Groß gedankt, der mich immer unterstützt und ermuntert hat. Und natürlich danke, dass ihre Wahl auf mich fiel. *Merci Mon Général!*

Ich hatte das große Glück einen zweiten Doktorvater zu haben. Auch Dr. Julian Eggert gilt mein ganz besonderer Dank für die vielen Gesprächsrunden, die mich auf dem richtigen Weg gehalten haben.

In diesem Zusammenhang geht mein Dank auch an Prof. Edgar Körner und Andreas Richter, die mich im Honda Research Institute herzlich aufgenommen haben und die finanziellen Rahmenbedingungen geschaffen haben.

Natürlich gilt mein Dank auch den vielen guten Geistern am Fachgebiet Neuroinformatik und Kognitive Robotik und dem Honda Research Institute Europe. Danke an die Administratorenarmee Sabine Schulz, Heike Groß und Burkhard Zittel. Insbesondere Sabine hat jetzt hoffentlich erstmal wieder Ruhe und muss mich nicht mehr gegen die aufmüpfige Technik verteidigen. Danke auch an Ute Schütz und Nevriye Memet. Beide haben eine gehörige bürokratische Last von mir genommen. Ute sei zusätzlich noch einmal für die vielen Gespräche gedankt, wenn mir mal wieder jemand die Nerven geraubt hat.

Weiterhin gilt meinen Kollegen bzw. Quasikollegen vom Fachgebiet Neuroinformatik und Kognitive Robotik und dem Honda Research Institute Offenbach ein Dank für die gute Arbeitsatmosphäre. Danke an Prof. Hans-Joachim Böhme, Dr. Andrea Scheidig, Dr. Klaus Debes, Dr.

Christoph Schröter, Nils Einecke, Erik Einhorn, Jens Kessler, Stefan Kirstein, Alexander König, Steffen Müller, Sven Rebhan, Erik Schaffernicht, Ronny Stricker und Micheal Volkhardt . Ich hoffe, Klaus ist mir nicht böse, dass er erst so weit unten genannt wird. Ohne ihn wäre vieles nicht möglich gewesen, einen besonderen Dank dafür. An Erik, Sven, Jens und Steffen geht nochmal ein besonderer Dank für die fachlichen und persönlichen Diskussionen.

In diesem Zusammenhang geht mein Dank auch an meine Studenten: Sandra Hesper, Antje Ober, Sören Strauß, Boris Lau, Konrad Schenk, Rene Matthias, Paul-Oliver Geue, Markus Eisenbach, Sebastian Hellwig, Sebastian Hommel, Hassan Tagharobi, Zhao Chen. Extra genannt sei Remo Pillat, der seine Arbeit zwar an der University of Central Florida erstellt hat, mir dafür aber bei meiner Reise nach Florida den einen oder anderen Geheimtipp geben konnte.

Bevor ich meine Tätigkeit am Fachgebiet Neuroinformatik und Kognitive Robotik aufnahm, hatte ich die Gelegenheit viele Jahre am Fachgebiet für Integrierte Hard- und Softwaresysteme als studentische Hilfskraft zu arbeiten. Dr. Karsten Henke, Dr. Heinz-Dietrich Wuttke und Jürgen Schmidt, die mir in den Jahren zu guten Freunden geworden sind, gilt gerade für diese Zeit mein herzlichster Dank. Sie waren eine wichtige Hilfe auf dem Weg zum Doktorand.

Auch bedanken möchte ich mich bei meinen langjährigen Freunden Friedemann Schoener, Wolfgang Steudel, Oliver Hofmann und Jingan Xu, besonders bei Friedemann für das eifrige Korrekturlesen. Danke nochmal an Jingan für die chinesische Übersetzung. Ich hoffe da steht auch wirklich der richtige Spruch.

Zu guter Letzt sei den wichtigsten Wegbegleitern, meinen Eltern, gedankt. Sie haben mir in all den Jahren den Rücken frei gehalten und mich bestärkt. Ganz herzlichen Dank dafür und für vieles mehr.

Inhaltsverzeichnis

Mathematische Notation	xxiii
1. Einleitung	1
1.1. Anspruch der Arbeit	3
1.2. Anwendungsszenario	4
1.2.1. Fazit	7
1.3. Gliederung der Arbeit	8
2. Motivation	13
2.1. Biologische Motivation	13
2.1.1. Simulander	14
2.1.2. Wahrnehmung von Bewegungen	15
2.1.3. Experimente mit Neugeborenen	17
2.1.4. Strukturen im menschlichen Gehirn	18
2.2. Verwandte Arbeiten	20
2.2.1. Anwendungsfelder	21
2.2.2. Methodik	23
2.2.2.1. Probabilistische Methoden	25
2.2.2.2. Neuronale Methoden	26
2.2.2.3. Analytische Methoden	27
2.2.2.4. Sonstige Verfahren	28
2.3. Fazit	29
3. Vorverarbeitung von Bewegungsabläufen	31
3.1. Datenakquisition	31
3.1.1. Point-Light Display	32

3.1.2.	Lasertracker	35
3.2.	Repräsentation von Bewegungstrajektorien	36
3.2.1.	Referenzsystem	40
3.2.1.1.	Kontextuelle Repräsentation	41
3.2.1.2.	Zielbezogene Repräsentation	42
3.2.1.3.	Trajektorienbezugssystem	42
3.2.2.	Abtastung	43
3.2.2.1.	Räumlich äquidistante Abtastung	44
3.2.2.2.	Zeitlich äquidistante Abtastung	44
3.2.3.	Koordinatentransformationen	45
3.2.4.	Auswahl der geeigneten Repräsentation	45
3.2.5.	Experimentelle Untersuchungen	50
3.2.6.	Fazit	55
3.3.	Bewertung der Ähnlichkeit von Trajektorien	57
3.3.1.	Fazit	60
4.	Echo State Netzwerke	63
4.1.	Vergleich von Prädiktionsalgorithmen	64
4.2.	Aufbau von ESN	65
4.2.1.	Funktionsweise	65
4.2.2.	Training und Anwendung	68
4.2.3.	Erweiterungen	71
4.3.	Experimentelle Evaluierung	72
4.3.1.	Referenzalgorithmen	73
4.3.2.	Gütekriterien	73
4.3.3.	Ergebnisse	76
4.4.	Anforderungen an den Realwelteinsatz	79
4.4.1.	Verkürzung der Beobachtungsphase	80
4.4.2.	Kompensation von Unterabtastung	82
4.4.3.	Prädiktion des Interaktionspartners	84
4.5.	Fazit	86
5.	Local Modelling	89
5.1.	Bestimmung der Einbettung	89
5.1.1.	Wahl der Einbettungsverzögerung	92
5.1.2.	Wahl der Einbettungsdimension	92
5.2.	Local Model Verfahren	93
5.2.1.	Local Average Model	96

5.2.2. Local Linear Model	97
5.3. Experimentelle Evaluierung	97
5.4. Anforderungen an den Realwelteinsatz	100
5.4.1. Verkürzung der Beobachtungsphase	100
5.4.2. Kompensation von Unterabtastung	102
5.4.3. Prädiktion des Interaktionspartners	104
5.5. Erweiterung der Local Models	105
5.5.1. Erweiterung der Wissensbasis	106
5.5.2. Erweiterung als Klassifikator	110
5.5.2.1. Learning Vector Quantisierung	110
5.5.2.2. Nearest-Neighbor-Klassifikation	111
5.5.2.3. Vergleichende Untersuchungen	112
5.6. Fazit	114
6. Spatio-temporale NMF	117
6.1. Nicht-negative Matrixzerlegung	121
6.1.1. Spärlichkeitskriterium	123
6.1.2. Transformationsinvarianz	125
6.2. Dekomposition von Bewegungsabläufen	126
6.2.1. Trainingsphase	129
6.2.2. Anwendungsphase	133
6.3. Experimentelle Evaluierung	136
6.3.1. Verschieben der Aktivitäten	137
6.3.2. Prädiktionsgenauigkeit	139
6.3.3. Erkennungsaufgabe	142
6.4. Fazit	144
7. Zusammenfassung und Ausblick	147
7.1. Szenariospezifische Anwendbarkeit	151
7.1.1. Prädiktionsaufgaben	151
7.1.2. Erkennungsaufgaben	153
7.2. Erweiterungsmöglichkeiten	154
A. Visuelle Bewegungssegmentierung	157
A.1. Markov Random Fields	159
A.2. Zusammensetzung der Energiefunktion	161
A.2.1. Berechnung des optischen Flusses	162
A.2.2. Integration von Farbinformationen	163
A.2.3. Integration von Tiefeninformationen	164

A.3. Verwerfen und Zusammenführen von Segmenten	165
A.4. Initialisierung	165
B. Verfahren zur Repräsentation von Zeit	167
B.1. Dimensionalität	167
B.2. Abtastung	170
B.2.1. Ereignis-basierte Abtastung	170
B.2.2. Informationsabhängige Abtastung	172
B.3. Funktionsapproximation	173
B.4. Experimentelle Evaluierung	175
C. Ähnlichkeit von Trajektorien	179
C.1. Experimentaldaten	179
C.2. \mathcal{L}_p -Normen	181
C.3. Korrelationskoeffizient	184
C.4. Sequence Search	185
C.4.1. Adaptive Piecewise Constant Approximation	186
C.4.2. Chebyshev-Approximation	189
C.5. Time Warping Algorithmen	190
C.6. Scan Matching Algorithmen	196
C.7. Zusammenfassung	199
D. Vergleich von Prädiktionsalgorithmen	203
D.1. Cluster Weighted Modeling	206
D.2. Autoregressive Models	207
D.2.1. Wiener Filter	208
D.2.2. Durbin-Levinson	209
D.2.3. Yule-Walker	209
D.3. Experimenteller Vergleich	210
D.3.1. Sinus und Lorenzattraktor	212
D.3.2. Realwelt Bewegungsdaten	213
D.3.3. Berechnungszeit	215
D.4. Fazit	215
E. Vergleich von Clusteralgorithmen	219
E.1. O-Cluster	221
E.2. Neuronale Ansätze	222
E.2.1. Neuronales Gas	222
E.2.2. Self Organizing Feature Map	223

E.2.3. Expanding Self Organizing Feature Maps	223
E.3. K-Means	223
E.4. SASH	224
E.5. Experimenteller Vergleich	225
E.6. Fazit	228

Abbildungsverzeichnis

1.1. Aufbau der Arbeit	10
2.1. Point-Light Display einer laufenden Person	16
2.2. Beteiligte Gehirnareale im menschlichen Gehirn	19
3.1. Bewegungstrajektorie eines Fußknöchels	33
3.2. Aufgezeichnete Trajektorien	35
3.3. Repräsentationsformen für Bewegungstrajektorien	37
3.4. Überblick über unterschiedliche Repräsentationsmöglichkeiten für Bewegungstrajektorien	39
3.5. Referenzsysteme	41
3.6. Trajektorienbezugssystem	42
3.7. Abtastmöglichkeiten	44
3.8. Eigentrajektorien	46
3.9. Interpretation der Transinformation	48
3.10. Transinformation der Repräsentationsformen	51
3.11. Paarweise Transinformation der Repräsentationen	53
3.12. Untersuchte Ähnlichkeitsmaße	58
4.1. Untersuchte Prädiktionsalgorithmen	64
4.2. Aufbau eines Echo State Networks	66
4.3. Funktionsweise eines ESN	67
4.4. Aufbau des modifizierten ESNs	70
4.5. Kurzschrittfehler KF_{log} von Varianten der ESNs auf Bewegungsdaten	77

4.6. Langschrittfehler LF_{log} von Varianten der ESNs auf Bewegungsdaten	78
4.7. Entwicklung des Langschrittfehlers	81
4.8. Langschrittfehler für unterschiedliche Vorverarbeitungen	83
4.9. Prädiktion des Rauschanteils	84
4.10. Trajektorienprädiktion mittels ESN	85
5.1. Beispiel für eine reguläre Einbettung	90
5.2. Beispiel einer Informationsfunktion $MI(T)$	91
5.3. Einfluss der Einbettungsverzögerung T auf benachbarte Punkte	93
5.4. Finden falscher Nachbarn durch Erhöhung von D	94
5.5. Grundprinzip des Local Modeling	94
5.6. Prädiktion mit dem Local Modeling	96
5.7. Kurzschrittfehler KF_{log} von Varianten des Local Model	98
5.8. Langschrittfehler LF_{log} von Varianten des Local Model .	99
5.9. Entwicklung des Langschrittfehlers	101
5.10. Langschrittfehler für unterschiedliche Vorverarbeitungen	103
5.11. Trajektorienprädiktion mittels Local Models	104
5.12. Erweiterung der Local Models	106
5.13. Prädiktion auf Basis des Clusteransatzes	108
5.14. Vergleich der Klassifikationsergebnisse	113
6.1. Prinzip der spatio-temporalen NMF	118
6.2. Basisdekomposition mittels PCA (a) und ICA (b)	120
6.3. Überführung von Bewegungstrajektorien in ihre Gridrepräsentation	128
6.4. Trainingsphase der Spatio-Temporalen NMF	130
6.5. Anwendungsphase der Spatio-Temporalen NMF	133
6.6. Verschiebung der Aktivitäten	135
6.7. Beispiele für Basisprimitive der Spatio-Temporalen NMF.	137
6.8. Konvergenzverhalten der Spatio-Temporalen NMF	138
6.9. Durchschnittlicher Ground-Truth-Fit	140
6.10. Prädiktionsgenauigkeit entlang einer Beispieltrajektorie	142
6.11. Vergleich der Prädiktionsergebnisse	143
A.1. Schematischer Ablauf der Bewegungssegmentierung	158
A.2. Markov Random Field zur Bildsegmentierung	159

B.1. Repräsentationsformen für Bewegungstrajektorien . . .	168
B.2. Einbeziehung der Zeit	168
B.3. Vergleich zur Unabhängigkeit der Raumdimensionen . .	169
B.4. Dynamic Instants auf einer Trajektorie	171
B.5. APCA Repräsentation einer Trajektorie.	172
C.1. Unterschiedliche Transformationen der Experimentaldaten	180
C.2. Referenzverlauf mit unterschiedlichen Modifikationen. .	182
C.3. Die zwei unterschiedlichen APCA-basierten Distanzmaße	187
C.4. Unterschiedliche Arten von Time Warping Verfahren . .	191
C.5. Pfadsuche in Distanzmatrix mittels DTW	193
C.6. Schematischer Ablauf des ICP Algorithmus.	197
D.1. Untersuchte Prädiktionsalgorithmen	204
D.2. Grundprinzip des Cluster Weighted Modeling	205
D.3. Sinustestdaten	211
D.4. Lorenz-Attraktor Testdaten	211
D.5. Fehler KF_{log} auf Sinusdaten und LF_{log} auf dem Lorenz- Attraktor	212
D.6. Fehler KF_{log} auf Bewegungsdaten	213
D.7. Fehler LF_{log} auf Bewegungsdaten	214
D.8. Trainings- und Prädiktionszeiten der Algorithmen . . .	216
E.1. Untersuchte Clusteralgorithmen	220
E.2. Unterteilung des Raumes auf Basis des O-Cluster- Verfahrens	221
E.3. Fehler zur Ground-Truth-Referenz	226
E.4. Rechenzeiten für das Ermitteln der Cluster	226
E.5. Rechenzeiten für einen Prädiktionsschritt	227

Mathematische Notation

Vektor- und Matrixnotation

$$\mathbf{v} = \begin{pmatrix} v^1 \\ \vdots \\ v^n \end{pmatrix}$$

Darstellung als Spaltenvektor

$$\mathbf{v}^T = (v_1, \dots, v_n)$$

Darstellung als Zeilenvektor

$$\mathbf{V} = \begin{pmatrix} v_1^1 & \cdots & v_m^1 \\ \vdots & \ddots & \vdots \\ v_1^n & \cdots & v_m^n \end{pmatrix} \\ = (v_i^j)_{i,j}$$

Darstellung als Matrix. Auf die Elemente v_j^i der Matrix wird über den Spaltenindex j und den Zeilenindex i zugegriffen.

Trajektoriennotation

$$\mathcal{T} = (\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{n-1})$$

Beschreibung einer Trajektorie mit den einzelnen Positionen

$\mathbf{s}_i^T = (x_i, y_i, z_i)$ bzw. $\mathbf{s}_i^T = (x_i, y_i)$ zum Zeitpunkt i . Die Reihenfolge des Auftretens entspricht der dargestellten Reihenfolge.

$$\mathcal{T} = \{\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{n-1}\}$$

Beschreibung einer Trajektorie mit den einzelnen Positionen $\mathbf{s}_i = (x_i, y_i, z_i)$ bzw. $\mathbf{s}_i = (x_i, y_i)$ zum Zeitpunkt i . Die Positionen werden als Menge aufgefasst. Somit spielt die Reihenfolge des Auftretens keine Rolle.

$$\mathbf{e}_t = (\mathbf{s}_t, \dots, \mathbf{s}_{t-(D-1)T})$$

Einbettungsvektor für den Zeitpunkt t mit der Einbettungsverzögerung T und Einbettungsdimension D

$$G = (g_i^j)_{i,j}$$

Gridrepräsentation einer Trajektorie (entspricht einem Binärbild)

$$g_i^j = \begin{cases} 1 & \text{Gridzelle Spalte } i \text{ Zeile } j \\ & \text{gehört zur Trajektorie} \\ 0 & \text{sonst} \end{cases}$$

Vergleichsmaße und spezielle Operatoren

$$KF = \frac{\sum_{p=1}^{N_{KF}} \frac{1}{p} NMSE_p}{\sum_{p=1}^{N_{KF}} \frac{1}{p}}$$

Kurzschrittfehler für die Evaluierung der Prädiktion mit $NMSE_i = (\mathbf{s}_i^{pred} - \mathbf{s}_i^{orig})^2$.

$$LF = \frac{\sum_{p=1}^N \frac{1}{\sqrt{p}} NMSE_p}{\sum_{p=1}^N \frac{1}{\sqrt{p}}}$$

Langschrittfehler für die Evaluierung der Prädiktion mit $NMSE_i = (\mathbf{s}_i^{pred} - \mathbf{s}_i^{orig})^2$

$$KF_{log} = \frac{1}{k} \sum_{i=1}^k \log_{10} KF_i$$

Mittelwert der logarithmierten Werte für die errechneten Kurzschrittfehler. Analog für den Langschrittfehler definiert.

$$D_{KF}^2 = \frac{1}{k} \sum_{i=1}^k (\log_{10} KF_i - KF_{log})^2$$

Standardabweichung der logarithmierten Werte für die errechneten Kurzschrittfehler. Analog für den Langschrittfehler.

$$\mathbf{C} = \mathbf{A} \odot \mathbf{B}$$

Elementweise Multiplikation
 $\forall i, j : C_j^i = A_j^i \cdot B_j^i$

Kapitel 1

Einleitung

Zukünftig werden mobile Service-Roboter in öffentlichen Gebäuden und Büroumgebungen eingesetzt, um Personen Informationen und andere Dienstleistungen zur Verfügung zu stellen. Dabei sollten die Roboter ein intelligentes Verhalten zeigen: solche Personen begrüßen, die möglicherweise mit ihnen interagieren wollen, Personen ignorieren, die kein Interaktionsinteresse haben und versuchen, unentschlossene Personen für eine Interaktion zu gewinnen.

Um intelligent reagieren zu können, muss der Roboter in der Lage sein, seine Umwelt zu erfassen und zu „verstehen“. Zur Erfassung der Umwelt verfügen soziale Assistenzroboter über eine Vielzahl an Sensoren. Darunter sind beispielsweise Laserscanner oder Sonarsensoren, aber auch Kameras, mit denen die Szene visuell ausgewertet werden kann. Moderne Algorithmen und schnelle Computersysteme machen es möglich, Objekte in der Szene robust zu erkennen und auch über eine Bildsequenz zu verfolgen.

Mit dem Schritt von der Bildanalyse zur Analyse von Bildsequenzen ist es möglich geworden, die Bewegungsabläufe einzelner Objekte aufzuzeichnen. Zusätzlich zu der rein statischen Analyse erkannter Objekte, enthalten diese Bewegungsabläufe (Trajektorien) wichtige Informationen. Diese Informationen können dazu beitragen, das Verständnis der Szene weiter zu erhöhen.

Beispielsweise ist es möglich, im Bereich der Überwachung von sicherheitskritischen Bereichen, ungewöhnlich handelnde Personen zu identifizieren und ihre Bewegungen vorherzubestimmen. Ebenso können im Bereich der häuslichen Betreuung durch mobile Serviceroboter Trajektorien benutzt werden, um anormales Verhalten zu erkennen und entsprechend zu reagieren. Für den Einsatz eines mobilen Assistenzroboters ist es wichtig, dass genügend Information über den Nutzer des Roboters bekannt ist. Beispielsweise kann so der Dialog sinnvoll an den Nutzer angepasst werden.

Damit mobile Roboter auch in belebter Umgebung operieren können, sind geeignete Navigationsstrategien notwendig, die die Bewegung der umgebenden Personen und Objekte mit einbeziehen. Um eine rein reaktive Navigation zu vermeiden, ist es ebenso von Interesse, die Bewegung der Personen und Objekte zu präzisieren. Somit kann eine geeignete Ausweichstrategie rechtzeitig gewählt werden.

Viele bekannte Verfahren nutzen dabei die Informationen der Trajektorien nur implizit durch Verwendung von neuronalen Netzen, Hidden Markov Modellen oder ähnlichem. Eine geeignete explizite Repräsentation der Trajektorien wird dabei oft vernachlässigt. An dieser Stelle soll diese Arbeit ansetzen. Ähnlich wie in der Bildverarbeitung scheint es sinnvoll, die beobachteten Daten – die Trajektorien – in geeigneter Weise aufzubereiten.

Dazu ist es notwendig die Gesamttrajektorie in einzelne Teiltrajektorien zu zerlegen. Die Dekomposition soll dabei so erfolgen, dass sich die einzelnen Teilstücke einfach repräsentieren lassen. Es ist dazu notwendig, die Punkte auf der Trajektorie zu finden, bei denen sich die Beschreibungsform der Trajektorie ändert.

An dieser Stelle ist die Frage gerechtfertigt, ob es sinnvoll ist, sich mit Trajektorien als Informationsquelle zu beschäftigen. Sind also die Informationen, die in den Bewegungen von Objekten bzw. von Personen enthalten sind, ausreichend, um die angesprochenen Aufgaben erfüllen zu können? Um auf diese Frage eine erste – wenn auch unvollständige – Antwort geben zu können, wird in Abschnitt 2.1 ein kleiner Überblick über die Repräsentation von Bewegungen beim biologischen Vorbild beleuchtet werden. Diese Betrachtung wird einen ersten Hinweis auf die Sinnfälligkeit der weiteren Arbeit geben.

1.1. Anspruch der Arbeit

Die vorliegende Dissertation ist im Fachgebiet Neuroinformatik und Kognitive Robotik an der Technischen Universität Ilmenau in Zusammenarbeit mit dem Honda Research Institute Europe GmbH im Rahmen des Forschungsprojektes *VISION* entstanden. Das Akronym *VISION* steht dabei für Dynamic VISual recognitION in multiobject scenes. Das Ziel des Projektes stellt die Erfassung, Verarbeitung und Speicherung von Objekten in dynamischen Umgebungen dar. Der Begriff *dynamische Umgebung* soll dabei ausdrücken, dass sich sowohl die zu erkennenden Objekte, als auch das Erfassungssystem in Bewegung befinden können. Dabei sollte sich nicht auf spezifische Objekte, wie Personen, Roboter, o. ä. beschränkt werden. Ein zentraler Punkt bei der Beschreibung sich bewegender Objekte stellt dabei die Untersuchung von Bewegungstrajektorien dar, welche den Kern dieser Arbeit bilden.

Als einführende Motivation soll in Abschnitt 1.2 ein Anwendungsszenario skizziert werden. Diese Arbeit erhebt entsprechend den Anforderungen des Projektes allerdings nicht den Anspruch, eine maßgeschneiderte Lösung für eine spezielle Anwendung zu erarbeiten. Ziel ist es in erster Linie, eine Palette an Werkzeugen und Herangehensweisen zu beleuchten, die in einer Vielzahl an Szenarien Anwendung finden können. Dabei sollen bekannte Verfahren aus unterschiedlichen, verwandten Forschungsfeldern auf ihre Eignung im Einsatz mit Bewegungstrajektorien untersucht und gegebenenfalls an die speziellen Bedürfnisse in diesem Einsatzfeld angepasst werden. Es soll dem Leser nach dem Studium der Arbeit möglich sein zu entscheiden, welches Verfahren für sein spezielles Anwendungsgebiet von Interesse ist.

Durch den Verzicht auf eine für ein spezielles Problem zugeschnittene Lösung werden in dieser Arbeit in erster Linie Verfahren untersucht, die nicht im aktuellen Fokus der anwendungsorientierten Forschung zur Bewegungsanalyse stehen. Es wird daher auch darauf verzichtet, etablierte Verfahren, wie beispielsweise Hidden Markov Modelle (HMM), aufzugreifen und entsprechend zu verbessern. Da eine solche Verbesserung in den meisten Fällen an dem jeweils gewählten Problem untersucht und spezifisch darauf angepasst wird, scheint dies im Zusammenhang dieser Arbeit auch nicht sinnvoll.

Bei der Beschreibung und Untersuchung von Bewegungsabläufen lassen sich zwei Herangehensweisen unterscheiden. Ähnlich wie im Bereich der Computergraphik, existieren generative und analytische Ansätze. Generative Ansätze dienen dabei in erster Linie zur Steuerung von anthropomorphen Robotersystemen. Ziel ist es dabei beispielsweise, Greif- oder auch Laufbewegungen erzeugen zu können. Insbesondere wenn ein menschlicher Lehrer imitiert werden soll, werden diese Algorithmen auch zur Erkennung von Bewegungen eingesetzt.

Im Fokus der analytischen Sichtweise stehen in erster Linie Klassifikations- und Prädiktionsprobleme. Es soll also Wissen aus der beobachteten Bewegung extrahiert werden. Dieses Wissen kann dann in aufbereiteter Form z. B. dem Klassifikator zur Verfügung gestellt werden, um den Trainingsaufwand zu minimieren oder das Ergebnis zu verbessern. Die analytische Beschreibung von Bewegungen wird in der Literatur weniger verfolgt. Sie stellt aber dennoch eine wichtige Grundlage für intelligentes Verhalten dar und wird deshalb im Rahmen dieser Arbeit genauer untersucht.

Das anvisierte Aufgabenspektrum umfasst zum Einen eine systematische Gegenüberstellung bestehender Ansätze. Insbesondere sollen unterschiedliche Beschreibungsmöglichkeiten von Trajektorien beleuchtet und die Möglichkeit einer geeigneten Auswahl vorgestellt werden. Zum Anderen sollen Verfahren vorgestellt werden, die in der Lage sind, allein aufgrund der Informationen aus der Bewegung diese zu klassifizieren. Dabei werden diese Verfahren erstmals auf realen Bewegungsdaten getestet. Darüber hinaus sollen die beobachteten Bewegungstrajektorien prädiziert werden, um auf den weiteren Verlauf der Handlung entsprechend eingehen zu können.

1.2. Anwendungsszenario

Trotz der Forderung nach Allgemeingültigkeit, soll in diesem Abschnitt ein Szenario vorgestellt werden, das unterschiedliche Anwendungsbeispiele aufzeigt. Diese sollen bei der Untersuchung der einzelnen Verfahren und Methoden dazu dienen, Anforderungen zu formulieren und um die Eignung der Algorithmen evaluieren zu können.

Das dieser Arbeit übergeordnete Ziel stellt die Verbesserung der Mensch-Roboter-Interaktion¹ dar. Das im Folgenden beschriebene Beispiel ist daher aus unterschiedlichen Anwendungsaspekten aufgebaut, bei denen die Interaktion zwischen dem Menschen und dem unterstützenden Roboter eine wesentliche Rolle spielt. Für solche Anwendungsfelder findet sich bereits eine breite Palette an Beispielen, darunter der Einsatz als Shoppingassistent [Gross und Böhme, 2000; Gross et al., 2008], als Haushaltshilfe [Stein et al., 2009] und als Museumsführer [Graf et al., 2000], um nur einige wenige zu nennen.

Als Beispiel für das Szenario soll ein Assistenzroboter dienen, der auch in der Lage ist manipulatorische Tätigkeiten durchzuführen. Dieser Roboter könnte als Empfangspersonal in einem Theater oder Museum eingesetzt werden. Dort soll er die Besucher begrüßen und, falls sie es wünschen, sich um ihre Garderobe kümmern

Ein wichtiger Gedanke dabei ist, dass der Roboter lernen muss, sich sozial zu verhalten, um von den Personen, mit denen er interagiert, akzeptiert zu werden. Dazu muss der Roboter in der Lage sein, subtile Hinweise, wie Körpersprache, interpretieren zu können. Ebenso spielt die Fähigkeit sich so zu bewegen, dass Interaktionspartner nicht verunsichert werden, eine wichtige Rolle.

Als erstes muss der Assistenzroboter feststellen, ob ein bestimmter Besucher die beschriebene Interaktion wünscht. Es wäre etwas befremdlich, wenn der Roboter jede detektierte Person ansprechen würde. Dies würde zum Einen unnötig Ressourcen binden, wenn der Roboter ständig mit nicht-interaktionswilligen Personen beschäftigt wäre. Zum Anderen würden sich Besucher, die in Eile sind, weil beispielsweise die Vorstellung schon angefangen hat, durch ein solches System gestört fühlen, wodurch die Akzeptanz von Servicerobotern stark leiden würde. Darüberhinaus soll anders als bei stationären Systemen, wie beispielsweise einem Fahrkartenautomaten, die mobile Einheit in der Lage sein, sich auf unschlüssige oder suchende Personen zuzubewegen.

Daher sollten intelligente Roboter in der Lage sein, ein natürliches menschenähnliches Verhalten zu zeigen. Zum Beispiel könnten sie dem potenziellen Interaktionspartner auf halbem Weg entgegenkommen. Es wä-

¹Human-Robot-Interaction (HRI)

re auch denkbar, dass der Roboter auf sich aufmerksam macht, um unentschlossene Menschen auf seine Fähigkeiten hinzuweisen.

Um all dies leisten zu können, muss der Roboter eigenständig in der Lage sein, zu entscheiden, wer Interesse an einer Interaktion hat und wer nicht. Es muss also eine Klassifikationsaufgabe gelöst werden. Der Mensch, als Vorbild, inferiert eine solche Intention anhand unterschiedlicher subtiler Hinweise, wie Mimik, Gestik und Körpersprache. Durch eine Limitierung der Kameraauflösung sind insbesondere Mimik und Gestik nicht auf weite Entfernung auszuwerten. Die Bewegungstrajektorie, ein Teil der Körpersprache, ist allerdings bekannt, sobald die Person von dem Roboter detektiert wird.

Abhängig davon, ob mit einer Person interagiert werden soll oder ihr aus dem Weg gegangen wird, muss eine entsprechende Strategie zur Navigation gewählt werden. Auf dem Weg zu seinem Interaktionspartner ist der Assistenzroboter mit einer Vielzahl von Hindernissen konfrontiert. Diese gilt es auf dem Weg zum anvisierten Ziel zu umfahren. Nicht alle dieser Hindernisse behalten statisch ihre Position bei. Gerade in der genannten Einsatzumgebung befinden sich auch dynamische Hindernisse. In erster Linie handelt es sich dabei um Menschen, aber möglicherweise auch andere bewegliche Objekte. Auch diesen Hindernissen gilt es auszuweichen. Allerdings müssen hierbei bestimmte Nebenbedingungen beachtet werden. Damit der Roboter auch als Assistent akzeptiert wird, sollte der Ausweichvorgang möglichst nicht abrupt erfolgen.

Insbesondere, wenn es sich bei dem dynamischen Hindernis um einen Menschen handelt, muss der Mensch in der Lage sein, das Ausweichen des Roboters richtig zu interpretieren. Ein Roboter, der mit maximaler Geschwindigkeit auf eine Person zufährt, um kurz vor der Person auszuweichen ist technisch sicher realisierbar, würde allerdings wesentlich schwerer akzeptiert werden. Sinnvoll ist in einer solchen Situation ein weiches Abbremsen und rechtzeitiges zur Seite fahren. Dazu ist ein frühzeitiges Erkennen und Prädizieren der Bewegung des dynamischen Hindernisses notwendig. Topp bezeichnet ein solches Vorgehen als *Socially Acceptable Navigation* [Topp und Christensen, 2005].

Ist der Roboter bei seinem Interaktionspartner angekommen soll er einen Dialog beginnen. Aufgrund der manipulatorischen Möglichkeiten des Roboters kann dieser mit einer Begrüßung durch Handschlag in-

itiiert werden. Um die Hand des Besuchers korrekt greifen zu können, muss deren Trajektorie vorhergesagt werden können und die Bewegung der Roboterhand entsprechend angepasst werden.

Während des Dialoges soll der Besucher im Fall eines Theaters über die aktuelle Vorstellung oder über besondere Ausstellungsstücke des Museums informiert werden. In Abschnitt 2.1.2 wird diskutiert, dass Menschen in der Lage sind, auch anhand der Körpersprache auf Emotionen zu schließen. Es ist also beispielsweise möglich zu entscheiden, ob ein Nutzer durch den geführten Dialog immer genervter reagiert. Dem Roboter ist es mit einem solchen Wissen möglich, den Dialog entsprechend anzupassen, um Reaktionen, wie z. B. bei aktuellen Fahrkartenautomaten zu vermeiden.

Zum Schluss soll je nach Bedarf noch die Garderobe entgegengenommen werden. Dabei kann auch anhand der Körpersprache entschieden werden, ob der Besucher seine Kleidung abgeben oder behalten möchte. Wie auch bei der Begrüßung mit Handschlag ist es essentiell, dass die Maschine in der Lage ist die Bewegungen der Personen, genauer gesagt die der Arme und Hände der Personen, zu erkennen und entsprechend vorherzusagen. Dies ist besonders von Interesse, da es hierbei nicht mehr um eine Akzeptanzfrage, sondern um eine mögliche Gefährdung des Interaktionspartners geht.

1.2.1. Fazit

Dieses etwas visionäre Szenario setzt sich aus einer Vielzahl unterschiedlicher Anwendungsaspekte zusammen. Seine Komplexität erlaubt es daher leider nicht eine vollständige Lösung im Rahmen einer einzigen Dissertation anzubieten. Daher wird sich diese Arbeit auf die Analyse der Bewegungstrajektorien beschränken. Dabei wird darauf geachtet, dass die vorgestellten Verfahren auch auf andere Szenarien übertragbar sind.

Um dieses Szenario umsetzen zu können, sind eine ganze Reihe Aspekte, die Analyse von Bewegungen betreffend, zu betrachten. Beim *Aspekt der Navigation* ist es für die Wahl einer geeigneten Navigationsstrategie wichtig, dass die Bewegung des dynamischen Hindernisses geeignet repräsentiert wird, um den weiteren Verlauf möglichst genau vorhersagen zu können. Im Rahmen dieser Arbeit soll die eigentliche Navigations-

strategie allerdings außer Acht gelassen werden und einzig die Vorhersage der Bewegung eine Rolle spielen.

Ein weiterer Aspekt stellt die *körperliche Interaktion* im Sinne z. B. des Handschlages dar. Dabei ist ähnlich wie beim Navigationsaspekt wichtig die Bewegung der Hand vorhersagen zu können. Allerdings reichen anders als bei der Navigation dazu nicht mehr die Daten im zweidimensionalen Raum aus. Vielmehr müssen dreidimensionale Daten verwendet werden. Im Rahmen der Arbeit werden die Verfahren zur Prädiktion von Bewegungen sowohl mit zwei- als auch mit dreidimensionalen Daten evaluiert.

Der letzte Aspekt ist die *Erkennung der Bewegungen*. In dieser Arbeit wird daher untersucht, wie und ob die Bewegungstrajektorien zur Klassifikation zu verwenden sind. Wie beschrieben sieht der Roboter sich unter Umständen mit unterschiedlichen Situationen und damit neuen Klassifikationsaufgaben konfrontiert. Darüber hinaus kann es sein, dass sich die Umwelt des Roboters ändert und das System sich somit adaptieren muss. Dazu muss möglicherweise nicht nur der Klassifikator nachtrainiert werden. Für die neue Situation kann sich auch eine andere Trajektorienrepräsentation als sinnvoll erweisen. Damit sollte die Auswahl der Repräsentationsform nicht vom Systementwickler getroffen werden, sondern durch das System selbst möglich sein.

1.3. Gliederung der Arbeit

Der prinzipielle Aufbau der Arbeit ist in Abbildung 1.1 dargestellt. Höher gelegene „Etagen“ des stilisierten Hauses, bauen dabei auf den darunterliegenden, im Sinne einer Anwendung der vorgestellten Verfahren, auf. Das „Dach“, als finaler Abschluss des Hauses, steht für das in Abschnitt 1.2 vorgestellte Anwendungsbeispiel mit den beiden untersuchten Methoden zur Prädiktion und Klassifikation. Die Abbildung wird in jedem der dargestellten Kapitel wiederholt werden. Dabei wird der jeweilige behandelte Aspekt hervorgehoben werden.

Nachdem in Kapitel 1 die Einordnung und Abgrenzung der Arbeit diskutiert wird, soll Kapitel 2 klären, in wie weit das Fokussieren auf Bewegungstrajektorien sinnvoll ist. Dazu werden sowohl Arbeiten aus dem

Bereich Biologie bzw. Psychologie, als auch anwendungsorientierte, ingenieurtechnische Projekte vorgestellt.

Das Kapitel 3 gliedert sich in drei Abschnitte, die Datenakquisition, die Repräsentation von Bewegungen und das Bestimmen der Ähnlichkeit von Trajektorien. Dabei wird damit begonnen, die in dieser Arbeit verwendeten realen Bewegungsdaten von Personen vorzustellen. In Anhang A wird eine Möglichkeit präsentiert, Trajektorien aufzuzeichnen, d. h. mit denen Objekte erkannt und verfolgt werden können. Dies geschieht mit Hilfe eines im Rahmen des Projektes entwickelten Verfahrens zur visuellen Bewegungssegmentierung beliebiger Objekte. Dieses auf Markov Random Fields basierende Verfahren stammt aus den Anfängen der Projektlaufzeit. Nachdem sich der Fokus des Projektes auf die Auswertung von Trajektorien verschoben hatte, wurde die Entwicklung dieses Verfahrens nicht weiter vorangetrieben. Es ist nur der Vollständigkeit halber im Anhang dieser Arbeit aufgeführt. Prinzipiell sind alle Verfahren und Untersuchungen in dieser Arbeit so ausgelegt worden, dass sie nicht von der Art der Datenaufnahme abhängig sind.

Als nächstes stellt Abschnitt 3.2 unterschiedliche Möglichkeiten der Repräsentation von Bewegungsabläufen vor. Da es je nach Aufgabenstellung sinnvoll ist, die eine oder andere Repräsentationsform zu wählen, wird außerdem ein Ansatz präsentiert, mit dem diese Auswahl automatisiert stattfinden kann. Das Vorgehen wird dabei an einem konkreten Beispiel erläutert, bei dem mit Hilfe eines realen Assistenzroboters Bewegungstrajektorien aufgezeichnet wurden. Zu Gunsten einer leichten Nachvollziehbarkeit des Beispiels erfolgt eine Beschränkung auf wenige Repräsentationsformen. Alle übrigen untersuchten Repräsentationsformen sind in den Anhang B ausgelagert. Im weiteren Verlauf der Arbeit wird zur Vereinfachung, wenn nicht anders erwähnt, mit einer Repräsentation im kartesischen Raum, wie sie direkt von üblichen Trackingsystemen stammt, gearbeitet. Dies geschieht, da diese Form der Repräsentation bei den umfangreichen experimentellen Untersuchungen gute Ergebnisse erzielen konnte. Abschnitt 3.2 ist daher für das weitere Verständnis der Arbeit nicht unbedingt notwendig. Ein versierter Leser kann das Kapitel ohne Probleme überspringen.

Für unterschiedliche Anwendungen, aber auch für viele Verfahren, spielt die Ähnlichkeit von Trajektorien eine große Rolle. Daher werden in Abschnitt 3.3 eine Reihe von Ähnlichkeitsmaßen kurz vorgestellt. Dabei

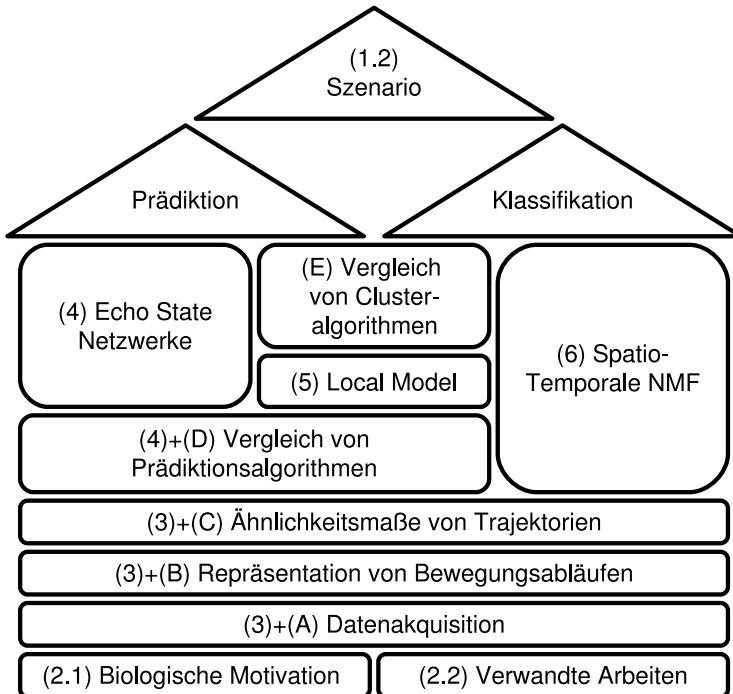


Abbildung 1.1.: Aufbau der Arbeit:

Die Abbildung veranschaulicht, wie die einzelnen Kapitel aufeinander aufbauen. Dabei kennzeichnen die Zahlen bzw. Buchstaben den jeweiligen Abschnitt, das Kapitel bzw. den entsprechenden Anhang.

spielen nicht nur übliche Metriken eine Rolle, die einen skalaren Wert für die Ähnlichkeit liefern. Vielmehr werden auch Verfahren betrachtet, die eine Aussage über die Art der Unähnlichkeit erlauben. Die ausführliche Untersuchung kann in Anhang C nachvollzogen werden.

In den Kapiteln 4 bis 6 werden Algorithmen vorgestellt, mit denen in erster Linie Prädiktions-, aber auch Klassifikationsprobleme gelöst werden können. Um die am besten geeigneten Verfahren aus dem Bereich der Zeitreihenanalyse für die Anwendung auf Bewegungsabläufen herauszufinden, wurden Verfahren aus unterschiedlichen methodischen Bereichen verglichen und erstmals auf realen Bewegungstrajektorien eingesetzt. Details dieser vergleichenden Untersuchungen wurden im Interesse des Leseflusses in Anhang D ausgelagert.

Die Echo State Netzwerke werden als erstes untersuchtes Verfahren in Kapitel 4 diskutiert. Die Grundidee ist dabei eine implizite Zerlegung des zeitlichen Signales mit Hilfe von zufällig ausgebildeten Teilnetzen. Im Rahmen dieser Arbeit konnte die Anzahl der Neuronen durch einen parallelen Trainingsprozess im Vergleich zu den Angaben in der Literatur stark reduziert werden.

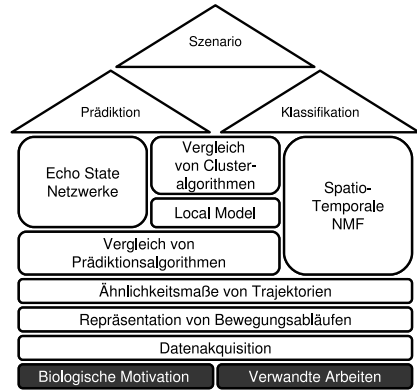
Das Kapitel 5 beschäftigt sich mit den Local Models, die auf Basis ähnlicher bereits beobachteter Bewegungsabläufe eine Voraussage zum weiteren Verlauf treffen können. Diese Arbeit entwickelt eine Lösung, wie es durch den Einsatz von Clusteralgorithmen möglich ist, die Fähigkeit zur Prädiktion um eine Wissensbasis zu erweitern. Damit ist die Prädiktion nicht mehr auf eine lange Beobachtungsphase angewiesen, während der keine Prädiktionen möglich sind. Details zur Auswahl eines für diese Zwecke geeigneten Clusterverfahrens sind in Anhang E ausgeführt. Darüberhinaus werden Möglichkeiten erarbeitet und untersucht, eine Klassifikationsaussage ableiten zu können.

Kapitel 6 präsentiert eine Weiterentwicklung der nicht-negativen Matrixzerlegung (NMF) zur Anwendung auf Zeitreihen. Die vorgestellte spatio-temporale NMF generiert ein Basissystem ähnlich einem Alphabet anhand entsprechender Trainingsdaten. Mit diesen Trainingsdaten können dann beobachtete Trajektorien zerlegt werden. Durch Ausnutzung bestimmter Eigenschaften der Zerlegung kann diese auch verwendet werden, um den weiteren Bewegungsverlauf zu präzisieren.

Die Arbeit wird in Kapitel 7 zusammengefasst. Ebenso erfolgt ein Ausblick auf eine mögliche Weiterführung des Themas.

Kapitel 2

Motivation



In diesem Kapitel soll ein Blick in unterschiedliche Forschungsfelder geworfen werden, um zu klären, inwieweit die Analyse von Bewegungsinformationen einen zusätzlichen Informationsgewinn ermöglichen kann. Zuerst werden einige Untersuchungen aus dem Bereich der Biologie bzw. Psychologie diskutiert. Danach sollen Arbeiten vorgestellt werden, die sich bereits mit der Nutzung von Bewegungsinformationen beschäftigen.

2.1. Biologische Motivation

An dieser Stelle wird beleuchtet, wo und wie die Analyse von Bewegungsabläufen ihre Entsprechung im biologischen Vorbild findet. Ziel ist es dabei, sowohl eine Motivation abzuleiten, als auch beobachtete Effekte mit den Mitteln dieser Arbeit nachzuvollziehen. Dies soll helfen, die noch immer nicht vollständig verstandenen Vorgänge in den Gehirnen von Tieren und Menschen begreifbarer zu machen.

Es wird weiterhin geklärt, ob bzw. bis zu welchem Grade das ausschließliche Auswerten von Bewegungsinformationen ausreichend ist, um hinreichende kognitive Leistungen erbringen zu können. Sicher verwendet eine komplexe Struktur, wie das Gehirn, ein Reihe von Informationen, um geeignete Hypothesen über die Umwelt und somit eine Vorstellung

dieser entwickeln zu können. Es wird sich allerdings zeigen, dass tatsächlich Verarbeitungsmechanismen existieren, die lediglich mit Bewegungsinformationen auskommen und kein weiteres Modellwissen voraussetzen.

Der Fokus der Aufarbeitung der Literatur liegt dabei nicht in erster Linie auf Arbeiten, die versuchen, die Struktur des Gehirns zu analysieren und die Funktionen einzelner Hirnareale zu erschließen. Vielmehr stehen an dieser Stelle Forschungsergebnisse aus dem Bereich Psychologie im Vordergrund, bei denen empirische Experimente bestimmte Verhaltensweisen aufzeigen.

2.1.1. Simulander

Ein sehr intensiv und tiefgründig untersuchtes Feld im Bereich der Nutzung von visuell erfassten Bewegungsinformationen, ist das Jagdverhalten des Schleuderzungensalamanders. Der Schleuderzungensalamander ist in der Lage, mit seiner Zunge Beute in der Größe von 0,05 cm auf eine Entfernung von 5 cm zu fangen. Dies entspricht einer Genauigkeit von 0,57°. Die Beute befindet sich dabei in Bewegung, so dass es notwendig ist, dass die Echse diese präzisieren können muss.

Mitte der neunziger Jahre wurde am Institut für theoretische Neurophysik der Universität Bremen ein System mit dem Namen *Simulander* entwickelt [Eurich et al., 1995; Eurich et al., 1997]. Dieses System besteht aus einer neuronalen Architektur, die den tatsächlich im Hirn des Salamanders gefundenen Strukturen und Funktionsprinzipien nachempfunden wurde. Das *Simulander-System* besteht dabei aus zwei Teilsystemen, die die notwendigen motorischen Reize auslösen, um eine Bewegung zum Fangen der Beute durchzuführen.

Der erste Teil (*Simulander I*) dient der Anpassung der Blickrichtung mit dem Ziel den Blickwinkel zwischen Beute und der virtuellen Amphibie zu minimieren. Die Größe der rezeptiven Felder im visuellen System sowie die Struktur des verwendeten neuronalen Netzes, wurden direkt aus Untersuchungen am biologischen Vorbild abgeleitet. Die Gewichte des neuronalen Netzes wurden dabei mit Hilfe einer Evolutionsstrategie, basierend auf einer zufälligen Suche im Parameterraum, adaptiert [Bäck et al., 1991; Rechenberg, 1973]. Die Verwendung einer Evolutionsstra-

tegie hat hierbei nichts mit einer biologischen Motivation zu tun. Es handelt sich dabei nur um die Wahl eines geeigneten Optimierungsalgorithmus.

Nachdem *Simulander I* in der Lage ist den Kopf in Richtung des Beutetieres zu drehen, fehlt noch das Ansteuern der Zunge zum Fangen der Beute. Diese Funktion übernimmt der zweite Teil der Architektur *Simulander II*. Auch in diesem Fall leitet sich der Aufbau des Netzes direkt aus Untersuchungen am biologischen Vorbild ab. Dabei wurde beispielsweise die Abschätzung der Entfernung mit Hilfe so genannter binokularer Neuronen [Eurich et al., 1997] vorgenommen.

Um die genaue Position der Beute und somit die Zielposition der Zunge ermitteln zu können, muss der Schleudertzungensalamander in der Lage sein, seine Umgebung dreidimensional zu erfassen. Dies geschieht durch die Nutzung rein visueller Informationen. Zu erwähnen ist dabei, dass das Tiefsehen hauptsächlich auf Basis von Disparitätsinformationen funktioniert. Die Akkommodation der Linse und die Konvergenz der Augen spielen dazu im Falle des Salamander nur im geringen Maße eine Rolle.

Das aus den Teilen *Simulander I und II* bestehende Gesamtsystem wurde ausgiebig getestet und mit dem biologischen Vorbild verglichen. Die Treffgenauigkeit des entwickelten künstlichen Systems beträgt dabei bei einer Entfernung von 5 cm ca. 2 mm.

2.1.2. Wahrnehmung von Bewegungen

Mit dem Blick auf *Simulander* ist zu erkennen, dass komplexes Verhalten durchaus ausschließlich auf visuell erfassten Bewegungsinformationen beruhen kann. Sicherlich besitzt der Salamander im Vergleich zum menschlichen Gehirn nur sehr einfache zerebrale Strukturen. Wie die im Folgenden vorgestellten Untersuchungen zeigen, benutzt auch der Mensch wahrgenommene Bewegungen für bestimmte Aufgaben.

In den siebziger Jahren veröffentlichte Gunnar Johansson seine Untersuchungen zur menschlichen Wahrnehmung [Johansson, 1973; Johansson, 1974]. Dabei präsentierte er den Probanden Aufnahmen sich bewegnender Personen. Diese Personen sind dabei allerdings nicht komplett oder

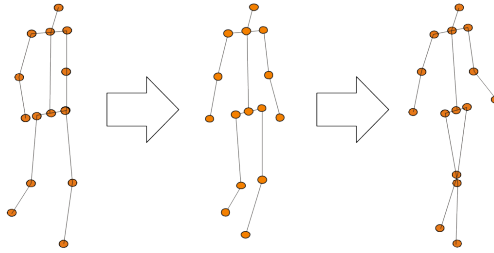


Abbildung 2.1.: Point-Light Display einer laufenden Person: *Zu sehen sind drei Einzelansichten, die aus der Sequenz einer gehenden Person stammen. Eigentlich bestehen die von [Pollick, 2004] aufgenommenen Point-Light Displays nur aus Punkten. Die Verbindungslinien wurden lediglich aus Visualisierungsgründen hinzugefügt.*

als Silhouette zu erkennen. Vielmehr sieht man nur eine Anzahl von Punkten, die an speziellen Stellen des Körpers platziert wurden (siehe dazu Abbildung 2.1).

Man bezeichnet diese Art von Aufnahmen als Point-Light Displays. Sie sind beispielsweise auch aus dem Bereich der computeranimierten Filme bekannt, wenn man versucht, reale Bewegungen auf die virtuellen Charaktere zu übertragen.

Johansson stellt dabei fest, dass die meisten Probanden in der Lage sind, diese Anordnung von Punkten als Person zu erkennen. Dabei spielt es allerdings schon eine Rolle, ob man nur ein Einzelbild oder eine längere Sequenz zu sehen bekommt. Wird nur ein Einzelbild präsentiert, fällt es wesentlich schwerer bzw. ist es unmöglich, den richtigen Zusammenhang zwischen den Punkten zu erkennen.

An dieser Stelle könnte man vermuten, dass das menschliche Hirn den räumlichen Zusammenhang der Punkte nutzt, um eventuell ein dreidimensionales Modell an die Lage der Punkte anzupassen. Dem widersprechen allerdings die Forschungen von Cutting.

[Cutting, 1978] entwickelte in Anlehnung an diese Untersuchung einen Algorithmus, mit dem Point-Light Displays künstlich generiert werden können. Dadurch ist es möglich, die Position der Punkte beliebig zu

verrauschen. Somit kann sichergestellt werden, dass ein eventuell existierendes Modell nicht mehr mit den Punkten zur Deckung gebracht werden kann. Dabei stellte man fest, dass die Lage der Punkte zueinander einen geringeren Einfluss hat. Wichtiger scheint hierbei eine gemeinsame Bewegung der Punkte zu sein.

Um das Jahr 2000 wurden diese Arbeiten von Frank E. Pollick wieder aufgegriffen und weiterentwickelt [Pollick, 2004]. Für die Forschungen von Pollick steht im Vordergrund, welche Informationen aus solchen spärlichen Daten noch extrahiert werden können. Dazu wurden unterschiedliche Bewegungsabläufe als Point-Light Display bestehend aus 15 Punkten aufgezeichnet. Es ist sicher leicht nachvollziehbar, dass sich aus solchen Daten recht einfach das Geschlecht der aufgezeichneten Person ermitteln lässt, da Männer und Frauen sich signifikant im Gang unterscheiden. Darüber hinaus kann sogar auf den emotionalen Zustand der Person geschlossen werden.

An dieser Stelle wird nun klar, dass Bewegungstrajektorien eine wichtige Informationsquelle darstellen. Ebenso scheint es, dass das menschliche Hirn in der Lage ist, mit solchen Informationen, ohne zusätzliche Unterstützung, umzugehen. Doch wie kann man nun solche Informationen nutzen? Ebenso ist bis hierhin offen geblieben, welche Areale im menschlichen Hirn dafür verantwortlich sind, diese Art von Daten auszuwerten.

2.1.3. Experimente mit Neugeborenen

Ein wichtiger Hinweis, wenn es um Bewegungsinformationen geht, kann in den frühen Entwicklungsstadien, sowohl von Menschen, als auch von anderen Tieren, gefunden werden. Es ist eine bekannte Tatsache, dass Kleinkinder zu Beginn des Lebens das Verhalten ihrer Eltern übernehmen. Aber welche ist die Information, die es den Neugeborenen ermöglicht, sich auf ihre Eltern zu konzentrieren, um die durchgeführten Handlungen nachahmen zu können? [Méary et al., 2007] beschreibt in seinen Arbeiten ein Experiment, welches zeigt, dass neugeborene Kinder ihre Aufmerksamkeit automatisch auf biologisch generierte Bewegungen, also Bewegungen von Lebewesen, lenken. Es muss also möglich sein, Bewegungen, die von einem biologischen System ausgeführt werden, zu erkennen und sie von anderen Bewegungsquellen zu unterscheiden. Da-

zu behauptet Méary, dass biologische Bewegungen durch dynamische Regelmäßigkeiten charakterisiert sind.

Ähnliche Experimente wurden auch von [Vallortigara et al., 2005] durchgeführt, dieses Mal allerdings nicht mit menschlichen Kindern, sondern mit frisch geschlüpften Küken. Als man den jungen Hühnern auf Bildschirmen die bereits diskutierten Point-Light Displays präsentierte, zeigten sie eine klare Präferenz für biologisch generierte Bewegungen. Ein interessanter Fakt ist dabei, dass diese Küken ihre Aufmerksamkeit nicht nur auf Darstellungen von Hennen, sondern auch auf die von potentiellen Räubern, wie z. B. Katzen, richten.

Für beide Experimente wurden Point-Light Display Sequenzen verwendet. Die erzielten Ergebnisse konnten allerdings nicht bestätigt werden, wenn den jeweiligen Probanden nur unbewegte Bilder gezeigt wurden.

2.1.4. Strukturen im menschlichen Gehirn

Bis hierhin wurden Veröffentlichungen diskutiert, welche die psychologischen Aspekte von Bewegungen untersuchen. Es ist noch offen, welche Areale im menschlichen Gehirn für die Verarbeitung solcher Daten verantwortlich sind. Der Einleitungsabschnitt des Artikels von [Lange und Lappe, 2006] gibt einen kurzen Überblick über den aktuellen Wissensstand, wie das menschliche Gehirn mit Bewegungsinformationen umgeht.

Der Sulcus Temporalis Superior (STS) wird oft mit der Wahrnehmung von biologischen Bewegungen in Verbindung gebracht [Bonda et al., 1996; Oram und Perrett, 1996; Puce et al., 1998; Grossman et al., 2000; Vaina et al., 2001; Beauchamp et al., 2002; Santi et al., 2003; Thompson et al., 2005]. Da er sowohl von Form- als auch aus Bewegungsarealen Erregungen empfängt, befindet er sich an einer idealen Position, um Resultate der Form- und Bewegungsverarbeitung integrieren zu können [Oram und Perrett, 1996; Vaina et al., 2001; Beauchamp et al., 2002]. Dagegen ist die Rolle anderer Hirnareale weniger eindeutig geklärt. Einige Studien haben eine selektive Aktivierung des Medialen Temporalen Areals (MT) [Vaina et al., 2001; Ptito et al., 2003] und des Occipitalen kinetische Feldes (KO) [Vaina et al., 2001; Santi et al., 2003] bei der Beobachtung biologischer Bewegungen feststellen können. Von beiden

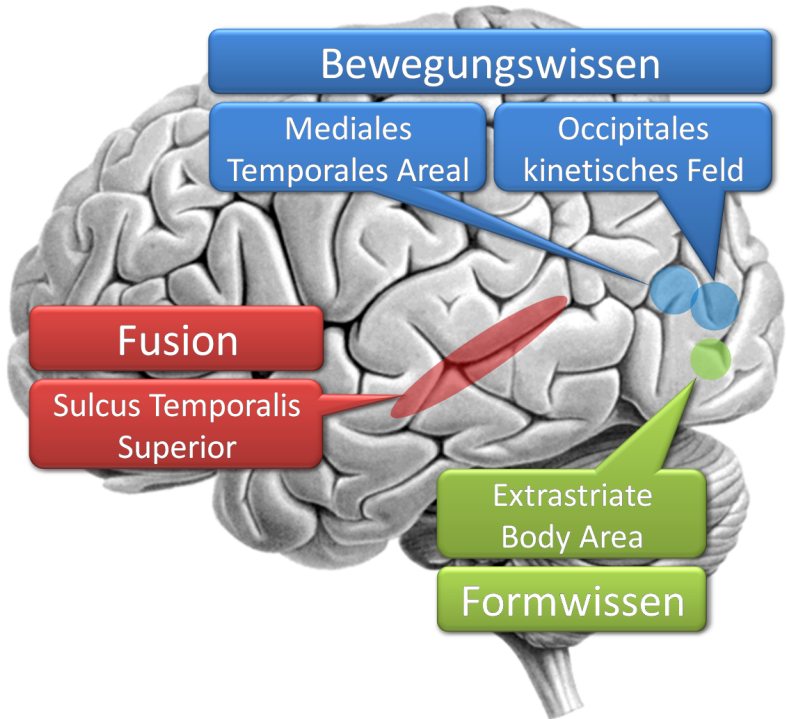


Abbildung 2.2.: Beteiligte Gehirnareale im menschlichen Gehirn: *Hypothesen, die durch die Wahrnehmung von Bewegung und statischer Informationen entstehen, werden unabhängig voneinander in unterschiedlichen Regionen (blau und grün gekennzeichnet) generiert. Beide Hypothesen scheinen sich jedoch gegenseitig zu unterstützen. Sie werden in einem weiteren Areal fusioniert (rot).*

Regionen wird vermutet, dass sie lokale Bewegungssignale verarbeiten. Andere Studien [Grossman et al., 2000; Downing et al., 2001] beobachten dieselben Aktivitäten, wenn man einen sogenannten „Scrambled Walker“ als Kontrollstimulus verwendet. Dabei werden Punkte mit identischen Bewegungssignalen präsentiert. Allerdings weisen die Punkte eine zufällige räumliche Anordnung auf, wodurch sich keine menschliche Gestalt mehr ergibt. Im Gegensatz dazu wird die Extrastriate Body Area (EBA) durch statische Bilder des menschlichen Körpers aktiviert [Downing et al., 2001]. Darüberhinaus ist die EBA bei der Präsentation von Point-Light Displays ebenfalls stärker aktiviert als bei dem „Scrambled Walker“-Stimulus [Downing et al., 2001; Grossman und Blake, 2002]. Die Lage der einzelnen Gehirnareale ist in Abbildung 2.2 noch einmal verdeutlicht.

An der gesamten Verarbeitung von Form- und Bewegungsinformationen sind zweifelsfrei mehr als die hier benannten Regionen beteiligt. Allerdings ist dieser kurze Überblick ausreichend, um zu verstehen, dass die Verarbeitung in unterschiedlichen Regionen und damit weitestgehend getrennt stattfindet. Somit ist eine ausschließliche Betrachtung von Trajektorien durchaus auch aus biologischer Sicht gerechtfertigt. Für die weiterführende Forschung, kann aber nicht auf die Formwahrnehmung verzichtet werden.

2.2. Verwandte Arbeiten

Die Recherche auf dem Gebiet der Biologie hat ergeben, dass das Gehirn von Menschen und einigen Tieren in der Lage ist, Bewegungsabläufe zu interpretieren und daraus Verhalten abzuleiten. Dieser Abschnitt soll dazu genutzt werden, Arbeiten vorzustellen, die bereits erfolgreich die Analyse komplexerer Bewegungsabläufe für technische Systeme einsetzen. Dabei sollen die Veröffentlichungen unter zwei wesentlichen Aspekten beleuchtet werden. Als Erstes soll die Frage beantwortet werden, welche Arten von Anwendungen auf die Analyse von Bewegungsmustern aufbauen. Da der Fokus dieser Arbeit im Bereich der mobilen Robotik angesiedelt ist, soll bei der Beleuchtung der Anwendungsfelder nur verwandte Themenbereich aufgeführt werden. Ansonsten besteht die Gefahr sich in der Vielzahl möglicher Applikationen zu verlieren. Die

jeweiligen Anwendungsfelder sind entsprechend kursiv hervorgehoben. Danach sollen die dabei zu Einsatz gekommenen Methoden vorgestellt werden. Dies kann und soll an dieser Stelle nur exemplarisch erfolgen. Es wird sich daher auf die am weitesten verbreiteten Methoden in erster Linie aus dem Bereich Robotik und deren näheren Umfeld konzentriert werden. Zur besseren Übersichtlichkeit bzw. Einordnung sind die Anwendungsfelder jeweils kursiv hervorgehoben.

2.2.1. Anwendungsfelder

Durch die zunehmende Angst vor terroristischen Anschlägen und die steigende Anzahl von gewalttätigen Übergriffen, wurden in den letzten Jahren viele staatlich subventionierte Programme zur intelligenten *Überwachung von öffentlichen Plätzen* ins Leben gerufen. Ein Beispiel hierfür ist das CASANDRA-Projekt [Zajdel et al., 2007]. Ziel des CASANDRA-Projektes ist es, aggressive Handlungen von Passanten, z.B. auf Bahnhöfen, zu erkennen.

In vielen Fällen steht allerdings nicht eine solch eindeutige Erkennungsaufgabe im Vordergrund. Für eine große Anzahl Überwachungsaufgaben, beispielsweise [Johnson und Hogg, 1996; Junejo und Foroosh, 2007], wird Auffälligkeit als Abweichung von normalen Bewegungsabläufen definiert. Dabei spielen allerdings nicht immer Überwachungen zum Erhöhen der Sicherheit eine Rolle. In [Suzuki et al., 2007] werden Geschäfte und eine Ladenpassage untersucht, um auf Basis von Bewegungstrajektorien die Platzierung von Produkten zu verbessern. Auch in Heim- oder Büroumgebungen werden Analysen von typischen Bewegungsabläufen durchgeführt [Bennewitz et al., 2005; Bennewitz et al., 2002].

Gerade bei Überwachungsaufgaben fällt eine sehr große Menge Datenmaterial an. Daher hat sich im Bereich Datenbanken eine Spezialisierung der *Time Series Databases (TSDB)* herausgebildet. Sie wird als *Motion Data Mining* [Khalid und Naftel, 2005] oder auch als *Moving Object Databases* [Yu und Kim, 2006] bezeichnet. Dieses Forschungsfeld untersucht, wie Bewegungsabläufe möglichst effizient gespeichert und wiedergefunden werden können. Dabei spielen Techniken der Repräsentation, aber auch Ähnlichkeitsmaße und Clusterverfahren eine wesentliche Rolle. In [Jin et al., 2004] werden beispielsweise solche Techniken

des Data Mining dazu eingesetzt, den Verkehrsfluss an Kreuzungen zu analysieren. Laut [Yu und Kim, 2006] gehören auch Mobilfunksysteme, Gefechtsplanung und Transportaufgaben zu den möglichen Anwendungsbereichen für diese Art von Datenbanken.

Ein sehr großes Anwendungsfeld für Bewegungstrajektorien stellt der Bereich der *Navigation* dar. Dabei wird einerseits versucht auf möglichst natürliche Art und Weise einen Bewegungspfad zu planen [Brogan und Johnson, 2003; Bruce und Gordon, 2004; Kuipers, 1983]. Viel wichtiger ist allerdings, die Erkennung der Bewegungen der Personen in deren Umfeld sich ein Assistenzroboter bewegen soll. Um andererseits in Umgebungen mit dynamischen Hindernissen navigieren zu können, ist es notwendig die Bewegung dieser Hindernisse vorhersagen zu können. Die Prädiktion muss in solchen Fällen über einen längeren Zeitraum stabile und brauchbare Ergebnisse liefern [Govea et al., 2004; Weser et al., 2006].

Eine etwas kuriosere Umsetzung der Erkennung der Bewegung von Personen und der entsprechenden Reaktion darauf findet sich in [Takeda et al., 2007]. In dieser Veröffentlichung wird ein Roboter beschrieben, der als Tanzpartner bzw. auch Tanzlehrer auftreten soll.

Die Idee der Prädiktion lässt sich nicht nur im Bereich der mobilen Assistenzrobotik anwenden. In [Rhodes et al., 2007] findet sich ein Beispiel, bei dem der Kurs von Schiffen in einer Hafeneinfahrt prädiziert wird. Das System soll einen Koordinator dabei unterstützen, Unfälle zu vermeiden.

Bis zu diesem Punkt wurden Publikationen vorgestellt, bei denen die Bewegungstrajektorien einem zweidimensionalen Pfad entsprechen. Die Analyse von Trajektorien ist allerdings nicht auf den zweidimensionalen Fall beschränkt. Insbesondere wenn es um die Beschreibung von Körperbewegungen geht, muss die Trajektorie im dreidimensionalen Raum repräsentiert sein.

Im dreidimensionalen Fall finden sich Anwendungsfelder, die denen der Navigation ähneln. Beispielweise wird in [Shimosaka et al., 2007] versucht, die Art der Bewegung zu erkennen. Es soll dabei erkannt werden, ob die Person steht, geht oder eine andere Tätigkeit ausführt. Eine komplexere *Beschreibung der Körperbewegung* bishin zu den Bewegungen der einzelnen Gelenke findet sich dagegen in [Ogawara et al., 2007]. In

[Otero et al., 2006] werden unterschiedliche Gesten auf Basis der Bewegungsabfolge erkannt. Darüber hinaus versucht man anthropomorphe Roboter zu konstruieren, die komplizierte Bewegungen durchführen sollen. Hierzu werden Modelle benutzt, die eine möglichst *natürliche Bewegung generieren* können [Bolder et al., 2007]. Dies lässt sich sogar soweit ausbauen, dass die Roboter in der Lage sind, durch *Imitationslernen* komplexe menschliche Bewegungsabläufe nachzuahmen [Schaal et al., 2003; Takano und Nakamura, 2006; Lee und Nakamura, 2007; Muehlig et al., 2009].

In der automatisierten Produktion soll die Zusammenarbeit zwischen Mensch und Roboter verbessert werden. Dies ist momentan durch die strikte Absicherung des Arbeitsbereiches des Roboters nicht möglich. Um die Zusammenarbeit zu ermöglichen muss der Sicherheitsbereich adaptiv gestaltet werden. Außerdem muss ständig die Position des Arbeiters bekannt sein und seine Handlung prädiziert werden, um Gefährdungen des Arbeiters durch den *Produktionsroboter* zu vermeiden [Hahn et al., 2008].

Zu guter Letzt sei die *Handschrifterkennung* als eine weitere Anwendung der Trajektorienanalyse erwähnt [Brakensiek, 2002; Williams et al., 2007; Dixon und Khosla, 2004]. Auf den ersten Blick können hierfür ähnliche Überlegungen getroffen werden, wie für die Analyse von Bewegungen. Genau genommen handelt es sich um nichts anderes, als die Bewegung des Stiftes bzw. der Hand. Allerdings müssen einige spezielle Eigenschaften bedacht werden. Beispielsweise kann der Stift abgesetzt werden. Dann wird die Handschrifttrajektorie an einer anderen räumlichen Position fortgeführt.

2.2.2. Methodik

Genauso vielfältig wie die Anwendungsfelder sind auch die Methoden, die dabei zum Einsatz kommen. Dabei kann festgestellt werden, dass es keine Tendenz zu einer bestimmten Methodik in den einzelnen Anwendungsfeldern gibt. Je nach Präferenzen kommen auch alle Arten von Methoden, d. h. probabilistische, neuronale oder auch analytische Ansätze, zum Einsatz. Tabelle 2.1 trägt die einzelnen Ansätze zusammen und ordnet sie entsprechend ein.

Probabilistisch	Neuronal	Analytisch	Sonstiges
<ul style="list-style-type: none"> ● HMM – [Lee et al., 2003] – [Suzuki et al., 2007] – [Bennewitz et al., 2005] – [Lee und Nakamura, 2007] – Hierarchische HMM <ul style="list-style-type: none"> * [Takano und Nakamura, 2006] – Abstract HMM <ul style="list-style-type: none"> * [Bui et al., 2001] – FHMM <ul style="list-style-type: none"> * [Williams et al., 2006] * [Williams et al., 2007] ● CRF <ul style="list-style-type: none"> – [Shimosaka et al., 2007] ● Dynamic Bayes Networks <ul style="list-style-type: none"> – [Zajdel und Kröse, 2005] 	<ul style="list-style-type: none"> ● [Baier, 2006] ● RNN <ul style="list-style-type: none"> – [Tani und Nolfi, 1999] ● GNG ● [Rhodes et al., 2007] ● Leaky-Integrator <ul style="list-style-type: none"> – [Johnson und Hogg, 1996] ● SOINN <ul style="list-style-type: none"> – [Okada und Hasegawa, 2007] 	<ul style="list-style-type: none"> ● Linear Dynamical Systems <ul style="list-style-type: none"> – [Dixon und Khosla, 2004] ● Dynamic Movement Primitives <ul style="list-style-type: none"> – [Schaal et al., 2003] ● Euler-Lagrange Gleichungen <ul style="list-style-type: none"> – [Oyama und Uno, 2006] ● Spatiotemporal Morphable Models <ul style="list-style-type: none"> – [Ilg und Giese, 2002] ● DTW <ul style="list-style-type: none"> – [Junejo und Foroosh, 2007] ● DFT <ul style="list-style-type: none"> – [Khalid und Naftel, 2005] 	<ul style="list-style-type: none"> ● Clustering <ul style="list-style-type: none"> – Pairwise Clustering <ul style="list-style-type: none"> * [Govea et al., 2004] – Greedy <ul style="list-style-type: none"> * [Jin et al., 2004] – K-Means <ul style="list-style-type: none"> * [Suzuki et al., 2007] – SOM <ul style="list-style-type: none"> * [Weser et al., 2006] * [Khalid und Naftel, 2005] ● Eventbasiert <ul style="list-style-type: none"> – Goals/Resting Places <ul style="list-style-type: none"> * [Bruce und Gordon, 2004] * [Bennewitz et al., 2005] – Krümmung <ul style="list-style-type: none"> * [Rao et al., 2002] * [Junejo und Foroosh, 2007] ● Unterraumprojektion <ul style="list-style-type: none"> – Blind Source Sep. <ul style="list-style-type: none"> * [Omlor und Giese, 2006] – PCA <ul style="list-style-type: none"> * [Troje, 2002] – Multidim. Scaling <ul style="list-style-type: none"> * [Suzuki et al., 2007]

Tabelle 2.1.: Vorgestellten Literatur zur Bewegungsanalyse

Da es nicht der Anspruch dieser Arbeit ist, ein konkretes Problem zu lösen, soll nicht die jeweilige Umsetzung in den einzelnen Arbeiten im Vordergrund stehen. Stattdessen sollen die einzelnen Methoden kurz genannt und ihre Übertragbarkeit auf andere Problemstellungen im Bereich der Bewegungsanalyse beschrieben werden.

2.2.2.1. Probabilistische Methoden

In der aktuellen Forschung erfreuen sich probabilistische Ansätze sehr großer Beliebtheit. Dementsprechend wird auch für die Bewegungsanalyse eine Vielzahl wahrscheinlichkeitsbasierter Ansätze untersucht. Dabei kommen in erster Linie Hidden Markov Modelle (HMM) zum Einsatz [Lee et al., 2003; Suzuki et al., 2007; Bennewitz et al., 2005; Lee und Nakamura, 2007]. Bei Hidden Markov Modellen handelt es sich um stochastische Prozesse 2. Ordnung, die durch ihre sequenzielle Modellierung zeitliche Zusammenhänge ausdrücken können.

In der Literatur existieren zahlreiche Spielarten, Modifikationen und Weiterentwicklungen. Eine Möglichkeit, unterschiedliche Zeithorizonte zu realisieren, bieten hierarchische HMM [Takano und Nakamura, 2006]. Im Falle der Abstract HMM [Bui et al., 2001] wird gezeigt, dass die zu Grunde liegende Markov-Annahme beliebig abstrahiert werden kann. Somit lassen sich nahezu beliebige Ebenen definieren, die jeweils einen anderen zeitlichen Kontext repräsentieren können. Mit den Factorial HMM [Williams et al., 2007; Williams et al., 2006] lassen sich komplexere Systeme aus mehreren einzelnen zusammensetzen, da die beobachtbaren zeitlichen Zustände durch mehrere versteckte Zustände beschrieben werden können. Die verallgemeinerte Form der HMM, die Dynamic Bayes Networks (DBN), kommen in [Zajdel und Kröse, 2005] zum Einsatz.

Ähnlich wie beim Dynamic Time Warping [Junejo und Foroosh, 2007; Muehlig et al., 2009; Sakoe und Chiba, 1990] modelliert ein HMM in erster Linie die Ausprägung einer Trajektorie. Damit geht allerdings die Aussage über die Dynamik der Bewegung verloren. Um dies zu verhindern, muss explizit die Geschwindigkeit der Trajektorie beobachtet werden.

Interessant ist die Verwendung der Conditional Random Fields (CRF) [Shimosaka et al., 2007], da es sich hierbei um ein ungerichtetes graphisches Modell handelt. Theoretische besteht also die Möglichkeit die Bewegung auch rückwärts durchzuführen.

Die beschriebenen grafischen Modelle sind in der Lage Trajektorien zu klassifizieren. Allerdings muss für jede Klasse ein grafisches Modell erstellt und trainiert werden. Ebenso werden HMM dazu eingesetzt, eine aktuell beobachtete Bewegung zu präzisieren. Dabei ist die Größe des Prädiktionshorizontes abhängig von Größe des Modells, d. h. von der Anzahl der Zustände. Auch bei der Prädiktion gilt, dass für jede Trajektorienart, die präzisiert werden soll, ein eigenes Modell erstellt werden muss.

2.2.2.2. Neuronale Methoden

Im Bereich der neuronalen Verfahren muss unterschieden werden, ob das entsprechende Netzwerk bereits in der Lage ist, zeitliche Abläufe zu modellieren. Eine große Gruppe solcher Netzwerke sind die Rekurrenten Neuronalen Netze (RNN) [Jordan, 1986; Elman, 1990]. Hierbei wird der zeitliche Kontext durch Rückführung der Ausgaben oder der Informationen aus den Hiddenschichten erzeugt. [Tani und Nolfi, 1999] gehen sogar noch einen Schritt weiter, in dem sie Hierarchien solcher rekurrenten Netze zur Beschreibung von Bewegungen einsetzen. In dem Ansatz von [Johnson und Hogg, 1996] wird der zeitliche Zusammenhang durch eine besondere Hiddenschicht geschaffen. Bei den Neuronen dieser Schicht handelt es sich um Leaky Integratoren, wodurch ähnlich wie bei den rekurrenten Netzen ein zeitliches Abklingen realisiert wird.

Interessanterweise kommen für die Bewegungsanalyse auch neuronale Verfahren zum Einsatz, die nicht direkt in der Lage sind zeitliche Informationen zu verarbeiten. In [Rhodes et al., 2007] kommt eine dem Growing Neural Gas (GNG) ähnelnde Variante zum Einsatz. Im Nachhinein werden die zeitlichen Übergänge zwischen den einzelnen Neuronen beschrieben. Dagegen wird von [Okada und Hasegawa, 2007] ein Self-organizing Incremental Neural Network (SOINN) verwendet. Dies wird auf Bewegungstrajektorien angewendet, die mit Hilfe des Dynamic Time Warpings im zeitlichen Ablauf einander angeglichen wurden.

Eine ausführliche Untersuchung zur Anwendung neuronaler Netze auf Bewegungsdaten wird in der Dissertation [Baier, 2006] vorgestellt.

Neuronale Verfahren besitzen genauso, wie die beschriebenen probabilistischen Verfahren, die Fähigkeit auf entsprechende Daten trainiert zu werden. Da es mit Neuronalen Netzen möglich ist, den Datenraum zu partitionieren, ist es meist ausreichend nur ein Netz, welches unterschiedliche Klasseninformationen ausgeben kann, zu trainieren.

2.2.2.3. Analytische Methoden

Die dritte Gruppe an Verfahren stellen die analytischen Ansätze dar. In einigen Arbeiten findet sich die Tendenz, Bewegungen mit Hilfe von Differenzialgleichungen zu beschreiben [Schaal et al., 2003; Dixon und Khosla, 2004; Oyama und Uno, 2006]. Auch wenn die Modelle parametrischer Natur sind, wird durch die Differenzialgleichung bis zu einem gewissen Grade Modellwissen eingebracht.

Eine analytische Alternative zu den HMM stellt das Dynamic Time Warping dar [Junejo und Foroosh, 2007; Sakoe und Chiba, 1990]. Dabei können Trajektorien mit unterschiedlichem zeitlichen Verlauf miteinander verglichen werden. Dazu werden sie bezüglich der zeitlichen Dimension lokal gestaucht oder gestreckt, bis die räumlichen Dimensionen eine minimale Distanz aufweisen.

In [Ilg und Giese, 2002] findet sich mit den Spatio-Temporal Morphable Models ein sehr interessanter Ansatz, der mehrere Aspekte versucht zu kombinieren. Durch ein dem Dynamic Time Warping ähnlichem Verfahren werden Korrespondenzen zwischen bekannten Trajektorien hergestellt. Auf deren Basis werden elementare Trajektorienstücke extrahiert. Diese elementaren Bewegungen werden dazu verwendet, eine aktuell beobachtete Trajektorie zu rekonstruieren.

Ein etabliertes Verfahren, wenn es um die Analyse von zeitlichen Abläufen geht, ist die Fourieranalyse. Insbesondere in ihrer diskreten Ausprägung der Diskreten Fouriertransformation findet sie auch im Bereich der Bewegungsanalyse Anwendung [Khalid und Naftel, 2005]. Sie soll an dieser Stelle als Vertreter für eine Beschreibung mit Hilfe von Basisfunktionen stehen. Dadurch ist es möglich, die Trajektorie in einen Parameterraum zu transformieren.

2.2.2.4. Sonstige Verfahren

Die Verfahren die in diesem Abschnitt vorgestellt werden, können zwar alle in die, den bisherigen drei Abschnitten entsprechenden, Kategorien einsortiert werden. Sie wurden aber dennoch herausgegriffen und zu Gruppen zusammengefasst.

Einen großen Anteil, insbesondere im Zusammenhang mit Time Series Databases, nehmen die Clusterverfahren ein [Govea et al., 2004; Jin et al., 2004; Suzuki et al., 2007]. Dabei wird versucht, ähnliche Bewegungsabläufe zusammenzufassen. Meist wird dabei auf die etablierten Clusterverfahren zurückgegriffen. Die eigentliche Charakterisierung von ähnlichen Abläufen muss durch eine geeignete Wahl der Repräsentation erfolgen. Durch die Verwendung von Standardansätzen zum Clustern geht der zeitliche Zusammenhang zwischen den Clustern verloren, da die Unterteilung auf Basis der räumlichen Lage erfolgt und keine kausale Beziehung zwischen den Clustern existiert. In [Weser et al., 2006; Khalid und Naftel, 2005] werden Möglichkeiten vorgestellt, diesen Nachteil zu kompensieren. Dazu werden Übergangswahrscheinlichkeiten zwischen den einzelnen Clustern definiert, die es ermöglichen eine zeitliche Abfolge herzustellen. Vergleichbare Ansätze wurden beispielsweise bereits unter den Namen Temporal Kohonen Map (TKM) [Chappell und Taylor, 1993] und Recurrent Self-organizing Feature Map (RSOM) [Koskela et al., 1998] publiziert.

Eine andere Möglichkeit Trajektorien zu segmentieren stellt das Festlegen von markanten Stellen dar. In [Bennewitz et al., 2005] und in [Bruce und Gordon, 2004] werden dabei ähnliche Konzepte verfolgt. Dazu werden bestimmte Punkte im Raum definiert, an denen sich Trajektorien häufen. Diese werden als Resting Places [Bennewitz et al., 2005] oder Goals [Bruce und Gordon, 2004] bezeichnet. Ein anderer Ansatz wird von [Rao et al., 2002; Junejo und Foroosh, 2007] verfolgt. Hierbei definieren bestimmte Eigenschaften einer einzelnen Trajektorie die markanten Punkte. Dazu wird in beiden Arbeiten die Krümmung verwendet.

Auch im Zusammenhang mit Bewegungsdaten fallen üblicherweise eine Menge Daten an. In den meisten Fällen sind nicht alle diese Daten informationstragend. Daher werden häufig Verfahren zu Unterraumprojektion eingesetzt. In [Omlor und Giese, 2006] werden unterschiedliche Methoden, wie Hauptkomponentenanalyse (PCA) und Independend

Component Analyse (ICA), verglichen. [Troje, 2002] geht sogar noch einen Schritt weiter und benutzt die PCA dazu sowohl die räumliche Beschaffenheit eines menschlichen Point-Light Displays, als auch dessen zeitliche Abfolge der Bewegungen zu beschreiben. Ein mit der PCA verwandtes Verfahren kommt in [Suzuki et al., 2007] zum Einsatz. Beim sogenannten Multidimensional Scaling (MDS) wird nicht wie bei der PCA die Unterraumprojektion auf Basis der originalen Daten errechnet. Stattdessen wird mit Hilfe der paarweisen Ähnlichkeiten der zur Verfügung stehenden Daten ein Raum aufgespannt. Damit ist dieses Verfahren und damit auch die Qualität seiner Ergebnisse sehr stark abhängig von der Wahl eines geeigneten Ähnlichkeitsmaßes.

Zum Schluss sei noch die Dissertation von [Stein, 2003] erwähnt. Auch diese Arbeit beschäftigt sich mit der Analyse von Bewegungsabläufen. Genauso wie in dieser Arbeit werden zu Beginn unterschiedliche Repräsentationsformen untersucht. Allerdings sind die betrachteten Verfahren nicht identisch mit den in dieser Arbeit aufgeführten Repräsentationsformen. Der Anspruch von [Stein, 2003] besteht ebenfalls darin, neue Wege zu beschreiten und sich nicht auf die etablierten Verfahren zurückzuziehen. Dabei kommt eine Repräsentationsform, die als Qualitative vektorielle Repräsentation bezeichnet wird, zum Einsatz. Mit dieser erfolgt eine eher semantische Beschreibung der Trajektorie. Mit Hilfe dieser semantischen Beschreibung können die Trajektorien abstrakter beschrieben werden indem unwesentliche lokale Details weggelassen werden. Dazu bedient er sich aus der Geographie stammenden Generalisierungsverfahren, wie z. B. der Rolland-Generalisierung, konvexen Hüllen-Generalisierung und der Puffergeneralisierung.

2.3. Fazit

Sowohl aus den biologischen, als auch den ingenieurtechnischen Arbeiten lässt sich ableiten, dass aus Bewegungstrajektorien eine große Menge an zusätzlichen Informationen gewonnen werden kann. Die Analyse von Bewegungen bildet daher einen zentralen Kern dieser Arbeit.

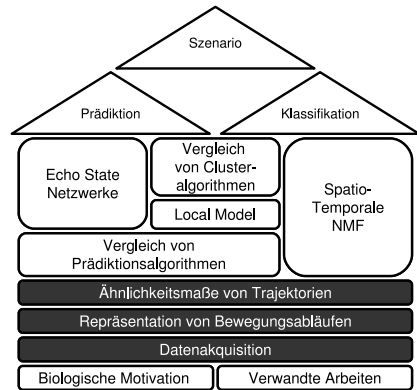
An der Aufarbeitung der bereits veröffentlichten Arbeiten ist zu erkennen, dass sich zur Bewegungsanalyse bereits eines breiten Spektrums an Methoden bedient wird. Es fehlt allerdings noch eine systemati-

sche Einordnung zur gezielteren Anwendung der einzelnen Verfahren. Dies lassen insbesondere Arbeiten, wie [Hahn et al., 2008] erkennen, die mit recht einfachen Mitteln, wie Templatematching, versuchen komplexe Problemstellungen, wie Prädiktionsaufgaben zu lösen. Im Rahmen dieser Arbeit wird für unterschiedliche Repräsentationsformen und Ähnlichkeitsmaße für Bewegungsdaten ein erster Ansatz zur Systematisierung vorgestellt.

Ähnlich wie in [Stein, 2003] soll auch in der vorliegenden Arbeit versucht werden, unter Anderem eine geeignete Vorverarbeitung für Bewegungsdaten zu finden. Darüberhinaus wird die Prädiktion und die Klassifikation einen zentralen Kern der Arbeit darstellen. Dabei sollen die recht gut untersuchten Methoden, wie z. B. HMM, ausgespart werden. Ziel der Arbeit soll es eher sein, neu Algorithmen für das Feld der Bewegungsanalyse zu erschließen.

Kapitel 3

Vorverarbeitung von Bewegungsabläufen



Bevor in den Kapiteln 4 bis 6 die Prädiktion und Klassifikation von Bewegungstrajektorien eine Rolle spielen wird, beschäftigt sich dieses Kapitel mit der Vorverarbeitung. Dabei spielt zum Einen in Abschnitt 3.2 die Frage eine Rolle, welche Repräsentationsform am besten geeignet für die Verarbeitung von Bewegungstrajektorien ist. Zum Anderen wird in Abschnitt 3.3 beleuchtet, wie man die Ähnlichkeit von Trajektorien ausdrücken kann. Bevor sich diesen beiden Problemen gewidmet wird, sollen als Erstes noch die für die Experimente in dieser Arbeit verwendeten Daten vorgestellt werden.

3.1. Datenakquisition

In Abschnitt 1.2 wurde ein recht komplexes Szenario beschrieben, bei dem unterschiedliche Aspekte im Rahmen dieser Arbeit untersucht werden sollen. Um die dazu notwendigen experimentellen Evaluierungen

durchführen zu können, werden im Rahmen dieser Arbeit im wesentlichen zwei unterschiedliche Datensätze verwendet. Diese sollen im folgenden kurz vorgestellt werden.

In Abschnitt 3.1.1 werden dreidimensionalen Daten der Bewegungen einzelner Gliedmaßen vorgestellt. Sie sollen helfen, die Aspekte zu beleuchten, die notwendig sind, wenn der Assistenzroboter mit seinem Gegenüber körperlich interagieren, z. B. ihm die Hand geben oder eine Tasse reichen, soll. Da für diese Daten auch Klasseninformationen vorhanden sind, werden sie auch eingesetzt, um die Möglichkeiten des Erkennens von Bewegungen zu untersuchen.

Anschließend werden in Abschnitt 3.1.2 zweidimensionale Trajektorien beschrieben. Da bei Navigationsaufgaben in erste Linie auf zweidimensionale Beschreibungen zurückgegriffen wird, werden diese Daten zum Verständnis solcher Problemstellungen herangezogen.

Zur Untersuchung unterschiedlicher Möglichkeiten zur Repräsentation von Trajektorien kam ein weiterer zweidimensionaler Datensatz zum Einsatz. Dieser wurde auf ähnliche Weise aufgezeichnet, wie in Abschnitt 3.1.2 beschrieben. Allerdings enthält er zusätzlich Informationen über das Interaktionsinteresse der sich bewegenden Person. Da dieser nur für die Untersuchungen zu Repräsentationsmöglichkeiten genutzt wird, erfolgt seine Vorstellung erst an entsprechender Stelle in Abschnitt 3.2.5.

3.1.1. Point-Light Display

Im Rahmen der Experimente in dieser Arbeit wurden an vielen Stellen die von Frank Pollick am Department of Psychology der University of Glasgow aufgezeichneten Bewegungsdaten verwendet. Diese stehen unter der Adresse http://paco.psy.gla.ac.uk/data_ptd.php zum Herunterladen frei zur Verfügung.

Dies scheint im Zusammenhang mit der Forderung nach einem Realweltainsatz erst einmal unvereinbar zu sein. Allerdings sollten bei sämtlichen Untersuchungen die jeweils betrachteten Verfahren im Vordergrund stehen. Deshalb wurden Daten verwendet, die mit einem Tracker aufge-

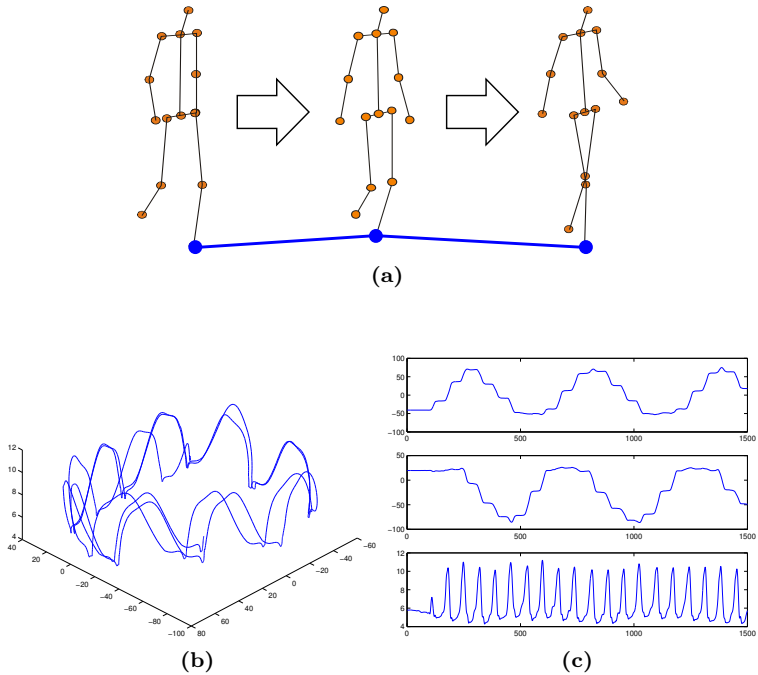


Abbildung 3.1.: Bewegungstrajektorie eines Fußknöchels: Die aufgezeichneten Daten des linken Fußknöchels (a) als Bewegung im Raum (b) und aufgetrennt in einzelne Dimensionen (c). Bewegungsdaten weisen häufig Regelmäßigkeiten auf, sind aber dennoch als chaotische Zeitreihen einzustufen, da sich ihr Verhalten unter Umständen innerhalb kürzester Zeit ändern kann. Betrachtet man die einzelnen Raumkoordinaten unabhängig voneinander, so kann man auf diesen 1-dimensionalen Daten operieren. Von oben sind die x -, y - und z -Koordinaten der Bewegung als zeitlicher Verlauf dargestellt. Die Einheiten der Wertachsen sind stets cm. ((b) und (c) entnommen aus [Strauß, 2008])

zeichnet wurden, der selbst möglich wenig Fehler produziert und somit die Experimente unter Umständen negativ beeinflusst.

Darüber hinaus sei zu erwähnen, dass durch die Verwendung der Datenbank auf reale Bewegungssequenzen zurückgegriffen werden konnte. Somit konnte auf eine ausschließlich auf synthetischen Daten durchgeführte Auswertung verzichtet werden. Ein Beispiel für die verwendeten Daten ist in Abbildung 3.1 dargestellt.

Für die Aufzeichnung der Daten wurden Personen mit Markern beklebt und mit einem von Optotrak stammenden 3D-Erfassungssystem mit einer Frequenz von 60 Hz aufgezeichnet. Die Marker wurden dabei an 15 Gelenken der Person befestigt, und es wird somit deren Bewegung erfasst. Die Daten enthalten Aufzeichnungen von 30 männlichen und weiblichen Personen, die unterschiedliche Aktionen ausführen. Darunter die folgenden:

- das Anheben eines Gegenstandes,
- eine Klopfbewegung,
- eine Wurfbewegung und
- das Laufen im Kreis.

Die einzelnen Aktionen wurden mit unterschiedlichen Emotionen der beteiligten Personen aufgezeichnet. Es sind die Emotionen:

- ärgerlich
- glücklich,
- traurig und
- neutral dargestellt.

Die Bewegungsdatenbank wurde auch für die in Abschnitt 2.1.2 beschriebenen Untersuchungen eingesetzt. Weitere Informationen über die Aufzeichnung der Daten finden sich in [Pollick et al., 2003].

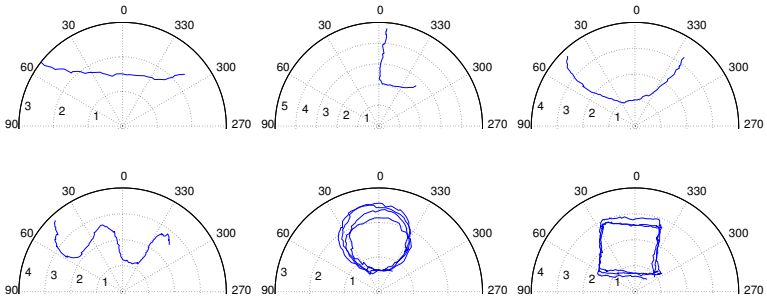


Abbildung 3.2.: Aufgezeichnete Trajektorien:

Beispiele für die mit Hilfe des Lasertrackers aufgezeichneten Bewegungsdaten in Polarkoordinaten. Der Ursprung entspricht dabei der Position des Roboters.

3.1.2. Lasertracker

Die in Abschnitt 3.1.1 vorgestellten Daten basieren zwar auf realen Aufnahmen von Personen, können aber durch die Methodik der Aufnahme nur zu einem gewissen Grade mit Daten verglichen werden, die von einem Assistenzroboter aufgezeichnet werden. Üblicherweise arbeiten Assistenzroboter mit Trackingsystemen, die nicht markerbasiert funktionieren. Dabei kommen neben visuellen Sensoren auch Laserscanner zum Einsatz.

Um einem solchen System gerecht zu werden, wurden mit einem vereinfachten Trackingalgorithmus auf Basis eines Laserscanners für einige Experimente zusätzliche Daten aufgezeichnet. Dazu wird der Laserscanner stationär positioniert und die Umgebung ohne Personen aufgezeichnet, um ein Hintergrundmodell zu berechnen. Bewegt sich eine Person durch den auf 5 m Radius beschränkten Erfassungsbereich des Lasers, weicht die Beobachtung des Lasers vom aufgezeichneten Hintergrundmodell ab. Diese Position der Änderung wird dann für jeden Zeitschritt aufgezeichnet, um die Trajektorie zu erhalten. Dabei werden die Daten mit einer Aufzeichnungsrate von 10 Hz erfasst. Da der Laser nur in einer Ebene Daten erfassen kann, sind die Daten nur im zweidimensionalen Raum verfügbar und repräsentieren die Projektion der

Position der Person auf die Bodenebene. Während die Daten aus Abschnitt 3.1.1 ca. 1500 bis 2500 Punkte je Trajektorie enthalten, wurden mit Hilfe des Lasertrackers 50 bis 300 Positionen umfassende Trajektorien aufgezeichnet. Dies entspricht einer Aufzeichnungsdauer von 5 bis 30 Sekunden. Insgesamt wurden 20 Trajektorien mit den in Abbildung 3.2 dargestellten Bewegungen aufgezeichnet und für die Experimente verwendet. Damit lassen sich auch mit diesen Daten verallgemeinerbare Ergebnisse erzielen.

Diese simple Herangehensweise garantiert, dass mögliche Fehler des Trackingalgorithmus gering gehalten werden können. Somit kann sich bei den Auswertungen auf die Prädiktionsalgorithmen konzentriert werden. Dadurch, dass auf die Nutzung beispielsweise eines Partikelfilters verzichtet wurde, weisen die aufgezeichneten Trajektorien ein gewisses Rauschen, verursacht durch die Sensorungenauigkeit, auf. Dies wird bewusst in Kauf genommen, um beurteilen zu können, wie die jeweiligen Verfahren damit zurechtkommen.

3.2. Repräsentation von Bewegungstrajektorien

Eine zentrale Fragestellung bei der Betrachtung von Bewegungsinformationen ist die Art ihrer Repräsentation. Mit Repräsentation soll in dieser Arbeit die Form der Speicherung einer Bewegungstrajektorie bezeichnet werden. Oder etwas anders formuliert, in welcher Weise muss die Trajektorie dem nachfolgenden Algorithmus übergeben werden, um diesen das Erfüllen seiner Aufgabe zu erleichtern. In den meisten Veröffentlichungen wird die Form der Repräsentation durch den Entwickler festgelegt. Dieser Abschnitt soll zum Einen helfen, diesen Entscheidungsprozess zu vereinfachen und bietet dazu eine Systematisierung an. Zum Anderen besteht natürlich der Wunsch, diese Entscheidung auf intelligente Weise durch das System durchführen zu lassen. Deshalb wird am Ende dieses Kapitels demonstriert, wie eine solche Auswahl automatisiert erfolgen kann.

Dabei soll in diesem Kapitel ein Beispiel helfen zu verstehen, welche Schritte durchgeführt werden müssen. Um das Beispiel besser nachvoll-

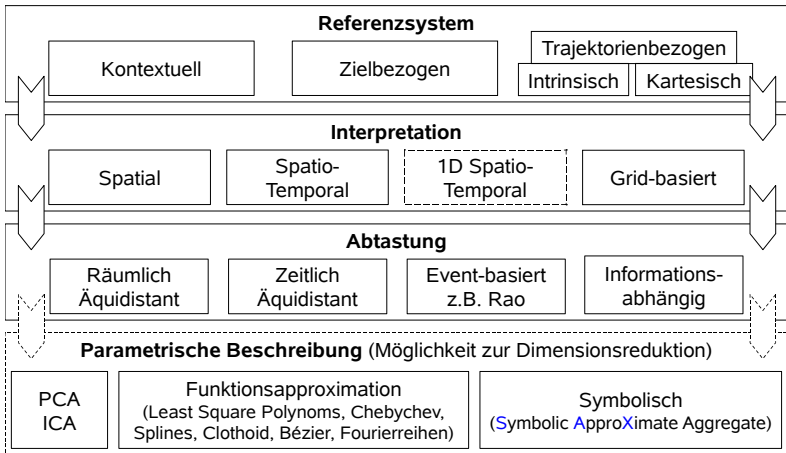


Abbildung 3.3.: Repräsentationsformen für Bewegungstrajektorien: *Durch die unterschiedlichen Repräsentationsformen werden auch unterschiedliche Aspekte der Bewegung in den Vordergrund gerückt.*

ziehen zu können, wird die Anzahl der Repräsentationsmöglichkeiten gering gehalten. Abbildung 3.4 zeigt dabei die in diesem Kapitel untersuchten Methoden und in welcher Reihenfolge sie aufeinander aufbauen. In Anhang B werden alle übrigen in Abbildung 3.3 vorgestellten und untersuchten Repräsentationsmöglichkeiten näher beleuchtet. Als Beispiel wird der in Abschnitt 1.2 beschriebene Erkennungsaspekt untersucht. Ziel ist es dabei, für eine beobachtete Person zu entscheiden, ob diese Person Interesse an einer Interaktion mit dem Serviceroboter hat oder nicht. Zu lösen ist somit ein binäres Klassifikationsproblem. Aus diesem Grund ist in Abbildung 3.4 als letzter Schritt (f) der Klassifikator dargestellt.

Die in diesem Kapitel gezeigten Untersuchungen wurden in [Müller et al., 2008] veröffentlicht. Eine ausführlichere Beschreibung aller durchgeführten Experimente findet sich in [Ober, 2007]¹.

¹Die Diplomarbeit wurde vom Autor im Rahmen des Projektes betreut.

Bevor sich der eigentlichen Problematik zugewandt werden kann, bleibt noch das Problem zu klären, wie die Bewegungsinformation aufgezeichnet wird. Prinzipiell muss dazu das entsprechende dynamische Objekt erkannt und in Ort und Zeit verfolgt werden können. Ein solches als Tracker bezeichnetes System, liefert im Idealfall zu jedem Zeitpunkt $i = 0, 1, \dots, n - 1$ die Position \mathbf{s}_i des betrachteten Objektes (z. B. der sich annähernden Person). In der Praxis bereitet das Tracken beispielsweise von Personen durchaus noch Probleme, so dass Daten unvollständig bzw. falsch zugeordnet vorliegen. Für die Betrachtungen im Rahmen dieser Arbeit soll allerdings der Idealfall angenommen werden. Es wird also davon ausgegangen, dass die Trajektorie vollständig und kontinuierlich vorliegt. Dort wo eine Abweichung vom Idealfall eine Rolle spielt, wird entsprechend darauf hingewiesen.

Die Position des Objektes wird entsprechend des Anwendungsgebietes des Trackers zu Verfügung gestellt. Manche kamerabasierten Systeme arbeiten beispielsweise direkt auf den Koordinaten des Eingabebildes. Wird für das Tracken ein Laserscanner verwendet, wird die Objektposition auf einer Ebene meist in Polarkoordinaten angegeben. Durch Verwendung von Sensoren, die in der Lage sind, die Umgebung dreidimensional zu erfassen, ist es möglich, die Position der Objekte auch in einem dreidimensionalen, meist kartesischen Raum anzugeben. Werden mehrere unterschiedliche Sensoren miteinander kombiniert, muss im Rahmen der Sensorfusion ein gemeinsames Koordinatensystem verwendet werden. Dazu werden die unterschiedlichen Koordinatensysteme durch eine geeignete Projektion in das gemeinsame Koordinatensystem überführt.

Um nicht jeden dieser Sonderfälle einzeln betrachten zu müssen, wird als weitere Vereinfachung angenommen, dass die Position der dynamischen Objekte im dreidimensionalen Raum $\mathbf{s}_i^T = (x_i, y_i, z_i)$ oder auch auf einer zweidimensionalen Ebene $\mathbf{s}_i^T = (x_i, y_i)$ bekannt ist. Für die Darstellung der Trajektorie wird diese als Vektor angesehen, dessen Komponenten die Positionen \mathbf{s}_i für n Zeitschritte enthalten:

$$\mathcal{T} = (\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{n-1}). \quad (3.1)$$

Dabei entspricht die Reihenfolge der Komponenten dem zeitlichen Auftreten der Positionen. Für alle Repräsentationsformen, die in diesem

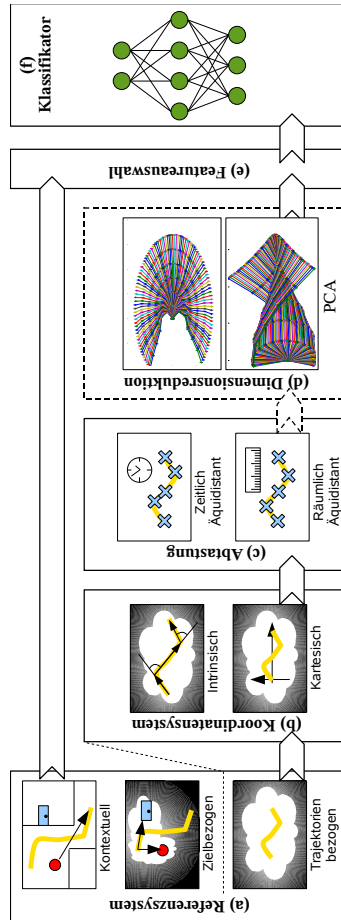


Abbildung 3.4.: Überblick über unterschiedliche Repräsentationsmöglichkeiten für Bewegungstrajektorien: Die Wahl des richtigen Referenzsystems (a), des Koordinatensystems (b) und der Abtastung (c) ist hochgradig aufgabenabhängig. Zusätzlich kann eine Dimensionsreduktion (d) durchgeführt werden. In (d) ist dabei der Einfluss von zwei der Eigentrajektorien auf die rekonstruierte Trajektorie dargestellt. Die richtige Kombination wird nach einer Auswahl durch Feature-Selektions-Algorithmen z. B. einem neuronalen Netz als Klassifikator präsentiert.

Kapitel vorgestellt werden, wird jeweils die Überführung in die entsprechende Variante, ausgehend von dieser Notation, vorgestellt.

Sämtliche Vorverarbeitungsschritte müssen in der Lage sein, online auf dem Assistenzroboter zu funktionieren. Dabei stellt allerdings nicht die Anforderung an die Rechenzeit das Hauptproblem dar. Viel wichtiger ist, dass sämtliche Methoden mit der Tatsache zurechtkommen müssen, dass während der Erkennungsaufgabe nur ein Teil der Trajektorie bekannt sein wird. Dies gilt insbesondere auch für den Klassifikator, der über das Interaktionsinteresse entscheiden soll. Damit nicht erst die gesamte Trajektorie bekannt sein muss, wird dieser mit Teilstücken der Trajektorie trainiert.

Solche Teilstücke werden erzeugt, in dem ein Fenster über die Trajektorie geschoben wird und nur der Inhalt des Fensters betrachtet wird. Dies ist in Abbildung 3.5 durch das farbliche Hervorheben des Endes der Trajektorie angedeutet. Bei der Bestimmung der Fenstergröße muss allerdings immer ein Kompromiss eingegangen werden. Auf der einen Seite braucht man eine adäquate Länge, um ausreichend viele Informationen zur Verfügung stellen zu können. Auf der anderen Seite muss die Erkennungsleistung schnell genug erfolgen können.

Für die vorgestellten Experimente ist die Größe des Fensters fest gewählt wurden. Um die Größe des Fensters entsprechend der bearbeiteten Daten zu bestimmen, können unterschiedliche Methoden aus dem Bereich der Zeitreihenanalyse verwendet werden. Eine genauere Beschreibung solcher Methoden findet sich in Abschnitt 5.1.

3.2.1. Referenzsystem

Die Situation des potenziellen Nutzers wird mehr oder weniger explizit bzw. implizit in den vom Tracker stammenden Daten wiedergespiegelt. Um unterschiedliche Sichtweisen auf diese Situation zu erlauben, sind unterschiedliche Formen der Repräsentation notwendig. Diese unterschiedlichen Formen werden erzeugt, in dem die in Abbildung 3.5 angedeuteten Referenzsysteme verwendet werden. Je nach gewählten Referenzsystem können vorher implizit vorhandene Informationen explizit zur Verfügung gestellt werden. An dieser Stelle sollen drei mögliche

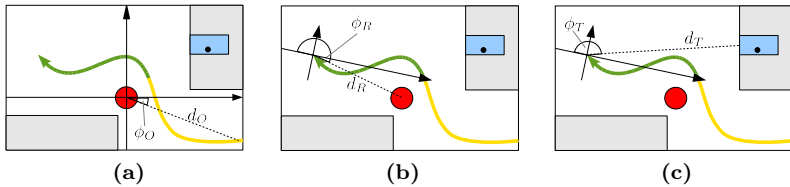


Abbildung 3.5.: Referenzsysteme:

Kontextuelle Informationen werden durch das spezifizieren des (a) Ausgangspunktes der Trajektorie (ϕ_O, d_O) zur Verfügung gestellt. Der Ursprung des Polarkoordinatensystems liegt dabei an der Position des Roboters (in der Grafik zeigt der Roboter nach rechts). Die (b) Position des Roboters (ϕ_R, d_R) oder (c) anderer potentieller Zielobjekte (z. B. Türen) (ϕ_T, d_T) werden dagegen in einem auf der Trajektorie zentrierten Koordinatensystem zur Verfügung gestellt.

Referenzsysteme eine Rolle spielen. Die Wahl wurde dabei so getroffen, dass sie für das angeführte Beispiel sinnvoll ist.

3.2.1.1. Kontextuelle Repräsentation

Die Erste unter den betrachteten Repräsentationsformen ist das kontextuelle Referenzsystem (siehe Abbildung 3.5 (a) bzw. 3.4 (a) oben). Dessen Ziel ist es, Informationen über die Umgebung zu modellieren. Für das gewählte Beispiel ist es sinnvoll, den Ausgangspunkt der betrachteten Trajektorie in einem Polarkoordinatensystem zu beschreiben, welches seinen Ursprung an der Position des Roboters besitzt. Dazu wird der erste detektierte Punkt (ϕ_O, d_O) der gesamten Trajektorie in Relation zur Position des Roboters angegeben. Somit wird explizit die Information zur Verfügung gestellt, aus welcher Richtung sich die Person dem Roboter genähert hat. Die meisten in der Literatur zu findenden Ansätze belassen es bei dieser globalen Form der Repräsentation. Im Folgenden sollen mögliche Alternativen aufgezeigt werden.

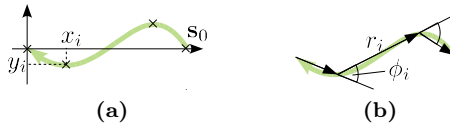


Abbildung 3.6.: Trajektorienbezugssystem:

Die linke Darstellung (a) zeigt einen Punkt auf der Trajektorie (x_i, y_i) , der in einem kartesischen Koordinatensystem liegt, während die rechte Abbildung (b) seine intrinsische Repräsentation veranschaulicht.

3.2.1.2. Zielbezogene Repräsentation

Insbesondere bei Fragestellungen zur Erkennung der Intention des potentiellen Nutzers erweist es sich als zweckmäßig, den Ursprung des Koordinatensystems zur Position der Person zu verschieben. Die Position möglicher Ziele können somit, wie in Abbildung 3.5 (b) und (c) gezeigt, relativ zur Person angegeben werden. Auch an dieser Stelle werden wieder Polarkoordinaten benutzt. Somit kann direkt durch Nutzung der Winkelinformation bestimmt werden, ob sich die Person auf das entsprechende Ziel zubewegt. Die möglichen Ziele sind nicht immer nur für die Aufgabenstellung relevante Objekte, wie beispielsweise der Assistenzroboter (ϕ_R, d_R) als Interaktionspartner (vergleiche Abbildung 3.5). Darüberhinaus ist es sinnvoll, z. B. die Lage der Türen bzw. Durchgänge (ϕ_T, d_T) zu kennen. Die entsprechenden Abstände d und der Winkel ϕ werden relativ zur Bewegungsrichtung angegeben. Im übertragenen Sinne versetzt man das System durch die zielbezogene Beschreibung in die Lage des Nutzers, um seine Absichten zu bestimmen.

3.2.1.3. Trajektorienbezugssystem

Um die Bewegung losgelöst von ihrer Lage in der Umgebung zu betrachten, wird ein ausschließlich auf die Trajektorie ausgerichtetes Bezugssystem verwendet. Somit wird der Fokus auf die Charakteristik der Bewegung gelegt. Diese Charakteristik wird unter anderem durch die Geschwindigkeit, die Beschleunigung und die Geradlinigkeit beschrie-

ben. Durch die Geradlinigkeit wird beispielsweise ausgesagt, wie zielgerichtet sich eine Person auf den Roboter zubewegt. An dieser Stelle sollen zwei unterschiedliche Möglichkeiten vorgestellt werden, wie man eine Bewegung in einem Trajektorienbezugssystem repräsentieren kann (siehe Abbildung 3.4 (b)).

Für die erste Möglichkeit (Abbildung 3.6 (a)) wird der Ursprung eines kartesischen Koordinatensystems in den zuletzt beobachteten Punkt der Trajektorie gelegt. Die Orientierung des Koordinatensystems wird durch den korrespondierenden ersten Punkt \mathbf{s}_0 im aktuell betrachteten Fenster bestimmt. Die x -Achse wird dabei so gelegt, dass dieser erste Punkt auf ihr liegt. Die y -Achse liegt entsprechend senkrecht dazu, was zumindest im zweidimensionalen Fall zu einer eindeutigen Lage des Koordinatensystems führt.

Die zweite Möglichkeit, welche in Abbildung 3.6 (b) zu sehen ist, kann mit einer Wegbeschreibung verglichen werden. Es wird beschrieben, wie man von einem Punkt auf der Trajektorie zum nächsten kommt. Es werden also die intrinsischen Eigenschaften der Trajektorie wiedergegeben. Dazu gibt man für jeden Punkt auf der Trajektorie den Abstand r_i und den Winkel ϕ_i zum vorhergehenden Punkt an. Der Winkel wird dabei entsprechend der jeweiligen Bewegungsrichtung angegeben.

3.2.2. Abtastung

Bewegungsdaten, die von einem Trackingsystem aufgezeichnet wurden, sind zwangsläufig bereits abgetastet. Die Abtastung orientiert sich dabei in den seltensten Fällen an der beobachteten Bewegung. Vielmehr richtet sich die Abtastung nach der Aktualisierungsrate der beteiligten Sensoren oder der Rechenzeit der benutzten Algorithmen. Aber welche Art der Abtastung ist für die Untersuchung von Bewegungen am besten geeignet? An dieser Stelle sollen zwei prinzipielle Strategien, wie in Abbildung 3.4 (c) bzw. Abbildung 3.7 angedeutet, gegenübergestellt werden.

Der Vollständigkeit halber sei noch erwähnt, dass sich sämtliche Betrachtungen zum Abtasten der Bewegung an das Shannonsche Abtasttheorem halten müssen. So ist garantiert, dass der originale Verlauf rekonstruiert werden kann, demnach also keine Information verloren geht.

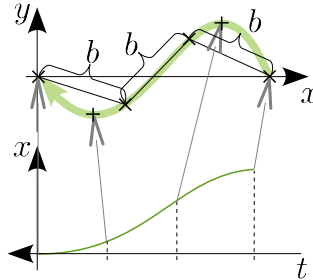


Abbildung 3.7.: Abtastmöglichkeiten:

Im oberen Graph wird die Trajektorie an vier Stellen (\times), die jeweils denselben Abstand b zueinander haben, abgetastet. Die untere Darstellung zeigt die x -Koordinate der Trajektorie gegen die Zeit aufgetragen. Äquidistante Abtastung bezüglich der t -Achse führt dann zu anderen Abtastpunkten auf der Trajektorie (+).

Betrachtet man natürliche Bewegungen, z. B. die eines Menschen, steckt die wesentliche Information ohnehin in den Anteilen niedriger Frequenz.

3.2.2.1. Räumlich äquidistante Abtastung

Das räumlich orientierte Abtasten hat den Vorteil, dass die Anzahl der Datenpunkte für gleiche zurückgelegte Wegstrecken konstant bleibt. Wenn eine Person beispielsweise an einer Stelle stehen bleibt, wird nur ein Datenpunkt für diese Stelle verwendet, anstelle einer Menge von Punkten an derselben räumlichen Position. Auf diese Weise wird die Form der Trajektorie in den Vordergrund gestellt und Variationen bezüglich der Geschwindigkeit eliminiert. Dies ist vergleichbar mit der Art und Weise, wie Hidden Markov Modelle (HMM) mit der Beobachtung einer Trajektorie umgehen.

3.2.2.2. Zeitlich äquidistante Abtastung

Als Alternative zur räumlich äquidistanten Abtastung ist es möglich, die Bewegung zu vorgegebenen festen Zeitintervallen zu repräsentieren. Üb-

licherweise liefern Trackingsystem diese Art der Abtastung. Durch diese Form der Repräsentation wird die Geschwindigkeit der Person implizit angegeben. Je nach gewählter Aufgabenstellung ist daher die eine oder andere Form des Abtastens von Vorteil.

3.2.3. Koordinatentransformationen

Durch die unterschiedlichen Repräsentationsformen ist es möglich, dass manche der Informationen korreliert vorliegen. Findet man diese Korrelationen, ist es möglich, die Dimensionalität der Eingangsdaten des Klassifikators stark zu reduzieren. Ein Verfahren, welches in solchen Fällen üblicherweise Anwendung findet, ist die Hauptkomponentenanalyse (HKA oder PCA). Sie liefert die Eigenvektoren und die dazugehörigen Eigenwerte der Kovarianzmatrix der Daten. Geometrisch interpretiert wird durch die Eigenvektoren ein neuer Raum aufgespannt. Die Lage der Datenpunkte in diesem Raum ergibt sich dabei so, dass eine minimale Korrelation zwischen den einzelnen Dimensionen des neuen Raumes vorliegt. Die Eigenwerte beschreiben die Größe der Varianz entlang der entsprechenden Eigenvektoren.

Im Falle von Bewegungsdaten wird jede durch die Fensterung ausgeschnittene Teiltrajektorie als einzelner Datenpunkt aufgefasst. Die berechneten Eigenvektoren werden entsprechend als Eigentrajektorien bezeichnet. In Abbildung 3.8 sind beispielhaft die ersten vier Eigentrajektorien dargestellt. Diese wurden aus der intrinsischen Repräsentation zeitlich äquidistant abgetasteter Daten berechnet. Es ist gut zu erkennen, dass sich, ähnlich wie bei der Fourieranalyse, ein System aus Bassettschwingungen herausgebildet hat.

3.2.4. Auswahl der geeigneten Repräsentation

Auch wenn in diesem Kapitel nur ein kleiner Teil der möglichen Repräsentationsformen von Bewegungsabläufen betrachtet wird, ist zu erkennen, dass es eine Vielzahl an Möglichkeiten gibt, eine beobachtete Bewegung einem Klassifikator zur Verfügung zu stellen. Es bleibt allerdings noch immer offen, welche dieser Möglichkeiten die richtige Wahl

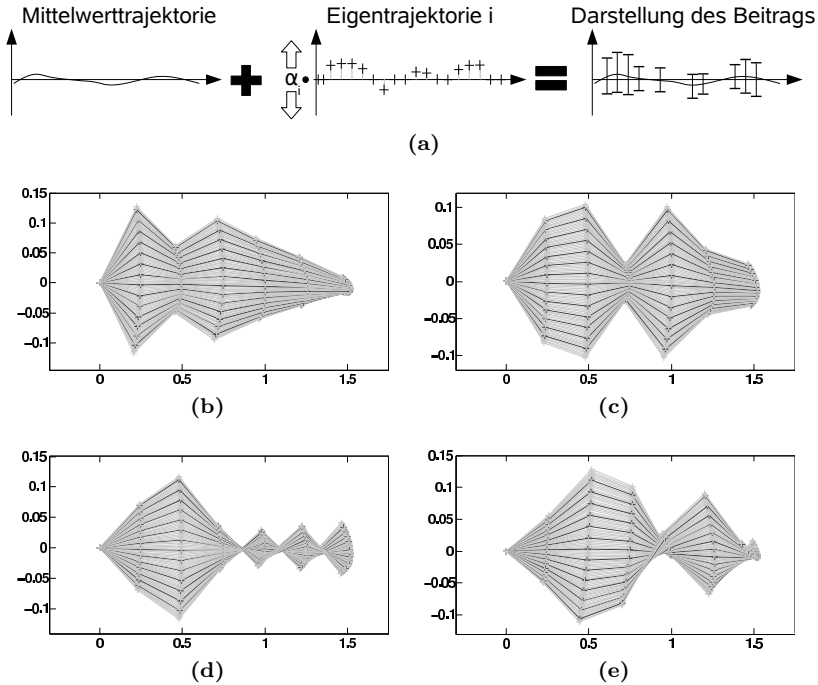


Abbildung 3.8.: Eigentrajektorien:

Die Abbildungen (b)-(e) zeigen den Beitrag der ersten vier Eigentrajektorien, wie in Abbildung (a) angedeutet. Dazu wurden bei der linearen Überlagerung alle Gewichte, außer dem Gewicht α_i der betrachteten Eigentrajektorie, zu null gewählt. Das Gewicht α_i der betrachteten Eigentrajektorie wurde von $-2\sqrt{e_i}$ bis $2\sqrt{e_i}$ variiert, wobei e_i der entsprechende Eigenwert, also die Varianz in der jeweiligen Dimension darstellt.

für die gegebene Aufgabenstellung ist bzw. wie man diese sinnvoll bestimmt.

Methoden, die dies ermöglichen, werden unter dem Begriff Feature Subset Selection (oder kurz Feature Selection, siehe Abbildung 3.4 (e)) dazu benutzt, aus einer großen Anzahl Features, jene auszuwählen, die den größten Informationsgehalt besitzen. Im Falle der Bewegungsrepräsentation wird jede Variable bzw. Dimension als einzelnes Feature angesehen.

Die Feature Selection Methoden helfen nicht nur, die geeignetste Repräsentation zu finden. Das Reduzieren der Daten kommt auch dem Klassifikator zugute. Durch eine große Menge an ausgewählten Features, ist entsprechend auch die Eingangsdimension des Klassifikators hoch. Damit beispielsweise ein Neuronales Netz einen solchen hochdimensionalen Eingaberaum hinreichend beschreiben kann, ist im Allgemeinen eine entsprechende Menge Neuronen notwendig. Um ein solch großes Netz entsprechend trainieren zu können, müssen genügend Trainingsbeispiele vorhanden sein, damit das Netz nicht zur Überanpassung neigt. Ebenso würde der Zeitaufwand für den Trainingsprozess stark ansteigen.

Feature Selection Methoden werden üblicherweise in Filter-, Wrapper- und Embedded-Ansätze unterschieden (vergleiche dazu [Guyon et al., 2006]). Sowohl die Wrapper- als auch die Embedded-Methoden bauen auf dem für das Problem notwendigen Lernalgorithmus auf, um die Features zu evaluieren. Dagegen sind die Filter-Methoden unabhängig von dem benutzten Klassifikator. Für das hier vorgestellte Problem wurde sich aufgrund dieser Unabhängigkeit für den Filter-Ansatz entschieden. Prinzipiell wird die Relevanz eines jeden Features durch statistische Maße, wie dem Pearson-Korrelationskoeffizient, der Fisher Diskriminanzanalyse oder auch andere informationstheoretische Maße, bestimmt.

In diesem speziellen Fall wurde die, beispielsweise in [Guyon et al., 2006, Kapitel 6] beschriebene Transinformation (Mutual Information, MI) gewählt. Die Transinformation drückt die Korrelation zwischen einem Feature x_i , z. B. der x -Koordinate des ersten Punktes auf der Trajektorie, und dem Ziel Y aus. Das Ziel wird dabei durch die Klassifikationsinformation beschrieben. In diesem konkreten Fall war die Klasseninformation dass Interaktionsinteresse, d. h. ob eine Person die Absicht hat mit dem Roboter zu interagieren. Dabei bedeuten große Werte der Transinformation eine starke Korrelation, während kleine einen Hinweis auf

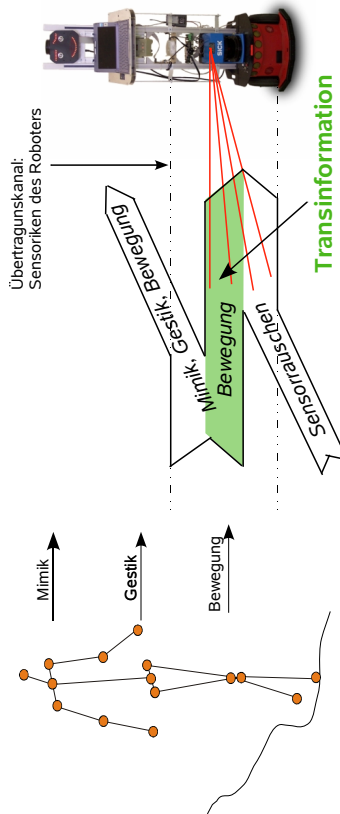


Abbildung 3.9.: Interpretation der Transinformation:

Die sich nähernde Person teilt über ihre Gestik, ihren Gesichtsausdruck und ihre Bewegungstrajektorie unbewusst ihr Interaktionsinteresse mit. Dabei wird in dieser Arbeit nur die Bewegungstrajektorie untersucht. Somit gehen Gesten und Gesichtsausdrücke verloren. Während des Aufzeichnens der Trajektorie werden neben dem üblichen Meßrauschen auch für die eigentliche Klassifikation irrelevante Bewegungen mit aufgezeichnet. Mit Hilfe der Mutual Information kann der entsprechend relevante Teil extrahiert werden. (Angepasst übernommen aus [Ober, 2007])

für das Problem irrelevante Features geben. Die Transinformation MI wird nach folgender auf der Kullback-Leibler Divergenz aufbauenden Gleichung berechnet:

$$MI(x_i, Y) = \int_{x_i} \int_Y P(x_i, Y) \log \frac{P(x_i, Y)}{P(x_i)P(Y)} dx_i dY \quad (3.2)$$

Eine grafische Veranschaulichung dieser Gleichung mit der Interpretation der Bedeutung der Transinformation im Kontext des gewählten Szenarios, findet sich in Abbildung 3.9.

Die Berechnung der Wahrscheinlichkeitsdichten stellt für den kontinuierlichen Fall eine große Herausforderung dar. In der praktischen Implementierung wird sie daher durch einen histogramm-basierten Ansatz approximiert. Dadurch vereinfachen sich auch die Integrale zu Summen über die Bins der Histogramme. Der interessierte Leser sei hierbei auf [Scott, 1992] verwiesen, um weitere Details zur Abschätzung mit Histogrammen oder anderen Ansätzen zu erfahren.

Die resultierende Transinformation gibt die Relevanz des jeweiligen Features in Bezug auf das entsprechende Klassifikationsproblem. Allerdings werden keine Aussagen über mögliche Redundanzen getroffen. Um diesen Nachteil zu überwinden, ist es möglich, zusätzlich die paarweise Transinformation $MI(x_i, x_j)$ für jede mögliche Featurekombination x_i und x_j zu berechnen. In diesem Fall würde ein hoher Wert eine Redundanz andeuten.

Einen Schritt weiter geht die Verbund-Transinformation (Joint Mutual Information, JMI), da sie zusätzlich ein weiteres Problem eliminiert. Es kann passieren, dass zwei Features, wenn sie allein auftreten, keinen Beitrag für das Klassifikationsproblem liefern. Treten sie allerdings gemeinsam auf, helfen sie das Problem zu entscheiden. Ein einfaches Beispiel hierfür ist das XOR-Problem. Die Verbund-Transinformation berechnet die Relevanz einer Menge $X = \{x_1, \dots, x_n\}$ von Features bezüglich eines Ziels Y :

$$JMI(X, Y) = \int_{X, Y} P(x_1, \dots, x_n, Y) \log \frac{P(x_1, \dots, x_n, Y)}{P(x_1, \dots, x_n)P(Y)} \quad (3.3)$$

Basierend auf diesen Maßen kann schließlich eine Teilmenge aus allen möglichen Repräsentationen ausgewählt werden. Mit Hilfe der ausgewählten Teilmenge wird dann der Klassifikator trainiert. Im folgenden Abschnitt soll dies einmal praktisch gezeigt werden.

3.2.5. Experimentelle Untersuchungen

Für die Durchführung der Experimente wurde die im Fachgebiet Neuroinformatik und kognitive Robotik zur Verfügung stehende Roboterplattform HOROS verwendet. Sie ist mit einer Vielzahl an Sensoren ausgestattet und ermöglicht so den Einsatz eines multimodalen Personentrackers. Dieser soll allerdings nicht weiter ausgeführt werden. Der interessierte Leser sei auf die Veröffentlichung [Scheidig et al., 2006] verwiesen. Mit Hilfe des Personentrackers werden alle 100 ms die Positionen der Personen im Umfeld von 6 m erfasst und aufgezeichnet.

Der Roboter wurde, wie auch in [Müller et al., 2008] beschrieben, an 15 nicht aufeinanderfolgenden Tagen für jeweils 6 Stunden an drei unterschiedlichen öffentlichen Plätzen aufgestellt. Durch das Variieren der Positionen ist es möglich, eine ortsunabhängige Aussage des Interaktionsinteresses abzuleiten. Um Interaktionen sowohl mit neuen als auch mit bekannten Nutzern zu ermöglichen, wurde der Roboter mit einer Applikation ausgestattet, die es ihm ermöglicht, als Informationsterminal zu fungieren. Dabei wurden alltägliche Informationen über die Speisepläne, Busfahrpläne und besondere Veranstaltungen zur Verfügung gestellt. Darüber hinaus wurde auch eine audio-basierte Geschlechterschätzung zur Unterhaltung der Nutzer integriert.

Um auf sich aufmerksam zu machen, wurde eine zufällige Begrüßung abgespielt, sobald eine Person vom Roboter detektiert wurde. Das Spektrum der Begrüßungen enthielt sowohl freundliche als auch provokative Äußerungen. Nachdem die Personen den Bereich um den Versuchsaufbau wieder verlassen hatten, wurden sie von den Experimentatoren um das Ausfüllen eines Fragebogens gebeten. Auf Details zu den Fragebögen, insbesondere zur Verwertbarkeit aus psychologischer Sicht wird in [Ober, 2007] eingegangen. Durch die Auswertung der Fragebögen war es möglich, für die jeweilige Person die aufgezeichneten Trajektorien mit einer Vielzahl an Informationen zu unterlegen, die dann als Trainings-

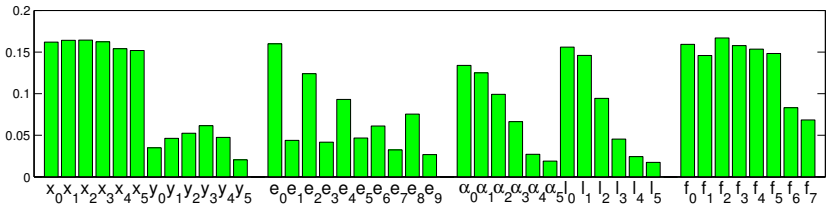


Abbildung 3.10.: Transinformation der Repräsentationsformen: Die Abbildung zeigt die Transinformation für die aus den Repräsentationen gewonnenen einzelnen in Abschnitt 3.2.1.3 und 3.2.3 vorgestellten Features bezüglich des Interaktionsinteresses. Die Spalten 1 bis 12 zeigen die neu abgetasteten Datenpunkte in der kartesischen Repräsentation (zuerst die x -Komponenten, dann die y -Komponenten). In den Spalten 13 bis 22 sind die Hauptkomponenten e_i für die kartesische Repräsentation dargestellt. Die intrinsische Repräsentation (Winkel α und Länge l der einzelnen Segmente) in den Spalten 23 bis 34. Die dazugehörigen Hauptkomponenten f_i in den Spalten 35 bis 42.

daten für den Klassifikator verwendet werden konnten. Exemplarisch sei in diesem Abschnitt das Interaktionsinteresse herausgegriffen.

Die so aufgezeichneten Trajektorien wurden in alle in diesem Kapitel vorgestellten Repräsentationsformen überführt. Anschließend wurde mit Hilfe der beschriebenen Feature Selection Algorithmen eine Teilmenge ausgewählt. In den durchgeführten Auswertungen wurden jeweils die möglichen Alternativen miteinander verglichen. Zuerst wurde die zeitlich äquidistante Abtastung der räumlichen Variante gegenübergestellt. Entgegen den Erwartungen hat dabei der Verlust der Geschwindigkeitsinformationen bei den räumlich äquidistanten Daten zu keiner Verschlechterung geführt. Beide Methoden haben ähnlich gut abgeschnitten. Um das Vorgehen besser illustrieren zu können, soll sich daher nur auf eine Methode, nämlich die räumlich äquidistante Abtastung konzentriert werden. Bei der Auswahl wurde eine Variante herausgegriffen, bei der besonders interessante Effekte auftreten. Weitere Details zu den hier nicht erwähnten Ergebnissen können in [Ober, 2007] nachvollzogen werden.

Abbildung 3.10 zeigt die in Abschnitt 3.2.4 beschriebene Transinformation zwischen den jeweiligen Features in der entsprechenden Repräsentation und dem Interesse der Person zu interagieren. Vergleicht man in der Abbildung die Resultate des kartesischen Trajektorienbezugs-systems (x_i, y_i) mit dem intrinsischen (α_i, l_i) (siehe Abschnitt 3.2.3), stellt man fest, dass in der kartesischen Repräsentationsform ähnlich viel Information über das Nutzerinteresse enthalten ist. Die Summe der Transinformationswerte für die entsprechenden Repräsentationen ist etwa gleich groß, wie man an der Fläche der Balken erkennen kann. Interessant ist allerdings, dass sich bei der kartesischen Repräsentation diese Information in der x -Koordinate befindet, während die y -Koordinate weniger Informationsgehalt hat. Bei der intrinsischen Kodierung sammelt sich die Information in den ersten Winkeln α_i bzw. Segmentlängen l_i . Für diese Methode scheint demnach die Beschreibung des Anfangs des Trajektorienteilstückes auszureichen.

Trägt man, wie in Abbildung 3.11 geschehen, die paarweise Transinformation der einzelnen Features gegeneinander ab, erkennt man bei dem kartesischen Trajektorienbezugs-system (Abbildung 3.11 (a)), dass ein hoher Grad an Redundanz bei dieser Darstellung auftritt. Schaut man sich beispielsweise den Wert (x_4, x_5) an, erkennt man eine große paarweise Transinformation und kommt zu dem Schluss, dass nur einer der beiden Werte benötigt wird, um das Klassifikationsproblem zu entscheiden. Im Vergleich dazu besitzt das Paar (y_0, y_4) einen niedrigen Wert, was auf eine geringere Redundanz hindeutet. Durch die auf die Trajektorie ausgerichtete Lage des Koordinatensystems wird die x -Koordinate in erster Linie durch die Vorwärtsbewegung der Person verursacht und besitzt daher eine größere Korrelation. Die y -Koordinate hingegen repräsentieren die Form bzw. Krümmung der Trajektorie und sind folglich weniger stark korreliert.

Die intrinsische Darstellung (Abbildung 3.11 (c)) weist eine geringere Redundanz auf, als die kartesische. Man beachte dabei die unterschiedliche Skalierung der Abbildung 3.11 (a) und (c). Auffällig ist jedoch, dass jeweils die ersten Winkel bzw. die Segmentlängen mehr paarweise Transinformation und somit eine geringere Unabhängigkeit aufweisen.

Um lineare Redundanzen zu beseitigen, kann die in Abschnitt 3.2.3 beschriebene Hauptkomponentenanalyse benutzt werden. In Abbildung 3.10 ist die Transinformation der signifikantesten Hauptkomponenten,

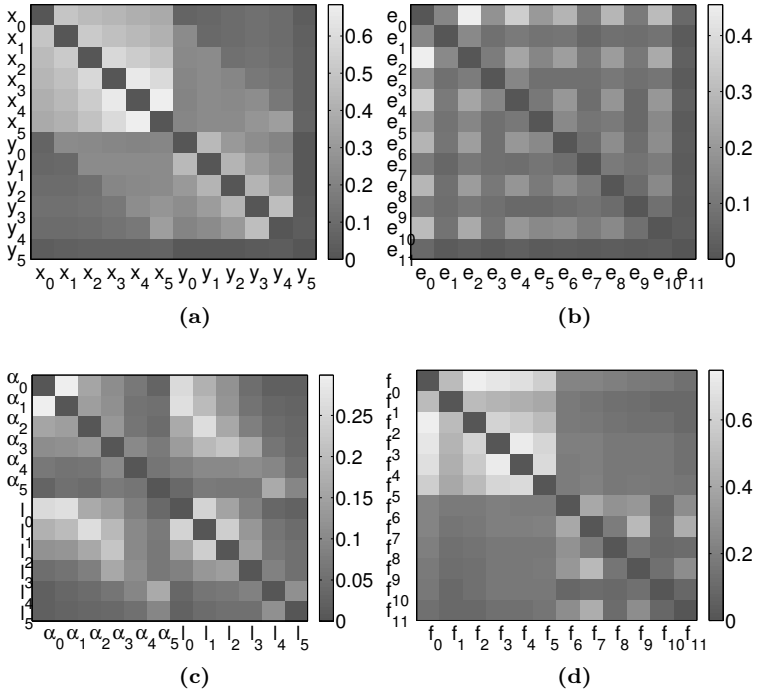


Abbildung 3.11.: Paarweise Transinformation der Repräsentationen: Die Abbildung zeigt die paarweise Transinformation für das kartesische Trajektorienbezugssystem (a), die dazugehörigen Hauptkomponenten (b), die intrinsische Repräsentation (c) und auch dazu die entsprechenden Hauptkomponenten (d). Helle Grauwerte stehen für höhere Werte und bedeuten, dass beide Features sich viel Informationsgehalt teilen, sie also zumindest teilweise redundant sind. Ein dunkler Grauwert bedeutet niedrige Werte und steht somit für Unabhängigkeit der betrachteten Features.

genauer gesagt der Projektionen der ursprünglichen Daten auf die Eigenvektoren e_i bzw. f_i dargestellt. Dabei handelt es sich bei e_i um die Hauptkomponenten des kartesischen Trajektorienbezugssystems und bei f_i um die der intrinsischen Repräsentation.

Anders als bei den meisten hier nicht dargestellten Experimenten tritt im Falle der Hauptkomponentenanalyse für das kartesische Trajektorienbezugssystem ein interessanter Effekt auf. Üblicherweise geht man davon aus, dass die Dimensionsreduktion am besten durchgeführt wird, in dem man sich auf die Dimensionen mit der größten Varianz konzentriert. Da die Hauptkomponenten in den Abbildungen nach der Varianz sortiert sind, würde man also einen steten Abfall der Transinformation erwarten, so wie es bei den Hauptkomponenten der intrinsischen Trajektorienbeschreibung der Fall ist. In Abbildung 3.10 ist allerdings zu beobachten, dass nur jede zweite Hauptkomponente einen guten Beitrag zur Klassifikation liefert. Dazu kommt, dass in Abbildung 3.11 (b) beobachtet werden kann, dass eben diese Hauptkomponenten trotz Dekorreliertheit paarweise redundant sind.

Die paarweise Transinformation der Hauptkomponenten, die aus den intrinsischen Daten erzeugt wurden, zeigen, dass diese Repräsentationsform eine sehr hohe Redundanz aufweist. Die Redundanz ist sogar wesentlich größer, als die der ursprünglichen Darstellung. Offensichtlich ist die Nutzung der Hauptkomponenten in diesem speziellen Fall keine sinnvolle Wahl.

Der Klassifikator wurde exemplarisch jeweils mit der diskutierten kartesischen Darstellung und der entsprechenden in den Eigentrajektorienraum überführten Form trainiert. Dabei wurden sowohl die komplette Featuremenge als auch die auf Basis der Transinformation bzw. paarweisen Transinformation ausgewählten Teilmengen trainiert. Durch die Auswahl von Teilmengen reduziert sich die Komplexität des als Klassifikator verwendeten Multilayer Perzeptrons. Damit sinkt der Trainingsaufwand und das Netz ist besser in der Lage zu generalisieren. Die in Tabelle 3.1 dargestellten Ergebnisse zeigen, dass die unterschiedlichen Teilmengen vergleichbare Ergebnisse, wie bei den vollständigen Daten, liefern. Der dargestellte Klassifikationsfehler ergibt sich als Durchschnitt über alle 4526 Teiltrajektorien des gesamten Testdatensatzes. Die Tabelle ist in vier Blöcke unterteilt, bei denen jeweils die Anzahl der selektierten Features sinkt. Block I stellt die maximal verfügbare Anzahl

an Features pro Repräsentationsform dar. Die ersten vier Hauptkomponenten in Block III liefern die besten Ergebnisse.

Bei der Betrachtung des Klassifikationsfehlers ist allerdings zu beachten, dass jeweils die gesamte Trajektorie von Beginn der Aufzeichnung für die Evaluierung verwendet wurde. Bei den meisten Trajektorien lässt sich allerdings am Beginn kaum ein bzw. gar kein Unterschied feststellen. Außerdem ist die Klassenzugehörigkeit für die gesamte Trajektorie definiert worden, da durch die Fragebögen nicht ermittelt werden konnte, ab wann die Person Interesse an einer Interaktion entwickelt hat.

3.2.6. Fazit

Die Wahl des richtigen Referenzsystems hat großen Einfluss auf die Klassifikationsaufgabe. Wenn beispielsweise der Roboter als Informationssystem allgemeine Informationen, wie dem Wetterbericht, zur Verfügung stellen würde, spielte die Richtung, aus der die Person gekommen ist, keine Rolle. Deshalb würde ein zielbezogenes Referenzsystem für die Entscheidung, eine Interaktion beginnen zu wollen, ausreichend sein. Wenn der Roboter allerdings Informationen über den Platz, an dem er sich befindet, präsentiert, wie z. B. in einem Museum, wären ankommende Personen die sinnvollere Zielgruppe für eine Interaktion. In einem solchen Fall ist die kontextuelle Repräsentation notwendig.

In diesem Abschnitt wurde daher nicht nur eine kleine Auswahl unterschiedlicher Repräsentationsformen vorgestellt. Darüberhinaus wurden Methoden untersucht, mit denen die für das jeweilige Problem geeignete Repräsentationsform ausgewählt werden kann. Auf diese Weise wurden erste Schritte unternommen, um die Vorverarbeitung von Bewegungsinformationen von der Entscheidung eines Experten zu entkoppeln. Allerdings kann, wie in [Ober, 2007] gezeigt wurde, die Entscheidung je nach gewähltem Informationsmaß unterschiedlich ausfallen.

Durch die Wahl des Erkennungsaspektes aus Abschnitt 1.2 sollte auch verdeutlicht werden, dass die methodische Untersuchung von Bewegungsmustern als sinnvoll einzuschätzen ist. Es konnte dabei gezeigt werden, dass mit unterschiedlichen Repräsentationsformen auch unterschiedliche Klassifikationsgenauigkeiten erreicht werden können. Die

	Features	Anzahl Neuronen		Fehler
		Inputschicht	Hidden-schicht	
I	$(x_0, y_0, \dots, x_5, y_5)$	12	16	0,2252
	(e_0, \dots, e_{11})	12	16	0,2454
	$(x_0, y_0, \dots, x_5, y_5)$	12	10	0,291
	(e_0, \dots, e_{11})	12	10	0,2362
II	$(x_1, y_1, \dots, x_3, y_3)$	6	10	0,3186
	(e_0, \dots, e_5)	6	10	0,2354
	(e_6, \dots, e_{11})	6	10	0,286
III	(x_2, y_2, x_3, y_3)	4	10	0,2057
	(e_0, \dots, e_3)	4	10	0,193
	(e_4, \dots, e_7)	4	10	0,2768
IV	(x_2, y_2)	2	10	0,1957
	(y_2, x_5)	2	10	0,230
	(e_0, e_1)	2	10	0,230
	(e_2, e_3)	2	10	0,280

Tabelle 3.1.: Klassifikationsfehler des zweischichtigen Multi-Layer Perzeptron: *Es wurden jeweils unterschiedliche Repräsentationsformen bzw. Teilmengen der Repräsentationsformen (siehe erste Spalte) verwendet, um das Netz zu trainieren. Klassifiziert wird das Nutzerinteresse für das betrachtete Trajektorienfenster. Die binäre Ausgabe des Netzes wird mit der bekannten Klasseninformation verglichen, um den Netzwerkfehler zu bestimmen. Die besten Ergebnisse werden bei der Auswahl der ersten vier Hauptkomponenten erzielt (hervorgehoben).*

Nutzung der Transinformation und der Verbundtransinformation ermöglicht es, eine geeignete Auswahl zu treffen.

Für die weitere Forschung ist es wichtig, dass der hier angedeutete Prozess vollständig automatisiert und in überschaubarer Zeit ablaufen kann. So wird es einem lernenden Robotersystem möglich sein, für neu gestellte Aufgaben aus der ihm zur Verfügung stehenden Sammlung an Repräsentationsformen die passende auszuwählen.

Um bei den weiteren Untersuchungen die Anzahl der durchzuführenden Experimente überschaubar zu halten, wird, wenn nicht anders erwähnt, auf einer Repräsentation im kartesischen Raum gearbeitet. In Kapitel 5 wird diese zusätzlich in den sogenannten Einbettungsraum überführt. Da diese Überführung prinzipiell nur eine bessere Repräsentation des Trajektorienverlaufes ermöglichen soll, gibt es keinerlei Einschränkungen, ob eine kartesische Repräsentation zu Grunde liegen muss.

In [Troje, 2002] wird der zeitliche Verlauf einer PCA-basierten Repräsentation weiter untersucht. Auf diese Weise ist es möglich ohne Expertenwissen ein Modell zu erstellen, welches den zeitliche Ablauf einer menschlichen Bewegung generieren kann. Da die PCA nur eine lineare Transformation der ursprünglichen Daten vornimmt, überrascht das erfolgreiche Abschneiden dieser Repräsentationsform nicht. Es könnte daher in weiterführenden Arbeiten die in den nachfolgenden Kapiteln vorgestellten Prädiktionsverfahren beispielsweise auch auf Basis einer PCA-basierten Repräsentation untersucht werden.

3.3. Bewertung der Ähnlichkeit von Trajektorien

Nachdem im letzten Abschnitt unterschiedliche Repräsentationsformen als Vorverarbeitungsstufe für die im Fokus dieser Arbeit stehenden Algorithmen näher beleuchtet wurden, soll sich dieser Abschnitt der Wahl eines geeigneten Ähnlichkeits- oder Distanzmaßes für Trajektorien widmen (siehe Abbildung 3.12). Eine Aussage über die Ähnlichkeit von beobachteten Trajektorien ist an vielen Stellen entscheidend. Beispielsweise wird in Kapitel 5 eine Gruppe von Verfahren vorgestellt, die auf einer Nachbarschaftsauswahl basieren. Um beurteilen zu können, wel-

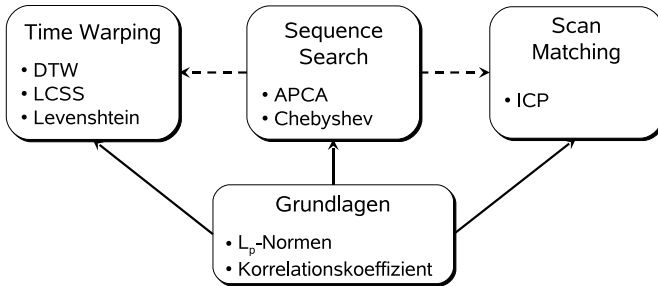


Abbildung 3.12.: Untersuchte Ähnlichkeitsmaße:

Die Abbildung zeigt die in diesem Kapitel vorgestellten Ähnlichkeitsmaße. Dabei bilden klassische Verfahren die Grundlage. Eine Erweiterung stellen die Sequence Search Algorithmen dar. Time Warping und Scan Matching bieten zusätzlich ein Möglichkeit, die Art der Unähnlichkeit zu bestimmen. Die gestrichelten Pfeile sollen andeuten, dass die Sequence Search Ähnlichkeitsmaße die Distanzmaße in den beiden anderen Kategorien austauschen können.

che Trajektorien im Sinne der Ähnlichkeit von Bewegungen benachbart sind, spielt das Distanzmaß eine entscheidende Rolle. Die Überführung in das entsprechende Ähnlichkeitsmaß, soll hierbei allerdings nicht weiter beleuchtet werden. In den meisten Fällen wird der Zusammenhang zwischen Distanz D und Ähnlichkeit S wie folgt definiert: $S(A, B) = e^{-D(A, B)}$. Eine weiterführende Diskussion findet sich beispielsweise in [Shepard, 1987].

Darüberhinaus kann ein geeignetes Verfahren zum Bestimmen der Ähnlichkeit auch dazu eingesetzt werden, mögliche Zusammengehörigkeiten, z. B. Interaktionen, zwischen den Objekten, deren Trajektorien untersucht werden, zu ermitteln. Dazu wird die Annahme getroffen, dass Objekte mit beginnender Interaktion eine gemeinsame Eigenschaft in ihren Bewegungen teilen. Am einfachsten lässt sich diese Idee nachvollziehen, wenn man sich vorstellt, dass beispielsweise eine Person einen Ball fängt und man die Trajektorien des Balles und der Hand betrachtet. Zu Anfang bewegen sich beide Objekte aufeinander zu. Ab dem Zeitpunkt des Fangens sind die Trajektorien des Balles und der Hand

identisch. Es ist allerdings auch denkbar, dass die gemeinsamen Bewegungen durch andere Bewegungen überlagert werden. Betrachtet man z. B. die Trajektorien der Hand und des Kopfes einer Person, teilen beide die Gesamtbewegung des Körpers. Darüberhinaus enthält insbesondere die Trajektorie der Hand noch eine intrinsische Bewegung, die keinen Schluss auf einen gemeinsamen Ursprung der Bewegungen zulässt.

Nicht immer ist es sinnvoll, die Ähnlichkeit zweier Bewegungen auf eine einzige skalare Größe zu reduzieren. Vielmehr würden manche Anwendungen durch eine Aussage über die Art der Unähnlichkeit stark profitieren. In dem speziellen Fall der Bewegungstrajektorien spielen dabei die Verschiebung in Raum Δx und Zeit Δt , die Rotation φ und die Skalierung s eine Rolle. Man könnte also ganz allgemein eine Art Distanzmaß definieren, welches für jede dieser drei Möglichkeiten eine gesonderte Aussage treffen kann:

$$\begin{pmatrix} \Delta x \\ \Delta t \\ \varphi \\ s \end{pmatrix} = \underset{\Delta t, \Delta x, \varphi, s}{\operatorname{argmin}} \left(d(\mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S} \cdot \mathcal{T}_1, \mathcal{T}_2) \right). \quad (3.4)$$

Dabei sollen die Matrizen \mathbf{T} , \mathbf{R} und \mathbf{S} andeuten, dass die Trajektorie \mathcal{T}_1 den entsprechenden Translationen, Rotationen bzw. Skalierungen unterzogen wird.

Bei der Suche nach einem geeigneten Maß, wird die Trajektorie auch in diesem Anhang als Zeitreihe aufgefasst. Damit handelt es sich also um eine Funktion, die jedem Zeitpunkt eine bestimmte Position zuordnet. Es kann somit auf Verfahren zur Bestimmung der Ähnlichkeit von Funktionen zurückgegriffen werden. Lässt man darüberhinaus die zeitliche Dimension außer acht, handelt es sich bei einer Trajektorie um eine Punktmenge. Damit kann auch in diesem Forschungsfeld nach geeigneten Verfahren gesucht werden.

Um einschätzen zu können, welche Verfahren zur Bestimmung der Ähnlichkeit geeignet sind, wurde die folgende Auswahl getroffen. Dazu gehören zum Einen Verfahren, dir nur eine Aussage über das Ausmaß der Unähnlichkeit treffen:

- \mathcal{L}_p -Normen

- Korrelationskoeffizient
- Adaptive Piecewise Constant Approximation
- Chebyshev-Approximation

Zum Anderen werden Verfahren betrachtet, die eine Aussage über die Art der Unähnlichkeit zulassen:

- Time Warping Verfahren
 - Dynamic Time Warping
 - Longest Common Subsequence
 - Levenshtein-Distanz
- Scan Matching Algorithmen am Beispiel des Iterative Closest Point Verfahrens

Diese Verfahren wurden auf Daten getestet, die künstlich bestimmten Änderungen unterworfen wurden. Dabei wurden Transformationen, wie Rotation, Skalierung und Translation angewendet. Somit kann beobachtet werden, welche Änderung der Bewegungstrajektorie welche Auswirkungen auf das Ähnlichkeitsmaß hat.

Um an dieser Stelle den Lesefluss nicht vom eigentlichen Kern der Arbeit abzulenken, befindet die Beschreibung der einzelnen Verfahren und der entsprechenden Experimente in Anhang C. Im folgenden Fazit sind die Ergebnisse der Experimente nur kurz zusammengefasst.

3.3.1. Fazit

Je nach Problemstellung muss sich für das entsprechende Verfahren entschieden werden. Tabelle 3.2 soll dabei als Entscheidungshilfe dienen. In der Tabelle wird in den ersten drei Spalten verdeutlicht, welche Dimension der Daten durch das jeweilige Verfahren verarbeitet werden können: eindimensionale Daten (1D), dreidimensionale Daten (3D) und Daten mit beliebiger Dimensionalität (nD). Die Unterscheidung zwischen Daten mit beliebiger Dimensionalität und mit drei Dimensionen ist, wie zu erkennen, wegen des untersuchten Scan Matching Verfahrens getroffen worden. Da hier mit jeder zusätzlichen Dimension die Anzahl der mög-

	1D	3D	nD	benötigt gleichlange Daten	Anzahl Parameter	Aussage über die Art der Unähnlichkeit
\mathcal{L}_p -Normen	×	×	×	×	0	–
Korrelationskoeffizient	×	E	E	×	0	–
APCA	×	–	–	*	1	–
Chebyshev-Approximation	×	–	–	–	1	–
Dynamic Time Warping	×	×	×	–	0	E
Longest Common Subsequence	×	×	×	–	>1	E
Levenshtein-Distanz	×	×	×	–	1	E
Scan Matching Algorithmus (ICP)	E	×	E	–	0	×

* beliebige Punkteanzahl, aber zeitliche Länge muss identisch sein

×	erfüllt
E	Erweiterung notwendig
–	nicht erfüllt

Tabelle 3.2.: Einordnung der Verfahren:

In der Tabelle sind die unterschiedlichen Ähnlichkeitsverfahren zusammengefasst und jeweils ihre Anforderung an die Daten dargestellt. Dabei wird in den ersten drei Spalten verdeutlicht, welche Dimension der Daten durch das jeweilige Verfahren verarbeitet werden können: eindimensionale Daten (1D), dreidimensionale Daten (3D) und Daten mit beliebiger Dimensionalität (nD).

lichen Transformationen stark ansteigt, muss je nach Problemstellung entschieden werden, ob sich der Mehraufwand im Sinne der Rechenzeit lohnt.

Die nächste Spalte gibt Auskunft, ob die Daten für den Vergleich bezüglich ihrer Ähnlichkeit dieselbe Anzahl an Datenpunkten besitzen müssen. Da im Falle der \mathcal{L}_p -Normen jeweils der paarweise Abstand berechnet wird, ist das Vergleichmaß auf gleichlange Teilstücke angewiesen. Eine Ausnahme bildet die Adaptive Piecewise Constant Approximation (APCA) (nachzulesen in Abschnitt C.4). Bei diesem Verfahren ist es nur notwendig, dass beide Trajektorien denselben Zeithorizont aufweisen, die Anzahl der Punkte spielt keine Rolle.

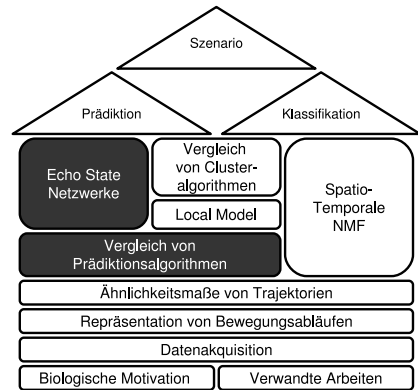
In der vorletzten Spalte ist die Anzahl der Parameter dargestellt. Um eine Vergleichbarkeit der berechneten Ähnlichkeiten zu ermöglichen, sollten keine Parameter existieren bzw. die existierenden Parameter konstant gelassen werden.

Die letzte Spalte scheint redundant zu der im letzten Abschnitt getroffenen Unterteilung, bezüglich einer Aussage über die Art der Unähnlichkeit zu sein. Durch sie soll allerdings gezeigt werden, dass die Time Warping Verfahren (Dynamic Time Warping, Longest Common Subsequence und Levenshtein Distanz) eigentlich als Distanzmaße im klassischen Sinne, die einen skalaren Wert liefern, konzipiert worden. Durch die Art der Bestimmung dieser Distanz lassen sie aber auch eine Aussage über Art der Unähnlichkeit zu.

Da unter den untersuchten Ähnlichkeitsmaßen keines als klarer Favorit überzeugen konnte, soll in den folgenden Kapiteln jeweils auf die Euklidische Distanz zurückgegriffen werden. Dabei kann aber insbesondere für weiterführende Untersuchung an den jeweiligen Stellen ein Austausch in Betracht gezogen werden.

Kapitel 4

Echo State Netzwerke



Nachdem sich das letzte Kapitel mit der Vorverarbeitung von Trajektorien beschäftigt haben, sollen in diesem und in den folgenden Kapiteln Verfahren zur weiteren Verarbeitung vorgestellt und untersucht werden. Dabei wird insbesondere auf die Anforderungen des in Abschnitt 1.2 vorgestellten Szenarios eingegangen werden, wobei in diesem Kapitel der Navigationsaspekt und damit die Prädiktion die größte Rolle spielen wird. Darüberhinaus wird in den folgenden Kapiteln die Klassifikation von Bewegungsabläufen betrachtet.

Gegenstand dieses Kapitels bilden dabei die Echo State Netzwerke. Sie stellen im Bereich neuronaler Netze ein etabliertes Verfahren zur Verarbeitung von Zeitreihen dar. Dabei stehen Problemstellungen wie Klassifikation und Prädiktion im Vordergrund. Zur Erfüllung des Navigations- und Interaktionsaspektes wird sich auf die Fähigkeit zur Prädiktion fokussiert.

Der Grundgedanke ist dabei, wie in Abschnitt 3.2 bereits dargelegt, die Interpretation der Bewegung als Zeitreihe. Somit lassen sich eine Vielzahl an Verfahren aus dem Bereich Zeitreihenanalyse¹ für diese Zwecke verwenden. Dabei bleibt zu untersuchen, inwieweit die bestehenden

¹Time Series Analysis

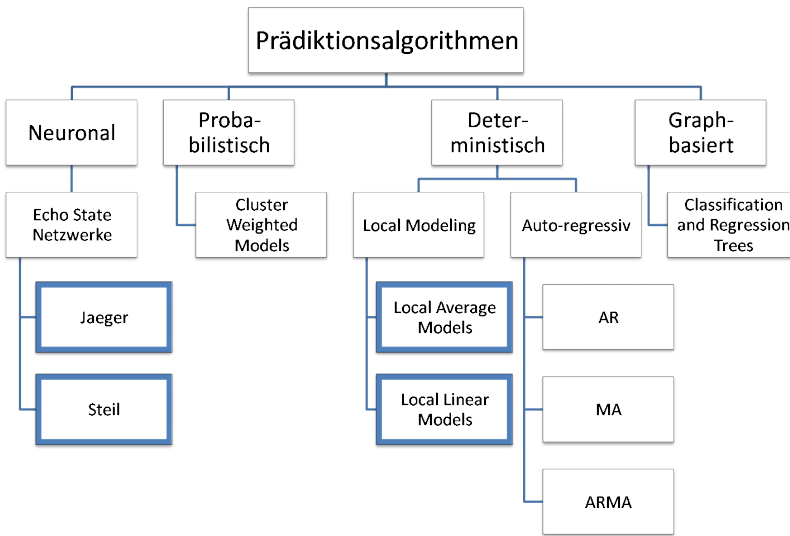


Abbildung 4.1.: Untersuchte Prädiktionsalgorithmen:

Die Abbildung zeigt die in [Strauß, 2008] untersuchten Prädiktionsalgorithmen. Die hervorgehobenen Verfahren haben sich als am vielversprechendsten erwiesen und werden daher in diesem und im folgenden Kapitel näher beleuchtet.

Verfahren auf das relativ spezielle Einsatzfeld angepasst werden können. Die entsprechenden Ergebnisse und Verbesserungsvorschläge wurden im Detail in [Hellbach et al., 2008] veröffentlicht.

4.1. Vergleich von Prädiktionsalgorithmen

In diesem Kapitel und in Kapitel 5 werden Verfahren zur Prädiktion von Zeitreihen vorgestellt. Diese wurden allerdings nicht beliebig, sondern auf Basis extensiver Experimente ausgewählt. Im Rahmen der Experimente wurden unterschiedliche Verfahren zum ersten mal bezüglich ihrer Fähigkeit zur Prädiktion von realen Bewegungen untersucht. Da-

bei wurde darauf geachtet, dass möglichst ein breites Spektrum an Forschungsfeldern abgedeckt wird. In Abbildung 4.1 sind die im Rahmen der Arbeit [Strauß, 2008] untersuchten Methoden und die Einordnung in ihre Forschungsfelder dargestellt.

In den vergleichenden Untersuchungen wurden vier unterschiedliche Algorithmen gegenübergestellt: Echo State Netzwerke, Local Models, Cluster Weighted Models und Autoregressive Modelle. Untersucht wurden dabei sowohl die Anforderungen an die Rechenzeit, als auch die Prädiktionsgüte. Bei der Prädiktionsgüte wurden sowohl Kurzzeit-, als auch Langzeitprädiktionen betrachtet. Bei allen Tests haben die Echo State Netzwerke sehr gute Ergebnisse erzielt. Kurz hinter den Echo State Netzwerken sind die Local Models einzuordnen. Beide Verfahren wurden für weitere Untersuchungen ausgewählt, die in diesem und im folgenden Kapitel aufgezeigt sind.

Auf die experimentellen Untersuchungen und die Beschreibung der weiteren Verfahren wird zum besseren Verständnis nicht weiter eingegangen. Details dazu finden sich in Anhang D. Ergebnisse dieser Arbeit und weiterführende Untersuchungen wurden in [Hellbach et al., 2008; Hellbach et al., 2009b] publiziert.

4.2. Aufbau von ESN

Es ist allgemein bekannt, dass Neuronale Netzwerke gut geeignet zur Approximation von Funktionen sind. Für die spezielle Aufgabe der Zeitreihenprädiktion kamen in den letzten Jahren besonders Echo State Netzwerke (ESN) zum Einsatz [Jaeger und Haas, 2004; Steil, 2007]. Sie stellen eine besondere Form der Rekurrenten Neuronalen Netze dar. Ihre besondere Anordnung der Neuronen macht sie mit einem Netzwerk aus Neurooszillatoren vergleichbar.

4.2.1. Funktionsweise

Echo State Netzwerke habe einige spezifische Eigenschaften, durch die sie sich von normalen Neuronalen Netzen unterscheiden. Die Hidden-

²Die Diplomarbeit wurde vom Autor im Rahmen des Projektes betreut.

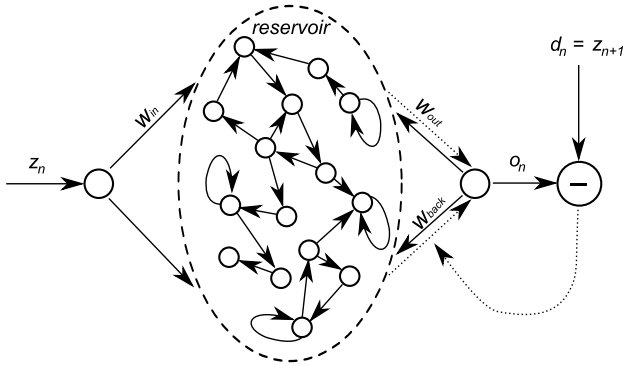


Abbildung 4.2.: Aufbau eines Echo State Networks:

Alle Gewichte werden zufällig initialisiert und sind - bis auf w_{out} - auch im Verlauf des Lernvorgangs unveränderlich. Zum Lernen von w_{out} wird zum jeweiligen Trainingsinput am Ausgabeneuron (und somit auch zur Berechnung der Feedbackkanten) die korrekte Ausgabe d_n erzwungen. (Entnommen aus [Strauß, 2008]²)

schicht besteht aus Neuronen, die, wie in Abbildung 4.2 angedeutet, zufällig auf beliebige Weise verknüpft sind. Es wird allerdings gefordert, dass nur eine geringe Anzahl an Verknüpfungen existiert. Dadurch entstehen in der Hiddenschicht Strukturen, die man sich als Teilnetze vorstellen kann, in denen es, wie es in Abbildung 4.3 angedeutet ist, zu Kreisschlüssen kommen kann. Jedes dieser Teilnetze wird, wie es bei Neurooszillatoren der Fall ist, zum Schwingen angeregt und generiert ein entsprechendes Signal in Richtung der Ausgabeschicht. Da hier sozusagen aus einem Vorrat an Schwingungen geschöpft werden kann, wird die Hiddenschicht auch als Reservoir bezeichnet.

Durch die angesprochenen zyklischen Verknüpfungen werden vergangene Zustände wie ein Echo im Netzwerk hin und hergeworfen. Durch dieses rekurrente Verhalten muss jeweils nur der aktuell betrachtete Punkt auf der Trajektorie dem Netz als Eingabe präsentiert werden.

Die Gewichte, die sich innerhalb des Reservoirs befinden, die also die Neuronen der Hiddenschicht miteinander verknüpfen, werden üblicherweise in der einer Matrix \mathbf{W}^r zusammengefasst. Wie bereits angedeutet

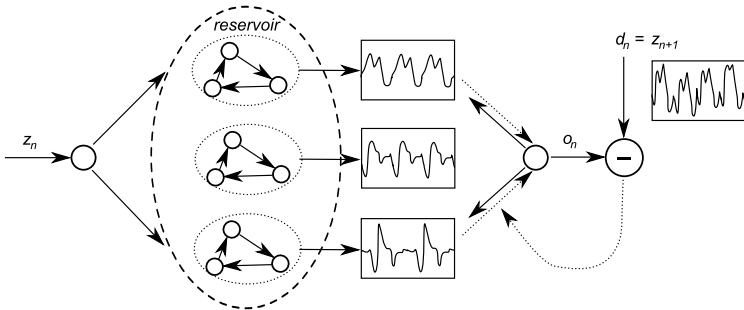


Abbildung 4.3.: Funktionsweise eines ESN:

Durch die niedrige Konnektivität im Reservoir entstehen weitestgehend unabhängige Neuronencluster, welche jeweils eigene charakteristische Outputs aufweisen. Diese werden kombiniert und die Gewichte w_{out} derart angepasst, dass das Originalsignal nachgebildet werden kann. (Entnommen aus [Strauß, 2008])

werden zur Initialisierung der Gewichte und somit an die Matrix \mathbf{W}^r bestimmte Bedingungen gestellt. In [Jaeger und Haas, 2004] wird erwähnt, dass der Spektralradius³ $spec$ der Matrix ein wichtiger Parameter des Netzwerkes ist und auf keinen Fall größer als 1.0 sein darf, um die Stabilität des Netzes nicht zu gefährden. Interessanterweise wird in [Steil, 2007] diskutiert, dass ein Spektralradius in der Nähe von 1.0 bzw. sogar geringfügig größere Werte als 1.0 zu besseren Ergebnissen führen. Nichtsdestotrotz kann diese Forderung auf einfache Weise erfüllt werden. Von der zufällig initialisierten Matrix \mathbf{W}^r wird der Spektralradius $spec_{old}$ bestimmt. Danach wird die Matrix mit dem Verhältnis aus gewünschtem und errechnetem Spektralradius skaliert.

$$\mathbf{W}^r \leftarrow \frac{spec}{spec_{old}} \cdot \mathbf{W}^r \quad (4.1)$$

Darüberhinaus existiert die Forderung nach Spärlichkeit der Matrix \mathbf{W}^r . Dadurch wird die bereits geschilderte geringe Verknüpfung des Reservoirs erreicht, wodurch es zur Ausbildung von teilnetzartigen Strukturen

³Der Spektralradius einer Matrix ist definiert als ihr größter Eigenwert.

kommt. Sowohl in [Jaeger und Haas, 2004] als auch in [Steil, 2007] wird vorgeschlagen, etwa 80% der Gewichte auf null zu setzen.

Eine weitere Besonderheit, die die Echo State Netze von den meisten anderen neuronalen Netzen abhebt, ist die Tatsache, dass ausschließlich die Gewichte \mathbf{w}_{out} zwischen Hiddenschicht und Ausgabeschicht trainiert werden. Alle anderen Gewichte, d. h. Input- \mathbf{w}_{in} , Reservoir- \mathbf{W}^r und Feedbackgewichte \mathbf{w}_{back} werden nur am Anfang zufällig gewählt und bleiben sonst konstant.

4.2.2. Training und Anwendung

Um Echo State Netzwerke zu trainieren, werden sie wie bereits beschrieben, zufällig initialisiert. Diesem zufällig initialisierten Netz wird dann Schritt für Schritt die zu trainierende Zeitreihe, d. h. die Trajektorie präsentiert. Die internen Zustände der Neuronen der Hiddenschicht \mathbf{r}_n zum Zeitschritt n ergeben sich aus der gewichteten Summe der vorangegangenen Zustände \mathbf{r}_n , der aktuellen Beobachtung \mathbf{s}_n und der rückgeführten Ausgabe des Netzes im vorangegangenen Zeitschritt \mathbf{o}_{n-1} :

$$\mathbf{r}_n = f_h(\mathbf{W}^r \cdot \mathbf{r}_{n-1} + \mathbf{w}_{in} \cdot \mathbf{s}_n + \mathbf{w}_{back} \cdot \mathbf{o}_{n-1}) \quad (4.2)$$

Die Bezeichnung der Gewichte kann in Abbildung 4.2 nachvollzogen werden. Die in der Gleichung als f_h bezeichnete Funktion beschreibt das Ausgabeverhalten der einzelnen Neuronen. Das Definieren einer unteren bzw. oberen Schranke durch die Ausgabefunktion verhindert bis zu einem gewissen Grad, dass sich die Werte im Reservoir aufschaukeln können. Üblicherweise wird in der Hiddenschicht auf die Fermifunktion oder den hyperbolischen Tangens zurückgegriffen. Die Berechnung der Ausgabe des Netzes, d. h. also des prädizierten Zeitschrittes $\mathbf{o}_n = \mathbf{s}_{n+1}$ berechnet sich dann einfach als:

$$\mathbf{o}_n = f_o(\mathbf{w}_{out} \cdot \mathbf{r}_n) \quad (4.3)$$

Für die Ausgabefunktion f_o der Ausgabeschicht kann wie in der Hiddenschicht ebenso eine sigmoidale Funktion gewählt werden. Da in der Ausgabeschicht das Begrenzen auf eine obere bzw. untere Schranke nicht

so wichtig ist, wie in der Hiddenschicht, kann auch eine lineare Ausgabefunktion gewählt werden.

Für das eigentliche Training, also das Adaptieren der Gewichte \mathbf{w}_{out} , wird als erstes eine Matrix \mathbf{R} aufgestellt. Diese Matrix enthält für jeden Zeitschritt $k = i, \dots, n$ die Zustände \mathbf{r}_k der Neuronen des Reservoirs als Spalteneintrag. Mit Hilfe der Matrix \mathbf{R} werden durch lineare Regression die Gewichte \mathbf{w}_{out} zwischen Hiddenschicht und Ausgabeschicht angepasst:

$$\mathbf{w}_{out} = (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{R}^T \mathbf{T} \quad (4.4)$$

Dabei stellt \mathbf{T} eine Aneinanderreihung der erwarteten Prädiktionen \mathbf{t}_k zum jeweiligen Zeitpunkt k dar. Das Training sollte allerdings nicht direkt mit dem zufällig initialisierten Netz begonnen werden. Das Netz braucht schließlich eine gewisse Zeit i , in der sich die Eingaben im Reservoir ausbreiten können und sich die Teilschwingungen aufgebaut haben. Erst wenn diese sogenannte transiente Phase überwunden ist, kann mit dem Training begonnen werden.

Wie am Aufbau der Matrix \mathbf{R} und somit der Gewichte \mathbf{w}_{out} zu erkennen ist, basiert das Training nur auf einer einzelnen Trajektorie. Dabei handelt es sich um den bereits beobachteten Teil der zu präzisierenden Trajektorie.

Nachdem die Gewichte auf die aktuell beobachtete Trajektorie adaptiert wurden, kann das Netz für die Prädiktion der weiteren Bewegung genutzt werden. Dazu wird dem Netz, nachdem die Aktivitäten im Reservoir zurückgesetzt wurden, die Bewegung nochmals von Anfang an präsentiert. Wenn der letzte beobachtete Wert erreicht ist, entspricht die Ausgabe des Netzes der Prädiktion um einen Zeitschritt. Um mehr als einen Zeitschritt präzisieren zu können, wird jeweils der letzte prädiizierte Wert dem Netzwerk wieder als Eingabe zugeführt. In den im Folgenden dargestellten Experimenten wurden so bis zu 800 Zeitschritte in die Zukunft prädiiziert.

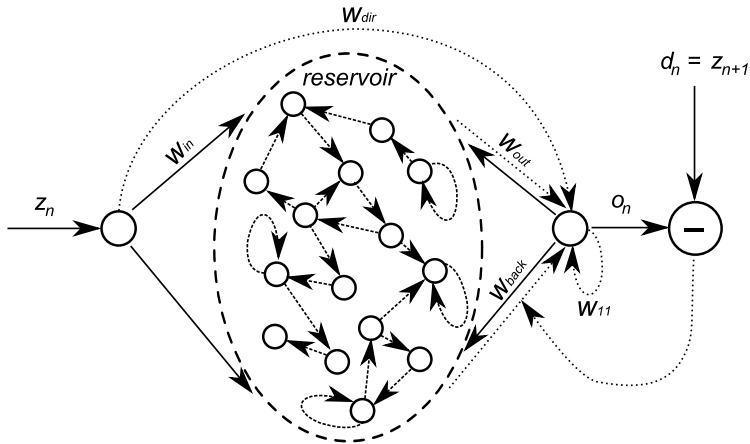


Abbildung 4.4.: Aufbau des modifizierten ESNs:

Als neue lernbare Kanten wurden w_{dir} und w_{11} hinzugefügt. Ein weiterer Unterschied zum ursprünglichen ESN (Abbildung 4.2) ist, dass das Reservoir nicht mehr statisch ist. Genau genommen sind es die Parameter der Fermi-Ausgabefunktionen der Reservoirneuronen die durch Intrinsic Plasticity angepasst werden. (Entnommen aus [Strauß, 2008])

4.2.3. Erweiterungen

In [Steil, 2007] wurden einige zusätzliche Erweiterungen zu den Standard Echo State Netzwerken eingeführt. Dazu gehören ein online-fähiger Lernalgorithmus und die sogenannte intrinsische Plastizität, durch die das Reservoir mit angepasst werden kann. Außerdem wurden, wie in Abbildung 4.4 zu sehen, zusätzliche Gewichte eingeführt. Zum Einen wird eine direkte Kante \mathbf{w}_{dir} zwischen Eingabe- und Ausgabeschicht hinzugefügt. Dadurch hat die Ausgabe die Chance, direkt auf den aktuellen Input reagieren zu können, und es entstehen keine Verzögerungen durch die Ausbreitung im Reservoir. Zum Anderen werden rekurrente Verbindungen \mathbf{w}_{11} an den Ausgabeneuronen vorgeschlagen.

Durch die intrinsische Plastizität kann das Reservoir flexibler auf die präsentierten Daten angepasst werden. Allerdings werden, anders als bei Neuronalen Netzen üblich nicht die Gewichte angepasst, sondern die Parameter der Ausgabefunktion. Im Falle von [Steil, 2007] handelt es sich um die Fermifunktion:

$$f_n(x) = \frac{1}{1 + e^{-a_n \cdot x - b_n}} \quad (4.5)$$

Angepasst werden für jedes Neuron n der Parameter a_n , der die Steilheit der Funktion bestimmt, und der Parameter b_n , die Funktion entlang der Abszissenachse verschiebt. Dadurch wird der Arbeitsbereich der Funktion so angepasst, dass die Neuronen in der Lage sind, tatsächlich Ausgaben zwischen 0 und 1 zu liefern und nicht in den asymptotischen Bereichen „festsitzen“. Das Adaptieren der Parameter der Fermifunktion erfolgt vor dem Training der Ausgabegewichte und sollte noch in der Einschwingphase des Reservoirs erfolgen. Ansonsten kann es passieren, dass das Netzwerk instabil wird.

Da man vom mathematischen Standpunkt die Parameter a_n und b_n auch als Teil der Gewichte interpretieren kann, hat eine Veränderung Auswirkungen auf den Spektralradius. Dies kann dazu führen, dass das Netz zu instabilen Zuständen tendiert.

Zusätzlich zu den in der Literatur bekannten Erweiterungen wurde in [Hellbach et al., 2008] eine selbstentwickelte Möglichkeit vorgestellt, die Tatsache, dass die Gewichte nicht angepasst werden, sinnvoll auszu-

nutzen. Durch die Vielzahl an Gewichten, die nicht trainiert werden, besitzen Echo State Netzwerke eine große Menge an Parametern, die im Vorfeld festgelegt werden müssen. Insbesondere durch die zufällige Initialisierung der Reservoirgewichte \mathbf{W}^r kommt es je nach Initialisierung zu unterschiedlich ausgeprägten Basisschwingungen. Damit darunter auch die sind, die für die aktuelle Beobachtung benötigt werden, muss die Anzahl der Neuronen im Reservoir entsprechend hoch gewählt sein. Gerade an dieser Stelle hilft also eine geschickte Initialisierung, die Anzahl der Neuronen und somit den Rechenaufwand zu reduzieren.

Um die für die aktuell beobachtete Bewegung am besten passende Initialisierung zu finden, werden mehrere Instanzen des Echo State Netzwerkes parallel trainiert. Für jede der Instanzen werden die Gewichte zufällig festgelegt. Während des Trainingsprozesses werden die Ausgaben des Netzes mit dem bekannten Verlauf der Trajektorie verglichen. Das Netz, welches dabei die besten Ergebnisse liefert, wird für die eigentliche Prädiktion verwendet [Hellbach et al., 2008]. Diese Art der Netzwerke wird für alle folgenden Experimente verwendet.

4.3. Experimentelle Evaluierung

Ziel der Prädiktion ist es, wie im Szenario in Abschnitt 1.2 beschrieben, dem Roboter zu ermöglichen, in Umgebungen mit dynamischen Hindernissen zu navigieren und mit Personen zu interagieren. Darüberhinaus soll die Interaktion beispielsweise auch eine Begrüßung per Handschlag beinhalten. Für das Beispiel der Begrüßung durch Handschlag, werden Prädiktionen der Bewegung im dreidimensionalen Raum benötigt. Um vergleichbare und nachvollziehbare Ergebnisse zu erzielen, wurden zur Evaluierung die in Abschnitt 3.1.1 präsentierten Realweltbewegungsdaten verwendet.

Aus den Bewegungsdaten 25 Zeitreihen zufällig ausgewählt. Dabei wurde darauf geachtet, dass von jedem Messpunkt (Kopf, Schulter etc.) und auch von jeder Bewegungsart (Laufen, Werfen, ...) mindestens eine Testzeitreihe verwendet wird. Somit kann gewährleistet werden, dass genügend Varianz in den Testdaten vorhanden, ist um die Ergebnisse verallgemeinern zu können.

Für die Experimente wurde im Durchschnitt ein Prädiktionshorizont von 500 Zeitschritten bei einer Gesamtlänge von ca. 1500 Zeitschritten verwendet. Dies entspricht einer Vorhersage von ca. 8.3 Sekunden, da die Daten mit 60 Hz aufgezeichnet wurden. Bei den meisten Trackingsystemen, wie Kalman- oder Partikelfiltern sind die Prädiktionen so ausgelegt, dass das System in der Lage ist für nur einige wenige Zeitschritte fehlerhafte Detektionen zu überbrücken. In solchen Fällen kann die Prädiktion nicht als zuverlässige Quelle für eine Langzeitprädiktion angesehen werden, wie sie beispielsweise für die Navigation oder die Begrüßung per Handschlag benötigt wird.

4.3.1. Referenzalgorithmen

Es wurden nicht nur die einzelnen Arten der Echo State Netzwerke miteinander verglichen, um die am besten geeignete Variante zu bestimmen. Es sollte darüberhinaus möglich sein, einschätzen zu können, in welchem Maße das Verwenden der Prädiktionen einen Vorteil bringt. Dazu wurden zwei unterschiedliche Referenzalgorithmen implementiert. Diese entsprechen den meist einfachen Annahmen für die Bewegungsmodelle in Kalman- oder Partikelfilter. Die erste Referenz stellte eine konstante Prädiktion dar. Dazu wird der letzte beobachtete Punkt auf der Trajektorie über den Prädiktionshorizont wiederholt. Als Zweites wird eine lineare Annahme getroffen. Im Prinzip entspricht dies einem Bewegungsmodell, welches eine konstante Geschwindigkeit annimmt. Für die vergleichenden Grafiken wurde jeweils der bessere der beiden Referenzalgorithmen verwendet und als rote Linie eingezeichnet.

4.3.2. Gütekriterien

Für den Vergleich der einzelnen Algorithmen ist es notwendig, dass das verwendete Fehlermaß auch einen Schluss auf die Anwendbarkeit für das definierte Szenario zulässt. Ziel ist es dabei, über einen relativ langen Zeitraum eine zuverlässige Prädiktion zu erhalten. Außerdem ist dabei wichtig zu wissen, ob auftretende Abweichungen möglicherweise erst am Ende des Prädiktionshorizonts entstehen. So kann beurteilt werden, ob die Algorithmen beispielsweise zu einer konstanten Abweichung neigen. Aus diesen Gründen wurden zwei Maße definiert. Beide basieren jeweils

auf der normalisierten mittleren quadratischen Abweichung⁴ entlang des N Zeitschritte langen Prädiktionshorizontes:

$$NMSE = \frac{1}{N \cdot \sigma^2} \sum_{i=1}^N (\mathbf{s}_i^{pred} - \mathbf{s}_i^{orig})^2 = \frac{1}{N} \sum_{i=1}^N NMSE_i \quad (4.6)$$

Dabei bezeichnet \mathbf{s}_i^{pred} den prädizierten Bewegungsablauf und \mathbf{s}_i^{orig} die tatsächliche Bewegung. Die verwendete Varianz σ entspricht dabei der größten Varianz aller Raumdimensionen der Originaltrajektorie. Bei dieser Wahl wird davon ausgegangen, dass Dimensionen mit großen Änderungen eine wichtigere Rolle bei der Prädiktion spielen.

Aus Gleichung 4.6 werden zwei Varianten abgeleitet. Bei der ersten handelt es sich um den Kurzschrittfehler (KF). Dieser ist verantwortlich, zu prüfen, ob die Prädiktion nach den ersten Schritten bereits sinnvolle Werte liefert, oder ob beispielsweise ein prinzipieller Fehler, wie eine konstante Verschiebung auftritt.

$$KF = \frac{\sum_{p=1}^{N_{KF}} \frac{1}{p} NMSE_p}{\sum_{p=1}^{N_{KF}} \frac{1}{p}} \quad (4.7)$$

Dabei wird die Länge des Prädiktionshorizonts relativ klein gewählt. Im Beispiel der verwendeten Experimentaldaten, bei denen Prädiktionen von maximal 500 Zeitschritten untersucht werden, werden in Formel 4.7 nur die ersten $N_{KF} = 75$ Zeitschritte betrachtet. Dies entspricht bei der verwendeten Aufzeichnungsrate von 60 Hz einer Prädiktion von 1.25 s.

Auf der anderen Seite wird durch den Langschrittfehler LF der gesamte Prädiktionshorizont betrachtet. Außerdem werden weiter in der Zukunft liegende Werte durch entsprechende Gewichtung weniger stark abgeschwächt.

$$LF = \frac{\sum_{p=1}^N \frac{1}{\sqrt{p}} NMSE_p}{\sum_{p=1}^N \frac{1}{\sqrt{p}}} \quad (4.8)$$

⁴Normalized Mean Square Error (NMSE)

Zur Erstellung der Grafiken wurde der mittlere Fehler und die Varianz aller Testdaten errechnet. Dabei sind zur Mittelung bzw. Varianzberechnung die logarithmierten Werte des Kurzschritt- bzw. Langschrittfehlers verwendet worden:

$$KF_{log} = \frac{1}{k} \sum_{i=1}^k \log_{10} KF_i \quad (4.9)$$

bzw.

$$D_{KF} = \sqrt{\frac{1}{k} \sum_{i=1}^k (\log_{10} KF_i - KF_{log})^2} \quad (4.10)$$

Somit werden negative Ausreißer abgeschwächt und besonders gute Prädiktionen bevorzugt. Zum Verständnis sei hier ein kleines Beispiel aufgeführt. Angenommen es existieren zwei Algorithmen, die folgende Fehler bei den Trajektorien \mathcal{T}_0 bis \mathcal{T}_3 liefern:

Algorithmus 1	\mathcal{T}_0	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	Mittelwert	Varianz
absolut	0,1	0,1	0,1	1000	250	433
logarithmisch	-1	-1	-1	3	0	1,7

Algorithmus 2						
absolut	\mathcal{T}_0	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	Mittelwert	Varianz
absolut	10	10	10	10	10	0
logarithmisch	1	1	1	1	1	0

Algorithmus 1 zeigt im Allgemeinen bessere Ergebnisse und scheitert nur an der Trajektorie \mathcal{T}_3 . Dagegen scheint Algorithmus 2 einen prinzipiellen Fehler zu begehen. An dieser Stelle wird deutlich, dass das logarithmische Maß die bessere Entscheidung ist. Bei der Interpretation der Diagramme ist zu beachten, dass die eingezeichnete Varianz die logarithmierte Varianz der Daten ist.

4.3.3. Ergebnisse

Im Gegensatz zu dem in Anhang D präsentierten allgemeinen Vergleich der Algorithmen soll dieser Abschnitt dazu genutzt werden, die einzelnen Varianten der Echo State Netzwerke zu beleuchten. Wie bereits in Abschnitt 4.2.3 erwähnt existieren in der Literatur im wesentlichen zwei unterschiedliche Arten an Echo State Netzwerken, die im Folgenden als Jäger-Netze [Jaeger und Haas, 2004] und als Steil-Netze [Steil, 2007] bezeichnet werden.

Für beide Varianten wurden, wie ebenfalls in Abschnitt 4.2.3 diskutiert, mehrere Netze parallel trainiert und jeweils das beste für die Prädiktion ausgewählt. Als Anzahl wurden 3 und 10 parallele Netzwerkinstanzen gewählt. In den Grafiken sind die Ergebnisse für die am besten abschneidende Parameterkombination dargestellt. In [Jaeger und Haas, 2004] wird vorgeschlagen, dass die Anzahl der Neuronen im Reservoir bis zu 2000 Stück betragen soll. Durch die parallele Instanziierung kann die Anzahl der Neuronen auf 15 bis 250 Neuronen reduziert werden. Bei 3 parallelen Netzwerken liegt die Anzahl von $3 \cdot 250 = 750$ immer noch deutlich unter diesem Wert. Somit kann nicht nur die Anwendungs-, sondern auch die Trainingsphase hinsichtlich der Rechenzeit profitieren.

In den Grafiken findet sich weiterhin eine Unterscheidung in je zwei Varianten, die als 1 und 2 bezeichnet wurden. Für die erste Variante werden ausschließlich die Werte der einzelnen Gewichte zufällig festgelegt. Bei Variante 2 werden zusätzlich die Anzahl der Reservoirneuronen, der Spektralradius und die Spärlichkeit des Reservoirs in vorgegebenen Intervallen zufällig bestimmt. Die feste Wahl der Parameter für Variante 1 erfolgte nach ausgiebigen Tests. Da die Ergebnisse, insbesondere bei den Jäger-Netzwerken, deutlich voneinander abweichen (siehe Abbildungen 4.5 und 4.6), lässt sich schlussfolgern, dass die Anzahl der parallel instanziierten Netzwerke noch zu gering ist. Entscheidet man sich dazu, die Anzahl der parallelen Netze weiter zu erhöhen, geht der dadurch erzielte Vorteil allerdings wieder verloren.

Für die Steil-Netzwerke wurde noch eine dritte Variante untersucht. Ähnlich der Idee von Temporal Delay Neural Networks (TDNN) [Zell, 1994] bzw. dem Einbettungsansatz der Local Models wurden dem Echo State Netzwerk nicht nur der letzte Wert der Zeitreihe \mathbf{s}_n , sondern zusätzlich noch der vorletzte Wert \mathbf{s}_{n-1} präsentiert. Insbesondere ist in

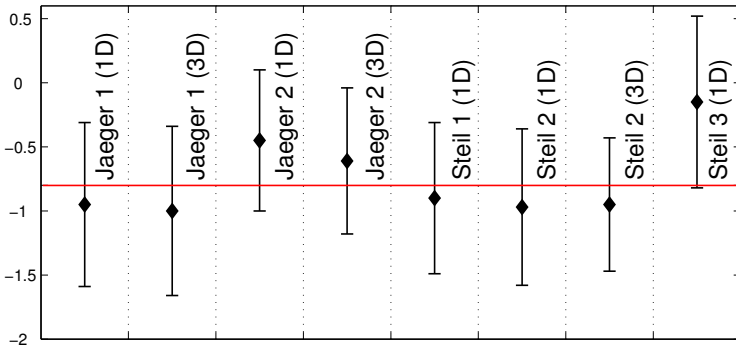


Abbildung 4.5.: Kurzschrittfehler $K_{F_{log}}$ von Varianten der ESNs auf Bewegungsdaten: Bei den Netzen mit der Struktur nach Jaeger (Spalten 1-4) schneiden jene mit fest gewählten Parametern besser ab. Die Steilvarianten mit mehr als einem Zeitschritt als Input verhalten sich schon im Kurzschrittbereich sehr instabil und erzielen verhältnismäßig schlechte Ergebnisse. Selbst die besten ESNs schlagen den als durchgehende horizontale Linie eingezeichneten Referenzalgorithmus nur knapp. (Entnommen aus [Strauß, 2008])

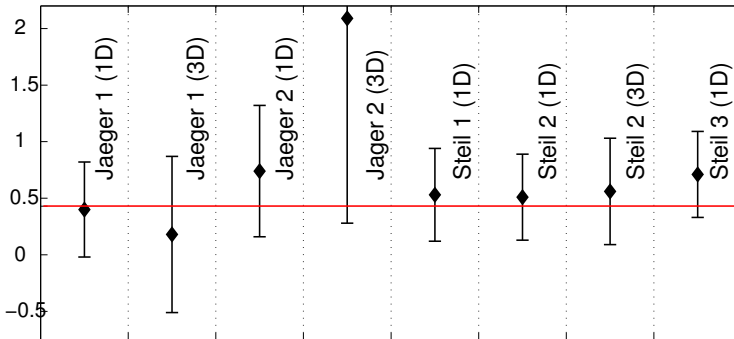


Abbildung 4.6.: Langschrittfehler LF_{log} von Varianten der ESNs auf Bewegungsdaten: Im Langschrittbereich erzielen die 3-dimensionalen Jaegernetze das beste Ergebnis, eine zufällige Parameterauswahl schneidet jedoch sehr schlecht ab. Die Steilnetze liegen alle relativ dicht zusammen, hier schneiden die 1-dimensionalen Varianten leicht besser ab, wobei der Unterschied zu den 3-dimensionalen mit steigender Neuronenzahl abnimmt. Mehr als den letzten Zeitreihenwert als Input bietet sich ebenfalls nur bei größerer Neuronenzahl an.

Abbildung 4.5 zu erkennen, dass sich die Ergebnisse eher verschlechtern.

Wie beschrieben sind die Ergebnisse auf dreidimensionalen Bewegungstrajektorien durchgeführt worden. Die entsprechenden Punkte der Bewegungstrajektorie im dreidimensionalen Raum wurden zum Einen direkt dem Netzwerk als Input präsentiert, wodurch auch direkt eine dreidimensionale Größe prädiziert werden konnte. In den Abbildungen 4.5 und 4.6 ist diese Variante mit (3D) gekennzeichnet. Zum Anderen wurden drei Netzwerke benutzt, um jeweils eine einzelne Dimension unabhängig von den anderen zu prädizieren. Diese Variante ist in den entsprechenden Abbildungen mit (1D) markiert. Wie zu erkennen ist, konnten dadurch, bis auf bei einem als Ausreißer anzusehendem Test, die Ergebnisse geringfügig verbessert werden.

4.4. Anforderungen an den Realwelteinsatz

Durch die in Abschnitt 4.3 diskutierte experimentelle Evaluierung konnte gezeigt werden, dass Echo State Netzwerke prinzipiell dazu geeignet sind, Bewegungsdaten zu prädizieren. Auch wenn es sich bei den verwendeten Datensätzen um Realweltdaten handelt, können diese durchaus als idealisiert angesehen werden. Zum Einen wurde ein Trackingsystem verwendet, bei dem leicht zu erkennende Markierungen die Grundlage bilden. Zum Anderen sind die Daten mit 60 Hz aufgezeichnet wurden. Bei den üblicherweise verwendeten Trackingsystemen liegt die Aufzeichnungsrate entweder sensor- oder auch rechenzeitbegingt wesentlich niedriger.

Gerade eine größere Aufzeichnungsrate bedeutet für das Netzwerk, dass in einem wesentlich kürzeren Zeitraum die für das Training und damit für den Wissenserwerb benötigte Datenmenge zur Verfügung steht. Da Echo State Netzwerke ausschließlich auf aktuell beobachteten Daten arbeiten, bedeutet das auch, dass entsprechend lange beobachtet werden muss. Im Falle von Bewegungsdaten ist das natürlich äußerst problematisch, da eben nicht über längere Zeiträume dieselbe Bewegung ausgeführt wird.

Geht man, wie in den Experimenten geschehen, von einer Trainingsphase von 1000 Lernschritten aus, entspricht das bei mit 60 Hz aufgezeichneten Daten einer Beobachtungszeit von ca. 17 Sekunden. Selbst diese relativ kurze Zeit ist in der Praxis bereits als äußerst kritisch anzusehen. Senkt man die Abtastrate auf die Hälfte, müsste die interagierende Person mehr als eine halbe Minute die gleiche Bewegung ausführen, um eine Prädiktion ableiten zu können.

Neben den bisher verwendeten Bewegungsdaten aus Abschnitt 3.1.1, werden im Folgenden Daten benutzt, die einen erhöhten Schwierigkeitsgrad hinsichtlich der Abtastfrequenz und des Sensorrauschens aufweisen. Die Aufzeichnung dieser Daten ist in Abschnitt 3.1.2 beschrieben.

4.4.1. Verkürzung der Beobachtungsphase

Als erstes soll untersucht werden, ob die Beobachtungsphase tatsächlich so lang gewählt werden muss, wie es für die Experimente aus Abschnitt 4.3.3 getan wurde. Dazu wurde eine mit dem Lasertracker aufgezeichnete Bewegungstrajektorie von unterschiedlichen Zeitpunkten aus prädictiert. Der Startpunkt der Prädiktion und damit die Größe der Trainingsdaten wurde sukzessive in die Zukunft verschoben. Der Prädiktionshorizont weist, um eine Vergleichbarkeit zu gewährleisten, für jeden Startpunkt dieselbe Größe auf.

In Abbildung 4.7 sind die Ergebnisse dargestellt. Während die obere Abbildung den Verlauf der Trajektorie darstellt, wird in der unteren der Verlauf des Langschrittfehlers für unterschiedliche Startzeitpunkte abgetragen. Exemplarisch wurde eine Trajektorie von einer Person gewählt, die vor dem Roboter hin und her läuft. Zu erkennen ist, dass die Beobachtung von ca. drei Viertel eines Zyklus ausreichen, um die Bewegung mit entsprechend kleinem Fehler vorherzusagen. Ab etwa 45 bis 50 Trainingsschritten bleibt der Fehler dann relativ konstant. Auch wenn es sich dabei quantitativ um wesentlich weniger Trainingsbeispiele handelt, werden durch die geringere Aufzeichnungsrate von 10 Hz trotzdem ca. 5 Sekunden für das Training benötigt.

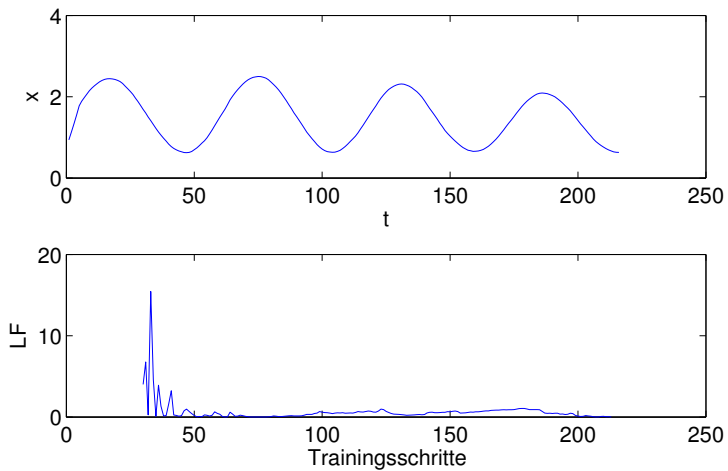


Abbildung 4.7.: Entwicklung des Langschrittfehlers:

In der unteren Grafik ist für die Anzahl der verwendeten Trainingsdaten der Langschrittfehler für die Trajektorie aus der oberen Grafik dargestellt. Zu erkennen ist, dass Echo State Netzwerke etwas weniger als eine volle Periode als Trainingsdaten benötigen, um brauchbare Ergebnisse für die Prädiktion zu liefern.

4.4.2. Kompensation von Unterabtastung

In [Helsper, 2008]⁵ konnte experimentell bestätigt werden, dass eine Unterabtastung der Trajektorien zu wesentlich schlechteren Ergebnissen für die Prädiktion führt. Dies ist damit zu erklären, dass wenn eine Trajektorie mit einer geringeren Aufzeichnungsrate erfasst wird, dem Reservoir der Echo State Netzwerke nicht genügend Zeitschritte zur Verfügung stehen, um die entsprechenden Teilschwingungen auszubilden. Es bleibt zu untersuchen, wie man dieser Verschlechterung entgegenwirken kann.

Die naheliegendste Idee ist die Interpolation der beobachteten Trajektorie, um zwischen den aufgezeichneten Punkten weitere Stützstellen einfügen zu können. Diese Stützstellen können nun ebenso für das Training genutzt werden. Die aufgezeichneten Punkte wurden stückweise mit kubischen Splines interpoliert. In Abbildung 4.8(b) ist zu erkennen, dass sich dadurch die Ergebnisse nur im Falle der Steil-Netzwerke verbessern lassen.

Schaut man sich die Prädiktion auf den originalen aufgezeichneten Daten noch einmal etwas genauer an (Abbildung 4.9), stellt man fest, dass gegen Ende der Prädiktion nur noch hochfrequente Anteile präsent sind. Hier liegt die Vermutung nahe, dass die Daten mit Sensorrauschen versehen sind. Um dies zu kompensieren, wurden die Daten mit einem zeitlichen Mittelwertfilter über 10 Zeitschritte bearbeitet. In Abbildung 4.8(c) ist zu erkennen, dass dies zu einer deutlichen Verbesserung der Ergebnisse führt.

Der Vollständigkeit halber ist in Abbildung 4.8(d) noch die Kombination aus Mittelwertfilter und Interpolation dargestellt. In Relation zur Prädiktion mit Hilfe der originalen Trajektorien können nur mit den Jäger-Netzwerken bessere Ergebnisse erzielt werden, allerdings verschlechtern sich die Ergebnisse im Vergleich zu Abbildung 4.8(c) wieder. Vergleicht man für die Jäger-Netzwerke die Änderung zwischen Abbildung 4.8(a) und (c) bzw. zwischen Abbildung 4.8(b) und (d) kann jeweils die selbe Differenz beobachtet werden, d. h. in beiden Fällen erbringt die Mittelwertfilterung den selben positiven Beitrag. Für das schlechte Abschneiden der Steil-Netzwerke kann nur vermutet werden, dass im Anpassen

⁵Die Studienarbeit wurde vom Autor im Rahmen des Projektes betreut.

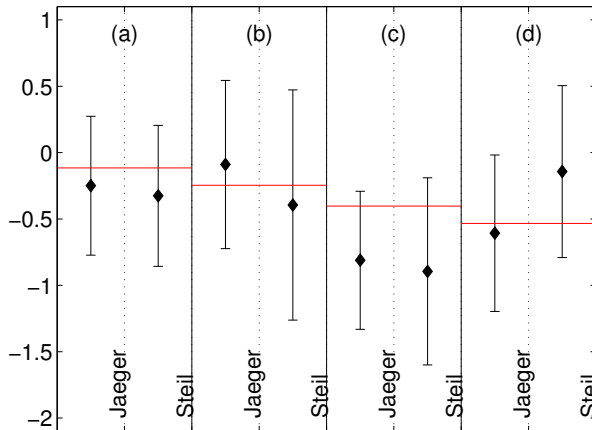


Abbildung 4.8.: Langschrittfehler für unterschiedliche Vorverarbeitungen: Die Grafik ist in vier mit (a) bis (d) gekennzeichnete Teilbereiche eingeteilt. Dargestellt wird jeweils der Langschrittfehler für die Jäger- und Steil-Variante der Echo State Netzwerke. (a) Die Prädiktion auf Basis der unveränderten Daten dient als Referenz. (b) Durch Interpolation wurde die Aufzeichnungsrates künstlich erhöht. (c) Die aufgezeichneten Daten wurden vor der Prädiktion mittelwertgefiltert. (d) Die Daten wurden mittelwertgefiltert und anschließend weitere Stützstellen interpoliert.

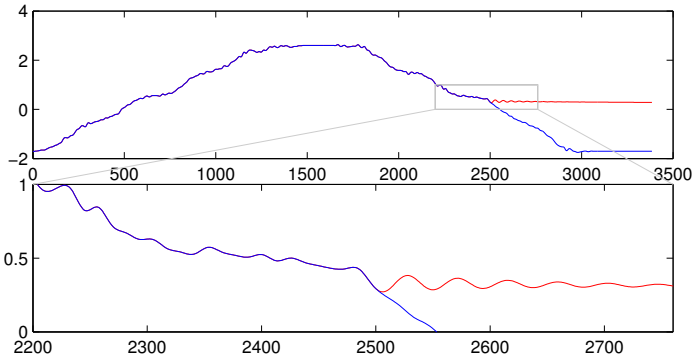


Abbildung 4.9.: Prädiktion des Rauschanteils:

Anstatt die eigentliche Bewegung zu prädizieren, wurde anscheinend durch das Echo State Netzwerk das Rauschen, d.h. der hochfrequente Anteil, gelernt. Die untere Grafik stellte eine Vergrößerung der oberen Grafik dar. Die Prädiktion startet jeweils beim 2500. Zeitschritt.

der Parameter der Fermifunktion (vergleiche Abschnitt 4.2.3) die Ursache zu suchen ist. Für das Anpassen der Parameter wird ein Teil des Trainingsprozesses verwendet. Während dieser Zeit ist das Netzwerk sehr instabil (siehe auch [Steil, 2007]). In Zusammenhang mit der Mittelwertfilterung und der Interpolation kann es sein, dass das Netzwerk in einem solchen instabilen Zustand hängen bleibt.

4.4.3. Prädiktion des Interaktionspartners

Nachdem bei sämtlichen Untersuchungen nur theoretische Fehlermaße verwendet wurden, soll zum Abschluss grafisch verdeutlicht werden, wozu die Prädiktion mit Echo State Netzwerken in der Lage ist. In Abbildung 4.10 ist die Prädiktion einer Auswahl der in Abschnitt 3.1.2 beschriebenen Daten dargestellt. Die Teilabbildungen (a)-(d) stellen dabei unabhängige Versuchsreihen dar. Dazu wurden die Echo State Netzwerk jeweils mit dem grau eingezeichneten Teil der Trajektorie trainiert. Der rot eingezeichnete prädizierte Teil der Trajektorie beruht einzig auf den

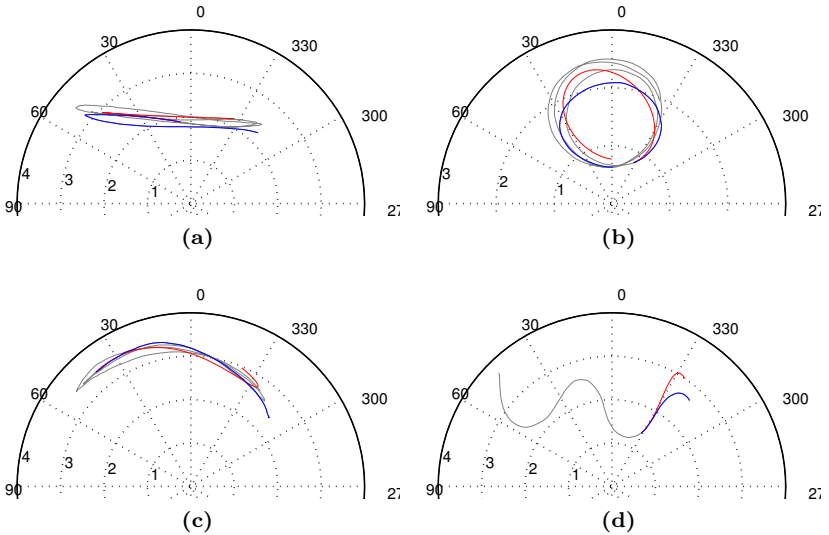


Abbildung 4.10.: Trajektorienprädiktion mittels ESN:

Mit Hilfe der Echo State Networks nach Jaeger und vorheriger Filterung der Trajektorien wurde der weitere Verlauf der Bewegungsdaten aus Abschnitt 3.1.2 prädiziert. Die vier Prädiktionen erfolgen dabei jeweils auf Basis der grau dargestellten Trainingsdaten. Der weitere (tatsächliche) Verlauf ist in blau und der prädizierte Verlauf in rot eingezeichnet.

trainierten Beobachtungen. Vergleicht man diese Prädiktion mit dem tatsächlichen Verlauf der Trajektorie (blau eingezeichnet), stellt man fest, dass sämtliche Prädiktion äußerst zufriedenstellende Ergebnisse liefern. Dabei soll nochmal ins Gedächtnis gerufen werden, dass sämtliche Prädiktionen ohne explizit definiertes Modell durchgeführt wurden. Ein solches Wissen über den weiteren Verlauf der Bewegung liefert einen großen Beitrag zur Lösung des in Abschnitt 1.2 beschriebenen Navigationsaspektes.

4.5. Fazit

Mit Hilfe von Echo State Netzwerken ist es möglich, den weiteren Verlauf von Bewegungstrajektorien vorherzusagen. Es wurden die beiden bekanntesten Vertreter der Echo State Netzwerke miteinander verglichen. Dabei hat sich gezeigt, dass sich die Version nach [Jaeger und Haas, 2004] besonders für Langzeitprädiktion eignet.

Zusätzlich zu den Ansätzen aus der Literatur konnte die Anzahl der Neuronen im Reservoir drastisch reduziert werden. Dazu wurden mehrere Netzwerkinstanzen parallel initialisiert und jeweils die beste für die Prädiktion verwendet. Das Reduzieren der Reservoir-Neuronen ermöglicht eine kürzere Berechnungszeit, was dem Einsatz in einem Echtzeitsystem, wie einem mobilen Roboter, begünstigt.

In dem in Abschnitt 1.2 vorgestellten Szenario wurden zwei Aspekte diskutiert, die auf eine Langzeitprädiktion aufbauen. Zum Einen sollte die Prädiktion genutzt werden, um die Navigationsstrategie zu verbessern. Zum Anderen benötigt die Begrüßung per Handschlag Kenntnis über den weiteren Bewegungsverlauf des Interaktionspartners. Mit den Echo State Netzwerken steht für beide Aspekte ein guter Kandidat zur Verfügung. Um sie einsetzen zu können, muss allerdings entschieden werden, ob die im Folgenden diskutierten Nachteile in Kauf genommen werden können.

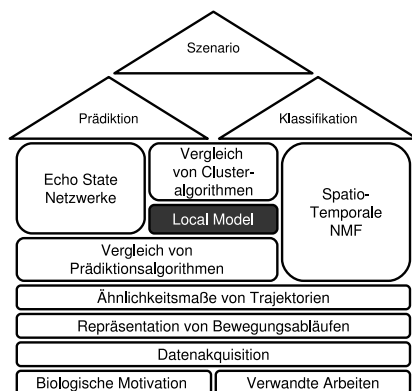
Prinzipiell lassen sich Echo State Netzwerke zwar zur Prädiktion einsetzen. Durch ihren oszillierenden Charakter werden die Prädiktionen nach einer gewissen Zeit unzuverlässig. Dies liegt daran, dass die Basis-schwingungen, die das Netz generiert, unterschiedlich schnell abklingen, wodurch die gelernte Überlagerung ihre Bedeutung verliert. Der Zeitrahmen, in dem das Netz stabile Prädiktionen liefert, reicht aber in den meisten Fällen vollkommen aus.

Darüberhinaus benötigen Echo State Netzwerke zum Prädizieren einer beobachteten Bewegung eine gewisse Zeit, um auf die Charakteristik der Bewegung trainiert zu werden. Dabei stellt nicht die Rechenleistung das Problem dar, vielmehr muss ein Gewisser Teil der Bewegung erst bekannt sein. Für den praxistauglichen Ansatz wäre es sinnvoll unterschiedliche Netze auf unterschiedliche Bewegungsarten off-line vorzutrainieren. Zu diesem Zwecke könnte man versuchen, wie es bereits beim

Training gemacht wird, mehrere Instanzen der Netze parallel zu halten. Im Trainingsprozess existiert allerdings eine Entscheidungsmöglichkeit, Netzinstanzen, die eine schlechte Prädiktion liefern, auszuschließen. Da diese auf mehreren Prädiktionsschritten beruht, bleibt fraglich, ob sie einfach auf den online Einsatz übertragen werden kann.

Kapitel 5

Local Modelling



In den experimentellen Evaluierungen, die in Abschnitt 4.1 kurz zusammengefasst sind, hat sich neben den Echo State Netzwerken ein weiteres Verfahren als geeignet erwiesen. Auch die Local Models stammen aus dem Forschungsfeld der Zeitreihenanalyse und werden zur Prädiktion eingesetzt. Die Prädiktion beruht dabei, wie auch bei Echo State Netzen, auf den aktuell beobachteten Daten und erfolgt ohne Modellannahme.

Anders als die Echo State Netzwerke bieten sie leichte Erweiterungsmöglichkeiten. In diesem Kapitel soll daher, neben dem eigentlichen Verfahren zur Prädiktion, die Möglichkeit der Nutzung bereits vorhandenen Wissens untersucht werden. Außerdem wird gezeigt, dass das Verfahren mit dieser Erweiterung in der Lage ist, die Bewegungen auch zu klassifizieren.

5.1. Bestimmung der Einbettung

Bei Echo State Netzwerken werden die Positionen für jeden Zeitschritt nacheinander dem Netz als Eingabe präsentiert. Das Reservoir sorgt dann dafür, dass der zeitliche Kontext erhalten bleibt. Die Local Mo-

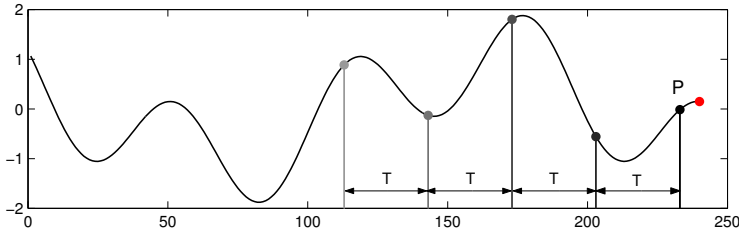


Abbildung 5.1.: Beispiel für eine reguläre Einbettung:

Die reguläre Einbettung des Punktes P besteht aus D Punkten (hier: 5), welche jeweils um T Zeitschritte versetzt sind. Im Bild am Ende der Trajektorie dargestellt ist der Output von P . Die Funktion ist eine Überlagerung von zwei Sinusschwingungen. (Entnommen aus [Strauß, 2008])

dels bieten einen solchen Mechanismus nicht. Daher muss die Trajektorie einer besonderen Form der Vorverarbeitung unterzogen werden. Sie entspricht im wesentlichen der Anwendung einer zeitlichen Fensterung¹, wie sie beispielsweise auch bei Time Delay Neural Networks (TDNN) [Zell, 1994] zum Einsatz kommt. Allerdings erfolgt das Festlegen der Fenstergröße und weiterer Parameter, wie zu sehen sein wird, automatisch.

Die Größe des Beobachtungsfensters berechnet sich als Produkt aus der Einbettungsverzögerung T und der Einbettungsdimension D . Die Bedeutung beider Parameter ist in Abbildung 5.1 dargestellt. Anders als beim üblichen Fensteransatz wird nicht jeder Punkt innerhalb des Fensters benutzt, sondern nur alle T Zeitschritte. Die Einbettungsdimension D trifft eine Aussage darüber, wie viele Punkte in die Einbettung einfließen. Ist, wie eben beschrieben, der zeitliche Abstand zwischen den Punkten konstant, spricht man von einer regulären Einbettung. In [Small, 2005] finden sich weitere Details dazu und darüberhinaus zur irregulären Einbettung.

¹Sliding Window Ansatz

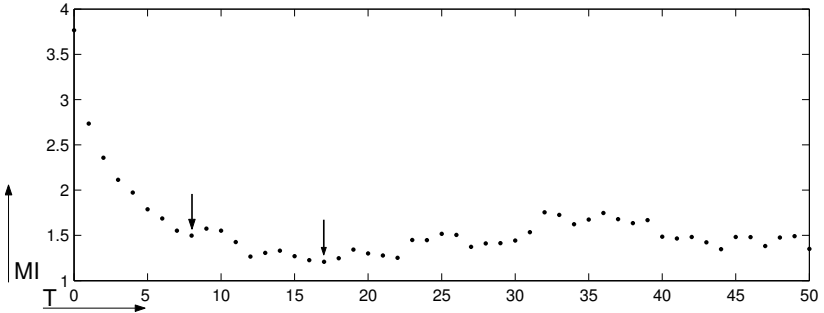


Abbildung 5.2.: Beispiel einer Informationsfunktion $MI(T)$:

Transformation MI in Abhängigkeit von der Einbettungsverzögerung T für die in Abbildung 5.1 dargestellte Funktion. Hierbei ist deutlich zu sehen, dass das erste Minimum (linker Pfeil) bei $T = 8$ nicht unbedingt die beste Wahl darstellt, da das Optimum bei $T = 17$ liegt (rechter Pfeil). (Entnommen aus [Strauß, 2008])

Aus mathematischer Sicht wird jeder Punkt \mathbf{s}_t der Trajektorie in einen $T \cdot D$ -dimensionalen Raum eingebettet:

$$\mathbf{e}_t = (\mathbf{s}_t, \mathbf{s}_{t-T}, \mathbf{s}_{t-2T}, \dots, \mathbf{s}_{t-(D-1)T}) \quad (5.1)$$

Da dies für jeden Punkt geschieht, wird auch das Abtasttheorem nach Shannon nicht verletzt. Die Einbettungsverzögerung T kann daher beliebig gewählt werden. Zu jeder Einbettung \mathbf{e}_t eines Punktes gehört seine Ausgabe \mathbf{o}_t . Dabei handelt es sich einfach um den weiteren Verlauf der Trajektorie $\mathbf{o}_t = \mathbf{s}_{t+1}$.

Bleibt noch die Frage offen, wie die beiden Parameter sinnvoll gewählt werden müssen. In der Literatur findet sich dazu ein weites Spektrum an Verfahren. Im Folgenden soll nur das jeweils praxisrelevanteste Verfahren erläutert werden.

5.1.1. Wahl der Einbettungsverzögerung

Um die Einbettungsverzögerung T zu bestimmen, wird auf die Transinformation MI zurückgegriffen. Wie bereits in Abschnitt 3.2.4 diskutiert, drückt die Transinformation aus, inwieweit zwei Parameter einen gemeinsamen Informationsgehalt aufweisen. Ist die Transinformation hoch sind die beiden untersuchten Größen redundant. Ist sie dagegen niedrig, tragen beide Größen Information bei. Ziel der Einbettung ist es, Punkte auf der Trajektorie auszuwählen, die möglichst wenig Redundanz aufweisen. Dazu wird theoretisch für jede Einbettungsverzögerung T der paarweise Informationsgehalt $MI(T)$ zwischen allen um T Zeitschritte auseinanderliegenden Punkten berechnet:

$$MI(T) = MI(\mathbf{s}_t, \mathbf{s}_{t-T}) = \sum_{\mathbf{s}_t} \sum_{\mathbf{s}_{t-T}} P(\mathbf{s}_t, \mathbf{s}_{t-T}) \log \frac{P(\mathbf{s}_t, \mathbf{s}_{t-T})}{P(\mathbf{s}_t)P(\mathbf{s}_{t-T})} \quad (5.2)$$

Die benötigten Wahrscheinlichkeiten $P(\mathbf{s}_t, \mathbf{s}_{t-T})$, $P(\mathbf{s}_t)$ und $P(\mathbf{s}_{t-T})$ werden, wie auch in Abschnitt 3.2.4, mit Hilfe von Histogrammen auf Basis der beobachteten Trajektorien bestimmt. Je mehr Trajektorien genutzt werden können, umso allgemeingültiger ist die Einbettung. Umgekehrt gilt, dass für entsprechende Teilprobleme auch individuelle Einbettungen bestimmt werden können.

Praktisch ist die Suche nach dem globalen Minimum zu aufwendig. Daher reicht es aus, wie in Abbildung 5.2 zu sehen, dass mit ansteigendem T das erste auftretende Minimum gewählt wird [Parlitz, 2006]. Dabei kann es allerdings passieren, dass es zu Überlappungen auf der eingebetteten Trajektorie kommt (siehe Abbildung 5.3). Solche Überlappungen können allerdings durch eine geeignete Wahl der Einbettungsdimension D wieder beseitigt werden.

5.1.2. Wahl der Einbettungsdimension

Um die Einbettungsdimension D bestimmen zu können, wird vorausgesetzt, dass die Einbettungsverzögerung T bereits bestimmt wurde. Wie im Zusammenhang mit Abbildung 5.3 bereits erwähnt, kann es im Einbettungsraum zu unerwünschten Überlappungen kommen. Man unterscheidet in diesem Kontext sogenannte wahre und falsche Nach-

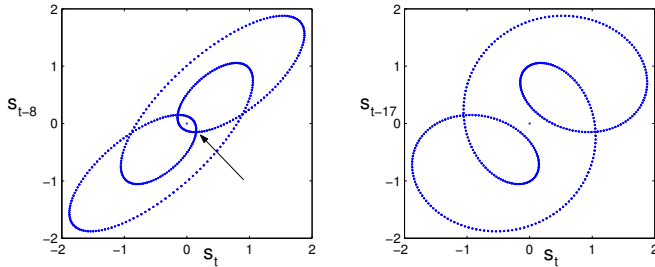


Abbildung 5.3.: Einfluss der Einbettungsverzögerung T auf benachbarte Punkte: Dargestellt ist die Funktion aus Abbildung 5.1 in unterschiedlichen Einbettungsräumen, bei unterschiedlicher Einbettungsverzögerung ($T = 8$ links, $T = 17$ rechts) und konstanter Einbettungsdimension $D = 2$. Die Bilder weisen einige Unterschiede auf. So gibt es bei dem rechten Bild im Gegensatz zum Linken keine Überschneidungen in der Mitte (Pfeil im linken Bild). (Entnommen aus [Strauß, 2008])

barn. Wahre Nachbarn sind im Einbettungsraum benachbarte Punkte, die auch im Sinne der eigentlichen Trajektorie aufeinanderfolgen. Bei falschen Nachbarn handelt es sich um die in Abbildung 5.3 dargestellten Überlappungen.

Das Verhältnis zwischen wahren und falschen Nachbarn bestimmt die Qualität der Einbettung und des darauf aufbauenden Prädiktionsverfahrens. In Abbildung 5.4 ist das Problem bei der Verwendung falscher Nachbarn kurz dargestellt. Da die meisten Verfahren, die den Einbettungsraum nutzen, darauf basieren, möglichst effizient nach ähnlichen Trajektorienverläufen zu suchen, würde durch die Auswahl von falschen Funktionsverläufen die Prädiktion fehlerhaft.

5.2. Local Model Verfahren

Wie eingangs bereits erwähnt, basiert die Gruppe der Local Model Verfahren [Engster und Parlitz, 2006] auf der im vorangegangenen Abschnitt beschriebenen Einbettung. Die prinzipielle Idee ist, wie in Abbildung

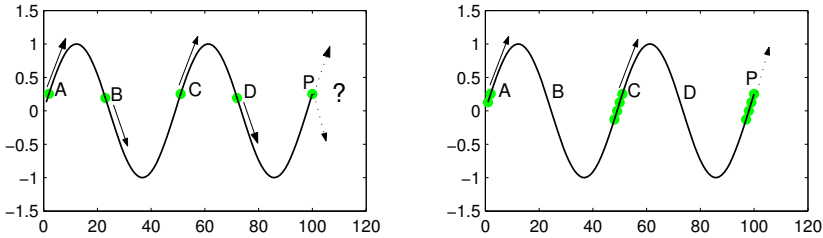


Abbildung 5.4.: Finden falscher Nachbarn durch Erhöhung von D : Bei Verwendung einer eindimensionalen Einbettung ($D = 1$) findet man bei der Nachbarsuche des Punkte P die vier Punkte A , B , C und D (linkes Bild). Bei einer Prädiktion für P ergibt sich aufgrund der falschen Nachbarn B und D dann ein Konflikt: Zwei Nachbarn sagen steigende Werte voraus, die anderen beiden hingegen fallende Werte. Durch Erhöhung der Einbettungsdimension können die falschen Nachbarn aufgrund ihrer unterschiedlichen Vorgeschichte erkannt werden (rechts). (Entnommen aus [Strauß, 2008])

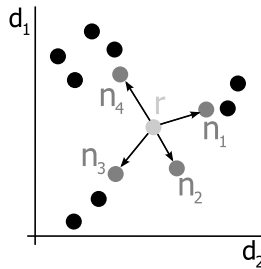


Abbildung 5.5.: Grundprinzip des Local Modeling:

Im dargestellten Einbettungsraum (d_1, d_2) wird beim Local Modeling nach den nächsten Nachbarn n_i des aktuellen Punktes r gesucht. Diese Nachbarn werden als Zustandsraumnachbarn behandelt und auf Basis deren Ausgaben wird die Prädiktion berechnet. In Abbildung 5.6 wird die Bedeutung der Nachbarschaftssuche veranschaulicht. (Entnommen aus [Strauß, 2008])

5.5 veranschaulicht, eine einfache Nachbarschaftssuche im Einbettungsraum für den letzten beobachteten Punkt \mathbf{e}_n , für den der weitere Verlauf prädiert werden soll. Dabei nutzt man die Tatsache aus, dass sich ähnliche Bewegungsabläufe im Einbettungsraum nah beieinander befinden (siehe Abbildung 5.6). Durch die Überführung in den Einbettungsraum, reicht als Distanzmaß die Euklidische Distanz aus. Theoretisch sind aber auch andere Distanzmaße verwendbar, wenn beispielsweise bestimmte Teile des Bewegungsablaufes in den Vordergrund gerückt werden sollen.

Zu jedem der gefundenen nächsten Nachbarn \mathbf{e}_i ist, wie beschrieben, eine Ausgabe \mathbf{o}_i zugeordnet. Im allgemeinen Fall des Local Models wird versucht, die Ausgabe durch eine polynomielle Approximation auf Basis der zugehörigen Einbettung \mathbf{e}_i zu beschreiben. Im Falle des in 5.2.2 beschriebenen Local Linear Model lautet die Annahme daher:

$$\mathbf{o}_i = v_0 + \sum_{j=1}^D \mathbf{e}_{i,j-1} v_j = (1, \mathbf{e}_i) \cdot \mathbf{v} \quad (5.3)$$

Die entsprechenden Polynomkoeffizienten v_0, \dots, v_D werden allerdings nicht für jeden gefundenen Nachbarn einzeln bestimmt. Sie werden vielmehr durch lineare Regression auf Basis aller Nachbarn bestimmt. Dabei werden die Nachbarn allerdings nicht gleichberechtigt zur Berechnung herangezogen. Um zu vermeiden, dass zu weit entfernte Nachbarn das Ergebnis verschlechtern, werden sie entsprechend ihrer Entfernung zum letzten Trajektorienabschnitt \mathbf{e}_n gewichtet. Die Polynomkoeffizienten $\mathbf{v} = (v_0, \dots, v_D)^T$ werden nach folgender Gleichung bestimmt:

$$\mathbf{v} = (\mathbf{P}_W^T \mathbf{P}_W)^{-1} \mathbf{P}_W^T \cdot \mathbf{o}_W \quad (5.4)$$

Dazu werden die gewichteten Polynome für jeden Nachbarn in der Matrix \mathbf{P}_W repräsentiert. Dabei steht in jedem Matrixelement ein Monom² des entsprechenden Polynoms. Wie auch bei der Matrix \mathbf{P}_W bezeichnet der Index W , dass auch die Ausgaben gewichtet in die Berechnung einfließen.

²Bei einem Monom handelt es sich um den elementaren Bestandteil eines Polynoms, der nur aus einem Term besteht.

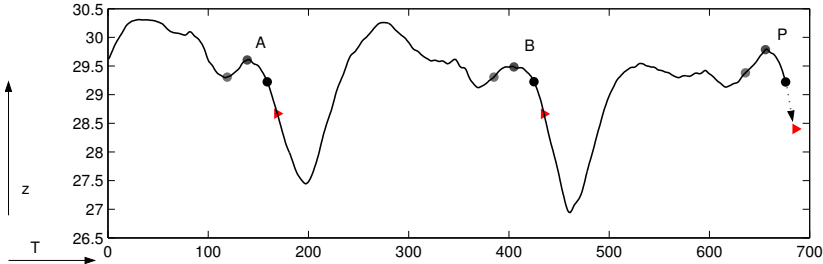


Abbildung 5.6.: Prädiktion mit dem Local Modeling:

Um eine Prädiktion für den Punkt P zu berechnen, werden zunächst die nächsten Nachbarn gemäß der gewählten Einbettung gesucht (A und B). Mit einer linearen Regression werden mit Hilfe der Nachbarn und deren Outputs (Dreiecke) die Koeffizienten für ein vorher festgelegtes Polynom geschätzt, mit dem die Vorhersage aus dem aktuellen Punkt berechnet wird. (Entnommen aus [Strauß, 2008])

In der Praxis ist der verwendete Polynomgrad niedrig. In den meisten Fällen reicht ein Polynomgrad von null bzw. eins. Man spricht in diesem Zusammenhang von Local Averaging Models bzw. Local Linear Models.

5.2.1. Local Average Model

Im ersten Fall vereinfacht sich das Polynom zu einer Konstante. Damit wird die lineare Regression aus Gleichung 5.4 zu einem gewichteten Mittel der weiteren Verläufe \mathbf{o}_i der N gefundenen Nachbarn:

$$\mathbf{v} = \mathbf{s}_{n+1}^{pred} = \frac{\sum_{i=1}^N w_i^2 \cdot \mathbf{o}_i}{\sum_{i=1}^N w_i^2} \quad (5.5)$$

Die ursprünglichen Polynomkoeffizienten \mathbf{v} nehmen damit außerdem direkt den Platz der Prädiktion \mathbf{s}_{n+1}^{pred} ein.

5.2.2. Local Linear Model

Bei der Benutzung des Local Linear Model, wird die Matrix mit den Einbettungen der N gefundenen Nachbarn gefüllt. Zusätzlich wird die erste Spalte der Matrix mit 1 aufgefüllt, um den konstanten Teil der Polynome zu beschreiben:

$$P = \begin{pmatrix} 1 & \mathbf{e}_1 \\ \vdots & \vdots \\ 1 & \mathbf{e}_N \end{pmatrix} \quad (5.6)$$

Im Anschluss wird die Matrix \mathbf{P} mit der in Abschnitt 5.2 beschriebenen Polynomkoeffizienten entfernungs-basierten Gewichtung der einzelnen Zeilen in die Matrix \mathbf{P}_W überführt. Die Koeffizienten \mathbf{v} werden dann entsprechend Gleichung 5.4 bestimmt und in das Polynom für den letzten beobachteten Punkt eingesetzt, um die Prädiktion zu berechnen:

$$\mathbf{s}_{n+1}^{pred} = v_o + \sum_{j=1}^D \mathbf{e}_{n,j-1} v_j \quad (5.7)$$

5.3. Experimentelle Evaluierung

Wie auch bei den Experimenten in Kapitel 4 ist das Ziel, die Eignung des Verfahrens für den Navigationsaspekt und den Interaktionsaspekt aus Abschnitt 1.2 zu demonstrieren. Es soll also auch hier gezeigt werden, inwieweit das Verfahren der Local Models in der Lage ist, eine möglichst gute Prädiktion zu berechnen. Darüberhinaus wird auch der Erkennungsaspekt beleuchtet und das Verfahren zur Klassifikation erweitert.

Dazu wird, wie auch in Abschnitt 4.3 geschehen, auf die Bewegungsdaten, die in Abschnitt 3.1.1 beschrieben sind, zurückgegriffen. Um die Vergleichbarkeit zu gewährleisten, werden die in Abschnitt 4.3.2 vorgestellten Langschritt- und Kurzschriftfehler gewählt. Somit wird die Fähigkeit zur Langzeitprädiktion beleuchtet, aber auch kontrolliert, ob systematische Fehler vorliegen.

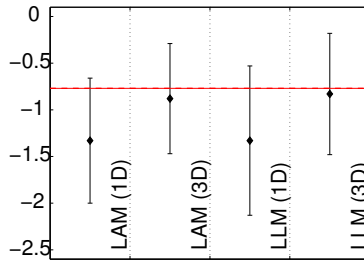


Abbildung 5.7.: Kurzschnittfehler $K_{F_{log}}$ von Varianten des Local Model: Dargestellt sind die durchschnittlichen logarithmischen Kurzfehler KF und deren Standardabweichung bei der Prädiktion der Bewegungsdaten aus Abschnitt 3.1.1 mit Local Average Model (LAM) und Local Linear Model (LLM), welche auf 1-dimensionalen Trajektorien operieren. Niedrigere Werte sind besser. Die horizontale Linie stellt den Mittelwert des Referenzalgorithmus (siehe Abschnitt 4.3.1) dar, welcher durch sämtliche Local Model Varianten unterboten wird.

Außerdem werden die in Abschnitt 4.3.1 eingeführten Referenzverfahren verwendet, um einschätzen zu können, ob man durch den Einsatz der Local Models Vorteile gegenüber der Verwendung klassischer Trackingalgorithmen erzielen kann.

Im Rahmen der Arbeit von [Strauß, 2008] wurden eine Vielzahl an Parameterkombinationen untersucht, auf die hier nicht weiter eingegangen werden soll. Dabei wurde beispielsweise auch die automatische Auswahl der Einbettungsverzögerung T und der Einbettungsdimension D genauer beleuchtet. Dabei konnte gezeigt werden, dass die in 5.1 vorgestellten Verfahren bereits sehr gute Ergebnisse ermöglichen. Allerdings war es möglich, durch zufällige Parameterauswahl bessere Prädiktionsergebnisse bezüglich des Kurzschnittfehlers zu erzielen. Beim Langschrittfehler lieferte die automatische Auswahl durchweg bessere Ergebnisse. Die durch automatische Auswahl festgelegten Parameter können daher nur als suboptimal aber ausreichend für Bewegungsdaten angesehen werden.

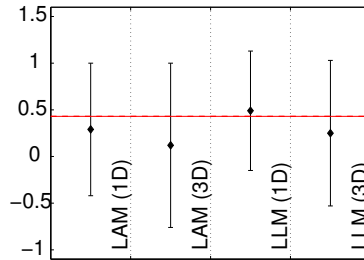


Abbildung 5.8.: Langschrittfehler LF_{log} von Varianten des Local Model: Die 3-dimensionalen Varianten des Local Model schneiden im Langschrittfehler (LF) wesentlich besser ab als die 1-dimensionalen und schlagen ebenfalls den Referenzalgorithmus deutlich. Wie auch bei Abbildung 5.7 wurden die Bewegungsdaten aus Abschnitt 3.1.1 prädiziert. Das LLM schneidet nicht wesentlich besser ab als das LAM und ist häufig sogar schlechter.

Exemplarisch soll in den Abbildungen 5.7 und 5.8 die Verwendung von drei- bzw. eindimensionalen Daten herausgegriffen werden. Genauso wie bei Echo State Netzwerken wurden die im dreidimensionalen Raum aufgezeichneten Bewegungstrajektorien dem Verfahren direkt präsentiert. Weiterhin wurden die Dimensionen als unabhängig angesehen und von drei Local Models individuell prädiziert. Anders als erwartet, ergeben sich bei der Prädiktion mit dreidimensionalen Trajektorien schlechtere Ergebnisse als bei der eindimensionalen Variante. Dies könnte auf die größere Periodendauer im dreidimensionalen Fall zurückzuführen sein. Da die unterschiedlichen Dimensionen jeweils andere Grundschwingungen aufweisen (siehe dazu Abbildung 3.1 auf Seite 33), dauert es entsprechend länger bis alle drei Dimensionen wieder „synchronisiert“ sind. Allerdings müsste dieser Effekt auch beim Langschrittfehler auftreten. Dass dies nicht geschieht, deutet darauf hin, dass die Verwendung von dreidimensionalen Daten auf Dauer gesehen stabilere Ergebnisse liefert und sich bezogen auf die kurze Prädiktion eher Offsetfehler ergeben haben.

Darüberhinaus fällt auf, dass die Verwendung von Local Linear Models keine signifikant anderen Ergebnisse gegenüber den Local Average

Models ergibt. Allerdings traten bei den, hier nicht gezeigten, Experimenten auf Sinus- und Lorenztrajektorien (vergleiche Abschnitt D.3) durchaus Unterschiede zwischen beiden Varianten auf. Daher sollten die Local Linear Models für zukünftige Untersuchungen nicht aus dem Auge verloren werden.

Auch für die Local Models konnte gezeigt werden, dass sie prinzipiell für die Prädiktion von Bewegungstrajektorien geeignet sind. Es bleibt allerdings noch offen, wie gut sie sich im Praxiseinsatz beweisen können.

5.4. Anforderungen an den Realwelteinsatz

Nachdem auch für die Local Models gezeigt werden konnte, dass sie geeignet sind, um Bewegungsdaten zu prädizieren, soll eine genauere Analyse bezüglich der Praxistauglichkeit erfolgen. Dazu werden die gleichen Experimente wie in Abschnitt 4.4 durchgeführt. Als Experimentaldaten werden wieder die, in Abschnitt 3.1.2 beschriebenen, mit dem Lasertracker aufgezeichneten Bewegungsdaten verwendet.

Bei den Local Models stellt die Verkürzung der Beobachtung auch ein großes Problem dar. Da die Anzahl der Nachbarn, auf denen die Prädiktion beruht, konstant ist, kann es durch die spärliche Besetzung im Einbettungsraum dazu kommen, dass nicht genügend Punkte in direkter Nachbarschaft vorhanden sind. In einem solchen Fall wird einer weiter entfernter Nachbar, also eine wesentlich unähnlichere Bewegung, benutzt, was zur Verschlechterung der Prädiktionsergebnisse führt. Ein möglicher Ausweg wäre, die Anzahl der auszuwählenden Nachbarn zu verringern. Da die Prädiktion allerdings immer eine Verallgemeinerung der bereits beobachteten Bewegungen darstellt, steht nicht mehr genügend Vielfalt zur Verfügung.

5.4.1. Verkürzung der Beobachtungsphase

In einer ersten Untersuchung soll betrachtet werden, wie viele Punkte mindestens beobachtet werden müssen, um eine geeignete Prädiktion ableiten zu können. Dazu wird auch wie schon bei den Echo State Netzwerken eine Trajektorie von unterschiedlichen Zeitpunkten an prä-

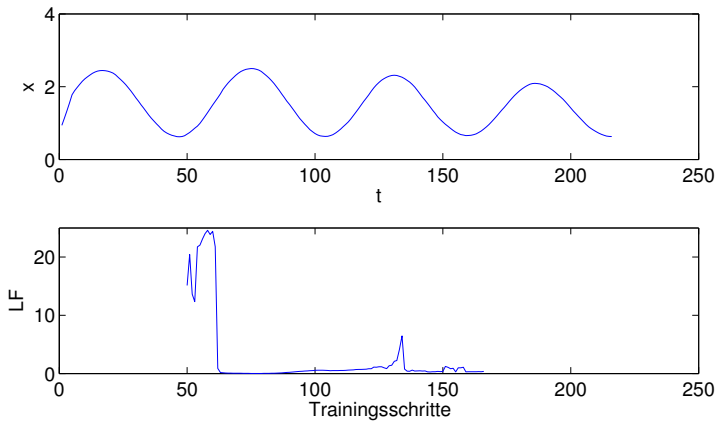


Abbildung 5.9.: Entwicklung des Langschrittfehlers:

In der unteren Grafik ist für die Anzahl der verwendeten Trainingsdaten der Langschrittfehler für die Trajektorie aus der oberen Grafik dargestellt. Zu erkennen ist, dass das Local Model erst nach einer vollen Periode brauchbare Ergebnisse für die Prädiktion liefert.

diziert. Für die Prädiktion stehen dann alle Punkte der Bewegung von dem Startpunkt an zur Verfügung. Für jeden Startpunkt wird der Langschrittfehler berechnet und grafisch dargestellt.

In Abbildung 5.9 ist zu erkennen, dass der Fehler erst ab einem vollen Zyklus sinkt. Dies ist durchaus nachvollziehbar, da vorher Verläufe prädiziert werden müssen, die noch nie beobachtet wurden. Interessant ist der Anstieg des Fehlers zum 130. Zeitschritt. Schaut man sich die Maxima der Bewegungstrajektorie etwas genauer an, stellt man fest, dass die Person im dritten Zyklus etwas früher umgekehrt ist. Hier wird offensichtlich durch die Local Models zu weit prädiziert, wodurch der Fehler ansteigt. Die Umkehrbewegung ist dagegen wieder bekannt und der Fehler sinkt.

5.4.2. Kompensation von Unterabtastung

Auch bei den Local Models bereitet eine geringere Abtastrate Probleme. Wie auch bei den Echo State Netzwerken wurde die geringe Abtastfrequenz durch Interpolation künstlich erhöht (Abbildung 5.10(b)). Dies führt allerdings, anders als bei den Echo State Netzwerken, zu einer Verbesserung der Ergebnisse für beide Varianten der Local Models.

Als nächstes wurde die bereits in Abschnitt 4.4.2 beschriebene Mittelwertfilterung durchgeführt (Abbildung 5.10(c)). Im Gegensatz zu Echo State Netzwerken wird allerdings nicht immer eine Verbesserung erzielt. Nur bei der Verwendung von Local Linear Models werden niedrigere Langschrittfehler erzielt. Dies könnte damit zu erklären sein, dass die Local Average Models nur eine Mittelung über die weiteren Verläufe durchführen und damit das Rauschen bereits eliminiert wird. Die Local Linear Models dagegen versuchen einen linearen Zusammenhang zwischen dem vergangen und dem weiteren Verlauf herzustellen. Durch die verrauschten Daten wird allerdings der lineare Zusammenhang der eigentlichen Bewegung verfälscht.

Die Kombination aus Mittelwertfilterung und Interpolation in Abbildung 5.10(d) bringt keine Verbesserung bezüglich des Langschrittfehlers. Die Ergebnisse sind mit denen aus Abbildung 5.10(c) vergleichbar.

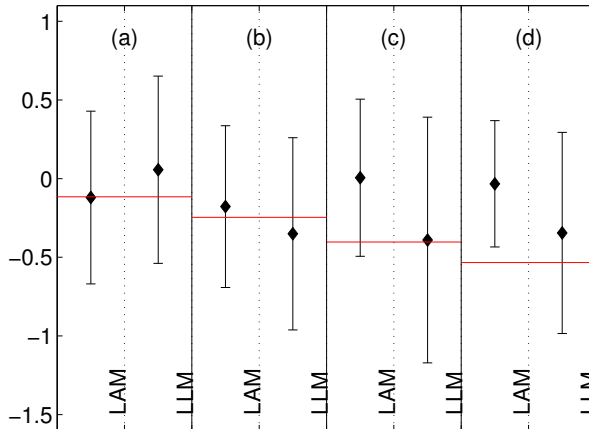


Abbildung 5.10.: Langschrittfehler für unterschiedliche Vorverarbeitungen: Die Grafik ist in vier mit (a) bis (d) gekennzeichnete Teilbereiche eingeteilt. Dargestellt wird jeweils der Langschrittfehler für das Local Average Model (LAM) und das Local Linear Model (LLM). (a) Die Prädiktion auf Basis der unveränderten Daten dient als Referenz. (b) Durch Interpolation wurde die Aufzeichnungsrate künstlich erhöht. (c) Die aufgezeichneten Daten wurden vor der Prädiktion mittelwertgefiltert. (d) Die Daten wurden mittelwertgefiltert und anschließend weitere Stützstellen interpoliert.

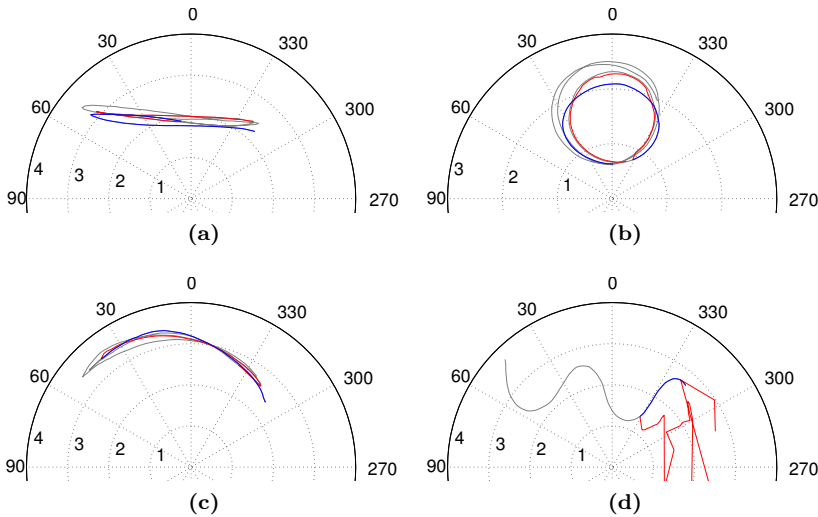


Abbildung 5.11.: Trajektorienprädiktion mittels Local Models: Mit Hilfe der Local Linear Models und vorheriger Filterung der Trajektorien wurde der weitere Verlauf der Bewegungsdaten aus Abschnitt 3.1.2 prädiziert. Die vier Prädiktionen erfolgen dabei jeweils auf Basis der grau dargestellten Trainingsdaten. Der weitere (tatsächliche) Verlauf ist in blau und der prädizierte Verlauf in rot eingezeichnet.

5.4.3. Prädiktion des Interaktionspartners

Ebenso wie bei den Echo State Netzwerken sollen die experimentellen Evaluierungen noch einmal grafisch veranschaulicht werden. Auch für die Local Models wurde die Prädiktion der Daten aus Abschnitt 3.1.2 in Abbildung 5.11 veranschaulicht. Genauso wie bei den Echo State Netzwerken, stellen die Teilabbildungen (a)-(d) jeweils unabhängige Versuchsreihen dar. Der rot eingezeichnete prädizierte Teil der Trajektorie beruht einzig auf dem bereits beobachteten Teil der Bewegung (grau eingezeichnet). Auch bei den Local Models zeigt der Vergleich mit dem blau eingezeichneten, tatsächlichen Verlauf der Bewegung äußerst zufriedenstellende Ergebnisse. Allerdings stellt man fest, dass bei

der Prädiktion der sinusförmigen Torkelbewegung durchaus noch Verbesserungspotential existiert. Hier liegt die Vermutung nahe, dass der Beobachtungszeitraum und damit das bekannte Wissen zu gering für eine stabile Prädiktion ist. Daher soll im Folgenden eine Möglichkeit zur Erweiterung der Wissensbasis vorgestellt werden.

5.5. Erweiterung der Local Models

Ebenso wie bei den Echo State Netzwerken wird bei den Local Models das Wissen für die Prädiktion aus der Vergangenheit gewonnen. Damit ist es notwendig, die zu prädizierende Bewegung entsprechend lange zu beobachten. Unter Umständen bedeutet das allerdings auch, dass eine Bewegung theoretisch so lange durchgeführt werden muss, dass eine Prädiktion eigentlich keinen Sinn mehr macht.

Wie in Abschnitt 5.4.1 und in [Schenk, 2009]³ experimentell nachvollzogen werden konnte, sind die Prädiktionen auf Basis der Local Models um so besser, je mehr zu der zu prädizierenden Bewegung ähnliche Verläufe bereits beobachtet wurden. Daher liegt der Ansatz nahe, nicht nur die aktuell beobachtete Bewegung für die Prädiktion heranzuziehen, sondern möglichst ein breites Spektrum an Bewegungen als Wissensbasis zur Verfügung zu stellen.

Für den normalen Linear Model Ansatz wird die Prädiktion ausschließlich auf Grundlage dem bereits beobachteten Teil der aktuellen Trajektorie durchgeführt. Der naheliegendste Ansatz an dieser Stelle ist, diese Wissensbasis beliebig zu erweitern. Die Local Models bieten mit ihrer Idee, ähnliche bereits bekannte Verläufe zu suchen, eine einfache Möglichkeit, solches zusätzliche Wissen einfließen zu lassen. Als problematisch erweist sich dabei, dass ein Anwachsen der Wissensbasis auch die Rechenzeit entsprechend ansteigen lässt.

Um das Durchsuchen der Wissensbasis gezielter durchführen zu können, sollte sie daher so aufgeteilt werden, dass eine systematische und somit schnelle Suche möglich ist. Um die Wissensbasis aufteilen zu können, wurden unterschiedliche Clusteralgorithmen untersucht. Im Folgenden

³Die Bachelorarbeit wurde vom Autor im Rahmen des Projektes betreut.

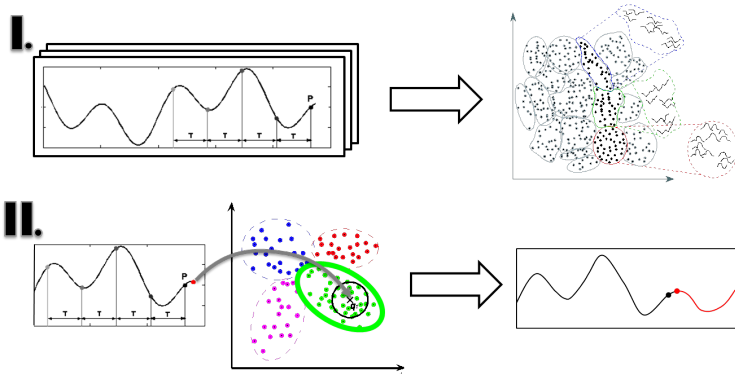


Abbildung 5.12.: Erweiterung der Local Models:

In der Trainingsphase (I.) werden alle Punkte der Trajektorien in den Einbettungsraum überführt und dort geclustert. Für eine aktuelle zu prädizierende Trajektorie (II.) wird der entsprechende Cluster ausgewählt und das Verfahren der Local Models angewendet.

soll aufgezeigt werden, wie diese Clusteralgorithmen in das Verfahren der Local Models eingebunden werden können.

5.5.1. Erweiterung der Wissensbasis

In Abbildung 5.12 ist der prinzipielle Ablauf für den erweiterten Local Model Algorithmus dargestellt. Der Ablauf ist, wie zu sehen, in zwei Phasen eingeteilt.

Die erste Phase kann als Trainingsphase bezeichnet werden. Im Falle des aktuellen Ansatzes wird sie nur einmal durchgeführt. Theoretisch besteht die Möglichkeit, den Trainingsprozess, d. h. das Erstellen der Cluster, auch während der Anwendung fortzuführen. Somit kann die Wissensbasis kontinuierlich erweitert und die Prädiktionsfähigkeit an neue Situationen angepasst werden.

Während des Trainings werden sämtliche Punkte aller bekannten Bewegungsabläufe in den Einbettungsraum überführt. Dies geschieht ge-

nauso, wie in Abschnitt 5.1 beschrieben. Auf Basis der eingebetteten Punkte wird der Einbettungsraum in disjunkte Cluster unterteilt. Dazu können neuronale Verfahren, wie Self Organizing Feature Maps und der Neuronale Gas Algorithmus verwendet werden. Beide Verfahren haben in den in Anhang E vorgestellten Experimenten die besten Resultate erzielt. Die Funktionsweise beider Verfahren kann in den Abschnitten E.2.2 und E.2.1 nachvollzogen werden.

Der Vorteil der Überführung in den Einbettungsraum besteht darin, dass sich ähnliche Bewegungsabläufe in räumlicher Nachbarschaft befinden. Damit werden durch das Clustern des Einbettungsraumes diese ähnlichen Bewegungsabläufe jeweils durch einen eigenen Cluster zusammengefasst. Um die Stabilität der Prädiktion der Local Models nicht zu gefährden, wird allerdings auf die Wahl eines Repräsentanten für jeden Cluster verzichtet. Statt dessen bleiben die in den Einbettungsraum überführten Punkte der Bewegung erhalten und stehen so für die Prädiktion zur Verfügung. Auf diese Weise wird verhindert, dass ähnlich schlechte Prädiktionsergebnisse wie bei Cluster Weighted Models erzielt werden (siehe Abschnitt D.1).

Die zweite Phase stellt die Prädiktion der aktuell beobachteten Trajektorie dar. Diese Phase wird für jeden zu prädizierenden Punkt durchgeführt. Dazu wird, genauso wie bei den eigentlichen Local Models, als erstes die aktuelle Beobachtung ebenfalls in den Einbettungsraum überführt. Bevor die Prädiktion berechnet werden kann, wird beim erweiterten Algorithmus als erstes der Cluster ausgewählt, in dessen Einzugsbereich die eingebettete aktuelle Beobachtung liegt. Im Falle der gewählten neuronalen Ansätze kann dazu auf die Bestimmung des Best-Matching-Neurons zurückgegriffen werden. Die Prädiktion selbst wird, wie in Abschnitt 5.2 beschrieben, durchgeführt. Dabei werden für die Bestimmung der zur Berechnung notwendigen Nachbarn nur Punkte aus dem aktuell gewählten Cluster herangezogen. Auf diese Weise wird nur der relevante Teil der Wissensbasis verwendet und der Suchaufwand enorm reduziert.

Abbildung 5.13 stellt mehrere Prädiktionsschritte mit dem erweiterten Local Model Ansatz dar. Dabei soll die Wahl des entsprechenden Clusters für die Prädiktion durch unterschiedliche Farben verdeutlicht werden. Es ist gut zu erkennen, dass zu den unterschiedlichen Cluster entsprechend bestimmte Teile der Bewegung gehören. In den Experimenten

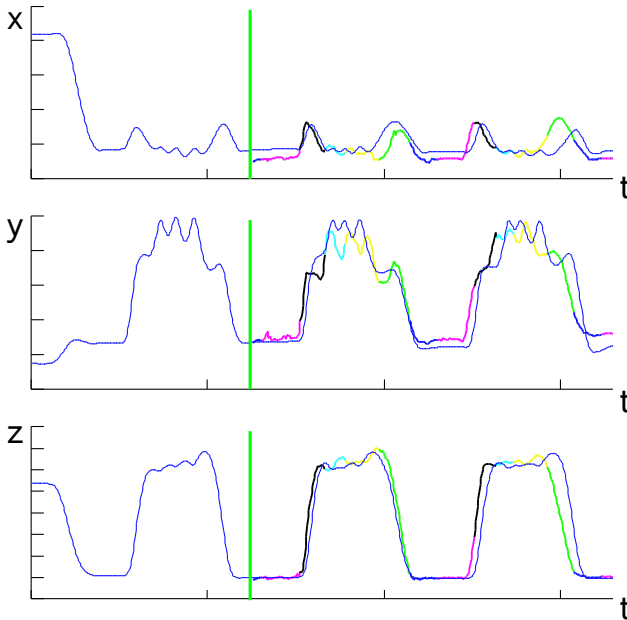


Abbildung 5.13.: Prädiktion auf Basis des Clusteransatzes: Die dickere Linie stellt den prädizierten Verlauf dar. Dieser beginnt ab der eingezeichneten vertikalen grünen Linie. Die unterschiedlichen Farben des prädizierten Verlaufs sollen dabei die Wahl der jeweiligen Cluster verdeutlichen. Wie zu erkennen ist, bleibt die Wahl des Clusters über mehrere Zeitschritte stabil.

in Anhang E wird gezeigt, dass die Verwendung der Clusterverfahren zu keiner gravierenden Verschlechterung der Prädiktionsresultate führt.

Abgrenzung zu anderen Arbeiten

Auf den ersten Blick erinnert die Verwendung von Clustern an bereits etablierte Verfahren. In Abschnitt D.1 wird das Cluster Weighted Modeling vorgestellt. Bei diesem Verfahren basiert die Prädiktion auf einer

vorherigen Clusterung der Daten. Allerdings ist die Idee, die der Erweiterung der Local Models zu Grunde liegen soll, eine andere.

Bei den Cluster Weighted Models werden die Daten geclustert und anschließend die Beschreibung der Cluster zur Prädiktion verwendet. Da die Cluster im wesentlichen über ihren Mittelwert beschrieben werden, entspricht diese Methode der Auswahl eines Repräsentanten. Da dieser Repräsentant, je nach Größe des Clusters, beliebig weit von der aktuellen Beobachtung entfernt sein kann, beeinflusst er auch entsprechend das Ergebnis der Prädiktion negativ. Außerdem werden von den Cluster Weighted Models alle Cluster zur Prädiktion benutzt, während für die hier präsentierte Erweiterung nur der am besten passende Cluster ausgewählt wird. Alle in diesen Cluster fallenden Daten können dann für die Prädiktion genutzt werden.

Ein Verfahren, welches einen ähnlichen Ansatz verfolgt, findet sich in [Rosenstein und Cohen, 1997]. Auch hier wird im Einbettungsraum eine Clusterung durchgeführt. Im Gegensatz zu dem vorgestellten Ansatz erfolgt die Clusterung allerdings nicht auf einer festen Wissensbasis. Sie wird statt dessen sukzessive aufgebaut und neu arrangiert. Dies hat zwar zur Folge, dass am Anfang nicht immer eine brauchbare Aussage abgeleitet werden kann, verfolgt aber eher den biologischen Gedanken des Lernens. Desweiteren wird die Prädiktion nicht wie bei den Local Models auf mehrere ähnliche Bewegungsabläufe gestützt. Stattdessen besitzt jeder Cluster einen Repräsentanten, mit bekannten weiteren Verlauf in die Zukunft. Für die Prädiktion wird der passende Cluster ausgewählt und bekannte weitere Verlauf verwendet.

Auch in [Loganathan und Huber, 2008] wird ein Ansatz zur Segmentierung von Bewegungstrajektorien vorgestellt. Dieser unterscheidet sich jedoch grundlegend von der hier vorgestellten Erweiterung. Das Ziel ist es dabei, in erster Linie Verhaltensübergänge zu bestimmen. Die bekannten Bewegungen werden mit Hilfe des EM-Algorithmus geclustert. Dabei wird davon ausgegangen, dass jede Bewegungsart in einen eigenen Cluster fällt. Die Verhaltensübergänge sind somit durch den Wechsel des Clusters definiert. Anders als in dieser Arbeit wird die Segmentierung im eigentlichen Koordinatensystem der Trajektorie und nicht im Einbettungsraum durchgeführt.

5.5.2. Erweiterung als Klassifikator

In Abbildung 5.13 konnte gezeigt werden, dass über einen längeren Zeitraum die Daten für die Prädiktion aus einem einzigen Cluster stammen. Daher liegt der Ansatz nahe, die ohnehin vorhandene Clusterung für eine Klassifikation nutzen zu können. Um die Klassifikation durchzuführen, wurden zwei unterschiedliche Ansätze miteinander verglichen.

5.5.2.1. Learning Vector Quantisierung

Der erste Ansatz ist eine logische Fortführung der verwendeten neuronalen Clusterverfahren. Will man anstelle einer unüberwachten Segmentierung eines Merkmalsraumes weitere vorhandene Informationen (z. B. Klassenzugehörigkeiten) einfließen lassen, findet man in der Literatur (z. B. [Kohonen, 1995]) die Learning Vector Quantisierung (LVQ). Vom Grundprinzip funktionieren diese genauso wie das Neuronale Gas. Zu Beginn wird eine feste Anzahl an Neuronen und Klassen definiert. In der Trainingsphase werden die zum Lernen verwendeten Beispiele nacheinander als Input präsentiert. Für jeden Input wird das Neuron bestimmt, welches die geringste Euklidische Distanz aufweist (Best-Matching-Neuron).

Da für das Trainingsbeispiel bekannt ist, zu welcher Klasse es gehört, kann dies für den Trainingsprozess genutzt werden. Deshalb wird, anders als beim Neuronalen Gas, bei der Adaption der Gewichte eine Unterscheidung gemacht. Diese ist Abhängig davon, ob das aktuell betrachtete Neuron zur gleichen Klasse wie der Datenpunkt gehören soll oder nicht. Gehören sowohl das Neuron, als auch der Datenpunkt zur selben Klasse, wird das Neuron in Richtung des Datenpunktes verschoben. Gehören beide zu unterschiedlichen Klassen, erfolgt eine Adaption des Neurons mit umgekehrten Vorzeichen, d. h. weg vom Trainingsbeispiel.

Bei diesem Vorgehen wird davon ausgegangen, dass die einzelnen Klassen zumindest partiell homogene Regionen bilden. Diese homogenen Regionen werden dann von den klassenspezifischen Neuronen quantisiert.

Bei der Anwendung auf das Klassifikationsproblem von Bewegungsabläufen wird genauso vorgegangen wie bei der Prädiktion mit Hilfe der geclusterten Daten. Vorteilhafterweise kann die, durch die Learning Vec-

tor Quantisierer erstellte Clusterung, wie in Abschnitt 5.5.1 beschrieben, ebenso für die Prädiktion eingesetzt werden.

5.5.2.2. Nearest-Neighbor-Klassifikation

Ein weiterer möglicher Ansatz besteht darin die Idee, die bei den Local Models der Prädiktion zugrunde liegt, auch für die Klassifikation zu nutzen. Im Falle der Prädiktion wird ein Zusammenhang zwischen den vergangenen Verläufen und dem zukünftigen Verlauf hergestellt. Um die Klassifikation durchführen zu können, muss ein Zusammenhang zwischen dem Trajektorienverlauf und der Klasseninformation angenommen werden. Dazu wird der Output (vergleiche Formel 5.3) durch einen Klassenvektor \mathbf{c}_n ersetzt, der eine Aussage über die Klasseninformation enthält, ersetzt:

$$\mathbf{c}_n = (c_n^1, \dots, c_n^K) \quad (5.8)$$

Dieser setzt sich aus den Komponenten

$$c_n^k = \begin{cases} \psi & \text{wenn } \mathbf{e}_n \text{ zur Klasse } k \text{ gehört} \\ -\psi & \text{wenn } \mathbf{e}_n \text{ nicht zur Klasse } k \text{ gehört} \end{cases} \quad (5.9)$$

zusammen. Der Vektor repräsentiert somit die Zugehörigkeit des Einbettungspunktes \mathbf{e}_n zu den jeweiligen Klassen $k = 1, \dots, K$ (z. B. laufen, werfen, klopfen, heben im Falle der Daten aus Abschnitt 3.1.1). Dabei ist ψ eine skalare Größe, mit der die Klassenzugehörigkeit ausgedrückt werden kann. Üblicherweise wird $\psi = 1$ gewählt.

Nachdem die nächsten Nachbarn bestimmt sind, lassen sich die Polynomkoeffizienten ähnlich wie in Gleichung 5.4 bestimmen:

$$\mathbf{v}_c = (\mathbf{P}_W^T \mathbf{P}_W)^{-1} \mathbf{P}_W^T \cdot \mathbf{C} \quad (5.10)$$

Dabei enthält die Matrix \mathbf{C} die Klassenvektoren für alle benutzten Nachbarn. Die Klassifikation für die aktuelle Beobachtung wird dann im Falle der Local Linear Models nach folgender Gleichung berechnet:

$$c_n = v_o + \sum_{j=1}^D \mathbf{e}_{n,j-1} v_i \quad (5.11)$$

und im Falle der Local Average Models gilt:

$$\mathbf{v}_c = \mathbf{c}_n = \frac{\sum_{i=1}^N w_i^2 \cdot \mathbf{c}_i}{\sum_{i=1}^N w_i^2} \quad (5.12)$$

In beiden Gleichungen 5.4 bzw. 5.10 wird dieselbe Pseudoinverse berechnet. Somit kann mit nur einem geringen Anstieg der Rechenzeit sowohl eine Prädiktion als auch eine Klassifikation berechnet werden.

Der berechnete Klassenvektor enthält in jeder seiner Komponenten für jede Klasse einen Wert für die Zugehörigkeit zur entsprechenden Klasse. Diese Werte können entweder direkt weiterverwendet werden, oder es muss eine Maximumauswahl erfolgen, um sich für eine Klasse zu entscheiden.

Zu guter Letzt sei noch angemerkt, dass im Falle der Local Average Models die Klassifikation einem k-Nearest-Neighbor-Klassifikator entspricht, bei dem das gewichtete Mittel der Klasseninformationen der nächsten Nachbarn berechnet wird. Die Local Linear Models sind mit einem Single-Layer Perzeptron mit Skalarproduktaktivierung vergleichbar, wobei das Training über lineare Regression erfolgt.

5.5.2.3. Vergleichende Untersuchungen

Für die Untersuchungen wurden die Daten aus Abschnitt 3.1.1 verwendet. Dabei wurden aus dem gesamten Datensatz 1200 Sequenzen von 30 unterschiedlichen Personen und mit unterschiedlichen Aktionen (Heben, Werfen, Gehen und Klopfen) ausgewählt. Sie zielten Ergebnisse lasse sich somit ohne Probleme verallgemeinern.

Um beide Verfahren vergleichen zu können, wurden sie jeweils mit unterschiedlichen Parametereinstellungen getestet. Aus Gründen der Übersichtlichkeit soll in Abbildung 5.14 nicht auf die einzelnen Untersuchungen eingegangen werden. Wichtiger an dieser Stelle ist der zu erkennende tendenzielle Unterschied zwischen beiden Ansätzen. Die Darstellung in Abbildung 5.14 verwendet eine ROC-Kurve, bei der die False-Positive-Rate (FPR) gegen die True-Positive-Rate (TPR) abgetragen wird. Die False-Positive-Rate

$$FPR = \frac{FP}{FP + TN} \quad (5.13)$$

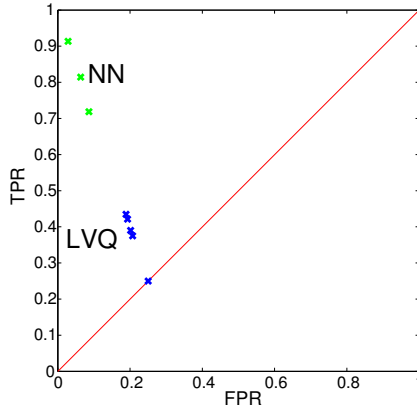


Abbildung 5.14.: Vergleich der Klassifikationsergebnisse: Die Darstellung zeigt ein ROC-Diagramm, bei dem die False-Positive-Rate (FPR) der True-Positive-Rate (TPR) gegenübergestellt werden. Ein optimaler Klassifikator würde ein Ergebnis in der linken oberen Ecke aufweisen. Die Bezeichnungen LVQ (blaue Punkte) und NN (grüne Punkte) stehen jeweils für die beiden getesteten Methoden, dem Learning Vector Quantisierer bzw. der Nearest-Neighbor-Klassifikation.

gibt das Verhältnis aus falsch klassifizierten positiven Beispielen (False Positives, FP) und allen als negativ klassifizierten Daten an (also zusätzlich die richtig klassifizierten Negativbeispiele, True Negatives, TN). In gleicher Weise gibt die True-Positive-Rate

$$TPR = \frac{TP}{TP + FN} \quad (5.14)$$

das Verhältnis aus richtig klassifizierten positiven Beispielen (True Positives, TP) und allen als positiv klassifizierten Daten an. Diesmal werden daher im Nenner die falsch klassifizierten Negativbeispiele (False Negatives, FN) addiert.

In der dargestellten ROC-Kurve ist klar zu erkennen, dass das LVQ-Verfahren deutlich schlechter abschneidet, als die Nearest-Neighbor-

Klassifikation. Die Ergebnisse, die sich dagegen mit dem Nearest-Neighbor-Klassifikator erzielen lassen, sind äußerst zufriedenstellend.

Bei der Suche nach dem Grund für die schlechten Ergebnisse hat sich herausgestellt, dass sich viele Einbettungspunkte aus dem Trainingsset auf nur wenige Cluster verteilt haben, wodurch vielen Clustern keine Einbettungspunkte zugeordnet waren. Ein möglicher Grund für dieses Phänomen könnte ein schlecht konditionierter Einbettungsraum sein. Interessant ist dabei, dass die Ungleichverteilung der Datenpunkte bei dem Neuronalen Gas Verfahren und den Self-Organizing-Feature-Maps nicht auftreten. Daher liegt die Vermutung nahe, dass die Hinzunahme der Klasseninformation beim LVQ-Verfahren die Ergebnisse verschlechtert.

5.6. Fazit

Es konnte gezeigt werden, dass Local Models für die Vorhersage von Bewegungstrajektorien geeignet sind. Damit sind auch sie in ihrer etablierten Form bereits ein guter Kandidat für den im Anwendungsszenario (Abschnitt 1.2) beschriebenen Navigationsaspekt und für das Beispiel der Begrüßung per Handschlag.

Allerdings weist der Standardalgorithmus noch Defizite auf, die bei der Verwendung mit Bewegungsdaten zu Problemen führen kann. Das Hauptproblem bildet dabei die Tatsache, dass die Prädiktion auf aktuellen Beobachtungen beruht. Insbesondere kommt es zu falschen Prädiktionen, wenn noch kein vergleichbarer Bewegungsablauf beobachtet wurde. Um dem entgegenzuwirken, wurden im Rahmen dieser Arbeit die Local Models um eine beliebig große Wissensbasis bereits bekannter Bewegungen erweitert. Diese Wissensbasis wurde mit Hilfe von Clusteralgorithmen unterteilt, damit die Nachbarschaftsuche, auf der die Local Models beruhen, effizienter durchgeführt werden kann. Damit wird der bereits bei Echo State Netzwerken beschriebene Nachteil, erst eine Weile beobachten zu müssen, bevor mit der Prädiktion begonnen werden kann, beseitigt.

In weiterführenden Untersuchungen wurde gezeigt, inwieweit die Wissensbasis auch dazu genutzt werden kann, um Aussagen über die ak-

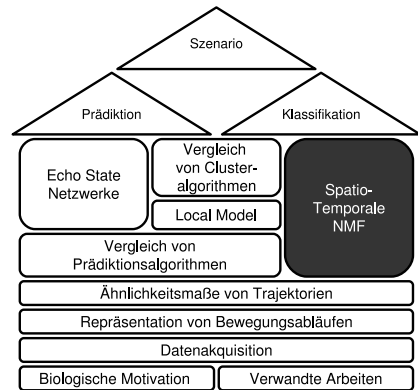
tuelle Bewegung ableiten zu können. Dazu wurde getestet, wie gut die aktuell durchgeführte Handlung klassifiziert werden kann. Für das Anwendungsszenario bedeutet dies, dass ein weiterer Teil der beschriebenen Aspekte gelöst werden kann. Dazu gehört z. B. die Erkennung des Interaktionsbedürfnisses des Nutzers, also letztendlich die Entscheidung ob ausgewichen werden muss oder ein Gespräch begonnen wird.

Allerdings besitzen auch Local Models ein paar Nachteile, für die entschieden werden muss, ob sie für den jeweiligen konkreten Anwendungsfall eine Rolle spielen. Beispielsweise ermöglichen Local Models auf deterministische Art und Weise nur eine einzelne Prädiktion des weiteren Verlaufs. Insbesondere für den Einsatz in komplexeren Systemen, sollten Möglichkeiten geschaffen werden, mehrere Hypothesen zu formulieren bzw. zumindest eine Unsicherheit für die Prädiktion anzugeben. Beispielsweise wäre es durchaus interessant, den polynomiellen Zusammenhang, der den Local Models zu Grunde liegt, mit Gauß-Prozessen zu approximieren.

Wie in den Experimenten gezeigt werden konnte, sind die Prädiktionsverfahren in der Lage, nahezu beliebige Bewegungen ohne explizites Modell vorherzusagen. In vielen aktuellen Trackingsystemen sind die Bewegungsmodelle eher simpel gehalten, um möglichst viele Fälle abdecken zu können. Dies geht bei bestimmten Bewegungen zu Lasten der Trackinggenauigkeit. Beispielsweise kann bei einer Winkbewegung der harte Umkehrpunkt große Probleme bereiten. In Abbildung 5.11 auf Seite 104 wurden auch Bewegungen untersucht, die einer Winkbewegung ähneln. Daher sollten weitere Untersuchungen erreichen können, diese Fähigkeit der Prädiktion als eine Art erweitertes Bewegungsmodell in den etablierten Trackingalgorithmen auf Basis von Kalman- oder Partikelfilter zu integrieren.

Kapitel 6

Spatio-temporale NMF



Der in diesem Kapitel vorgestellte Algorithmus basiert auf dem bekannten Verfahren der Nicht-Negativen Matrixzerlegung (NMF), welches auf einer grid-basierten Repräsentation von Trajektorien angewendet wird. Wie in Abbildung 6.1 zu sehen, werden ausgehend von einer Menge an Trainingsbeispielen Basisprimitive generiert. Ziel des Verfahrens ist, eine Dekomposition im Sinne eines Alphabetes für Bewegungstrajektorien zu erzeugen. Die erzeugten Basisprimitive werden dann benutzt, um eine beobachtete Trajektorie zu rekonstruieren. Die Information über die Rekonstruktion kann wiederum zur Klassifikation benutzt werden. Weiterhin ermöglicht eine Erweiterung des Rekonstruktionsansatzes, dass die beobachtete Bewegung weiter in die Zukunft präzidiert werden kann. Dabei geht der vorgeschlagene Ansatz über die herkömmlichen Methoden, die zur Trajektorienprädiktion (wie z. B. beim Tracken) verwendet werden, hinaus, da kein explizites Bewegungsmodell angewendet wird. Der Algorithmus ist vielmehr in der Lage, an die beobachtete Situation bzw. Bewegung anzupassen, indem die Basisprimitive neu trainiert werden. Die in diesem Kapitel vorgestellten Ergebnisse zur Prädikti-

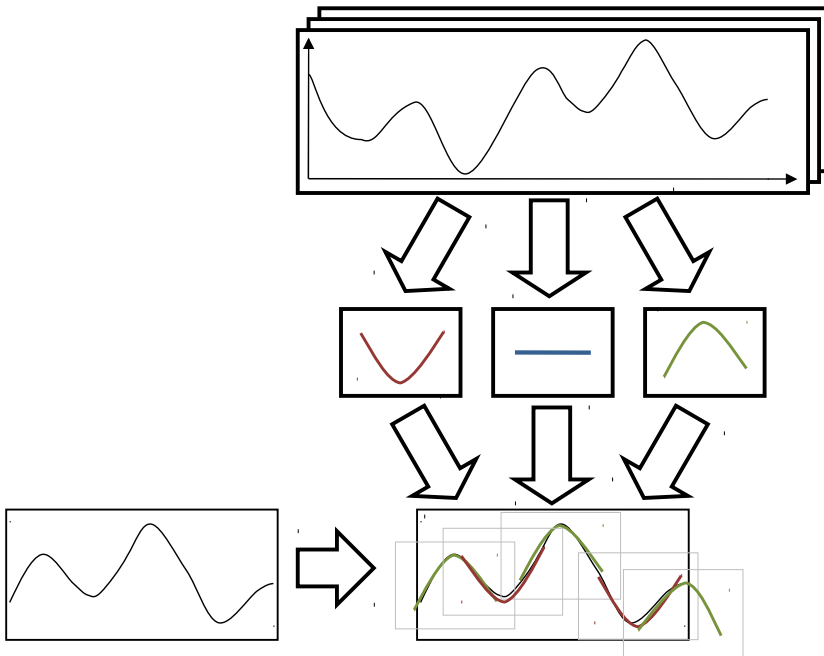


Abbildung 6.1.: Prinzip der spatio-temporalen NMF:

Aus einer Menge von Trainingstrajektorien werden verschiedene Basisprimitive automatisch erstellt. Eine beobachtete Trajektorie kann dann mit den erstellten Basisprimitive rekonstruiert werden.

on von Bewegungsdaten wurde bereits in [Eggert et al., 2009; Hellbach et al., 2009a] publiziert.

Ein interessantes Konzept, welches sich mit einer prinzipiellen Herangehensweise zur Dekomposition beschäftigt, wird in [Cemgil et al., 2006] vorgestellt. Ähnlich wie beim Klavierspielen werden die Elemente des Basisalphabetes – die Noten – überlagert, um eine Beobachtung (das Musikstück) rekonstruieren zu können. Dabei ist es möglich, vergleichbar mit dem Spielen eines Akkordes, die Basisprimitive gleichzeitig zu aktivieren. Die gesamte Rekonstruktion ergibt sich dann durch das Hintereinanderreihen der Basisprimitive. Da die Information über der Abfolge der Basisprimitive wesentlich niedrigdimensionaler als die ursprüngliche Beobachtung ist kann sie für die weitere Verarbeitung, z. B. zur Klassifikation, benutzt werden. Das ursprünglichen Piano-Model von [Cemgil et al., 2006] basiert aufgrund der Anwendung im Bereich Musik auf einem vorgegebenen Basisalphabet. Die hier vorgestellte Umsetzung ist dagegen in der Lage, die Basisprimitive aus einer gegebenen Menge von Trainingsdaten zu lernen. In [Williams et al., 2007] wird gezeigt, dass diese konzeptionelle Idee leicht auf die Dekomposition von Handschrifttrajektorien übertragbar ist. In diesem Kapitel soll die Idee nun für die Dekomposition von Bewegungstrajektorien aufgegriffen und weiterentwickelt werden.

Wenn es um die Dekomposition in Basisprimitive bzw. in Basissignale geht, findet man ein große Anzahl an Verfahren im Bereich der Blind Source Separation. Die am universell einsetzbarsten und daher auch prominentesten Verfahren bilden dabei die Hauptkomponentenanalyse (Principal Component Analysis – PCA) und die Independent Component Analysis (ICA). In Abbildung 6.2 sind für diese beiden Verfahren die entsprechenden Basisprimitive dargestellt. Darüber hinaus existiert mit der Nicht-Negativen Matrixzerlegung (Non-negative Matrix Factorization – NMF) [Lee und Seung, 2001] ein weiteres vielversprechendes Verfahren. Das erzeugte System von Basisvektoren ist bei der NMF nicht orthogonal. Dies ist im Zusammenhang mit Bewegungsdaten durchaus sinnvoll. Es ist dadurch möglich, dass die Basisprimitive – also die Elementarbewegungen, in die die Trajektorie zerlegt werden soll – in einem Teil des Verlaufs identisch sind und sich dafür an anderen Stellen spezialisieren.

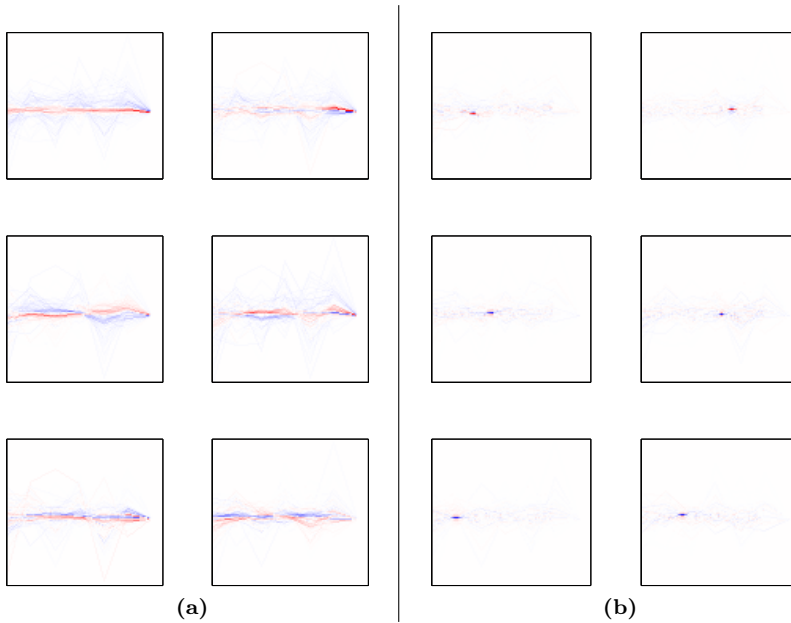


Abbildung 6.2.: Basisdekomposition mittels PCA (a) und ICA (b): Die dargestellten Basisprimitive wurden mit Hilfe der PCA (a) bzw. der ICA (b) auf Basis der Bewegungsdaten aus Abschnitt 3.1.1 erstellt. Dargestellt werden jeweils die ersten 6 Eigenvektoren bzw. unabhängigen Komponenten in farblicher Kodierung von blau (negativ) über weiß (null) zu rot (positiv). Bei der PCA (a) fällt auf, dass in jedem der Eigenvektoren jeweils ein Anteil existiert, mit dem eine andere Art der Bewegung ausgelöscht werden kann. Die ICA (b) hingegen konnte nur eine triviale Lösung generieren. Für jeden Abschnitt der Trajektorie hat sich eine unabhängige Komponente herausgebildet.

6.1. Nicht-negative Matrixzerlegung

Mit Hilfe der nicht-negativen Matrixzerlegung soll, wie auch z.B. mit Hilfe der PCA und ICA, das Source Separation Problem gelöst werden. Dabei soll ein durch die Überlagerung unterschiedlicher Quellen entstandenes Signal wieder in seine einzelnen Komponenten bzw. Basisprimitive zerlegt werden. Im vorliegenden Fall wird die Bewegungstrajektorie als eben dieses Signal betrachtet. Das Ziel ist es also, einen gegebenen Trainingsdatensatz in seine Basisprimitive zu zerlegen:

$$\mathbf{V} \approx \mathbf{W} \cdot \mathbf{H} \quad (6.1)$$

Dabei wird jedes Trainingsbeispiel als Spaltenvektor \mathbf{V}_i in der Matrix \mathbf{V} repräsentiert. Jede Spalte der Matrix \mathbf{W} steht wiederum für eine der Basisprimitive. In der Matrix \mathbf{H} bestimmt das Element H_i^j , in welchem Maß die Basisprimitive \mathbf{W}_j aktiv sein muss, um das Trainingsbeispiel \mathbf{V}_i rekonstruieren zu können. Da es sich bei der NMF um ein iteratives Verfahren handelt, stellt das Produkt aus \mathbf{W} und \mathbf{H} nur eine approximative Näherung an die Trainingsdaten \mathbf{V} dar. Dieses Produkt wird im Weiteren als Rekonstruktion $\mathbf{R} = \mathbf{W} \cdot \mathbf{H}$ bezeichnet.

Im Gegensatz zur PCA bzw. ICA erzielt man mit der NMF eine Dekomposition, die nur aus nicht-negativen Elementen besteht. Das bedeutet, dass Basisprimitive der Rekonstruktion ausschließlich hinzugefügt werden können. Es ist also nicht möglich, fehlerhafte Abschnitte zu eliminieren, die durch die Überlagerung einer nur teilweise passenden Primitive entstanden sind. Dies hat zur Konsequenz, dass ein Basisalphabet entsteht, bei dem jeder Buchstabe sehr spezielle Eigenschaften abdeckt. Für bestimmte Anwendungen, wie z. B. die Gesichtserkennung [Rajapakse und Wyse, 2003], ist diese Eigenart der NMF durchaus von Vorteil.

Um die Dekomposition zu erzeugen, werden optimierungs-basierte Ansätze verwendet. Dazu ist es als erstes notwendig, eine Energiefunktion E zu definieren:

$$E(\mathbf{W}, \mathbf{H}) = \frac{1}{2} \|\mathbf{V} - \mathbf{W} \cdot \mathbf{H}\|^2 \quad (6.2)$$

Durch Minimieren dieser Energiefunktion ist es nun möglich, Rekonstruktion unter Benutzung der Matrizen \mathbf{W} und \mathbf{H} zu erzeugen. Diese Rekonstruktion soll dabei so ähnlich wie möglich zu den Trainingsdaten \mathbf{V} sein, was wiederum die Differenz in Gleichung 6.2 gegen Null streben lässt. Im Falle der Standardformulierung der NMF sind keine weiteren Forderungen an die Rekonstruktion gegeben.

Wie man in Gleichung 6.2 sieht, hängt die Energiefunktion von den beiden unbekanntenen Matrizen \mathbf{W} und \mathbf{H} ab. Da diese beiden Matrizen üblicherweise eine große Anzahl an Elementen aufweisen, scheint das Lösen des Optimierungsproblems recht aufwendig zu sein. Erfreulicherweise kann jedes Trainingsbeispiel unabhängig von den anderen betrachtet werden:

$$\mathbf{V}_i \approx \sum_j H_i^j \cdot \mathbf{W}_j \quad (6.3)$$

Darüber hinaus können beide Matrizen alternierend angepasst werden. Dies hat zur Folge, dass sich der Raum, in dem der Optimierungsprozess stattfindet, verkleinert wird. Haben beispielsweise die Matrizen \mathbf{H} und \mathbf{W} p bzw. q Elemente, so würde der Optimierungsprozess ohne alternierendes Anpassen der Matrizen in einem $p \cdot q$ -dimensionalen Raum stattfinden. Durch das abwechselnde Anpassen beider Matrizen muss zuerst in einem p - und dann in einem q -dimensionalen Raum optimiert werden. Dadurch wird die Komplexität des Optimierungsprozesses stark reduziert, und es sind weniger Trainingsbeispiele notwendig, um den gesamten Raum abzudecken. Der Standard-NMF-Algorithmus ist wie folgt formuliert:

1. Berechnen der Rekonstruktion

$$\mathbf{R}_i = \sum_j H_i^j \mathbf{W}_j \quad (6.4)$$

2. Anpassen der Aktivitäten

$$H_i^j \leftarrow H_i^j \odot \frac{\mathbf{V}_i^T \mathbf{W}_j}{\mathbf{R}_i^T \mathbf{W}_j} \quad (6.5)$$

3. Berechnen der Rekonstruktion mit den neu berechneten Aktivitäten

$$\mathbf{R}_i = \sum_j H_i^j \mathbf{W}_j \quad (6.6)$$

4. Anpassen der Basisvektoren

$$\mathbf{W}_j \leftarrow \mathbf{W}_j \odot \frac{\sum_i H_i^j \mathbf{V}_i}{\sum_i H_i^j \mathbf{R}_i} \quad (6.7)$$

Dabei steht die Operation \odot für eine komponentenweise Multiplikation der Vektoren. Die Schritte 1 bis 4 werden iteriert, bis ein vorher definiertes Konvergenzkriterium erreicht ist. Dabei kann, wie üblich, die bereits definierte Energiefunktion benutzt werden, beispielsweise durch Festlegen einer Schwelle, unter die die Energie fallen muss. Ebenso ist es möglich abzubrechen, wenn sich die Energie von Schritt zu Schritt nur noch geringfügig ändert.

Der hier beschriebene Algorithmus nutzt die als multiplikative Updateregel bezeichnete Variante der NMF. Neben dieser Variante gibt es noch andere Möglichkeiten, die Anpassung der Aktivitäten und Basisvektoren zu berechnen. Diese sollen in dieser Arbeit allerdings nicht weiter beleuchtet werden. Es sei an dieser Stelle stattdessen auf eine sehr verständliche Zusammenfassung von [Berry et al., 2007] verwiesen.

6.1.1. Spärlichkeitskriterium

Wie in Gleichung 6.2 zu erkennen ist, ist die Energiefunktion in der Standardvariante der NMF auf eine sehr elementare Art und Weise definiert. Dies resultiert in einer Dekomposition, die beliebige Lösungen ohne besondere Charakteristiken zulässt. Unter Umständen kann das beispielsweise zu redundanten Informationen führen. Dies gilt insbesondere, wenn die Anzahl der Basisprimitive höher gewählt wird, als eigentlich zur Lösung des jeweils gegebenen Problems nötig gewesen wäre. Um solche Nachteile zu kompensieren, ist es sinnvoll, der Gleichung eine Nebenbedingung hinzuzufügen, die eine spärliche Aktivitätsmatrix \mathbf{H} fordert, wie es bereits in [Eggert und Körner, 2004] vorgestellt wurde. Eine solche Bedingung hilft, die Tatsache zu vermeiden, dass un-

terschiedliche Basisprimitive zur gleichen Zeit aktiviert werden können und somit überlagert werden.

Die erreichte Auswirkung auf die Dekomposition scheint auf den ersten Blick, dem an das Pianomodell angelehnten Ansatz zu widersprechen. Allerdings bedeutet in diesem Fall eine Aktivierung zur selben Zeit und am selben Ort, dass sich identische Basisprimitive herausbilden müssen. Sinnvoll wird die Möglichkeit einer Aktivierung zur selben Zeit und am selben Ort erst im Falle der Prädiktion, wenn mehrere Hypothesen generiert werden sollen.

$$E(\mathbf{W}, \mathbf{H}) = \frac{1}{2} \|\mathbf{V} - \overline{\mathbf{W}} \cdot \mathbf{H}\|^2 + \lambda \sum_{i,j} H_i^j \quad (6.8)$$

Der Einfluss der Spärlichkeitsbedingung kann durch den Parameter λ kontrolliert werden. An dieser Stelle soll ausschließlich der in Gleichung 6.8 gezeigte Spezialfall von Interesse sein. Eine weiterführende Erläuterung dazu findet sich in [Eggert und Körner, 2004]. Die algorithmische Beschreibung ist ähnlich zum Algorithmus der Standard NMF. Der einzige Schritt, der hinzugefügt werden muss, ist die Normierung der Basisprimitive in $\overline{\mathbf{W}}$. Die Normierung wird an dieser Stelle deshalb nötig, da sich sonst eine nicht spärliche Matrix \mathbf{H} herausbilden könnte. Denn der Spärlichkeitsterm wird auch minimal, wenn die Matrix \mathbf{H} viele kleine Komponenten enthält. Diese würden dann zum Rekonstruieren der Eingabe \mathbf{V} durch entsprechend große Komponenten in der Matrix \mathbf{W} kompensiert. Durch Normierung wird die Größe der Komponenten von \mathbf{W} beschränkt und somit die Herausbildung der Spärlichkeit in der Matrix \mathbf{H} begünstigt.

Um die Änderungen im Vergleich zur Standard NMF zu verdeutlichen, sind die gleich gebliebenen Teile der Berechnung grau dargestellt.

1. Normierung der Basisvektoren

$$\overline{\mathbf{W}}_j = \frac{\mathbf{W}_j}{\|\mathbf{W}_j\|} \quad (6.9)$$

2. Berechnen der Rekonstruktion

$$\mathbf{R}_i = \sum_j H_i^j \overline{\mathbf{W}}_j \quad (6.10)$$

3. Anpassen der Aktivitäten

$$H_i^j \leftarrow H_i^j \odot \frac{\mathbf{V}_i^T \overline{\mathbf{W}}_j}{\mathbf{R}_i^T \overline{\mathbf{W}}_j + \lambda} \quad (6.11)$$

4. Berechnen der Rekonstruktion mit den neu berechneten Aktivitäten

$$\mathbf{R}_i = \sum_j H_i^j \overline{\mathbf{W}}_j \quad (6.12)$$

5. Anpassen der Basisvektoren

$$\mathbf{W}_j \leftarrow \mathbf{W}_j \odot \frac{\sum_i H_i^j [\mathbf{V}_i + (\mathbf{R}_i^T \overline{\mathbf{W}}_j) \overline{\mathbf{W}}_j]}{\sum_i H_i^j [\mathbf{R}_i + (\mathbf{V}_i^T \overline{\mathbf{W}}_j) \overline{\mathbf{W}}_j]} \quad (6.13)$$

6.1.2. Transformationsinvarianz

Neben dem Spärlichkeitskriterium wurde in [Eggert et al., 2004] eine zusätzliche Erweiterung der NMF publiziert. Das Konzept der Transformationsinvarianz erlaubt es, die Basisprimitive beliebig zu verschieben, zu rotieren oder zu skalieren, um den Input zu rekonstruieren. Auf diese Weise ist es nicht notwendig, jede mögliche, in den Trainingsdaten auftretende, Transformation mit einem zusätzlichen Basisvektor zu modellieren. Erreicht wird dies, indem eine Transformationsmatrix \mathbf{T} der Formulierung der Dekomposition hinzugefügt wird.

$$\mathbf{V} \approx \mathbf{T} \cdot \mathbf{W} \cdot \mathbf{H} \quad (6.14)$$

Dabei ist zu beachten, dass die Aktivitätsmatrix \mathbf{H} so angepasst werden muss, dass jede mögliche Transformation durch eine eigene Aktivität abgedeckt werden kann. Dazu wird die Matrix \mathbf{H} zu einem Aktivitätstensor \mathbf{H}^m erweitert, wobei für jede Art der affinen Transformation (Rotation, Translation und Skalierung) eine entsprechende Stufe im Tensor existiert. Bei der Notation \mathbf{H}^m beschreibt der Index m einen

Vektor, der den jeweiligen Transformationsparameter (Rotationswinkel, Verschiebungsoffset und Skalierungsfaktor) beschreibt.

$$\mathbf{V}_i \approx \sum_j \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \cdot \mathbf{T}^{\mathbf{m}} \cdot \mathbf{W}_j \quad (6.15)$$

Bei der Berechnung muss nun für jede erlaubte Transformation die korrespondierende Aktivität individuell trainiert werden.

1. Berechnen der Rekonstruktion

$$\mathbf{R}_i = \sum_j \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{T}^{\mathbf{m}} \mathbf{W}_j \quad (6.16)$$

2. Anpassen der Aktivitäten

$$H_i^j \leftarrow H_i^j \odot \frac{\mathbf{V}_i^T \mathbf{T}^{\mathbf{m}} \mathbf{W}_j}{\mathbf{R}_i^T \mathbf{T}^{\mathbf{m}} \mathbf{W}_j} \quad (6.17)$$

3. Berechnen der Rekonstruktion mit den neu berechneten Aktivitäten

$$\mathbf{R}_i = \sum_j \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{T}^{\mathbf{m}} \mathbf{W}_j \quad (6.18)$$

4. Anpassen der Basisvektoren

$$\mathbf{W}_j \leftarrow \mathbf{W}_j \odot \frac{\sum_i \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{V}_i^T \mathbf{T}^{\mathbf{m}}}{\sum_i \sum_{\mathbf{m}} H_i^{j,\mathbf{m}} \mathbf{R}_i^T \mathbf{T}^{\mathbf{m}}} \quad (6.19)$$

Das Einführen der Matrix T hat aus mathematischer Sicht noch einen weiteren positiven Nebeneffekt. Wählt man die Dimensionen der Matrix geeignet, ist es möglich, dass die Basisprimitive kleinere Dimension haben können als die zu rekonstruierenden Trainingsbeispiele.

6.2. Dekomposition von Bewegungsabläufen

Um in der Lage zu sein, Trajektorien von in der Umgebung befindlichen dynamischen Objekten präzisieren zu können, ist es notwendig, diese Objekte zu identifizieren und ihre Bewegung verfolgen zu können.

Wie auch in den vorangegangenen Kapiteln wird zur Vereinfachung ein idealer Tracker angenommen, der in der Lage ist, objektspezifische kontinuierliche Trajektorien in Echtzeit zu liefern. Die so erhaltene Bewegungstrajektorie wird, wie bereits in Abschnitt 3.2 eingeführt, nun als Zeitreihe \mathcal{T} interpretiert, bei der für jeden Zeitpunkt $i = 0, 1, \dots, n-1$ eine Position $\mathbf{s}_i = (x_i, y_i, z_i)$ existiert:

$$\mathcal{T} = (\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{n-1}). \quad (6.20)$$

Es wäre nun möglich den Vektor \mathcal{T} dem NMF-Verfahren direkt zu präsentieren. Dies könnte allerdings zu einem ungewollten Verhalten führen, wenn man versucht, Basiskomponenten zu überlagern. Man könnte sich zwei Basisvektoren vorstellen: einen, der ein Linksabbiegen und ein anderer, der ein Rechtsabbiegen repräsentiert. Da die Vektoren additiv überlagert werden, würde in der Rekonstruktion eine Geradeausbewegung entstehen.

Ziel ist es, auf eine Menge von Basisprimitive zurückgreifen zu können, die eine nach der Anderen aneinandergereiht werden können. Weiterhin ist es für eine Prädiktionsaufgabe sinnvoll, mehrere alternative Hypothesen generieren zu können. Um diese Ziele erreichen zu können, wird die x - t -Trajektorie in eine Gridrepräsentation überführt, wie es in Abbildung 6.3 verdeutlicht wird. In der Gridrepräsentation repräsentiert jede Gridzelle (x_i, t_j) eine bestimmte Position (räumliche Koordinate) x_i zu einem bestimmten Zeitpunkt t_j . Die meisten aktuellen Navigationsverfahren, angefangen mit [Elfes, 1989], basieren auf Gridkarten. Dadurch wird mit diesem Ansatz die Möglichkeit einer Schnittstelle geschaffen, um die Prädiktionen auf einfache Weise in die Navigationsstrategie integrieren zu können.

Das entstandene Grid wird nun dem, um das Spärlichkeitskriterium und die Translationsinvarianz erweiterten, NMF-Algorithmus (vergleiche Abschnitt 6.1.1 bzw. 6.1.2) präsentiert. Die Nutzung der Gridrepräsentation für Bewegungstrajektorien unterstützt dabei den nichtnegativen Charakter der Basiskomponenten und ihrer Aktivitäten. Da es sich bei dem Grid um eine zweidimensionale Struktur handelt, stellt sich das Problem, wie diese zweidimensionale Struktur nun wiederum als Spaltenvektor in den entsprechenden Matrizen repräsentiert werden muss. Es wird sich allerdings zeigen, dass durch bestimmte Vereinfachungen



Abbildung 6.3.: Überführung von Bewegungstrajektorien in ihre Gridrepräsentation: *Eine Gridzelle wird auf 1 gesetzt, wenn sie sich auf dem Pfad der Trajektorie befindet und sonst auf 0. Jede Dimension der Bewegung wird dabei einzeln betrachtet. Während der Prädiktionsphase können mehrere alternative Hypothesen generiert werden, indem unterschiedliche Basisprimitive überlagert werden. In der Grafik ist dies durch die grauen Trajektorien auf der rechten Seite des Grids angedeutet.*

chungen des NMF-Verfahrens dieses Problem nicht gesondert betrachtet werden muss.

An dieser Stelle sei zu erwähnen, dass die Überführung in die Gridrepräsentation für jede der Dimensionen der Position s_i einzeln durchgeführt werden muss. Jedes der so entstandenen Grids wird nun durch die NMF unabhängig von den anderen verarbeitet. Jede Dimension unabhängig von den anderen zu betrachten, ist eine gängige Technik, um die Komplexität der Bewegungsanalyse zu reduzieren und dadurch unter anderem Rechenzeit zu sparen (vergleiche [Naftel und Khalid, 2006]). Theoretisch bietet der NMF-Ansatz die Möglichkeit, mehrdimensionale Gridrepräsentationen zu verarbeiten. Darauf soll an dieser Stelle allerdings erst einmal verzichtet werden.

Wendet man einen Algorithmus zur Basisdekomposition auf Bewegungsdaten an, ist es klar, dass es möglich sein muss, die entstehenden Bewegungsprimitive bestimmten Transformationen zu unterziehen, um sie wieder zur gesamten Trajektorie rekombinieren zu können. Um beispielsweise ein Linksabbiegen am Ende eines geraden Ganges modellieren zu können, bedarf es eigentlich nur zweier unterschiedlicher Basisprimitive: einem geraden Stück und einer Linkskurve. Um nun die Gesamttrajektorie zu rekonstruieren, muss die Primitive zum Linksabbiegen an das Ende der geradlinigen Bewegung verschoben werden. Die Erweiterung um die Transformationsinvarianz bietet sich also für den Einsatz auf Bewegungsdaten an. Für das in dieser Arbeit untersuchte Verfahren wird

sich allerdings auf die Translation als spezielle Art der Transformation konzentriert. Dies ermöglicht eine starke Vereinfachung des Algorithmus, was wiederum einem Einsatz unter Echtzeitbedingungen zu Gute kommt.

Das Spärlichkeitskriterium unterstützt das Vermeiden von trivialen Lösungen, die im Zusammenspiel mit der Translationsinvarianz entstehen können. Beispielsweise ist eine mögliche Lösung eine Basiskomponente, die nur an einer einzigen Gridposition gefüllt ist. Diese kann dann durch die entsprechenden Verschiebungen hintereinander gereiht werden. Somit würde die Trajektorie einfach als eine Art Kopie in die Aktivitätsmuster übernommen werden. Durch das Spärlichkeitskriterium wird nun die Aktivitätsmatrix ausgedünnt, wodurch diese Trivillösung wesentlich weniger wahrscheinlich wird.

6.2.1. Trainingsphase

Das Ziel der Trainingsphase ist es, eine vorher festgelegte Anzahl von Basisprimitive zu generieren, mit denen es möglich ist, den gegebenen Trainingsdatensatz sinnvoll zu dekomponieren (siehe Abbildung 6.4). Die so ermittelten Basisprimitive sollen dann in der Anwendungsphase dazu benutzt werden, die beobachtete und noch unbekannte Trajektorie zu beschreiben. Damit ist natürlich klar, dass der Trainingsdatensatz ein möglichst breites Spektrum des entsprechenden Anwendungsfeldes abdecken muss.

Die Trajektorien des Trainingsdatensatzes werden, wie in Abschnitt 6.2 vorgestellt, in ihre Gridrepräsentation überführt. Die Grids werden dann als Eingabe folgendem vereinfachten NMF-Algorithmus präsentiert:

1. Normierung der Basisvektoren

$$\overline{\mathbf{W}}_j = \frac{\mathbf{W}_j}{\|\mathbf{W}_j\|} \quad (6.21)$$

2. Berechnen der Rekonstruktion

$$\mathbf{R}_i^{\mathbf{k}} = \sum_j \sum_{\mathbf{n}} \overline{\mathbf{W}}_j^{\mathbf{k}-\mathbf{n}} H_i^{j,\mathbf{n}} \quad (6.22)$$

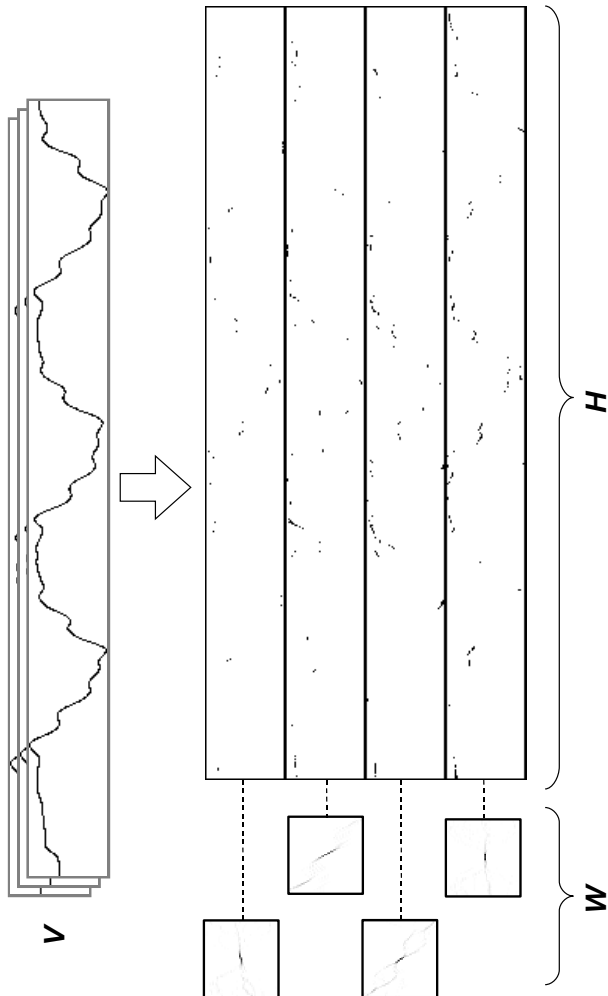


Abbildung 6.4.: Trainingsphase der Spatio-Temporalen NMF: Gegeben sei eine Menge von Trainingsbeispielen in der Matrix V . Der beschriebene Algorithmus berechnet daraus die Basisprimitive W und die zugehörigen Aktivitäten H .

3. Anpassen der Aktivitäten

$$H_i^{j,m} \leftarrow H_i^{j,m} \odot \frac{\sum_{\mathbf{n}} \mathbf{V}_i^{m+n} \overline{\mathbf{W}}_j^{\mathbf{n}}}{\sum_{\mathbf{n}} \mathbf{R}_i^{m+n} \overline{\mathbf{W}}_j^{\mathbf{n}} + \lambda} \quad (6.23)$$

4. Berechnen der Rekonstruktion mit den neu berechneten Aktivitäten

$$\mathbf{R}_i^{\mathbf{k}} = \sum_j \sum_{\mathbf{n}} \overline{\mathbf{W}}_j^{\mathbf{k}-\mathbf{n}} H_i^{j,\mathbf{n}} \quad (6.24)$$

5. Anpassen der Basisvektoren

$$\mathbf{W}_j^{\mathbf{k}} \leftarrow \mathbf{W}_j^{\mathbf{k}} \odot \frac{\nabla_{\overline{\mathbf{W}}_j} F^-(\mathbf{k})}{\nabla_{\overline{\mathbf{W}}_j} F^+(\mathbf{k})} \quad (6.25)$$

$$\begin{aligned} \nabla_{\overline{\mathbf{W}}_j} F^-(\mathbf{k}) &= \sum_i \sum_{\mathbf{n}} \mathbf{V}_i^{\mathbf{k}+\mathbf{n}} H_i^{j,\mathbf{n}} \\ &+ \sum_{\mathbf{l}} \left(\overline{\mathbf{W}}_j \overline{\mathbf{W}}_j^T \right)_{\mathbf{l}}^{\mathbf{k}} \sum_i \sum_{\mathbf{n}} \mathbf{R}_i^{\mathbf{l}+\mathbf{n}} H_i^{j,\mathbf{n}} \end{aligned} \quad (6.26)$$

$$\begin{aligned} \nabla_{\overline{\mathbf{W}}_j} F^+(\mathbf{k}) &= \sum_i \sum_{\mathbf{n}} \mathbf{R}_i^{\mathbf{k}+\mathbf{n}} H_i^{j,\mathbf{n}} \\ &+ \sum_{\mathbf{l}} \left(\overline{\mathbf{W}}_j \overline{\mathbf{W}}_j^T \right)_{\mathbf{l}}^{\mathbf{k}} \sum_i \sum_{\mathbf{n}} \mathbf{V}_i^{\mathbf{l}+\mathbf{n}} H_i^{j,\mathbf{n}} \end{aligned} \quad (6.27)$$

Bevor die dargestellten Verarbeitungsschritte iteriert werden können, müssen die Basiskomponenten \mathbf{W}_j , und Aktivitäten \mathbf{H}_i als Startpunkt für den Optimierungsprozess initialisiert werden. Dabei werden die Elemente der Basiskomponenten \mathbf{W}_j gleichverteilt zwischen 0 und 1 gezogen. Die Elemente der Aktivitätsmatrizen \mathbf{H}_i werden ebenfalls einer Gleichverteilung folgend zufällig bestimmt, allerdings im Intervall $[0, (p \cdot q)^{-1}]$, wenn $\mathbf{H}_i^{\mathbf{k}} \in \mathbb{R}^{p \times q}$.

Die Gridrepräsentation wird direkt ohne vorherige Umwandlung in eine vektorielle Form verarbeitet¹. Durch die Beschränkung auf Translation als einzig mögliche Transformation, ergeben sich einige interessante Vereinfachungen (vergleiche Abschnitt 6.1.2). Schaut man sich die innere der beiden Summen in der Berechnung der Rekonstruktion etwas genauer an, stellt man fest, dass es sich dabei um eine Faltung handelt:

$$\sum_{\mathbf{n}} \overline{\mathbf{W}}_j^{\mathbf{k}-\mathbf{n}} H_i^{j,\mathbf{n}} \quad (6.28)$$

Dies hat zur Konsequenz, dass hierfür die entsprechenden bekannten Beschleunigungen durch Transformation in den Frequenzraum mittels Fourieranalyse und anschließende Multiplikation benutzt werden können.

Ebenso können die Summen

$$\sum_{\mathbf{n}} \mathbf{V}_i^{\mathbf{k}+\mathbf{n}} H_i^{j,\mathbf{n}} \text{ und } \sum_{\mathbf{n}} \mathbf{R}_i^{\mathbf{k}+\mathbf{n}} H_i^{j,\mathbf{n}} \quad (6.29)$$

beschleunigt werden, da es sich hierbei um Korrelationsberechnungen handelt.

Neben den berechneten Basisprimitiven liefert der NMF-Algorithmus außerdem eine Aussage, wie die einzelnen Trainingsbeispiele unter Nutzung der Basisprimitive dekomponiert werden können. Diese Aussage kann für das Training des Klassifikators direkt weiterbenutzt werden.

Um möglichst eindeutige Basisprimitive zu erzeugen, sollte im Rahmen der Trainingsphase der Einfluss des Spärlichkeitskriteriums möglichst hoch sein. Dies geschieht durch eine geeignete Wahl des Parameters λ in Gleichung 6.8. Allerdings ist dabei zu beachten, dass die Wahl eines zu großen λ den Optimierungsprozess negativ beeinflussen kann. Dies wird klar, wenn man bedenkt, dass das Spärlichkeitskriterium (rechter Summand in Gleichung 6.8) minimal wird, wenn alle Elemente der Matrix

¹Rein formal werden damit die Matrizen \mathbf{V} , \mathbf{R} und \mathbf{W} zu Tensoren. Dies soll allerdings nicht weiter verfolgt werden. Man stelle sich hier einfach die einzelnen Grids bzw. Basisprimitive vor, auf die durch geeignete Indizierung zugegriffen werden kann. Insbesondere geschieht dies durch den jeweils unteren Index. Der oberere Index ist daher eine Indizierung innerhalb des jeweiligen Grids.

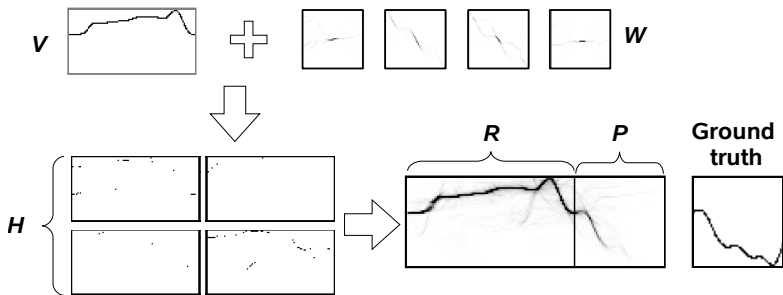


Abbildung 6.5.: Anwendungsphase der Spatio-Temporalen NMF: Die während der Trainingsphase generierten Basisprimitive W werden benutzt, um die beobachtete Trajektorie V zu rekonstruieren (Matrix R). Dies resultiert in einer Menge von spärlichen Aktivierungen (eine für jede Basisprimitive), die beschreiben, an welchem Ort und zu welchem Zeitpunkt eine bestimmte Primitive benutzt wird. Neben der Rekonstruktion der beobachteten Trajektorie ist es außerdem möglich, eine gewisse Anzahl von Zeitschritten vorherzusagen. Dazu wird die Matrix R um den Prädiktionshorizont P erweitert.

H Null sind. Um allerdings überhaupt eine Rekonstruktion generieren zu können, müssen eine gewisse Anzahl von Elementen ungleich Null sein. Durch die Wahl von λ muss also ein geeignetes Verhältnis der beiden Summanden hergestellt werden. Dies geschieht momentan noch per Hand. Es ist allerdings ohne Probleme möglich, dies mit entsprechenden Methoden (z. B. mit genetischen Algorithmen) zu automatisieren.

6.2.2. Anwendungsphase

Wie in Abbildung 6.5 angedeutet, wird in der Trainingsphase ein Satz Basisprimitive generiert. Diese sollen nun benutzt werden, um eine beobachtete und noch unbekannte Bewegung zu dekomponieren. Dazu wird angenommen, dass die Position eines dynamischen Objektes (z. B. einer Person) kontinuierlich verfolgt wird. Es kommt also zu jedem Zeitschritt eine neue Beobachtung hinzu. Um die beobachtete Bewegung

der NMF als Eingabe präsentieren zu können, wird nun ein Sliding-Window-Ansatz benutzt, um immer die jeweils n letzten Beobachtungen zu benutzen. Das so entstandene Zeitfenster der Trajektorie wird auf die bereits in Abschnitt 6.2 vorgestellte Art und Weise in eine Grid-Repräsentation überführt. Für dieses Grid wird nun die Aktivierung der bereits bekannten Basisprimitive ermittelt, in dem auf Basis des NMF-Ansatzes die Eingabe rekonstruiert wird. Im Prinzip werden dieselben Schritte wie in der Trainingsphase (Abschnitt 6.2.1) berechnet. Da aber die Basisprimitive bekannt sind und nicht mehr verändert werden müssen, entfallen die Schritte 4 und 5 aus Abschnitt 6.2.1:

1. Normierung der Basisvektoren

$$\overline{\mathbf{W}}_j = \frac{\mathbf{W}_j}{\|\mathbf{W}_j\|} \quad (6.30)$$

2. Berechnen der Rekonstruktion

$$R^k = \sum_j \sum_n \overline{\mathbf{W}}_j^{k-n} H^{j,n} \quad (6.31)$$

3. Anpassen der Aktivitäten

$$H^{j,m} \leftarrow H^{j,m} \odot \frac{\sum_n V^{m+n} \overline{\mathbf{W}}_j^n}{\sum_n R^{m+n} \overline{\mathbf{W}}_j^n + \lambda} \quad (6.32)$$

In der Anwendungsphase hängt die Wahl von λ von der zu lösenden Aufgabenstellung ab. Soll eine beobachtete Trajektorie prädiziert werden, ist es sinnvoll, dass mehrere alternative Hypothesen generiert werden können. Dazu müssen unterschiedliche Basisprimitive zeitnah aktiviert werden können. Die Matrix \mathbf{H} darf also weniger spärlich sein; λ kann demnach sehr klein bzw. sogar zu Null gewählt werden. Für eine Klassifikationsaufgabe sollte die Zerlegung möglichst eindeutig sein.

Anders als in der Trainingsphase reagiert der Optimierungsprozess nicht so sensibel, da in der Anwendungsphase nur noch eine der beiden Matrizen angepasst wird. Erfahrungsgemäß kann hier mit einem größeren Wert für λ gearbeitet werden. Allerdings gilt das in Abschnitt 6.2.1 diskutierte hier ebenso. Der Einfluss des Spärlichkeitskriteriums darf nicht zu groß werden.

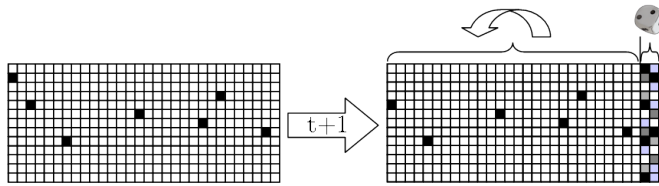


Abbildung 6.6.: Verschiebung der Aktivitäten:

Die Aktivitäten \mathbf{H}^j werden als Initialisierung des Optimierungsprozesses im nächsten Zeitschritt benutzt. Dazu werden sie um den entsprechenden Zeithorizont verschoben. Der unbekannte Teil wird zufällig aufgefüllt.

Standardmäßig impliziert eine neue Beobachtung als Eingabe für den NMF-Algorithmus eine neue zufällige Initialisierung des Optimierungsproblems. In diesem Falle hat die neue Beobachtung aber spezielle Eigenschaften. Sie besteht ja aus einem Zeitfenster, an welches die zuletzt beobachtete Position angefügt wurde. Da ein ansteigender Spaltenindex für einen späteren Zeitpunkt steht, bedeutet das, dass die Trajektorie mit zunehmender Zeit weiter nach links im Grid verschoben wird. Bei identischer Initialisierung findet man diese Verschiebung auch in den berechneten Aktivitäten zum nächsten Zeitschritt wieder. Um die Anzahl der Optimierungsschritte zu reduzieren, erscheint es sinnvoll, sich diese Eigenschaft zu Nutze zu machen. Dazu werden die Aktivitäten \mathbf{H}^j für die entsprechende Basiskomponente \mathbf{W}_j um die dem Zeitschritt entsprechende Gridanzahl nach links verschoben. Der noch unbekannte Teil der Aktivitäten, der für die neue Beobachtung verantwortlich zeichnet, wird zufällig aufgefüllt (siehe Abbildung 6.6). Dieses Aktivitätsmuster dient nun als Initialisierung für den Optimierungsprozess.

Um in der Lage zu sein, eine beobachtete Trajektorie vorhersagen zu können, muss der bis hierhin vorgestellte Algorithmus noch ein wenig erweitert werden. Dadurch, dass die Standard-NMF um die Translationsinvarianz erweitert wurde, können die Basisprimitive an eine beliebige Stelle im Grid verschoben werden. Dies hat zur Konsequenz, dass sie oder Teile von ihnen über den Rand hinausragen können. Bis zu diesem Punkt wurde die Rekonstruktion \mathbf{R} so definiert, dass sie dieselbe Dimension hat wie die Eingabematrix \mathbf{V} . Die durch das Herausragen

der Basiskomponenten entstandenen Zusatzinformationen mussten also abgeschnitten und damit verworfen werden. Um die Trajektorie in die Zukunft präzisieren zu können, wird nun einfach das Rekonstruktionsgrid \mathbf{R} auf der rechten Seite bzw. in die Zukunft erweitert (siehe Matrix \mathbf{P} in Abbildung 6.5). Auf diese Weise steht die zuvor abgeschnittene Information für die Prädiktion zur Verfügung.

Um einen maximalen Prädiktionshorizont zu erreichen, muss die Basis-komponente möglichst weit in die Zukunft hineinreichen können. Wie bereits erwähnt, wird zur Berechnung der Rekonstruktion eine Faltung zwischen den Aktivitäten und den Basisprimitive als Faltungskern durchgeführt. Es erweist sich an dieser Stelle als zweckmäßig, den Aufpunkt des Faltungskerns nicht in die Mitte der Basisprimitive, sondern an den linken Rand zu legen.

6.3. Experimentelle Evaluierung

Da für die Spatio-Temporale NMF eine andere Form der Bewegungsrepräsentation benutzt wird, ist es nicht ohne Probleme möglich, dieselbe Evaluierung, wie in Kapitel 4 und 5 beschrieben, durchzuführen. In den beiden Kapiteln liegt die Prädiktion direkt als Trajektorie vor, so dass ein direkter Vergleich möglich ist. Bei der Spatio-Temporalen NMF liegt die Trajektorie in der beschriebenen Grid-Repräsentation vor. Da eine Überführung in die Standardvariante der Trajektorienrepräsentation durchaus weitere Quellen für Fehler oder Ungenauigkeiten generieren kann, soll zur Evaluierung des eigentlichen Verfahrens direkt auf der Grid-Repräsentation gearbeitet werden. Insbesondere, da es die Darstellung als Grid erlaubt, Mehrdeutigkeiten zu kodieren, ginge durch eine Überführung in die eindeutige Trajektoriendarstellung Information verloren.

In den folgenden zwei Teilabschnitten sollen die wesentlichen Aspekte des vorgestellten Algorithmus genauer untersucht werden. Als erstes soll gezeigt werden, dass das Verschieben der Aktivitäten für die Prädiktion des nächsten Zeitschrittes hinsichtlich einer Entwicklung in Richtung eines echtzeitfähigen Einsatzes, einen enormen Zugewinn bringt. Danach soll die Qualität der Prädiktion genauer unter die Lupe genommen wer-

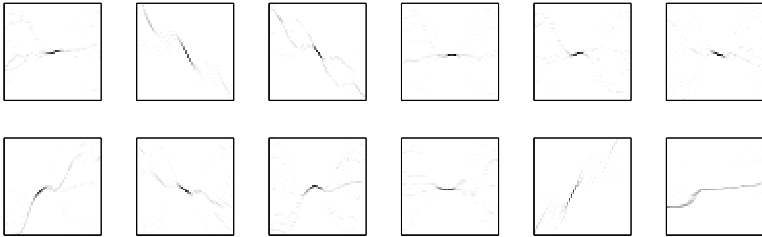


Abbildung 6.7.: Beispiele für Basisprimitive der Spatio-Temporalen NMF.: Die Werte jeder Gridzelle sind als Grauwert kodiert und reichen dabei von weiß für niedrige bis schwarz für hohe Werte. Ein spezieller Wert steht dabei für den Einfluss der jeweiligen Zelle. Somit eignen sich hellgraue Teile gut für eine unscharfe Überlagerung, während dunkelgrau bis schwarze Werte für ein eindeutigen Abschnitt auf der Trajektorie stehen.

den. Zur Evaluierung werden die bereits bekannten in Abschnitt 3.1.1 vorgestellten Bewegungsdaten von Personen verwendet.

Für die Durchführung der Experimente wurde die Größe der Basisprimitive mit 50×50 Gridzellen gewählt, wobei die Größe der Eingabedaten während des Trainings 500×50 Gridzellen für jede der Trajektorien des Trainingsdatensatzes betrug. Während der Anwendungsphase wurde die Größe auf 100×50 Gridzellen verkleinert. Die 100 Gridzellen entsprechen dabei dem Zeithorizont, auf dem die Prädiktion beruhen soll. Sinnvollerweise sollte eine Gridspalte, also ein Zeitschritt, in der Gridrepräsentation auch einem Zeitschritt des Trackingsystems entsprechen. Ein Beispiel für die trainierten Basisprimitive findet sich in Abbildung 6.7.

6.3.1. Verschieben der Aktivitäten

In Abschnitt 6.2.2 (vergleiche Abbildung 6.6) wurde diskutiert, dass die Informationen, die im vorangegangenen Zeitschritt berechnet wurden, als Initialisierung für den aktuellen Zeitschritt benutzt werden können. Ab-

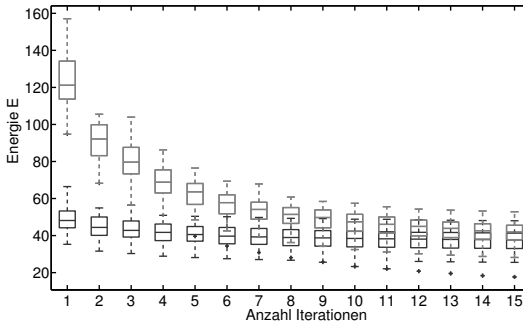


Abbildung 6.8.: Konvergenzverhalten der Spatio-Temporalen NMF: Die Kastengrafik zeigt die Charakteristik der Konvergenz der Energiefunktion (vergleiche Formel 6.2) für 15 Iterationsschritte. Um den oberen (hellgrauen) Graphen zur erzeugen, wurden die Aktivitäten zu jedem Zeitschritt zufällig initialisiert. Bei der unteren (dunkelgrauen) Kurve wurden dagegen die zeitlich verschobenen Aktivitäten des vorangegangenen Zeitschrittes benutzt.

bildung 6.8 stellt die in Gleichung 6.2 definierte Energiefunktion als Verlauf über die einzelnen Iterationsschritte (Schritte 1 bis 3 in Abschnitt 6.2.2) in einem Zeitschritt, d.h. für eine Prädiktion dar. Damit eine allgemeingültige Aussage entsteht das Experiment für mehrere Prädiktionen durchgeführt und Mittelwert und Varianz der Energiefunktion dargestellt.

In der Grafik werden beide Varianten, also mit zufälliger Initialisierung (obere blaue Kurve) und mit Initialisierung durch Verschieben der Aktivitäten (untere rote Kurve), gegenübergestellt. Die Anzahl der dargestellten Iterationsschritte beträgt 15. Diese geringe Anzahl reicht bereits aus, um den positiven Effekt zu beobachten. Es ist klar zu erkennen, dass der Algorithmus um etwa 10 Iterationsschritte schneller konvergiert. Für die Berechnung eines einzelnen Iterationsschrittes benötigt der Algorithmus 92 ms auf einem Rechner mit einem Intel T2050 Prozessor mit 1.6 GHz. Damit bietet die Verschiebung der Aktivitäten eine nicht geringe Einsparmöglichkeit hinsichtlich der Rechenzeit.

6.3.2. Prädiktionsgenauigkeit

Um die Qualität der Prädiktion evaluieren zu können, wird die Prädiktion mit der Gridrepräsentation des tatsächlichen Trajektorienverlaufes \mathbf{G} verglichen. Dieses Grid wird im Folgenden als Ground-Truth-Trajektorie bezeichnet. Das gewählte Vergleichsmaß S_{GT} soll eine Aussage treffen, wie gut die Prädiktion an der Stelle der Ground-Truth Trajektorie ist. Es sei daher wie folgt definiert:

$$\begin{aligned} S_{GT} &= \frac{1}{|T|} \sum_{t \in T} \frac{\sum_i (\bar{\mathbf{P}}_t \odot \mathbf{G}_t)^i}{\sum_i \mathbf{G}_t^i} \quad \text{mit } \bar{\mathbf{P}}_t = \frac{\mathbf{P}_t}{|\mathbf{P}_t|} \\ &= \frac{1}{|T|} \sum_{t \in T} \frac{\bar{\mathbf{P}}_t^T \cdot \mathbf{G}_t}{\sum_i \mathbf{G}_t^i} \end{aligned} \quad (6.33)$$

Hierbei wird für jede Zeitscheibe (Spalte in Gridrepräsentation) der Wert der Prädiktion an der Position der Ground-Truth-Trajektorie akkumuliert. Die zur Normierung verwendete Summe $\sum_i \mathbf{G}_t^i$ repräsentiert die Anzahl der belegten Gridzellen zum Zeitpunkt t . Dies ist notwendig, da mehrere (benachbarte) Gridzellen in einer Zeitscheibe von der Trajektorie belegt werden könnten. Dies geschieht, weil bei der Überführung in die Gridrepräsentation zwischen den Stützstellen der originalen Trajektorie mit Hilfe des Bresenham Algorithmus [Bresenham, 1965] interpoliert wird. Weiterhin wird durch die Länge des Prädiktionshorizontes dividiert, um unterschiedlich lange Prädiktionen vergleichbar zu machen. S_{GT} beträgt eins, wenn gilt $\mathbf{P} = \mathbf{G}$. Ansonsten gilt $S_{GT} \leq 1$, da die Einträge in \mathbf{P} den Wert eins nicht überschreiten.

Ein Wert von 1.0 steht für eine perfekte Prädiktion über den gesamten Prädiktionshorizont mit nur einer einzigen Prädiktionshypothese. Der Wert sinkt signifikant, sobald mehrere parallele Hypothesen entstehen.

Das Prädiktionsfenster enthält Überlagerungen der Basisprimitive und zwar den Anteil, der aus dem eigentlichen Rekonstruktionsteil herausragt. Ist nun also jeweils der Anfang unterschiedlicher Basistrajektorien geeignet, den letzten Teil der Trajektorie zu rekonstruieren, entstehen entsprechende Überlagerungen bzw. alternative Hypothesen im Prädiktionsteil. Diese Hypothesen sind allerdings in keiner Weise normiert.

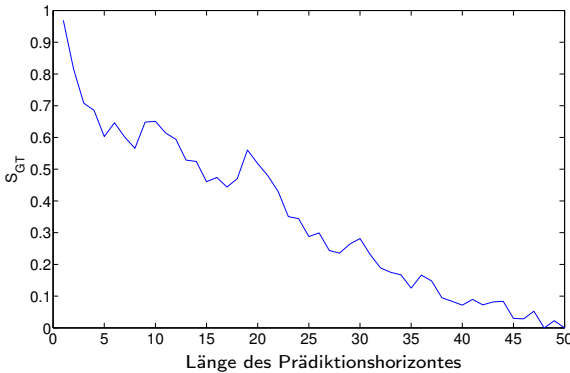


Abbildung 6.9.: Durchschnittlicher Ground-Truth-Fit:

Die mittlere Korrelation S_{GT} (siehe Gleichung 6.33) zwischen der Ground-Truth Trajektorie und der Prädiktion für unterschiedliche Längen des Prädiktionshorizontes abgetragen. Ein Wert von 1.0 steht dabei für eine perfekte Prädiktion über den jeweils betrachteten Prädiktionshorizont. Wie man es erwarten würde, sinkt die Genauigkeit der Prädiktion bei längeren Prädiktionen.

Eine starke Hypothese (hohe Werte) lässt darauf schließen, dass die entsprechende Basistrajektorie entsprechend gut für die Rekonstruktion geeignet ist. So ist es beispielsweise auch möglich, nur eine Prädiktion zu erzeugen, die allerdings sehr schwach ausgeprägt ist. Um solche Unterschiede ausgleichen zu können, wird die Prädiktionsmatrix \mathbf{P} spaltenweise, also für jede Zeitscheibe, normiert.

Die Basisprimitive können maximal um ihre zeitliche Breite aus dem rekonstruierten Grid \mathbf{R} herausgeschoben werden. Somit ergibt sich ein maximaler Prädiktionshorizont, der der Breite der Basisprimitive entspricht. Theoretisch könnte man, ähnlich wie bei ESN und LM, das bereits prädierte auch in die Rekonstruktion schieben, um dann, basierend auf der bereits durchgeführten Prädiktion, den Prädiktionshorizont zu erweitern. Dies soll aber für diese Auswertung keine Rolle spielen, da anders als bei ESN und LM, das Verfahren bereits ohne diesen Trick mehrere Schritte vorhersagen kann. Bei der gewählten Größe der Basisprimitive von 50×50 Gridzellen ist mit einer Auflösung von 60 Zeitschritten pro Sekunde eine Prädiktion von 0.83 Sekunden möglich. Die in den Beispieldaten vorliegenden 60 Hz werden allerdings meist

unterboten. Somit könnte eine wesentlich längere Prädiktion in Realzeit erreicht werden.

Um die Ergebnisse der Evaluierung besser einordnen zu können, sei noch erwähnt, dass der theoretisch maximale Prädiktionshorizont praktisch nicht erreicht werden kann, da die Basisprimitive eine verlässliche Grundlage im bekannten Teil brauchen und somit nicht beliebig in die Zukunft verschoben werden können. In Abbildung 6.9 ist dies dargestellt. Dazu wurde jeweils nur ein Prädiktionshorizont einer bestimmten Länge (abgetragen auf der x-Achse) betrachtet. Wie zu erwarten, sinkt die Prädiktionsgenauigkeit mit längerem Prädiktionshorizont. Nichtsdestotrotz fällt aber auf, dass die Abnahme recht gutmütig ist. Es kommt also zu keinem plötzlichen Zusammenbruch der Prädiktion. Da hier über mehrere Prädiktionen jeweils ein mittlerer Wert berechnet wurde, kann davon ausgegangen werden, dass das Ergebnis verallgemeinerbar ist. Allerdings ist zu beachten, dass das Resultat abhängig von den verwendeten Basisprimitiven ist. Wenn nur sehr kurze Basisprimitive ausgebildet werden, kann auch nur ein sehr kleiner Prädiktionshorizont erzielt werden. Bei den Experimenten kam nur ein Satz an Basisprimitiven zum Einsatz.

Für Abbildung 6.10 wurde eine beliebige Trajektorie ausgewählt und deren Verlauf zu verschiedenen Zeitpunkten prädiziert. Die Ergebnisse wurden als Histogramm zusammengefasst. Hierbei wurde der gesamte theoretisch mögliche Prädiktionshorizont zu Grunde gelegt. Zu beachten ist, dass die Prädiktionsfähigkeit recht stabil ist und keine Werte unter 0.1 erreicht werden. In gutmütigen Fällen werden sogar Werte von 0.4 erzielt.

Um die Prädiktion einschätzen zu können, ist in Abbildung 6.11 ein Vergleich zwischen den Prädiktionsergebnissen der Spatio-Temporalen NMF und zweier Kalmanfilter [Kalman, 1960] dargestellt. Dieser Vergleich erfüllt denselben Zweck, wie das Referenzverfahren in Kapitel 4 und 5. Zusätzlich wird durch die Verwendung des Kalmanfilters eine Modellierung der Unsicherheit ermöglicht. Beim ersten Kalmanfilter (Abbildung 6.11 (b)) wird die Geschwindigkeit mitgeschätzt und im Systemmodell einbezogen. Für den zweiten Kalmanfilter (Abbildung 6.11 (c)) wird zusätzlich noch die Beschleunigung für das Modell benutzt. Ab der grünen Linie erfolgt nur noch eine Prädiktion. Für den Kalmanfilter wird ab diesem Zeitpunkt der Updateschritt weggelassen.

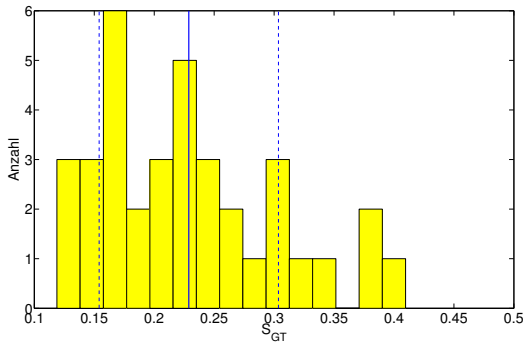
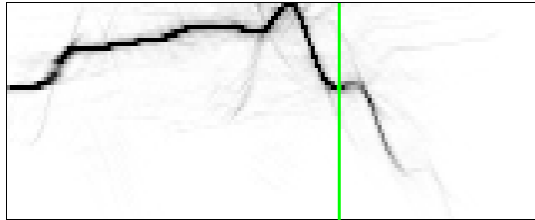


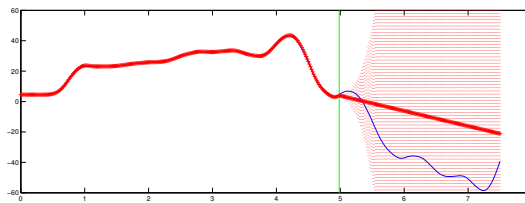
Abbildung 6.10.: Prädiktionsgenauigkeit entlang einer Beispieltrajektorie: Die Grafik zeigt den Ground-Truth-Fit mehrerer Prädiktionen entlang einer Beispieltrajektorie als Histogramm. Insgesamt wurden 36 Prädiktionen durchgeführt. Dies geschah jeweils zu jedem 10. Zeitschritt der Trajektorie. Ein Wert von 1.0 steht für eine perfekte Prädiktion über den gesamten Prädiktionshorizont. Die durchgehende und die gestrichelten vertikalen Linien sollen den Mittelwert bzw. die Varianz darstellen.

6.3.3. Erkennungsaufgabe

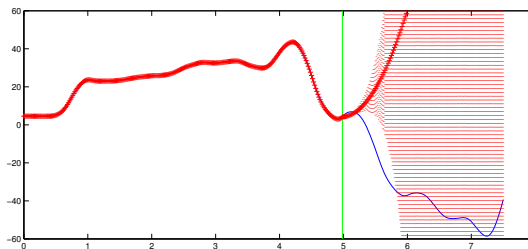
Dieser Teil der Evaluierung soll die prinzipielle Möglichkeit der Klassifikation, basierend auf den spärlichen Aktivitäten des vorgestellten Algorithmus, zeigen. Dabei soll nicht nach einem idealen Klassifikator gesucht, sondern eher untersucht, ob die spärliche Repräsentation der Aktivitäten für eine Erkennungsaufgabe geeignet ist. Ziel soll es daher sein, die jeweils durchgeführte Aktion der Bewegungsdaten aus Abschnitt 3.1.1 (werfen, laufen, klopfen, heben) zu klassifizieren. Dazu wurde in MatLab ein sogenanntes Layered Recurrent Neural Network trainiert. Dabei handelt es sich um eine mehrschichtige Version des bekannten Elman-Netzwerkes. Zum Trainieren wurde ein Standard error back-propagation Algorithmus verwendet. Der Klassifikator erzielte bei dem 4-Klassen-Problem eine Genauigkeit von 40%. Das schlechte Abschneiden liegt an der noch immer großen Eingabedimension, da für die ersten Tests die Aktivitäten direkt verwendet wurden. Für die weitere Forschung muss der spärliche Charakter weiter in den Vordergrund



(a)



(b)



(c)

Abbildung 6.11.: Vergleich der Prädiktionsergebnisse:

In den drei Grafiken sind die Prädiktionsergebnisse der Spatio-Temporalen NMF (a), und eines Kalmanfilters mit Geschwindigkeitsmodell (b) und eines Kalmanfilters mit Beschleunigungsmodell (c) dargestellt. Die Prädiktion beginnt jeweils ab der eingezeichneten grünen Linie. Die gepunktet roten Linien bei den Grafiken des Kalmanfilters veranschaulichen die Kovarianzschätzung.

gerückt werden. Dies unterstützt dann auch den Lernalgorithmus, da momentan die große Anzahl der Eingabewerte, die null sind, Probleme bereitet.

6.4. Fazit

In diesem Kapitel wurde ein Ansatz zur Dekomposition von Bewegungsabläufen vorgestellt. Dabei wurde als Grundlage auf die nicht-negative Matrixzerlegung zurückgegriffen. Um die nicht-negative Matrixzerlegung auf die speziellen Anforderungen im Zusammenhang mit der Bewegungsanalyse anzupassen, wurde das Spärlichkeitskriterium mit der Transformationsinvarianz kombiniert. Die Dekomposition der Trajektorien erfolgt in einer grid-basierten Repräsentationsform. Da die Basis-komponenten datengetrieben bestimmt werden, kann das Verfahren auf beliebigen Arten von Bewegungen Anwendung finden.

Im Experimenterteil konnte gezeigt werden, dass das Verschieben der Aktivitäten klare Vorteile im Hinblick auf einen Echtzeiteinsatz bringt. Ebenso konnte nachgewiesen werden, dass mit dem Verfahren eine Prädiktion der weiteren Bewegung möglich ist. Da diese Prädiktion durch das Überlagern unterschiedlicher Basiskomponenten zustande kommt, ist es möglich, mehrere Hypothesen für den weiteren Verlauf zu generieren. Somit ist eine wichtige Grundlage zum Lösen des in Abschnitt 1.2 beschriebenen Navigationsaspektes geschaffen.

Darüberhinaus wurde demonstriert, dass es prinzipiell möglich ist, die Aktivierungsmuster zum Lösen der Erkennungsaufgabe einzusetzen. Auch wenn die vorgestellten Ergebnisse noch unzureichend sind, ist dennoch das Potential zu erkennen. Um die Klassifikation weiter zu verbessern, muss die Spärlichkeit in den Aktivierungen forciert werden. Momentan kommt es noch zu verwaschenen Aktivierungszeitpunkten. Darüberhinaus sollte die Spärlichkeit für den Klassifikator besser genutzt werden. Die aktuelle Variante direkt auf den Aktivierungsmatrizen zu klassifizieren, hat sich, wie gezeigt, als nicht sinnvoll erwiesen.

Das Forcieren der Spärlichkeit hat weiterhin den Vorteil, dass klare Zeitpunkte der Aktivierung definiert werden können. Da diese Zeitpunkte den Wechsel der Bewegung symbolisieren, können sie als markante

Punkte für die Aufmerksamkeitssteuerung angesehen werden. Dies wurde allerdings nicht weiter untersucht und bietet einen möglichen Ansatzpunkt für weitere Forschung.

Im Gegensatz zu Echo State Netzwerken und Local Models wird bei der Spatio-Temporalen NMF die Prädiktion in einem Schritt errechnet. Der maximale Prädiktionshorizont wird dabei durch die Länge der Basisprimitive festgelegt. Um einen längeren Zeitraum prädizieren zu können, wäre es möglich, das Prinzip der Mehrschrittprädiktion, welches auch bei Echo State Netzwerken und Local Models zum Einsatz kommt, zu verwenden. Dazu muss einfach der prädizierte Teil der Gridrepräsentation als neue Beobachtung angesehen werden. Für weitere Untersuchungen wäre dabei interessant, wie sich der Spärlichkeitsparameters auf die Anzahl der parallelen Hypothesen auswirkt.

Kapitel 7

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde sich mit der Analyse von Bewegungstrajektorien beschäftigt. Dabei stand die Anwendbarkeit im Bereich der mobilen Assistenzrobotik im Vordergrund. Insbesondere sollte untersucht werden, inwieweit Informationen aus den Bewegungstrajektorien zur Unterstützung der Interaktion zwischen Mensch und Roboter genutzt werden können.

Die Motivation dieser Arbeit, sich ausschließlich mit Bewegungsabläufen zu beschäftigen, wurde in den Fähigkeiten und dem Verhalten von Tieren und Menschen gefunden. In der Biologie finden sich viele Beispiele, dass die Bewegungsanalyse eine wichtige Rolle spielt. Aus ihr werden zusätzliche Informationen für höhere kognitive Fähigkeiten gewonnen.

Der Blick in die aktuelle Forschung im Umfeld der mobilen Robotik verrät, dass auch hier bereits die Erkenntnis gereift ist, die Analyse von Bewegungen zu nutzen. Allerdings können an vielen Stellen nur sehr spezielle Lösungsvorschläge gefunden werden. In vielen Fällen wird eher auf Designerwissen zurückgegriffen und beispielsweise das Wissen über zu erwartenden Bewegungen explizit modelliert. Dadurch entstehen in den meisten Fällen Konzepte, die nur schwer auf andere Probleme übertragen werden können. Außerdem ist bei vielen Arbeiten zu beobachten, dass noch keine einheitliche Systematisierung existiert. Diese Arbeit hat

sich daher auf systematischem Wege dem Problem gestellt und Verfahren untersucht und entwickelt, die ohne Probleme auf eine Vielzahl von Anwendungsfeldern übertragbar sind.

Am Anfang der Arbeit wurde ein begleitendes Beispiel eines Empfangsroboters in einem Theater vorgestellt. Mit diesem konnte eingegrenzt werden, für welche Aufgabe bzw. Aspekte des Beispiels die Auswertung von Bewegungen relevant ist. Aufgrund dieser Teilaspekte kristallisierten sich Verfahren zur Prädiktion bzw. zur Klassifikation von Bewegungsabläufen als Hauptkern der Arbeit heraus. Die Teilaspekte des Beispiels werden in Abschnitt 7.1 noch einmal aufgegriffen, und es wird beleuchtet, ob die untersuchten Verfahren für die einzelnen Aspekte Anwendung finden können.

Bevor sich diesem Hauptkern zugewandt wurde, musste die Frage geklärt werden, welcher Vorverarbeitung die Trajektorie unterzogen werden muss. Als erstes wurde daher eine große Auswahl an Repräsentationsmöglichkeiten von Trajektorien systematisch vorgestellt und untersucht. Dabei ergab sich, dass sich, je nachdem welches Ziel im Rahmen der Aufgabenstellung verfolgt wird, jeweils andere Repräsentationsformen als geeignet erweisen. Es kann daher keine allgemeingültige Empfehlung bezüglich der Repräsentation gegeben werden. Je nach Aufgabenstellung muss individuell entschieden werden, wie die Vorverarbeitung der Trajektorie erfolgen soll. Um diese Entscheidung zu erleichtern, beschreibt die Arbeit ein Verfahren zur Merkmalsauswahl, das es ermöglicht, eine Wahl unabhängig vom Designerwissen zu treffen.

In vielen Verfahren spielen Distanz- bzw. Ähnlichkeitsmaße eine entscheidende Rolle. In den meisten Fällen wird dabei auf die Euklidische Distanz zurückgegriffen. In dieser Arbeit wurde die Frage untersucht, welche Alternativen zur Euklidischen Distanz bei der Anwendung auf realen Bewegungstrajektorien existieren und auch sinnvoll eingesetzt werden können. Dabei wurden klassische Distanzmaße näher beleuchtet. Diese geben aber nur einen skalaren Wert, der die Stärke der Unähnlichkeit beschreibt, wieder. Besonders bei Bewegungsinformationen, sind aber auch Informationen über die Art der Unähnlichkeit wünschenswert. Deshalb wurden darüberhinaus noch Methoden vorgestellt, mit denen eine Aussage über Verschiebung, Rotation und Skalierung der zu vergleichenden Bewegungen getroffen werden kann. Eine Vielzahl der untersuchten Methoden kam dabei erstmals für den Vergleich von realen

Bewegungstrajektorien zum Einsatz. Da die Ergebnisse der Vergleiche der Distanzmaße keinen Nachteil der Euklidischen Distanz aufzeigen, wurde für die weiteren Untersuchungen die Euklidische Distanz verwendet.

Wie bereits erwähnt bilden den eigentlichen Kern der Arbeit drei Ansätze, die unter anderem die Prädiktion von Bewegungen ermöglichen. Üblicherweise werden Prädiktionsverfahren aus dem Bereich der Zeitreihenanalyse mit synthetischen chaotischen oder periodischen Signalverläufen evaluiert. In dieser Arbeit wurden die unterschiedlichen Prädiktionsverfahren erstmals auf realen zwei- bzw. dreidimensionalen Bewegungstrajektorien vergleichend untersucht. Es konnte dabei gezeigt werden, dass diese Verfahren für die Prädiktion von Bewegungstrajektorien bereits sehr gute Ergebnisse erzielen können. Um die Ergebnisse weiter verbessern zu können, wurden unter anderem unterschiedliche Vorverarbeitungen der Trajektorie untersucht. Damit konnten die Prädiktionsalgorithmen für die Zusammenarbeit mit Trackingsystemen mit niedriger Auszeichnungsrate optimiert werden.

Als erstes Verfahren wurden Echo State Netzwerke beleuchtet. Diese Art der rekurrenten neuronalen Netze ist in ihrer Struktur stark an das biologische Vorbild angelehnt. Im Wesentlichen stellen sie aus ingenieurtechnischer Sicht ein System von Oszillatoren dar. Damit liegt ihre Fähigkeit darin, Signale zu generieren, die eine Überlagerung von periodischen Basissignalen darstellen. Diese Fähigkeit wird für die Prädiktion ausgenutzt. Es konnte experimentell nachgewiesen werden, dass Echo State Netzwerke in der Lage sind, auch Bewegungstrajektorien zu prädizieren. Dabei konnte im Rahmen dieser Arbeit, durch den besonderen Charakter der Bewegungstrajektorien, die Anzahl der Reservoirneuronen erheblich reduziert werden. Zusätzlich zu den unter idealen Bedingungen aufgezeichneten Körperteiltrackerdaten wurden auch Daten eines Personentrackingsystem verwendet. Mit diesen Daten wurde evaluiert, welche Vorverarbeitungsschritte notwendig sind, um eine bestmögliche Prädiktion zu erhalten.

Ähnlich gute Ergebnisse, wie mit Echo State Netzwerken, konnten mit den Local Models Verfahren erzielt werden. Die prinzipielle Idee der Local Models besteht darin, aus bereits beobachteten Bewegungsabläufen auf die aktuelle Bewegung zu schließen und sie entsprechend zu prädizieren. Auch bei den Local Models wurden neben der prinzipiellen

Eignung zur Prädiktion von Bewegungen zusätzlich notwendige Vorverarbeitungsschritte untersucht, die helfen, die Prädiktion zu verbessern. Die Prädiktion der Local Models basiert auf der Auswahl ähnlicher, bereits bekannter Bewegungsabläufe. Prinzipiell können diese Beobachtungen von unterschiedlichen Trajektorien stammen. Diese Möglichkeit wurde durch den in dieser Arbeit vorgestellten Ansatz ausgenutzt, um die Bewegungswissensbasis beliebig zu erweitern. Da ein beliebiges Hinzufügen von Bewegungsabläufen einen negativen Einfluss auf die Berechnungszeit hat, musste eine Möglichkeit gefunden werden, die Wissensbasis sinnvoll einzubinden. Die Lösung besteht darin, eine Clusterung des Einbettungsraums durchzuführen. Durch die Clusterung konnte nicht nur die Prädiktion unterstützt werden. Die so zur Verfügung stehende Wissensbasis kann für jede bekannte Bewegung zusätzliche Informationen, wie die Zugehörigkeit zu einer bestimmten Klasse (z. B. über die Art der Bewegung) speichern. In dieser Arbeit wurde ein Ansatz vorgestellt, eine solche zusätzliche Information ohne erheblichen Mehraufwand bezüglich der Rechenzeit auszunutzen. Es wurde damit zusätzlich die Möglichkeit geschaffen, die aktuelle Beobachtung zu klassifizieren.

Als letztes Verfahren stellt diese Arbeit ein neu entwickeltes Verfahren vor, welches die nicht-negative Matrixfaktorisierung erweitert, um mit raum-zeitlichen Trajektorien angewendet werden zu können. Die Nicht-negative Matrixfaktorisierung wird insbesondere in der Bildverarbeitung dazu verwendet, um das Quellentrennungsproblem zu lösen. Dazu werden die Bilddaten in Basisbilder zerlegt. Die Basisbilder sind allerdings nicht fest vorgegeben, sondern können entsprechend trainiert werden. Bei der Anwendung auf Bewegungsdaten werden die Trajektorien in eine raum-zeitliche Gridrepräsentation überführt. Diese Form der Repräsentation hat den Vorteil, dass es möglich ist, mehrere Hypothesen abbilden zu können. Durch die Verwendung einer Spärlichkeitsforderung und das Zulassen affiner Transformationen der Basiskomponenten bilden sich einzelne Trajektorienabschnitte als Basissignale heraus. Diese Trajektorienabschnitte können dann hintereinander verknüpft werden, um die Trajektorie zu reproduzieren, d. h. die Charakteristik der Trajektorie mit Hilfe des Basisalphabetes zu beschreiben. Über die reine Beschreibung der Trajektorie hinaus, schafft diese Arbeit durch eine spezielle Randbetrachtung die Möglichkeit den weiteren Verlauf der Bewegung präzisieren zu können. Zusätzlich kann die Information, wie die

einzelnen Basistrajektorien aneinandergereiht werden, zum Klassifizieren genutzt werden.

Nach der Zusammenfassung der in dieser Arbeit untersuchten und entwickelten Methoden bleibt noch zu klären, welches Verfahren sich für welchen Einsatzzweck besonders eignet. Dies soll im Folgenden anhand des bereits erwähnten begleitenden Beispiels erfolgen.

7.1. Szenariospezifische Anwendbarkeit

In Abschnitt 1.2 wurde ein Anwendungsbeispiel vorgestellt. Am Ende der Arbeit soll zusammengefasst werden, inwieweit die Anforderungen der einzelnen Aspekte der Anwendung durch die vorgestellten Verfahren erfüllt werden. Dabei stehen in erster Linie die Verfahren aus den Kapiteln 4 bis 6 im Mittelpunkt.

Da die Intention dieser Arbeit nicht darin bestand, eine vollständige Lösung für ein konkretes Szenario zu schaffen, soll im Folgenden darauf eingegangen werden, wie die vorgestellten Algorithmen sich in die jeweiligen Anwendungsbeispiele einfügen. Insofern ist dieser Abschnitt auch als Anregung für weiterführende Arbeiten gedacht.

7.1.1. Prädiktionsaufgaben

Die Intention des Beispiels zur Navigation ist es zu zeigen, dass aus Trajektorien gewonnenes Wissen für die Navigation insbesondere in Umgebungen mit dynamischen Hindernissen notwendig ist. Die Grundidee geht dabei davon aus, dass eine Prädiktion der Trajektorien der dynamischen Objekte verwendet wird, um die Navigationsstrategie entsprechend anzupassen. Diese Idee lässt sich auch leicht auf Interaktionsaufgaben zwischen Mensch und Roboter anwenden. Der wesentliche Unterschied besteht darin, dass im Falle der Navigation in erster Linie zweidimensionale Daten beobachtet werden, wo hingegen bei dem Beispiel der Begrüßung per Handschlag eher dreidimensionale Daten präzisiert werden müssen.

Echo State Netzwerke bieten eine gute Möglichkeit, um Bewegungstrajektorien zu präzidieren. Allerdings besteht bei ihnen das Problem, dass die Trajektorie eine gewisse Zeit beobachtet werden muss. Um dieses Problem im Sinne der Navigationsaufgabe zu umgehen, ist es möglich eine Kombination aus einer reaktiven und einer adaptiven Navigationsstrategie zu wählen. Reicht die Zeit für die notwendige Beobachtung nicht aus, wird durch einfaches Ausweichen oder Stoppen auf das Gegenüber reagiert. Kommt eine Prädiktion zustande, kann der Bewegungspfad des Roboters entsprechend umgeplant bzw. die Geschwindigkeit angepasst werden.

Bei den Local Models müssen solche Überlegungen nicht getroffen werden. Durch die in dieser Arbeit entwickelte Erweiterung, die die Local Models mit einer Wissensbasis an bekannte Bewegungen ausstattet, kann frühzeitig mit der Prädiktion begonnen werden. Durch zusätzliche Informationen, die der Wissensbasis mitgegeben werden können, kann auch entschieden werden, ob die Person eher an einer Interaktion interessiert ist oder nicht. Somit kann die Navigationsstrategie zwischen Ausweichen und Abfangen angepasst werden.

Das dritte Verfahren der Spatio-Temporalen NMF basiert auf einer Gridrepräsentation der Trajektorien. Da viele Navigationsverfahren ebenso auf grid-basierte Ansätze zurückgreifen, bietet sich die Integration des Verfahrens an. Durch die Gridrepräsentation werden darüber hinaus auch mehrere mögliche Hypothesen gebildet. Bei der Planung des Roboterbewegungspfad es können diese entsprechend der Stärke ihrer Ausprägung berücksichtigt werden.

Alle drei Verfahren sind in der Lage die Prädiktion von Bewegungstrajektorien durchzuführen. Damit wurden der Navigations- und Interaktionsaspekt aus Abschnitt 1.2 gelöst. Die Wahl des einzelnen Verfahrens hängt stark von der konkreten Anwendung ab. Local Models sind dann gut geeignet, wenn eine geeignet große Wissensbasis zur Verfügung steht. Bei der spatio-temporalen NMF bietet sich die Repräsentation der Bewegung als Grid für die Integration in grid-basierte Navigationsalgorithmen an. Echo State Netzwerke spielen ihre Vorteil aus, wenn die Prädiktion ausschließlich auf der Beobachtung erfolgen soll.

7.1.2. Erkennungsaufgaben

Im Rahmen der Interaktion mit mobilen Assistenzrobotern soll es möglich sein auch aus Bewegungstrajektorien bestimmte Aussagen über die Person und ihre Handlungen ableiten zu können. In Kapitel 3 wurde bereits an einem konkreten Erkennungsszenario gezeigt, wie die Auswahl einer geeigneten Repräsentation erfolgen soll. Der Einfachheit halber wurde für die Klassifikationsaufgabe auf ein Multi-Layer Perzeptron zurückgegriffen. Da ein Multi-Layer Perzeptron selbst nicht in der Lage ist, zeitliche Abläufe zu codieren, wurde hier entsprechend eine Vorverarbeitung in einem definierten Zeitfenster durchgeführt.

Für Echo State Netzwerke wurde die Möglichkeit, eine Klassifikation durchzuführen, nicht weiter untersucht. Theoretisch kann die Ausgabeschicht auch als Klassifikator genutzt werden. Im Prinzip würden die Ausgabeneuronen dann die unterschiedlichen Zustände des Reservoirs den entsprechenden Klassen zuordnen. Da die Zustände des Reservoirs beliebig komplex sein können, bleibt zu prüfen, ob eine einzelne Schicht nach dem Reservoir für die Klassifikation von Bewegungsabläufen ausreichend ist.

Mit der Erweiterung der Local Models um eine zusätzliche Wissensbasis ist es möglich, je nach enthaltenem zusätzlichem Wissen, eine Klassifikation durchzuführen. In Abschnitt 5.5.2 konnte experimentell bestätigt werden, dass durch diesen Ansatz gute Ergebnisse zu erzielen sind.

Im Falle der Spatio-Temporalen NMF besteht zumindest theoretisch die Möglichkeit, eine Klassifikation aufgrund der Abfolge der Basisprimitiven durchzuführen. In den Experimenten konnten allerdings nicht wirklich überzeugende Klassifikationsergebnisse erreicht werden, wobei hier vermutlich gerade die anvisierte Spärlichkeit zu den schlechten Resultaten geführt hat. Hier existiert durchaus noch Potenzial, die Aktivitäten auf sinnvolle Art und Weise dem Klassifikator zur Verfügung zu stellen. Hierzu sind weitere Untersuchungen notwendig und erfolgversprechend.

Die vorliegenden Ergebnisse lassen das Local Model als das Mittel der Wahl erscheinen, wenn es um das Lösen eines Erkennungsproblems geht. Die Ergebnisse der spatio-temporalen NMF sind auf den ersten Blick nicht wirklich überzeugend. Wie in Abschnitt 6.3.3 diskutiert wurde, besteht dabei allerdings noch Verbesserungspotenzial.

7.2. Erweiterungsmöglichkeiten

In Kapitel 3 wurde eine Reihe von möglichen Repräsentationsformen für Bewegungsabläufe untersucht. Dabei konnte geschlussfolgert werden, dass die Art der Repräsentation sehr stark von der jeweiligen Aufgabenstellung abhängt. In diesem Zusammenhang wurde auf Basis der Transinformation und Verbundtransinformation eine Variante zur semi-automatischen Auswahl der geeigneten Repräsentationsform vorgestellt. Für den Einsatz in einem mobilen Assistenzroboter ist dies momentan noch ausreichend, da in vielen Fällen eine vordefinierte Aufgabe gelöst werden soll. Wenn in zukünftigen Systemen allerdings mehr auf die Fähigkeit des Lernens bzw. Explorierens Wert gelegt wird, brauchen diese Systeme bereits eine breite Palette an möglichen Repräsentationsformen, aus denen vollautomatisch die entsprechend Passende ausgewählt werden kann. Um dies zu ermöglichen, sollten weitere Untersuchungen zur Featureauswahl durchgeführt werden.

Der größte Teil der Untersuchungen in dieser Arbeit beschränkte sich auf einzelne Trajektorien. Einzig im Kapitel C wurden Ähnlichkeitsmaße untersucht, mit denen es möglich ist, mehrere Trajektorien miteinander zu vergleichen. Allein mit diesem Ähnlichkeitsmaß ist es allerdings nicht möglich, kausale Zusammenhänge abzuleiten. Stellt man sich z. B. zwei Personen vor, die sich gegenseitig einen Ball zuwerfen, nützt die Aussage über die Ähnlichkeit der jeweiligen Handtrajektorien wenig. Dies gilt insbesondere dann, wenn beide eine andere Wurftechnik anwenden. Beide Trajektorien stehen aber dennoch in Beziehung. Um solche Beziehungen ermitteln zu können, muss also auf den zeitlichen Zusammenhang der beiden Trajektorien eingegangen werden. Ein einfacher, aber vielversprechender Ansatz bietet die Granger-Kausalität, die ein Vergleichsmaß für die zeitliche Abhängigkeit von Signalen zur Verfügung stellt. Das Maß selbst wurde für die Untersuchung ökonomischer Zeitreihen entwickelt, aber auch bereits für die Analyse von Biosignalen benutzt. In der Dissertation [Hesse, 2009] findet sich zu den einzelnen Varianten der Granger-Kausalität ein verständlicher Überblick.

In Rao [Rao et al., 2002] wird eine Möglichkeit vorgestellt, eine Trajektorie in Abschnitte zu unterteilen. Dazu wird die Krümmung benutzt um signifikante Stellen zu ermitteln, bei denen die Unterteilung erfolgt. Diese Punkte beschreiben Ereignisse (events) bei denen die Bewegung von

einem Zustand in einen anderen wechselt. Je nach Sichtweise bzw. Aufgabenstellung ist ein solcher Wechsel jeweils anderes ausgeprägt. Will man beispielsweise eine mögliche Verschlechterung des Gesundheitszustandes einer Person erkennen, könnte der Beginn einer Abweichung von einem normalen Verhalten bzw. der üblichen Bewegungsweise der Person ein wichtiger Indikator sein. Dadurch sollte das System veranlasst werden, weitere und möglicherweise aufwändigere Klassifikationen durchzuführen. Ebenso existiert eine Vielzahl an Beispielen, bei denen die Richtungsänderung eine Rolle spielt. In solchen Fällen reicht möglicherweise das Betrachten der Krümmung, wie in [Rao et al., 2002] vorgestellt, aus. Aber auch die in dieser Arbeit untersuchten Verfahren sind theoretisch in der Lage, signifikante Punkte zu bestimmen.

Bei der Vorstellung des begleitenden Beispiels wurde bereits darauf hingewiesen, dass für die Umsetzung des kompletten Szenarios weitere Forschungsanstrengungen notwendig sind. Im Rahmen dieser Arbeit wurde für die beleuchteten Aspekte mehr als nur eine Grundlage geschaffen. Damit stellt sie einen wichtigen ersten Schritt zur Integration und Akzeptanz mobiler Serviceroboter in dynamischen Umgebungen dar.

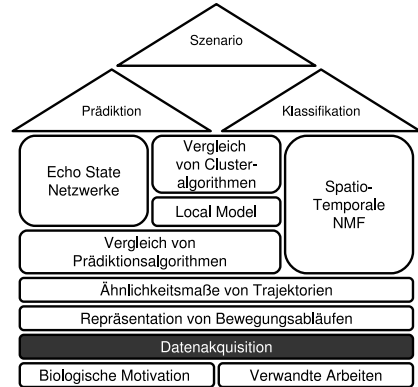
Nichtsdestotrotz ist es aber eben nur der erste Schritt und weitere müssen folgen. In den auf dieser Arbeit aufbauenden Projekten sollte der Fokus auf der Integration der Bewegungsprädiktionen in die Navigationsstrategie stehen. Wie geht man also mit dem prädierten Wissen um?

Darüberhinaus wurden in dieser Arbeit die Erkennungsaspekte ausschließlich auf Basis der Bewegungstrajektorien untersucht. Wie auch schon bei der Betrachtung der biologischen Motivation klar geworden ist, sollte die Bewegung nur einer der vielen Hinweise für eine endgültige Entscheidung sein. Auch wenn, wie diese Arbeit zeigen konnte, dass in der Bewegung wesentliche Information enthalten ist, macht es das Gehirn vor, dass sich eine Entscheidung, beispielsweise zur Auswahl einer Handlungsstrategie, auf mehrere Informationen stützen sollte.

Wenn die Forschung im Bereich mobile Robotik diese weiterführenden Fragestellungen gelöst hat, muss letztendlich der Mensch entscheiden, ob er, wie im Beispiel, von einem Empfangsroboter zu seinem Theater- oder Museumsbesuch begrüßt werden möchte.

Anhang A

Visuelle Bewegungs- segmentierung



Bewegungssegmentierung stellt in vielen Bereichen der Bildverarbeitung ein wesentliches Schlüsselement dar. Beispielsweise kann bei Algorithmen zur Videokompression die Datenmenge drastisch reduziert werden, wenn die Regionen mit kohärenter Bewegung bekannt sind [Torres et al., 1996]. Ebenso spielt die Bewegungsinformation bei der Auswertung von Szenen, wie z. B. bei der Katalogisierung von Videos [Snoek und Worring, 2005] oder in Überwachungsszenarien [Hu et al., 2004] eine Rolle. Darüberhinaus ermöglicht es mobilen Robotern, neben der Position auch die Geschwindigkeit sich bewegender Objekte zu kennen. Somit ist gerade in hochgradig dynamischen Umgebungen (z. B. in Menschenmengen) eine effektivere Hindernisvermeidung möglich.

In Abbildung A.1 ist schematisch der Ablauf der Segmentierung für einen Zeitschritt dargestellt. An dieser Stelle soll ausschließlich die Funktionsweise des Algorithmus von Interesse sein. Eine experimentelle Evaluierung findet sich in [Lau, 2007]¹ und [Hellbach et al., 2007].

¹Die Diplomarbeit wurde vom Autor im Rahmen des Projektes betreut.

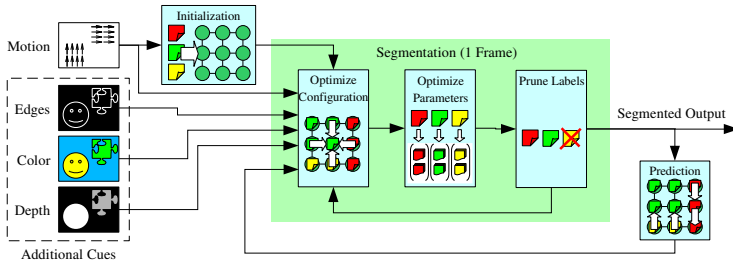


Abbildung A.1.: Schematischer Ablauf der Bewegungssegmentierung: *Der optische Fluss wird zur Initialisierung verwendet. Neben dem optischen Fluss können auch andere Cues eingebunden werden. Die Segmentierung des Bildes geschieht durch Optimierung eines Markov Random Fields. Die notwendigen Prototypenparameter werden nach erfolgter Segmentierung festgelegt. Das Ergebnis wird im nächsten Zeitschritt zur Initialisierung verwendet.*

Das Ziel der Segmentierung ist es, Hypothesen von sich bewegenden Objekten zu generieren. Dabei soll die Art des Objekts (z. B. eine Person oder ein Auto) keine Rolle spielen. Deshalb beruht das System in erster Linie auf dem optischen Fluss, der die aus Bilddaten gewonnene Bewegungsinformation repräsentiert. Dieser wird durch weitere beliebig erweiterbare Cues unterstützt.

Kern des Systems stellt ein zweistufiger Optimierungsprozess auf Basis von Markov Random Fields. Im ersten Schritt wird eine möglichst optimale Segmentierung entsprechend vorgegebener Eigenschaften für die Segmente gesucht. Diese Eigenschaften werden auch als Prototypenparameter bezeichnet. Sie umfassen mindestens den Bewegungsvektor $(\Delta x, \Delta y)$, der die durchschnittliche Bewegung der Segmente angibt. Sofern andere Cues eingebunden sind, können hier weitere Parameter ergänzt und zur Segmentierung herangezogen werden, wie beispielsweise Farbe oder Tiefeninformation.

Um zu vermeiden, dass diese Parameter durch einen Designer festgelegt werden müssen, werden im zweiten Schritt die Werte entsprechend der gefundenen Segmentierung angepasst. Somit ist der Algorithmus in der

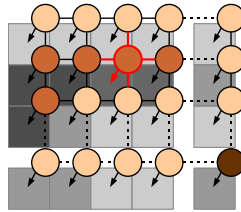


Abbildung A.2.: Markov Random Field zur Bildsegmentierung: *Jeder Knoten des Markov Random Field Graphen ist einem Pixel zugeordnet. Die den Knoten im Graphen zugeordneten Labels stehen dabei für die Zuordnung zu den entsprechenden Segmenten.*

Lage, sich an beliebige Objekte anzupassen und diese, sofern sie sich bewegen, zu segmentieren.

Im folgenden Abschnitt soll in verkürzter Weise das Prinzip der Markov Random Fields erläutert werden. Dem mit Markov Random Fields vertrauten Leser sei daher empfohlen, in Abschnitt A.2 weiterzulesen.

A.1. Markov Random Fields

Prinzipiell werden Markov Random Fields dazu verwendet, den Knoten eines beliebigen Graphen eine Labelinformation zuzuordnen (siehe Abbildung A.2). Die gefundene Zuordnung der Beschreibung durch Labels ist im Sinne der jeweils gestellten Aufgabe eine optimale Zuordnung. Dadurch sind Markov Random Fields vielfältig einsetzbar.

Bei der Anwendung auf Bilddaten definiert man den Graphen im allgemeinen so, dass für jeden Pixel ein Knoten im Graphen existiert. Die Verknüpfung der Knoten durch Kanten erfolgt entsprechend der Nachbarschaft der Pixel. In [Li, 2001] findet sich eine sehr gute Zusammenfassung zur Anwendung von Markov Random Fields im Bereich Bildverarbeitung. Dort können auch weitere mathematische Details nachvollzogen werden.

Neben dem Konzept der Markov Random Fields existiert eine weitere Art von Random Fields, die auf einer Gibbsverteilung beruht. Diese werden daher als Gibbs Random Fields bezeichnet. Bei den Markov Random Fields werden nur lokale Wahrscheinlichkeiten beschrieben, d. h. es wird angegeben, wie wahrscheinlich die Zuordnung des jeweiligen Pixels zu einem bestimmten Segment ist. Aus diesen einzelnen lokalen Wahrscheinlichkeiten muss dann die Gesamtwahrscheinlichkeit berechnet werden. Dagegen existiert bei den Gibbs Random Fields eine explizite Formulierung der Gesamtwahrscheinlichkeit für das Auftreten einer bestimmten Segmentierung des Bildes. Beide Arten der Random Fields lassen sich verlustfrei ineinander überführen. Eine Richtung des Beweises des sogenannten Hammersley-Clifford Theorems ist in [Li, 2001] dargestellt.

Der Vorteil der Darstellung durch Gibbs Random Fields ist die Möglichkeit zur Vereinfachung durch Modellierung mit Hilfe der Gibbs Verteilung:

$$P(\mathbf{f}) = Z^{-1} \cdot e^{-\frac{1}{T}U(\mathbf{f})}, \quad \text{mit } Z = \sum_{\mathbf{f} \in \mathcal{F}} e^{-\frac{1}{T}U(\mathbf{f})}. \quad (\text{A.1})$$

Üblicherweise sucht man nach der Segmentierung \mathbf{f} , für die die angegebene Wahrscheinlichkeit $P(\mathbf{f})$ maximal wird. Da es sich bei Z um eine Normalisierungskonstante handelt, ist die Wahrscheinlichkeit $P(\mathbf{f})$ dann maximal, wenn der Exponent $\frac{1}{T}U(\mathbf{f})$ minimal ist. Bei dem Divisor T handelt es sich um die sogenannte Temperaturkonstante, die die Schärfe der Verteilung steuert. Diese ist in den meisten Fällen sogar zu $T = 1$ gewählt. Somit ist nur eine Betrachtung der Energiefunktion $U(\mathbf{f})$ notwendig.

Mit Hilfe der Energiefunktion können spezielle Anforderungen und Bedingungen des Anwendungsfalles in das Modell integriert werden. Dabei ist darauf zu achten, dass sich an die vorher definierte Graphenstruktur orientiert wird. Die dabei einzuhaltenden Einschränkungen sind in [Li, 2001] verständlich zusammengefasst.

A.2. Zusammensetzung der Energiefunktion

Die für das Gibbs Random Field zu definierende Energiefunktion ist, wie für Segmentierungsaufgaben üblich, aus zwei Summanden zusammengesetzt. Der erste Summand soll dafür sorgen, dass möglichst homogene Regionen entstehen können:

$$REG(x, y, l) = \sum_{x'=x\pm 1} \bar{\delta}(l, f(x', y)) + \sum_{y'=y\pm 1} \bar{\delta}(l, f(x, y')) \quad (\text{A.2})$$

$$\text{mit } \bar{\delta}(a, b) = \begin{cases} 1 & \text{wenn } a \neq b \\ 0 & \text{sonst} \end{cases} \quad (\text{A.3})$$

Dazu werden in der Nachbarschaft des aktuellen Pixels (x, y) die vergebenen Segmentlabel $f(\cdot, \cdot)$ mit dem Segmentlabel l des aktuellen Pixels verglichen. Sind diese unterschiedlich, wird die Gesamtenergie erhöht, die Segmentierung erhält sozusagen eine Bestrafung. Um eine Änderung der Segmentierung an den Objektgrenzen zu ermöglichen, wird zusätzlich das Kantenbild $e(\cdot, \cdot)$ benutzt. Dabei wird die Annahme ausgenutzt, dass die Objektgrenze mit einer Kante im Kantenbild zusammenfällt. Da auch Kanten im zu segmentierenden Objekt selbst vorhanden sein können, kann der Einfluss der Kantenannahme über den Parameter α_{edg} gesteuert werden.

$$\begin{aligned} REG(x, y, l) = & \sum_{x'=x\pm 1} \bar{\delta}(l, f(x', y)) \cdot \left(1 - \alpha_{edg} \cdot \bar{\delta}(e(x, y), e(x', y))\right) \\ & + \sum_{y'=y\pm 1} \bar{\delta}(l, f(x, y')) \cdot \left(1 - \alpha_{edg} \cdot \bar{\delta}(e(x, y), e(x, y'))\right) \end{aligned} \quad (\text{A.4})$$

Der zweite, als Fidelity-Term FID bezeichnete Summand beschreibt, wie gut ein vergebenes Label zu den beobachteten Daten passt. Im Falle des vorgestellten Systems setzt dieser sich aus der gewichteten Summe der Energien der verwendeten Cues für Bewegung FID_{mot} , Farbe FID_{col} und Tiefe FID_{dep} zusammen.

$$\begin{aligned}
 FID(x, y, l) := & \alpha_{mot} \cdot FID_{mot}(x, y, l) \\
 & + \alpha_{col} \cdot FID_{col}(x, y, l) \\
 & + \alpha_{dep} \cdot FID_{dep}(x, y, l)
 \end{aligned} \tag{A.5}$$

Nach dem Aufstellen der Energiegleichung wird die Segmentierung durch Minimieren des folgenden Ausdruck ermittelt:

$$f'(x, y) := \arg \min_l (FID(x, y, l) + \alpha_{reg} \cdot REG(x, y, l)) \tag{A.6}$$

Dabei kann durch Ausnutzung der Markov Eigenschaft die Optimierung lokal an der Position (x, y) des jeweiligen Pixels erfolgen. Diese unabhängige Betrachtung der Pixel ermöglicht die Umsetzung des Algorithmus auf hochparallelen Architekturen, wie beispielsweise auf Grafikbeschleunigungshardware. Da in dieser Arbeit der methodische Aspekt im Vordergrund stehen soll, wird darauf nicht weiter eingegangen. Eine detaillierte Untersuchung ist in [Mühlig, 2007]² zu finden.

A.2.1. Berechnung des optischen Flusses

Die Berechnung des optischen Flusses erfolgt mit Hilfe eines korrelationsbasierten Ansatzes auf Basis der Summe der quadratischen Differenzen (SSD³). Dazu wird für jeden Pixel (x, y) im Bild I zum Zeitpunkt $t-1$ dessen korrespondierender Pixel im Bild I' zum Zeitpunkt t gesucht. Dies geschieht, indem in einem definierten Suchfenster um den Pixel für jede zulässige Verschiebung Δx bzw. Δy jeweils der quadratische Fehler berechnet wird. Um vorhandenes Bildrauschen zu kompensieren und um eine zuverlässigere Aussage ableiten zu können, wird die Differenz nicht nur auf Basis eines einzelnen Intensitätspaares berechnet. Vielmehr werden die Differenzen in einer als Korrelationsfenster bezeichneten Region um die Pixel summiert:

²Die Studienarbeit wurde vom Autor im Rahmen des Projektes betreut.

³Sum of Squared Differences

$$SSD(x, y, \Delta x, \Delta y) := \frac{1}{N^2} \sum_{i=x-k}^{x+k} \sum_{j=y-k}^{y+k} \left(I(i, j) - I'(i + \Delta x, j + \Delta y) \right)^2 \quad (\text{A.7})$$

Dabei steht N für die Größe des Korrelationsfensters und es gilt $k = \frac{N-1}{2}$.

Anders als bei klassischen Algorithmen zur Bestimmung des optischen Flusses, wird das SSD-Gebirge nicht verworfen, sondern im Optimierungsprozess weiterverwendet:

$$FID_{mot}(x, y, l) := SSD(x, y, \Delta x_l, \Delta y_l) \quad (\text{A.8})$$

Dabei bezeichnet Δx_l bzw. Δy_l die Verschiebung, die als Prototypenparameter dem jeweiligen Segment l zugeordnet ist. Es wird also angenommen, dass der Pixel (x, y) den selben optischen Fluss aufweist, wie das Segment l . Ist diese Annahme falsch, weist das SSD-Gebirge an dieser Stelle einen hohen Wert auf bzw. umgekehrt. Durch Nutzung des gesamten SSD-Gebirges können Mehrdeutigkeiten im SSD-Gebirge ausgenutzt werden, indem eindeutige Segmentinformationen der Nachbarpixel mit Hilfe des Regularisierungsterms weiterpropagiert werden. Das Aperturproblem wird dadurch abgeschwächt.

Die Bestimmung der Prototypenparameter Δx_l bzw. Δy_l erfolgt, nachdem der Optimierungsprozess des Markov Random Fields abgeschlossen ist. Die Parameter entsprechen dabei der am häufigsten auftretenden Verschiebung aller Pixel eines Segmentes \mathcal{R}_l :

$$(\Delta x_l, \Delta y_l) := \arg \min_{(\Delta x, \Delta y)} \sum_{(x, y) \in \mathcal{R}_l} SSD(x, y, \Delta x, \Delta y) \quad (\text{A.9})$$

A.2.2. Integration von Farbinformationen

Da die berechneten SSD-Gebirge nicht frei von Rauschen sind, ist es notwendig, weitere Anhaltspunkte für die Segmentierung zur Verfügung zu stellen. Ein solcher Anhaltspunkt bietet die Farbgebung der Objekte.

Darüberhinaus soll es auf diese Weise möglich sein, stoppende Objekte weiter segmentieren zu können.

Zur Einbeziehung der Farbinformationen werden der Farbton (Hue) und die Farbsättigung (Saturation) herangezogen. Genauer gesagt wird die Verteilung der beiden Farbkanäle über das segmentierte Objekt in Form eines Histogramms $c_l(h, s)$ benutzt. Von der Energiefunktion wird nun der Histogrammwert abgezogen, der zu den Farbwerten $H(x, y)$ und $S(x, y)$ des aktuellen Pixels (x, y) gehört. Je ähnlicher sich diese sind, desto kleiner wird also die Energie:

$$FID_{col}(x, y, l) := -c_l(H(x, y), S(x, y)) \quad (\text{A.10})$$

Für die Adaption der Prototypenparameter wird entsprechend für jedes Segment \mathcal{R}_l ein normalisiertes Histogramm entsprechend der aktuellen Zuordnung der Pixel (x, y) erstellt:

$$\forall h, s : c_l(h, s) = \frac{|\{(x, y) \in \mathcal{R}_l | (H(x, y) = h) \wedge (S(x, y) = s)\}|}{|\mathcal{R}_l|} \quad (\text{A.11})$$

A.2.3. Integration von Tiefeninformationen

Ebenso wie die Farbinformationen sollen auch die Tiefeninformationen helfen, das Segmentierungsergebnis zu verbessern. Dabei wird angenommen, dass das segmentierte Objekt eine homogene Tiefe aufweist. Die verwendete Tiefeninformation stammt dabei aus der Disparitätskarte einer Stereokamera. In die Energiegleichung geht die Abweichung der Disparität $d(x, y)$ des betrachteten Pixels von der Disparität des Segmentes d_l ein:

$$FID_{dep}(x, y, l) = \begin{cases} (d(x, y) - d_l)^2 & \text{bei vorhandener} \\ & \text{Disparitätsinformation} \\ 1 & \text{sonst} \end{cases} \quad (\text{A.12})$$

Für die Bestimmung der Prototypenparameter wird die durchschnittliche Disparität aller Pixel des Segmentes berechnet:

$$d_l = \frac{\sum_{(x,y) \in \mathcal{R}_l} d(x,y)}{|\mathcal{R}_l|} \quad (\text{A.13})$$

A.3. Verwerfen und Zusammenführen von Segmenten

Die beiden Optimierungsschritte: Festlegen der Segmentierung und Bestimmen der Segmentierungsparameter erfolgt in einem Zeitschritt, bis keine Änderungen mehr vorgenommen werden oder bis eine maximale Anzahl an Iterationszyklen überschritten ist. Nach jedem Zyklus wird geprüft, ob allen existierenden Segmentbeschreibungen auch Pixel zugeordnet sind. Ist dies nicht der Fall wird die entsprechende Segmentinformation verworfen. Dies ist insbesondere dann wichtig, wenn Objekte das Bild verlassen.

Außerdem werden Segmente, die mit denselben Parametern beschrieben sind, zu einem einzigen Segment vereinigt. In beiden Fällen reduziert sich so die Anzahl der zur Verfügung stehenden Segmentklassen. Daher wird für den folgenden Zeitschritt die Anzahl der Segmente wieder auf die vorher definierte Menge gesetzt. So ist es möglich, auf neu auftauchende Objekte reagieren zu können.

A.4. Initialisierung

Die Initialisierung des Markov Random Field erfolgt auf zwei unterschiedliche Weisen. Ist der Pixel noch keinem Segment zugeordnet, erfolgt eine Initialisierung auf Basis des optischen Flusses. Bei der Segmentierung von Bildsequenzen wird dagegen die Information aus der vorangegangenen Segmentierung benutzt:

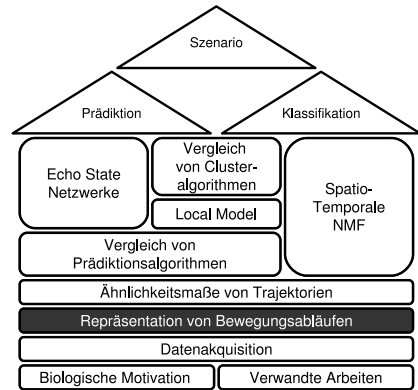
$$\forall x, y : f_t(x + \Delta x_l, y + \Delta y_l) := f_{t-1}(x, y) \text{ mit } l = f_{t-1}(x, y) \quad (\text{A.14})$$

Die Segmentierungsinformation f des Pixels (x, y) zum Zeitpunkt $t - 1$ wird dabei an die neue Position zum Zeitpunkt t verschoben. Dazu wird die Bewegungsinformation Δx_l bzw. Δy_l des zugewiesenen Segmentes l benutzt.

Üblicherweise werden bei Markov Random Fields Informationen aus dem Bildraum zur Segmentierung benutzt. An dieser Stelle wird die Strategie durch die Nutzung des optischen Flusses, aber auch durch eine geschickte Initialisierung, um zeitliches Wissen erweitert. Dadurch kann beim Verfolgen der Objekte auf einen gesonderten Tracker (z. B. Kalman Filter [Kalman, 1960]) verzichtet werden.

Anhang B

Verfahren zur Repräsentation von Zeit



In Abschnitt 3.2 wurde bereits eine Auswahl an Repräsentationsformen vorgestellt und untersucht. Die Untersuchungen wurden dabei am Beispiel der Erkennung des Nutzerinteresses durchgeführt. Nachfolgend werden weitere mögliche Repräsentationsformen kurz eingeführt und ihre Eignung für das gewählte Beispiel zusammengefasst (siehe Abbildung B.1). Ein vollständiger Überblick über die durchgeführten Experimente kann in [Hellbach und Wollenschläger, 2009] nachgeschlagen werden.

B.1. Dimensionalität

Trajektorien stellen per Definition den zeitlichen Verlauf einer bestimmten untersuchten Messgröße dar. Im Falle von Bewegungen sind dies die räumlichen Koordinaten. Je nach verwendeten Trackingsystem handelt sich dabei in den meisten Fällen um zwei- bis dreidimensionale Größen. Diese können direkt in dem Raum des Trackingsystems untersucht werden, wie es in Abbildung B.2 (a) dargestellt ist. Auf diese Weise geht jedoch die zeitliche Abfolge der einzelnen Positionen verloren.

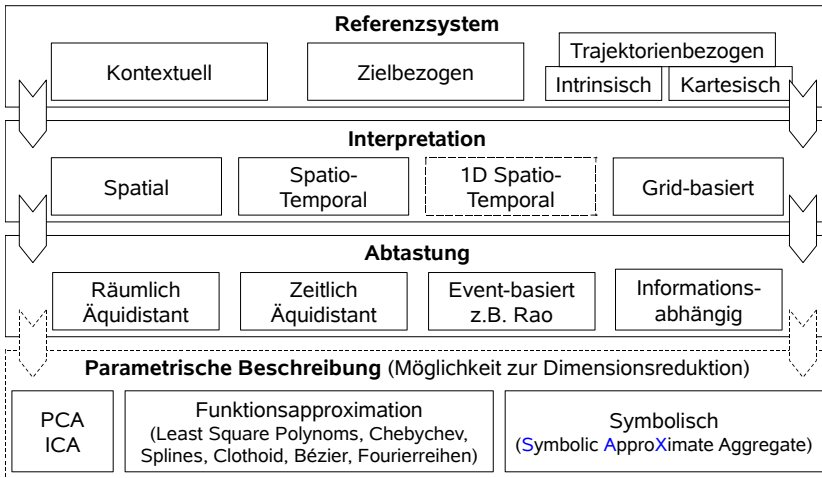


Abbildung B.1.: Repräsentationsformen für Bewegungstrajektorien: Durch die unterschiedlichen Repräsentationsformen werden auch unterschiedliche Aspekte der Bewegung in den Vordergrund gerückt.

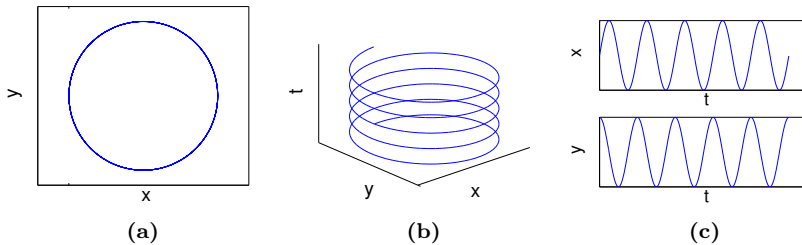


Abbildung B.2.: Einbeziehung der Zeit:

Drei mögliche Repräsentationsformen von Trajektorien gezeigt am Beispiel einer Kreisbewegung. In (a) ist die ausschließlich räumliche zu sehen, während in (b) die Zeitachse eine zusätzliche Dimension bildet. In vielen Fällen werden die Dimensionen unabhängig voneinander betrachtet (c).

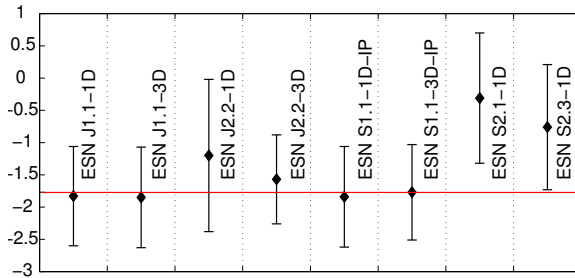


Abbildung B.3.: Vergleich zur Unabhängigkeit der Raumdimensionen: Für die einzelnen Varianten der Echo State Netzwerke wurden die einzelnen räumlichen Dimensionen jeweils unabhängig (1D) und als Verbund (3D) prädiziert. In zwei der vier Varianten schneiden die gemeinsame Betrachtung besser ab. Die Ergebnisse unterscheiden sich allerdings nicht signifikant.

Aus diesem Grund wird die Zeit als weitere Dimension hinzugefügt (Abbildung B.2 (b)). Auch wenn das Hinzufügen einer weiteren Dimension aus mathematischer Sicht keine Problem darstellt, nimmt die zeitliche Dimension eine Sonderstellung ein. Insbesondere bei zeitlich äquidistanten Daten können viele Berechnungen stark vereinfacht werden. Will man darüberhinaus die Ähnlichkeit von einzelnen Trajektorienabschnitten bestimmen, muss die Zeit gesondert behandelt werden. Ein Möglichkeit wäre, die Zeit wegzulassen, wie im letzten Absatz beschrieben. Durch den Wegfall der Reihenfolge würden invers ausgeführte Bewegungen als identisch klassifiziert werden können. Spielt die zeitliche Abfolge allerdings eine Rolle, muss für die zu vergleichenden Abschnitte eine gemeinsame Zeitskala festgelegt werden.

Durch die steigende Anzahl an Dimensionen, nimmt natürlich auch der Berechnungsaufwand zu. In vielen Veröffentlichungen werden die räumlichen Dimensionen als unabhängig angenommen und, wie in Abbildung B.2 (c) dargestellt, ihr zeitlicher Verlauf einzeln untersucht (vergleiche [Naftel und Khalid, 2006]). Kritisch zu sehen ist, dass die Annahme, die räumlichen Dimensionen seien unabhängig, in den meisten Fällen nicht zutrifft. Bei einer solchen Betrachtungsweise geht die in der Korrelation der Dimensionen steckende Information verloren.

Im Rahmen der Experimente zu den Prädiktionsverfahren in den Kapiteln 4 und 5 wurde auch die Möglichkeit der unabhängigen Betrachtung mit der gemeinsamen Betrachtung verglichen (siehe Abbildung B.3). Werden die Dimensionen unabhängig voneinander betrachtet, werden die Ergebnisse in zwei der vier untersuchten Varianten des Echo State Netzwerkes geringfügig schlechter. Die einzelne Betrachtung ist in diesem Fall also durchaus legitim.

B.2. Abtastung

Im Abschnitt 3.2 wurden bereits zwei Methoden zur Abtastung von Zeitreihen und damit auch von Trajektorien diskutiert. Dabei wurde in der zeitlichen bzw. räumlichen Variante jeweils äquidistant abgetastet. Der Abstand zwischen den Abtastpunkten muss dazu vorher so festgelegt werden, dass alle wichtigen Informationen der Trajektorie erfasst werden können. Somit werden Abschnitte, die weniger Information tragen, zwangsläufig überabgetastet.

Sinnvoll scheint an dieser Stelle, die Abtastabstände an die aktuellen beobachteten Daten anzupassen. Dazu seien aus der Literatur exemplarisch zwei Verfahren herausgegriffen und ihre jeweiligen Prinzipien kurz geschildert.

B.2.1. Ereignis-basierte Abtastung

Einer der wenigen Ansätze, der bereits in der Trajektorienanalyse zum Einsatz gekommen ist, findet sich in [Rao et al., 2002]. In der Publikation wird davon ausgegangen, dass die Analyse menschlicher Bewegungen auf signifikanten Änderungen in Richtung und Geschwindigkeit basieren sollte. Diese signifikanten Änderungen werden von den Autoren als *Dynamic Instants* (dynamische Augenblicke) oder auch als *Events* bezeichnet.

Zur Bestimmung der *Events* wird zuerst eine Glättung der Bewegung durchgeführt, um vorhandenes Rauschen zu entfernen. Für die geglättete Trajektorie wird die Krümmung κ berechnet:

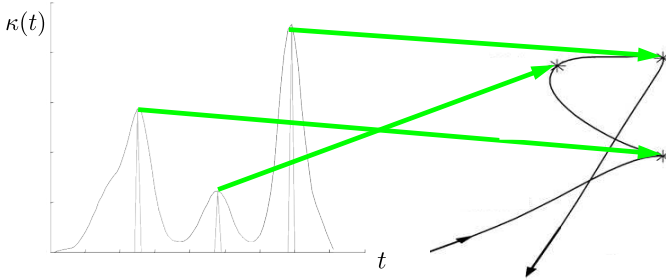


Abbildung B.4.: Dynamic Instants auf einer Trajektorie:
 Zur Bestimmung signifikanter Änderung der Bewegung werden die stärksten Krümmungen bestimmt. (entnommen und angepasst aus [Rao et al., 2002])

$$\kappa(t) = \frac{\sqrt{y''(t)^2 + x''(t)^2 + (x'(t)y''(t) - x''(t)y'(t))^2}}{(\sqrt{y''(t)^2 + x''(t)^2 + 1})^3} \quad (\text{B.1})$$

Dabei stellen $x'(t)$ und $x''(t)$ die erste und zweite Ableitung der Trajektorie zum Zeitpunkt t dar. In [Rao et al., 2002] wird die Krümmung für Trajektorien im zweidimensionalen Raum unter Hinzunahme der zeitlichen Dimension betrachtet. Da die Abtastintervalle in der zeitlichen Dimension als konstant angenommen werden, vereinfacht sich die Berechnung, wie in B.1 angegeben.

Als nächsten Schritt werden die lokalen Maxima des Krümmungsverlaufes über die Zeit bestimmt. Es werden also zum Ermitteln der *Events* die Stellen mit möglichst starker Krümmung gesucht. Die prinzipielle Vorgehensweise ist in Abbildung B.4 noch einmal dargestellt.

Experimente mit diesem Ansatz haben gezeigt, dass sich die Glättung der Trajektorie durchaus als problematisch erweist. Je nach Intensität werden entsprechen viele bzw. wenige *Events* gefunden. Darüber hinaus gehen durchaus auch Informationen verloren, da für jeden Zeitpunkt die hochdimensionalen Trajektorien auf eine skalare Größe abgebildet werden und erst anhand dieser die *Events* bestimmt werden.

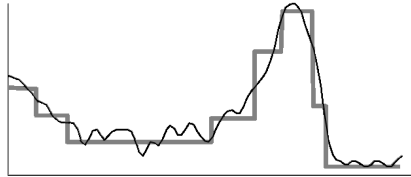


Abbildung B.5.: APCA Repräsentation einer Trajektorie. In Abhängigkeit von der Stärke der Änderung werden unterschiedlich große Bereiche zusammengefasst.

B.2.2. Informationsabhängige Abtastung

Das nächste Verfahren ist in der Literatur unter den Namen *Adaptive Piecewise Constant Approximation (APCA)* oder auch *Segmented Mean* zu finden [Chakrabarti et al., 2001; Cai und Ng, 2004]. Das aus dem Bereich der *Time Series Databases (TDSB)* stammende Verfahren nutzt implizit den Informationsgehalt der einzelnen Trajektorienabschnitte, um die Abtastgenauigkeit festzulegen. Dabei werden Abschnitte mit wenigen Änderungen zu großen Segmenten zusammengefasst, während Bereiche mit großen Änderungen entsprechend viele Stützstellen enthalten (siehe Abbildung B.5).

Dazu wird die zweidimensional Funktion bzw. die Trajektorie zuerst in gleichgroße Segmente aufgeteilt und für jedes Segment der Mittelwert über die enthaltenen Funktionswerte berechnet. Danach werden benachbarte Segmente zusammengefasst, wenn sich ihre Mittelwerte nur geringfügig unterscheiden. Dies wird solange wiederholt, bis eine definierte Anzahl Segmente erreicht ist.

Während bei einer zeitlich äquidistanten Abtastung die Information über die Segmentlänge, im Falle der Trajektorien der Abtastabstände, weggelassen werden kann, muss sie für die APCA explizit mit angegeben werden. Geschieht dies nicht, kann die Trajektorie nicht mehr eindeutig reproduziert werden.

B.3. Funktionsapproximation

In Abschnitt 3.2 wurde mit Hilfe der PCA die vektorielle Repräsentation der Trajektorie durch Rotation und Skalierung in einen anderen Raum projiziert. Ein ähnliches Prinzip liegt den im Folgenden vorgestellten Verfahren zu Grunde. Sämtliche Verfahren stammen aus dem Bereich der Funktionsapproximation. Während bei der PCA das Basissystem in der Form der Eigenvektoren anhand der Daten bestimmt wird, nutzen gängige Funktionsapproximationsalgorithmen fest definierte Basiskomponenten.

Um die Algorithmen für die Anwendung auf Trajektorien nutzen zu können, werden definierte Abschnitte der Trajektorie als Funktion $f(t)$ über die Zeit t aufgefasst. Der Einfachheit halber wird angenommen, dass die Funktion $f(t)$ in einen eindimensionalen Raum abbildet. Dies entspricht der eindimensionalen raum-zeitlichen Approximation aus Abschnitt B.1. Damit beliebige Funktionen approximiert werden können, besitzt jedes der Verfahren eine Anzahl von Parametern, die entsprechend der gegebenen Funktion bzw. dem Trajektorienverlauf angepasst werden. Somit stellen sie eine eindeutige Beschreibungsmöglichkeit dar.

In der Literatur finden sich bereits Ansätze, alternative Repräsentationsformen durch Nutzung von Approximationsalgorithmen zu erzeugen. An dieser Stelle soll allerdings nicht auf die mathematischen Details eingegangen werden. Da dies in gängigen Numerikbüchern, wie z. B. in [Neundorf, 2002], wesentlich ausführlicher getan wird. Insbesondere finden sich dort Details zu den Randbedingungen, wie die Wahl des Polynomgrades.

Eine der einfachsten Möglichkeit zur Approximation durch Polynome stellt die Methode der kleinsten Fehlerquadrate (Least Square Fit) dar [Naftel und Khalid, 2006]. Das Basissystem wird durch die einzelnen Monome $1, x, x^2, \dots, x^n$ des Polynoms $P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ gebildet. Die Komponenten des Basissystems sind allerdings nicht orthogonal, so dass sie zwangsläufig Redundanzen abbilden. Während dies in Kapitel 6 durch die spezielle Form der Repräsentation durchaus gewollt war, können sie im allgemeinen Fall durchaus zu Problemen führen.

Eine Weiterführung ist die Beschreibung mit Hilfe von Splines [Nunziati et al., 2005]. Durch Splines werden nur relativ kurze Abschnitte beschrieben. Dafür werden zusätzlich Randbedingungen bezüglich der Differenzierbarkeit zwischen benachbarten Abschnitten getroffen. Dadurch werden die benachbarten Abschnitte nicht mehr losgelöst voneinander betrachtet.

Um den Nachteil der nicht orthogonalen Basisfunktionen zu umgehen, stehen Chebyshev-Polynome zur Verfügung [Naftel und Khalid, 2006; Cai und Ng, 2004]. Bei dieser Approximationsform sind die Basispolynome so ausgelegt, dass sie orthogonal sind, d. h. ihr paarweises Skalarprodukt null ist. Darüberhinaus lassen sich die Koeffizienten, mit denen die Basispolynome gewichtet summiert werden, durch einen recheneffizienten Algorithmus bestimmen.

Ähnlich wie auch bei Chebyshev-Polynomen steht auch bei Béziersplines ein orthogonales Basissystem in Form der Bernsteinpolynome zur Verfügung. Durch geeignete Wahl der Kontrollpunkte können, wie auch bei Splines, Randbedingungen an den Übergängen zwischen benachbarten Segmenten getroffen werden.

Auch wenn aus mathematischer Sicht mit Hilfe von Polynomen gute Approximationsergebnisse erzielt werden können, ist die Interpretation der Überlagerung der einzelnen Basiskomponenten im Falle von Bewegungsdaten recht schwierig. Da Bewegungsdaten in erster Linie eine Überlagerung von Schwingungen darstellen, bietet die Fourierreihenentwicklung eine sinnvolle Alternative [Cai und Ng, 2004; Chakrabarti et al., 2001; Keogh et al., 2000; Naftel und Khalid, 2006]. Das Basissystem wird hier durch trigonometrische Funktionen unterschiedlicher Frequenz aufgestellt, die darüberhinaus noch orthogonal sind.

Zum Schluss seien noch kurz Clothoide erwähnt. Sie beschreiben die Krümmungsänderung einer Kurve, ausgehend von einer Anfangskrümmung. Um die gesamte Funktion approximieren zu können, muss sie aus Clothoidenteilstücken zusammengesetzt werden [Shin und Singh, 1990; Schröter et al., 2007]. Mit Hilfe der Anfangskrümmung lassen sich entsprechende Differenzierbarkeitsbedingungen stellen. Ursprünglich wurden Clothoide im Straßenbau eingesetzt, um besonders einfach zu fahrende Kurven bauen zu können. Sie wurden allerdings auch bereits erfolgreich für Pfadplanungsaufgaben eingesetzt.

B.4. Experimentelle Evaluierung

In den Tabellen B.1 und B.2 sind die Ergebnisse der Evaluierung zusammengefasst. Dabei entsprechen helle Farben schlechten Ergebnissen. Je dunkler die Färbung der Tabellenzelle, desto besser hat die jeweilige Kombination abgeschnitten.

HKA	Kartesisch		Intrinsisch		Intr.-kart.	
	ohne	mit	ohne	mit	ohne	mit
Spatial äquidistant						
Rohdaten	hell	hell	hell	hell	hell	hell
Splines	hell	hell	hell	hell	hell	hell
Bézierkurven	hell	hell	hell	hell	hell	hell
Chebyshev-Entwicklung	hell	hell	hell	hell	hell	hell
Fourier-Transformation	hell	hell	hell	hell	hell	hell
Least Squares Polynomfit	hell	hell	hell	hell	hell	hell
Taylor-Entwicklung	hell	hell	hell	hell	hell	hell
Temporal äquidistant						
Rohdaten	hell	hell	hell	hell	hell	hell
Splines	hell	hell	hell	hell	hell	hell
Bézierkurven	hell	hell	hell	hell	hell	hell
Chebyshev-Entwicklung	hell	hell	hell	hell	hell	hell
Fourier-Transformation	hell	hell	hell	hell	hell	hell
Least Squares Polynomfit	hell	hell	hell	hell	hell	hell
Taylor-Entwicklung	hell	hell	hell	hell	hell	hell

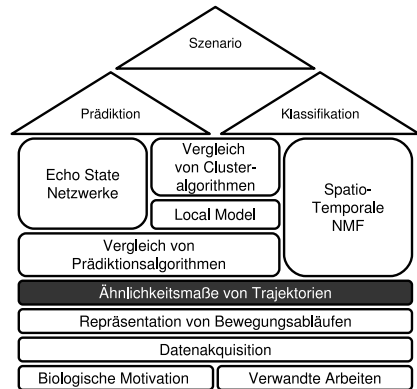
Tabelle B.1.: Eignung der Verfahren zur Bewegungsrepräsentation (Teil 1): Von ungeeignet (hell) bis gut geeignet (dunkel). Bei spatialer Abtastung sind Rohdaten verhältnismäßig gut geeignet. Die besten Ergebnisse erzielt die HKA der kartesisch kodierten Daten, insbesondere bei Repräsentation durch Splines und Bézierkurven. Zeitlich äquidistante Abtastung liefert insgesamt die besten Ergebnisse, insb. bei kartesischer oder intrinsisch-kartesischer Kodierung und HKA.

HKA	Kartesisch		Intrinsisch		Intr.-kart.	
	ohne	mit	ohne	mit	ohne	mit
Event-basiert						
Rohdaten		■			■	■
Splines						
Bézierkurven						
Chebyshev-Entwicklung		■				
Fourier-Transformation		■				
Least Squares Polynomfit						
Taylor-Entwicklung						
APCA-basiert						
Rohdaten		■			■	■
Splines						
Bézierkurven						
Chebyshev-Entwicklung		■				
Fourier-Transformation				■		
Least Squares Polynomfit						
Taylor-Entwicklung						

Tabelle B.2.: Eignung der Verfahren zur Bewegungsrepräsentation (Teil 2): Von ungeeignet (hell) bis gut geeignet (dunkel). Bei spatialer Abtastung sind Rohdaten verhältnismäßig gut geeignet. Die besten Ergebnisse erzielt die HKA der kartesisch kodierten Daten, insbesondere bei Repräsentation durch Splines und Bézierkurven. Event-basierte Kodierung liefert gute Ergebnisse bei der HKA auf kartesischer und intrinsisch-kartesischer Kodierung. APCA-basierte Verfahren zeigen nur vereinzelt gute Ergebnisse.

Anhang C

Ähnlichkeit von Trajektorien



Nachdem in Abschnitt 3.3 nur kurz diskutiert wurde, welche Möglichkeiten für die Bestimmung der Ähnlichkeit existieren, soll dieser Anhang dazu dienen die Verfahren im Detail vorzustellen und einige experimentellen Untersuchungen zu präsentieren.

In diesem Anhang sollen daher eine Vielzahl möglicher Metriken und Verfahren untersucht und verglichen werden. Als Fazit wird eine Übersicht entstehen, die es ermöglicht, für das entsprechende Anwendungsgebiet eine geeignete Wahl zu treffen. Zu Beginn werden eine Reihe klassischer Metriken gegenübergestellt. Die im Weiteren untersuchten Verfahren ermöglichen zwar eine wesentlich genauere Aussage, bauen aber auf den untersuchten klassischen Metriken auf.

C.1. Experimentaldaten

Da die Ergebnisse der durchgeführten Experimente in den jeweiligen Abschnitten an entsprechender Stelle präsentiert werden, sollen in diesem Abschnitt kurz die Voraussetzungen der Evaluierung diskutiert werden. Außer dem in Abschnitt C.2 gezeigten Beispiel, wurden sämtliche Unter-

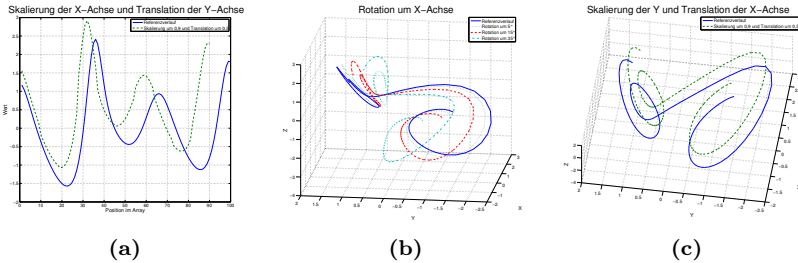


Abbildung C.1.: Unterschiedliche Transformationen der Experimentaldaten: Die 1D (a) bzw. 3D (b), (c) Daten wurden unterschiedlichen Transformationen, wie Rotation (b), Skalierung (a), (c) und Translation (a), (c) unterzogen.

suchungen auf der Trajektorie des dreidimensionalen Lorenzattraktors durchgeführt. Der Lorenzattraktor wurde gewählt, da es sich bei ihm um eine Zeitreihe mit chaotischem Verhalten handelt und er an vielen Stellen zur Evaluierung von Verfahren zur Zeitreihenanalyse eingesetzt wird [Shumway und Stoffer, 2000]. Bei allen Experimenten wurde ein konstantes Zeitfenster von 100 Schritten gewählt. Für die Untersuchungen, bei denen eindimensionale Daten verwendet werden, wurde jeweils die Y-Dimension des Lorenzattraktors herausgegriffen.

Getestet wurden unterschiedliche Transformationen, wie Rotation, Skalierung und Translation, die jeweils mit dem unveränderten Referenzverlauf verglichen wurden. Dabei wurden nicht nur elementare Transformationen untersucht, sondern auch Kombinationen in Betracht gezogen. In Abbildung C.1 sind drei Beispiele für die durchgeführten Veränderungen angegeben.

Die in dieser Arbeit vorgestellten Experimente stellen nur einen kleinen Auszug bzw. eine Zusammenfassung der Experimente in [Hellwig, 2009]¹ dar. In dieser im Fachgebiet Neuroinformatik und kognitive Robotik an der TU Ilmenau entstandenen Arbeit können die Ergebnisse der ausführlichen Experimente im Detail nachvollzogen werden.

¹Die Studienarbeit wurde vom Autor im Rahmen des Projektes betreut.

C.2. \mathcal{L}_p -Normen

Im Sinne der Definition der \mathcal{L}_p -Normen beschreiben sie Eigenschaften des Vektors, auf den sie angewendet werden. Den bekanntesten Vertreter stellt dabei die \mathcal{L}_2 -Norm dar, die auch unter dem Namen Euklidische Distanz bekannt ist. Sie beschreibt die Länge des gegebenen Vektors. Benutzt man nun den Differenzvektor zwischen zwei L -dimensionalen Vektoren \mathbf{x} und \mathbf{y} kann man Aussagen über die Lage der beiden Vektoren treffen.

$$\mathcal{L}_p(\mathbf{x} - \mathbf{y}) = \left(\sum_{i=1}^L |x_i - y_i|^p \right)^{1/p} \quad (\text{C.1})$$

Im Falle der Euklidischen Norm handelt es sich dabei um deren Abstand im klassischen Sinne. Definiert man die zu vergleichenden Trajektorien wie gewohnt als $\mathcal{T}_1 = (\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{n-1})$ und $\mathcal{T}_2 = (\mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_{n-1})$ mit jeweils L -dimensionalen Positionen \mathbf{s}_i , bzw. \mathbf{r}_i , ergibt sich:

$$\mathcal{L}_p(\mathcal{T}_1, \mathcal{T}_2) = \left(\sum_{i=1}^L \sum_{j=0}^{n-1} |\mathbf{s}_j^i - \mathbf{r}_j^i|^p \right)^{1/p} \quad (\text{C.2})$$

In [Faloutsos et al., 1997] werden die Eigenschaften der unterschiedlichen \mathcal{L}_p -Normen im Kontext der Abfrage von Zeitreihen in Datenbanken genauer untersucht und miteinander verglichen. An dieser Stelle sollen nur die wesentlichen Vertreter kurz erwähnt werden:

- \mathcal{L}_0 -Norm: Die auch als Minimum-Norm bezeichnete Norm stellt einen Spezialfall dar. Sie beschreibt die maximale Annäherung der beiden Trajektorien.
 - \mathcal{L}_1 -Norm: Sie kann auch unter den Begriffen City-Block Distanz oder auch Manhattan Norm gefunden werden. Sie ist optimal, wenn die Fehler additiv sind, und gilt als robust gegenüber einzelnen Ausreißern.
 - \mathcal{L}_2 -Norm: Sie ist bekannt unter dem Namen Euklidische Distanz und beschreibt den intuitiven Abstands begriff.
-

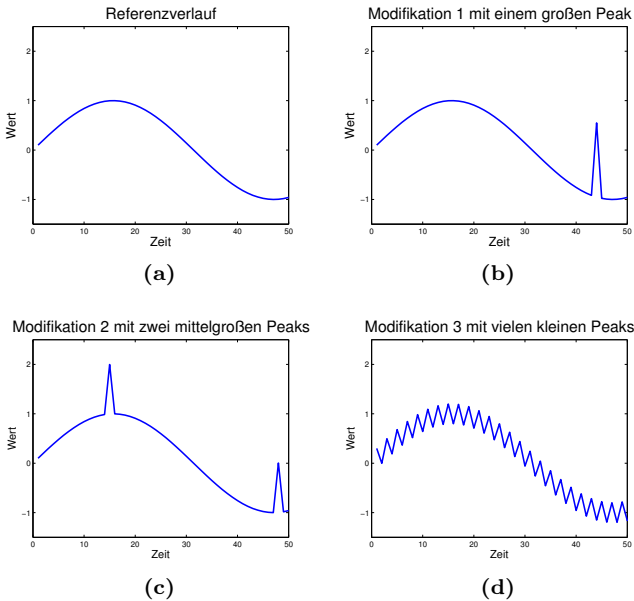


Abbildung C.2.: Referenzverlauf mit unterschiedlichen Modifikationen.: *Der Referenzverlauf (a) wurde jeweils um einen großen Ausreißer (b), zwei mittelgroße Ausreißer (c) und vielen kleinen Ausreißer (d) modifiziert, um die unterschiedlichen Auswirkungen dieser Änderung auf die \mathcal{L}_p -Normen zu demonstrieren.*

Norm	1 Peak	2 Peaks	viele Peaks
\mathcal{L}_1 -Norm	1.5	2.0	30.0
\mathcal{L}_2 -Norm	1.5	1.4	2.4
\mathcal{L}_∞ -Norm	1.5	1.0	0.2

Tabelle C.1.: Ergebnisse für unterschiedliche \mathcal{L}_p -Normen:

Es wurde die entsprechende Distanz zwischen dem Referenzverlauf und dem jeweiligen modifizierten Verlauf berechnet (siehe Abbildung C.2). Da es sich jeweils um Distanzmaße handelt, stehen kleinere Werte für eine größere Ähnlichkeit.

- \mathcal{L}_∞ -Norm: Die auch unter dem Begriff Maximum-Norm bekannte Norm gilt ebenfalls als Spezialfall und beschreibt die maximale Abweichung zwischen den Trajektorien.

Mit einem kleinen Beispiel soll nun verdeutlicht werden, wie die unterschiedlichen Normen auf spezielle Unähnlichkeiten reagieren. Dazu wurde, wie in Abbildung C.2 dargestellt, ein Referenzsignal auf unterschiedliche Art und Weise modifiziert. Die Ergebnisse des Vergleichs zwischen Referenzsignal und der entsprechenden Modifikation sind in Tabelle C.1 dargestellt. Wie man sieht, wird je nach verwendeter \mathcal{L}_p -Norm eine andere Veränderung als am Ähnlichsten errechnet. Bei der \mathcal{L}_1 -Norm schneidet die Variante mit nur einem Ausreißer am besten ab, da die Summe der Abstände der zwei Ausreißer größer ist, als der Abstand des einen Ausreißers. Bei der \mathcal{L}_2 -Norm gehen die Abstände nun quadratisch ein, so dass der reaktiv große Abstand des einen Ausreißers einen größeren Einfluss gewinnt. Die große Anzahl der Ausreißer mit jeweils kleinem Abstand kommen der \mathcal{L}_∞ -Norm zu Gute, da hier eben nur nach dem maximalen Abstand gesucht wird, die Anzahl aber keine Rolle spielt. Die \mathcal{L}_0 -Norm würde in allen Fällen eine Distanz von 0 berechnen, da es mindestens eine Stelle gibt, an der die Trajektorien identisch sind.

C.3. Korrelationskoeffizient

In der Wahrscheinlichkeitstheorie existieren unterschiedliche Maße, um die Ähnlichkeit von Datenverteilungen zu beschreiben. Da es sich bei Wahrscheinlichkeitsverteilungen um Funktionsverläufe handelt, sollte es keine Schwierigkeiten bereiten, die definierten Maße auf Bewegungstrajektorien zu übertragen. Dabei werden die Trajektorien auch hier wieder als Zeitreihen interpretiert.

Der prominenteste Vertreter, mit dem die Ähnlichkeit von Wahrscheinlichkeitsverteilungen ausgedrückt werden kann, ist der Korrelationskoeffizient. Er findet beispielsweise in [Li et al., 1996] Anwendung, um in einem Datenbanksystem Teilsequenzen von Zeitreihen wiederfinden zu können. Solche Datenbanken werden benutzt, um beispielsweise EKG-Daten speichern und durchsuchen zu können. Der Korrelationskoeffizient berechnet sich dabei wie folgt:

$$Kor(\mathbf{x}, \mathbf{y}) = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (\text{C.3})$$

mit den jeweiligen empirischen Mittelwerten

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{und} \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i .$$

Der Korrelationskoeffizient beschreibt die lineare Abhängigkeit zweier Funktionsverläufe und nimmt dabei Werte zwischen -1 und 1 an. Während ein Korrelationskoeffizient von 0 bedeutet, dass die Funktion keine lineare Abhängigkeit aufweist, stehen -1 bzw. +1 für eine vollständige negative bzw. positive Abhängigkeit. In Tabelle C.2 wird der Korrelationskoeffizient für unterschiedliche Transformationen der Trajektorie berechnet. Dabei zeigt sich, wie zu erwarten, dass eine Translation und eine Skalierung entlang der Raumachse keinen Einfluss haben. In beiden Fällen handelt es sich um eine lineare Transformation der Daten. Da bei der Berechnung des Korrelationskoeffizienten punktweise vorgegangen wird, kann bei einer zeitlichen Verschiebung nicht mehr der eigentliche Zusammenhang gefunden werden.

Translation Raumachse $Kor(x, y)$	Translation Zeitachse $Kor(x, y)$	Skalierung Raumachse $Kor(x, y)$
0.5 1	5 0.5	0.9 1
2 1	10 -0.2	0.7 1
5 1	25 -0.1	0.5 1

Tabelle C.2.: Berechnung des Korrelationskoeffizienten:
Der Korrelationskoeffizient wurde für unterschiedliche Transformationen der Experimentaldaten berechnet. Auffällig ist, dass Translation und Skalierung entlang der Raumachse keinen Einfluss haben.

Der Korrelationskoeffizient dient, wie bereits erwähnt, dazu, die lineare Abhängigkeit zweier Vektoren zu bestimmen. Daher ist er auf eindimensionalen Daten definiert. Es gibt allerdings die Möglichkeit, ihn auf mehrdimensionale Daten zu erweitern, indem man die Korrelationskoeffizienten jeder Dimension des einen Datensatzes zu jeder Dimension des anderen Datensatzes bestimmt. Hierbei entsteht allerdings eine Matrix von Korrelationswerten und nicht ein einzelner Distanzwert.

C.4. Sequence Search

Die nächste Gruppe von Verfahren zum Bestimmen von Ähnlichkeiten stammt aus dem Bereich der Datenbanken bzw. des Dataminings. In ihrem ursprünglichen Anwendungsfeld dienen sie dazu, Datensequenzen, wie z. B. Zeitverläufe, Bewegungstrajektorien und Börsendaten, effizient zu speichern bzw. wiederzufinden. In der Gruppe der Sequence Search Verfahren, die sich in das große Forschungsfeld der Time Series Databases (TSDB) einordnet, existiert eine beliebig große Anzahl an Verfahren, mit denen die Ähnlichkeit zwischen jeweils zwei Zeitreihen bestimmt werden kann. An dieser Stelle soll daher ein Verfahren, welches

in der entsprechenden Literatur als Referenzverfahren gehandelt wird, erläutert werden. Darüber hinaus ist exemplarisch ein weit verbreitetes Verfahren untersucht worden, welches die Zeitreihe auf Grundlage eines bestehenden Basissystems in einen Parameterraum transformiert.

C.4.1. Adaptive Piecewise Constant Approximation

Das in diesem Abschnitt vorgestellte Distanzmaß basiert auf der in Abschnitt B.2.2 bereits eingeführten Adaptive Piecewise Constant Approximation (APCA). Die Trajektorie wird also nicht mehr wie gewohnt als $\mathcal{T}_1 = (\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{n-1})$ repräsentiert. Es werden vielmehr Abschnitte zusammengefasst und segmentweise konstant approximiert, in denen keine wesentlichen Änderungen der Position stattfinden. Da die Abtastung keiner festen Regel mehr folgt, reicht es nicht mehr aus, nur die räumliche Information zu speichern. Es muss jeweils noch erkennbar sein, zu welchem Zeitpunkt das entsprechende Segment beginnt: $\mathcal{T}_1^{APCA} = (\langle \bar{\mathbf{s}}_i, t_i \rangle)_{0 \leq i < N}$.

Will man nun die so repräsentierte Trajektorie \mathcal{T}_1 mit einer weiteren Trajektorie $\mathcal{T}_2 = (\mathbf{r}_i)_{0 \leq i < m}$ vergleichen, findet man in [Chakrabarti et al., 2002] zwei Möglichkeiten dies zu tun. Zum Einen kann man, wie in Abbildung C.3(a) zu sehen, das Zeitraster der Vergleichstrajektorie \mathcal{T}_2 benutzen. Dazu wird für jeden Zeitpunkt der Abstand zum zeitlich korrespondierenden Segment der Trajektorie \mathcal{T}_1^{APCA} zum Berechnen der Distanz benutzt:

$$D_{AE}(\mathcal{T}_1^{APCA}, \mathcal{T}_2) = \sqrt{\sum_{i=1}^{N-1} \sum_{k=t_{i-1}}^{t_i} (\bar{\mathbf{s}}_i - \mathbf{r}_k)^2} \quad (\text{C.4})$$

Das Kürzel *AE* soll dabei für *approximate Euklidian* stehen.

Die zweite Möglichkeit einer Definition ist in Abbildung C.3(b) zu sehen. Die Vergleichstrajektorie \mathcal{T}_2 wird dazu derselben Segmentierung unterzogen wie die APCA-Repräsentation der Trajektorie \mathcal{T}_1^{APCA} . Dadurch entsteht die Repräsentation $\mathcal{T}'_2 = (\langle \bar{\mathbf{r}}_i, t_i \rangle)_{0 \leq i < N}$, wobei die Zeiten t_0 bis t_{N-1} identisch zu denen in \mathcal{T}_1^{APCA} sind. Die Distanz berechnet sich nun wie folgt:

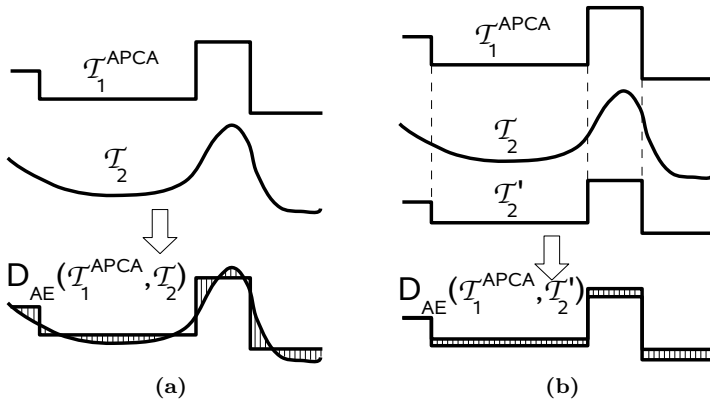


Abbildung C.3.: Die zwei unterschiedlichen APCA-basierten Distanzmaße: Die Trajektorien \mathcal{T}_1^{APCA} und \mathcal{T}_2 sollen verglichen werden, wobei \mathcal{T}_1^{APCA} bereits der APCA unterzogen wurde. Die linke Grafik (a) stellt die approximierte euklidische Variante $D_{AE}(\mathcal{T}_1^{APCA}, \mathcal{T}_2)$ des Distanzmaßes dar. Zur Berechnung wird für jeden Punkt auf \mathcal{T}_2 wird sein Abstand zu \mathcal{T}_1^{APCA} ermittelt. In der rechten Grafik (b) ist die Berechnung einer unteren Schranke $D_{LB}(Q, C)$ für die Distanz angegeben. Dazu wird \mathcal{T}_2 derselben Abtastung unterzogen, wie \mathcal{T}_1^{APCA} und anschließend die abschnittswise Abstände ermittelt. (Bilder entnommen aus [Chakrabarti et al., 2002])

Translation Zeitachse	D_{AE}	D_{LB}	\mathcal{L}_2 -Norm
5	9.7	8.6	9.6
10	15.1	14.3	15.1
25	13.4	13.2	13.5

Tabelle C.3.: Vergleich der beiden Varianten der APCA-Distanz: Die Distanz D_{AE} approximiert die Euklidische Distanz recht gut. Allerdings bildet sie, im Gegensatz zur Distanz D_{LB} keine untere Schranke der Euklidischen Distanz.

$$D_{LB}(\mathcal{T}_1^{APCA}, \mathcal{T}_2') = \sqrt{\sum_{i=1}^{N-1} (t_i - t_{i-1})(\bar{\mathbf{s}}_i - \bar{\mathbf{r}}_i)^2} \quad (\text{C.5})$$

Wie zu sehen ist, geht der räumliche Abstand der Segmente, wie bei der Euklidische Distanz üblich, quadratisch in die Berechnung ein. Zusätzlich wird der Segmentabstand allerdings noch mit der der Dauer des Segmentes gewichtet. Dadurch dass für \mathcal{T}_2 dieselbe Abtastung wie für \mathcal{T}_1 benutzt wird, kann es unter Umständen zum Verlust von Informationen kommen. Diese Distanz ist demnach eine wesentlich größere Approximation, als die Distanz D_{AE} . Allerdings liefert sie eine untere Schranke für die Euklidische Distanz. Daher rührt auch die Abkürzung *LB* für *lower-bounding Measure*. Dies spielt allerdings nur im Falle der Datenbanksuche eine Rolle, da man möglichst keine ähnliche Trajektorie in der Datenbank übersehen will. Exemplarisch sind beide Distanzen in Tabelle C.3 verglichen.

Vergleichbar mit dem Vorgehen der APCA wird in [Perng et al., 2000] ein auf Landmarken basiertes Verfahren vorgestellt. Bei diesem Verfahren, wird die Trajektorie ebenso auf nicht äquidistante Weise abgetastet. Als Landmarken werden dabei lokale Extrempunkte unterschiedlicher Ordnung bezeichnet.

C.4.2. Chebyshev-Approximation

Ein weiterer Vertreter aus dem Bereich Datenbanken verwendet die in Abschnitt B.3 präsentierte Möglichkeit der Repräsentation durch Chebyshev-Koeffizienten. Die Trajektorien $\mathcal{T}_1 = (\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{n-1})$ und $\mathcal{T}_2 = (\mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_{m-1})$ werden jeweils als Funktionsverläufe interpretiert. Diese Funktionsverläufe werden mit Hilfe der Chebyshev-Polynome approximiert. Die so berechneten Koeffizienten zum Überlagern der Polynome repräsentieren die beiden Trajektorien: $\mathcal{T}_1^{Cheb} = (a_0, a_1, \dots, a_{N-1})$ bzw. $\mathcal{T}_2^{Cheb} = (b_0, b_1, \dots, b_{N-1})$.

Zur Berechnung der Distanz wird, wie in [Cai und Ng, 2004] angegeben, eine skalierte Euklidische Distanz im Raum der Chebyshev-Koeffizienten verwendet.

$$D_{Cheb}(\mathcal{T}_1^{Cheb}, \mathcal{T}_2^{Cheb}) = \sqrt{\frac{\pi}{2} \sum_{i=0}^{N-1} (a_i - b_i)^2} \quad (\text{C.6})$$

Da die Chebyshev-Polynome orthogonal zueinander sind, kann dies ohne Einschränkung erfolgen. Durch die Verwendung der Chebyshev-Approximation ist es möglich, auch unterschiedlich lange Trajektorien miteinander zu vergleichen. Es muss nur darauf geachtet werden, dass beide Approximationen denselben Grad haben bzw. dass beide mit dem gewählten Grad hinreichend gut approximiert werden können.

Ein Beispiel für eine Distanzberechnung findet sich in Tabelle C.4. Dabei werden die Auswirkungen auf die Distanz bei unterschiedlichen Verschiebungen entlang der Raumachse verglichen. Es ist zu erkennen, dass bei größer werdender Translation die Distanz entsprechend wächst. Beim Blick auf die Chebyshev-Polynome fällt auf, dass das erste Polynom $P_0(t) = 1$ konstant ist. Damit hat die Verschiebung entlang der Raumachse theoretisch nur Auswirkung auf den ersten Chebyshev-Koeffizienten. Somit ist es möglich, ein bezüglich der Raumachse translationsinvariantes Distanzmaß wie folgt zu definieren:

$$D_{Cheb2}(\mathcal{T}_1^{Cheb}, \mathcal{T}_2^{Cheb}) = \sqrt{\frac{\pi}{2} \sum_{i=1}^{N-1} (a_i - b_i)^2} \quad (\text{C.7})$$

Translation Raumachse	D_{Cheb}	D_{Cheb2}
0.5	1.3	0.0
2	5.0	0.0
5	12.5	0.0

Tabelle C.4.: Chebyshev-Distanz bei unterschiedlichen Translationen: *Bei Translationen entlang der Raumachse wächst die Distanz mit größer werdender Verschiebung. Allerdings wirkt sich eine Verschiebung entlang der Raumachse nur auf den ersten Chebyshev-Koeffizienten aus. Lässt man diesen, wie bei D_{Cheb2} geschehen, bei der Distanzberechnung weg, ergibt sich kein Unterschied in der Distanz.*

Das auf Basis der Chebyshev-Approximation definierte Distanzmaß, basiert auf einer Zerlegung, die ein vorgegebenes Basisalphabet benutzt. Ein ähnliches Vorgehen findet man in [Agrawal et al., 1993]. Allerdings werden in dieser Veröffentlichung die Koeffizienten bzw. die Energien der diskreten Fouriertransformation benutzt.

C.5. Time Warping Algorithmen

In den ersten beiden Abschnitten (C.2 und C.3) wurden Ähnlichkeiten durch klassische Distanzmaße ausgedrückt. In der Einleitung zu diesem Kapitel wurde bereits erwähnt, dass es für bestimmte Anwendungen sinnvoll ist, darüberhinaus eine Aussage über die Art der Unähnlichkeit ableiten zu können. Darum sollen in den nächsten Abschnitten Verfahren untersucht werden, die eine solche Aussage ermöglichen. Dabei ist festzustellen, dass im Kern der zu besprechenden Verfahren eine Distanzfunktion, wie sie in Abschnitt C.2 besprochen wurde, genutzt wird. In den meisten Fällen wird sogar die Euklidische Distanz eingesetzt. Es ist aber durchaus vorstellbar, dass die verwendete Distanz durch andere

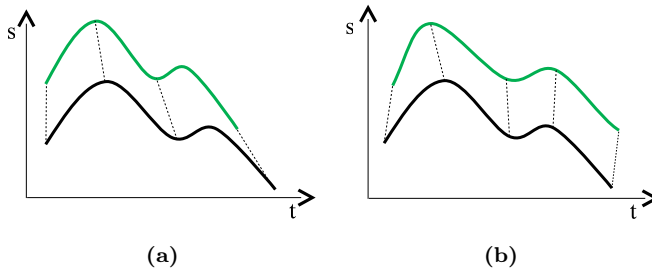


Abbildung C.4.: Unterschiedliche Arten von Time Warping Verfahren: Während bei den statischen Verfahren (a) ein konstanter zeitlicher Skalierungsfaktor zum Überführen der einen Trajektorie in die andere benutzt wird, findet bei den dynamischen Verfahren ein für jeden Zeitpunkt individuell festgelegte Skalierung statt.

ersetzt wird. Insbesondere gelten dann die in Abschnitt C.1 zusammengefassten Experimente analog.

Eine Gruppe von Verfahren, die über das reine Vergleichen von Trajektorien hinaus gehen, stellen die Time Warping Verfahren dar. Dabei soll exemplarisch der bekannteste Vertreter, das Dynamic Time Warping [Sakoe und Chiba, 1990], erläutert werden und am Ende des Abschnittes kurz auf die Unterschiede zu anderen Verfahren eingegangen werden.

Prinzipiell lassen sich, wie in Abbildung C.4 veranschaulicht, Time Warping Algorithmen in zwei Kategorien einteilen. Bei der Gruppe der statischen Verfahren wird eine Trajektorie mit einem konstanten Faktor entlang der zeitlichen Achse skaliert, bis der Start und Endpunkt zum selben Zeitpunkt liegen. Bei den dynamischen Verfahren hingegen wird dieser Skalierungsfaktor für jeden Zeitpunkt individuell festgelegt. Es findet also eine nicht-lineare zeitliche Transformation statt. Dies führt dazu, dass jeder Punkt der einen Trajektorien zur zeitlichen Position des korrespondierenden Punktes der anderen Trajektorie verschoben wird.

Das Dynamic Time Warping fällt, wie der Name verrät, in die Kategorie der dynamischen Verfahren. Der Algorithmus erwartet die zwei zu vergleichenden Signale in vektorieller Notation. Um nun Trajekto-

rien vergleichen zu können, werden sie in gewohnter Weise als $\mathcal{T}_1 = (\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{n-1})$ und $\mathcal{T}_2 = (\mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_{m-1})$ präsentiert. Anders als bei den Normen in Abschnitt C.2, müssen die Trajektorien allerdings nicht dieselbe Anzahl an Stützstellen besitzen.

Als nächstes wird eine Matrix \mathbf{V} benötigt, die für jedes mögliche Punktepaar $(\mathbf{s}_i, \mathbf{r}_j)$ dessen Euklidische Distanz enthält. Somit ist die $n \times m$ -Matrix wie folgt:

$$\mathbf{V} = \begin{pmatrix} v_1^1 & \cdots & v_m^1 \\ \vdots & \ddots & \vdots \\ v_1^n & \cdots & v_m^n \end{pmatrix} \quad ; \quad \forall i, j : v_j^i = \|\mathbf{s}_i, \mathbf{r}_j\|_2 \quad (\text{C.8})$$

In der Matrix \mathbf{V} muss nun ein Pfad vom linken oberen Element v_1^1 zum rechten unteren Element v_m^n gefunden werden. Ist die Summe der besuchten Elemente dabei minimal, beschreibt dieser Pfad die optimale Zuordnung zwischen den beiden Trajektorien. Zum Finden des optimalen Pfades wird das Paradigma der dynamischen Programmierung eingesetzt. Damit lässt sich die Summe der Punktabstände entlang des gefundenen Pfades ermitteln. Sie entspricht dem Abstand der Trajektorien unter Beachtung der optimalen Punktkorrespondenz und geht damit über die simple Bestimmung der Distanzen aus Abschnitt C.2 hinaus. Für die beiden gegebenen Trajektorien berechnet sich die Distanz also folgendermaßen:

$$DTW(\mathcal{T}_1, \mathcal{T}_2) = dtw(n, m) \quad (\text{C.9})$$

Dabei ist die Funktion $dtw(\cdot, \cdot)$ rekursiv definiert:

$$dtw(i, j) = \begin{cases} \infty & \text{für } i = 0 \vee j = 0 \\ v_j^i & \text{für } i = 1 \wedge j = 1 \\ v_j^i + \min \begin{pmatrix} dtw(i-1, j-1), \\ dtw(i-1, j), \\ dtw(i, j-1) \end{pmatrix} & \text{sonst} \end{cases} \quad (\text{C.10})$$

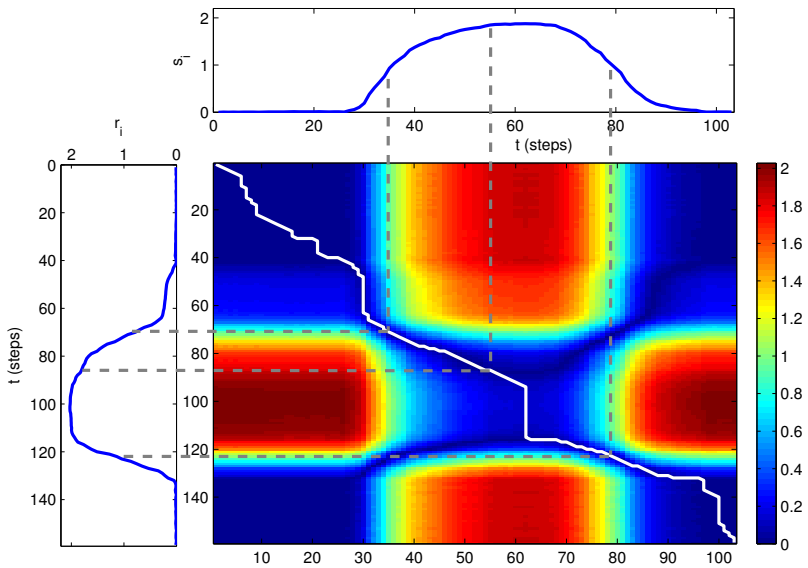


Abbildung C.5.: Pfadsuche in Distanzmatrix mittels DTW: Für jede paarweise Kombination von Punkten der beiden zu vergleichenden Trajektorien wird die Distanz ausgerechnet und in die Matrix \mathbf{V} eingetragen (hier farblich veranschaulicht). In der Matrix wird nun ein zusammenhängender Pfad gesucht, bei dem die Summe der Distanzen minimal ist.

In Abbildung C.5 soll der Algorithmus grafisch verdeutlicht werden. In der unteren rechten Ecke wird begonnen. Von dort aus wird, wie in Formel C.10 beschrieben, nach links, oben oder links oben geschaut, welcher der Einträge einen minimalen Wert aufweist. Da dies für dieses Element nicht bekannt ist, muss die Rechnung rekursiv fortgesetzt werden, bis die linke obere Ecke erreicht ist. Entsprechend der Lage des Pfades in der Matrix sind unterschiedliche Zeitpunkte auf beiden Trajektorien einander zugeordnet.

Neben der Aussage über die Distanz ist aber auch der eigentliche Pfad von Interesse. Er gibt eine Aussage, welche Punkte auf der einen Trajek-

torie mit welchen Punkten auf der anderen Trajektorie assoziiert sind. Dazu merkt man sich bei der Berechnung des Minimums in Gleichung C.10, welcher der drei Rekursionszweige das Minimum liefert. Somit ist Ende nicht nur die finale Distanz $DTW(\mathcal{T}_1, \mathcal{T}_2)$ bekannt, sondern auch welche Summanden und damit welche Zuordnungen zu dieser Distanz geführt haben.

Neben dem Dynamic Time Warping findet man in der Literatur noch andere Verfahren, deren Ziel die zeitliche Skalierung ist. Die algorithmische Umsetzung ist dabei ähnlich oder zumindest prinzipiell vergleichbar. Exemplarisch seien an dieser Stelle zwei Vertreter herausgegriffen. Zum einen das in [Hahn et al., 2008] vorgestellte, auf der Levenshtein-Distanz basierte Verfahren. Auf der anderen Seite das Longest Common Subsequence (LCSS) Verfahren von [Vlachos et al., 2002].

Diese beiden Verfahren wurden deshalb ausgewählt, da bei beiden der Distanzbegriff anderes aufgefasst wird, als beim ursprünglichen Dynamic Time Warping (DTW). Während beim DTW aus der errechneten Matrix die Summe der Euklidischen Abstände der jeweils korrespondierenden Punkte abgelesen werden kann, definiert sich die Ähnlichkeit bei den Verfahren aus [Hahn et al., 2008] und [Vlachos et al., 2002] aus der Anzahl der notwendigen Änderungen, um eine Trajektorie in die andere zu überführen. Ein solches, als Editdistanz bezeichnetes Maß, ist ursprünglich beim Vergleich von Zeichenketten zu finden. In [Hahn et al., 2008] wurde allerdings gezeigt, dass solche Arten von Distanzen auch für den Einsatz mit Bewegungstrajektorien anwendbar sind.

Für die Berechnung der Levenshtein-Distanz wird eine Matrix aufgebaut, die für jede Punktekombination die Information enthält, welche Anzahl an Änderungen (Löschen, Hinzufügen, Verschieben von Punkten) durchgeführt werden muss. Auch hier wird, wie in Gleichung C.10, das Wissen der bereits berechneten Nachbarn benutzt, um eine Aussage über die jeweilige Punktekombination berechnen zu können.

Die Berechnung der Longest Common Subsequence gibt als Ergebnis die Anzahl der Punkte zurück, die einander zugeordnet werden konnten. Im Gegensatz zum DTW ist es somit möglich, dass Punkte nicht zugeordnet werden müssen. Dies ist besonders von Vorteil, wenn sich Ausreißer in den Daten befinden.

Translation Raumachse	Levenshtein $e = 0.4$	Levenshtein $e = 1.5$
0.5	30	0
2	97	61
5	100	100

Tabelle C.5.: Levenshtein-Distanz bei unterschiedlicher Verschiebungen: Berechnet wurde die Levenshtein-Distanz bei unterschiedlichen Verschiebungen der Daten entlang der Raumachse. Dabei fällt auf, dass die Distanz stark abhängig ist von dem gewählten Schwellwert e , mit dem entschieden wird, ob eine Änderung notwendig ist oder nicht.

Beide Verfahren haben gemeinsam, dass sie für die Entscheidung, ob ein bestimmtes Punktepaar als korrespondierend angesehen wird, den Abstand der beiden Punkte benutzen. Damit ist allerdings die Angabe eines Schwellwertes notwendig, der somit einen starken Einfluss auf die Aussage der ermittelten Distanz hat (vergleiche Tabelle C.5).

Ein unter Umständen entscheidender Nachteil der Time Warping Verfahren ist, dass durch die Skalierung in der zeitlichen Dimension Informationen über die zeitliche Dynamik verloren gehen. Damit werden beispielsweise Trajektorien, die mit unterschiedlicher Geschwindigkeit ausgeführt werden, als ähnlich klassifiziert, wenn sie einen ähnlichen Verlauf haben. Auf den ersten Blick scheint dies durchaus sinnvoll zu sein. Für Anwendungen, bei denen allerdings die Geschwindigkeit eine Rolle spielt, liefert die Anwendung der Time Warping Verfahren keine brauchbaren Ergebnisse. Stellt man sich beispielsweise die Bewegung des Armes eines Tennisspielers vor, dann existiert durchaus ein wichtiger Unterschied, z. B. für den Spielverlauf und die anzuwendende Strategie, wie intensiv der Schwung des Armes beim Schlagen des Balles ist.

C.6. Scan Matching Algorithmen

Im Bereich Bildverarbeitung und insbesondere auch im Bereich Robotik existieren Problemstellungen, die es notwendig machen, einen gemessenen Sensoreindruck mit einem anderen in Deckung zu bringen. Dazu bedient man sich Algorithmen aus dem Bereich des Scan Matching. Einer der als Referenz geltenden Algorithmen ist dabei der Iterative Closest Point Algorithmus (ICP) [Besl und McKay, 1992]. Darüberhinaus existiert eine Vielzahl weiterer Algorithmen, die allerdings alle nach einem ähnlichen Schema ablaufen. Aus diesem Gründen soll an dieser Stelle auch nur der ICP Algorithmus näher beleuchtet werden.

Prinzipiell wird bei den Scan Matching Algorithmen versucht, einen gegebenen Sensoreindruck durch geeignete affine Transformationen aus Rotation und Verschiebung mit einem Modell in Übereinstimmung zu bringen. Sowohl der Sensoreindruck als auch das Modell sind dabei als Punktemengen gegeben. Das Übertragen auf Trajektorien ist dadurch recht einfach zu lösen, in dem die Punkte entlang der Trajektorie als Menge aufgefasst werden $\mathcal{T}_1 = \{\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{n-1}\}$ und $\mathcal{T}_2 = \{\mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_{n-1}\}$. Für eine potenzielle Erweiterung kann es sich durchaus als sinnvoll erweisen, den zeitlichen Kontext der Punkte in die Berechnung mit einfließen zu lassen.

Jede Iteration k des ICP Algorithmus läuft dabei, wie in Abbildung C.6 dargestellt, in vier Teilschritten ab. Zuerst erfolgt eine Bestimmung der korrespondierenden Punkte. Dabei wird jedem Punkt der Trajektorie $\mathcal{T}_1(k)$ der Punkt mit dem jeweils geringsten Abstand auf der anderen Trajektorie $\mathcal{T}_2(k)$ zugeordnet. Es entsteht eine Zuordnung zwischen den Punkten der Trajektorie, die in der Grafik als \mathcal{C} bezeichnet ist.

Im nächsten Schritt versucht man die Punkte der Trajektorie $\mathcal{T}_2(k)$ so zu rotieren und zu verschieben, dass sich der Abstand zu den korrespondierenden Punkten der Trajektorie $\mathcal{T}_1(k)$ jeweils minimiert. Für die gesamte Punktwolke muss dabei natürlich dieselbe Rotation bzw. Translation angewendet werden. Prinzipiell wird, um die Parameter der Rotation \mathbf{R} und der Translation \mathbf{T} bestimmen zu können, eine Energie- bzw. Fehlerfunktion definiert:

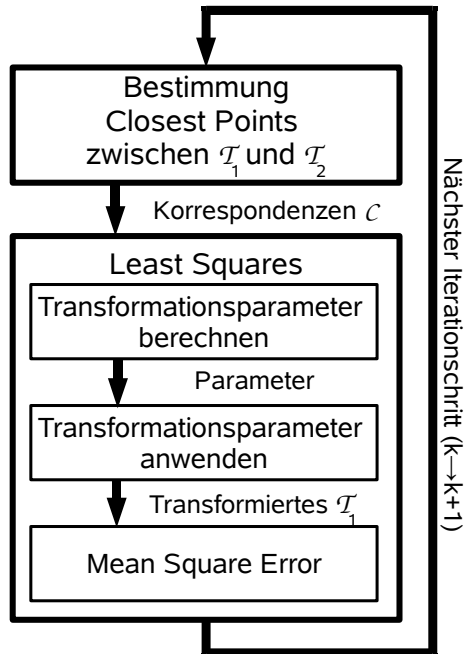


Abbildung C.6.: Schematischer Ablauf des ICP Algorithmus.: Jede Iteration läuft in vier Teilschritten ab. Diese werden solange wiederholt, bis die Änderung des mittleren quadratischen Fehlers unter eine definierte Schwelle gefallen ist.

$$f(\mathbf{R}, \mathbf{T}) = \frac{1}{n} \sum_{i=0}^{n-1} \|\mathbf{s}_i - \mathbf{R}\mathbf{r}_i + \mathbf{T}\|_2 \quad (\text{C.11})$$

Für diese Funktion muss nun das globale Minimum gefunden werden, um die optimalen Rotation und Translation zu erhalten:

$$(\mathbf{R}^*, \mathbf{T}^*) = \underset{\mathbf{R}, \mathbf{T}}{\operatorname{argmin}} f(\mathbf{R}, \mathbf{T}) \quad (\text{C.12})$$

In den meisten Fällen, wie auch in [Besl und McKay, 1992], wird dazu eine Berechnungsvorschrift hergeleitet, mit der nicht der gesamte Parameterraum auf der Suche nach dem Minimum untersucht werden muss. In [Besl und McKay, 1992] selbst wird zwar ein auf Quaternionen basierender Ansatz vorgestellt, der die statistische Charakteristik der beiden Trajektorien ausnutzt. Für mehr als drei Dimensionen wird allerdings auf die Singulärwertzerlegung verwiesen. Dies soll allerdings hier im Einzelnen nicht weiter beleuchtet werden.

Als nächstes wird die berechnete Transformation auf die Trajektorie $\mathcal{T}_2(k)$ angewendet. Zum Schluss wird überprüft, ob die Änderung des mittleren quadratischen Fehlers bereits unter eine vorher definierte Schwelle gefallen ist. Ist dies nicht der Fall, wird wieder mit der Korrespondenzsuche begonnen. Diesmal allerdings mit der rotierten und verschobenen Trajektorie $\mathcal{T}_2(k+1)$.

In Tabelle C.6 sind die Resultate auf rotierten bzw. auf skalierten Daten dargestellt. Wie zu erwarten bereiten die rotierten Daten keine Probleme. Die Rotation wird durch den Algorithmus richtig berechnet und die Distanz sinkt auf null. In der Tabelle scheint eine Rotation in entgegengesetzte Richtung errechnet wurden zu sein. Dies ist allerdings nicht der Fall. Die Trajektorie \mathcal{T}_2 wurde erzeugt, indem die Trajektorie \mathcal{T}_1 der in der ersten Spalte aufgeführten Rotation unterzogen wurde. Damit beide wieder deckungsgleich sind, muss die Trajektorie \mathcal{T}_2 dementsprechend wieder zurückrotiert werden.

Interessant ist, was bei den skalierten Daten passiert. Da die Skalierung im Algorithmus nicht vorgesehen ist, muss sie durch die beiden anderen Formen der Transformation (Rotation und Translation) abgefangen

werden. Da dies nicht gelingen kann, ergeben sich falsche Werte bei den Parametern und die Distanz sinkt nicht auf null. Beim Beispiel des Tennisspielers, der seinen Schlag mit unterschiedlicher Dynamik ausführt, würde dieses Verfahren daher keine brauchbaren Ergebnisse liefern.

C.7. Zusammenfassung

Die \mathcal{L}_p -Normen sind ein Distanzmaß, das theoretisch für beliebige Dimensionen definiert ist. Es können allerdings nur gleichlange Daten verglichen werden. Die \mathcal{L}_p -Normen bilden die Grundlage für viele andere Verfahren. In den meisten Fällen wird dabei die \mathcal{L}_2 -Norm verwendet.

Der Korrelationskoeffizient gibt an, wie stark zwei Verläufe linear voneinander abhängen. Das bedeutet, dass Translationen und Skalierungen entlang der Raumachse, also der eigentlichen Featureachse, keinen Einfluss haben. Der Korrelationskoeffizient kann dabei nur mit gleichlangen Daten umgehen.

Basierend auf der Adaptive Piecewise Constant Approximation (APCA) sind zwei Distanzmaße definiert worden. Vorteil dabei ist, dass beide Distanzmaße im Gegensatz zur Euklidischen Distanz auf Trajektorien mit unterschiedlicher Punktzahl arbeiten können. Eines der beiden Distanzmaße approximiert die Euklidische Distanz sehr gut, während das andere eine untere Schranke darstellt. Da für die Anwendung mit Trajektorien die Beschränktheit nach unten keine Rolle spielt, ist die erste Variante als geeigneter anzusehen.

Durch die Verwendung der Chebyshev-Distanz wird es möglich, unterschiedlich lange Trajektorien zu vergleichen. Besonders interessant ist die Modellierung der räumlichen Verschiebung. Durch Vernachlässigen des ersten Koeffizienten lässt sich ein translationsinvariantes Distanzmaß berechnen. Durch die Gesamtzahl der berechneten Koeffizienten lässt sich die Approximationsgüte bzw. auch die Glattheit der Funktion bestimmen. Somit ist es möglich, durch geeignete Wahl der Anzahl der Koeffizienten fehlerhafte Distanzberechnung durch Datenrauschen zu eliminieren bzw. zumindest einzudämmen.

Das Dynamic Time Warping nimmt Zuordnungen auf der Zeitachse vor und versucht dabei, die Distanz zu minimieren. Das heißt Verschiebung

Rotation <i>x</i> -achse	Distanz	$T = (\Delta x, \Delta y, \Delta z)$	$R = (\varphi_x, \varphi_y, \varphi_z)$
5	D=0	$T = (0, 0, 0)$	$R = (-5, 0, 0)$
15	D=0	$T = (0, 0, 0)$	$R = (-14.8, 0, 0)$
35	D=0	$T = (0, 0, 0)$	$R = (-32.9, 0, 0)$
Skalierung <i>x</i> -achse			
0.9	D=0.108	$T = (0, 0, 0)$	$R = (-2.7, 0, -0.3)$
0.7	D=0.181	$T = (0, 0, 0)$	$R = (-10.3, -0.6, -1.4)$
0.5	D=0.165	$T = (0, 0, 0)$	$R = (-18.2, -0.4, -2.2)$

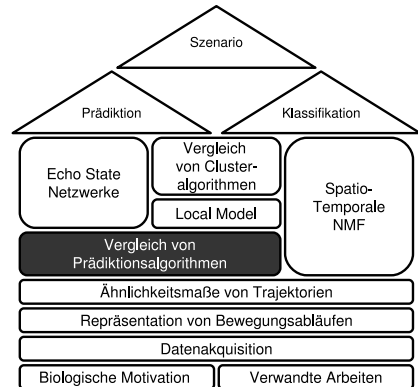
Tabelle C.6.: Vergleich der Resultate des ICP-Algorithmus: *Der ICP wurde zum Einen auf rotierten Daten getestet, aber auch auf Daten, die skaliert waren. Die rotierten Daten bereiten keine Probleme, da der Algorithmus mit Rotation umgehen kann. Bei den skalierten Daten muss die Skalierung durch die beiden zur Verfügung stehenden Transformationen abgefangen werden.*

gen und Skalierungen auf der Zeitachse werden kompensiert. Alle anderen Transformationen werden durch dieses Verfahren nicht erfasst. Im Gegensatz zum Longest Common Subsequence Algorithmus und zur Levenshtein-Distanz wird kein Parameter benötigt, der Einfluss auf die Distanz haben könnte. Neben einem skalarem Wert, der Aussage über die Distanz gibt, ist es ebenso möglich, die Punktkorrespondenzen zu erhalten, mit denen eine Trajektorie in die andere überführt werden kann.

Nach der Berechnung des Iterative Closest Points Algorithmus sind sowohl die Distanz als auch die dazugehörigen Transformationsparameter bekannt, wobei lediglich Translationen und Rotationen berücksichtigt werden. Durch die vorherigen Transformationen werden Translationen und Rotationen in allen Dimensionen kompensiert. Der ICP ist in der Lage, unterschiedlich lange Daten zu vergleichen. Prinzipiell spricht nichts dagegen den ICP-Algorithmus noch um die Skalierung zu erweitern. Dies würde bei raum-zeitlichen Daten im Gegensatz zu Sensoreindrücken durchaus Sinn machen.

Anhang D

Vergleich von Prädiktionsalgorithmen



In den Kapiteln 4 und 5 wurden Verfahren zur Prädiktion von Zeitreihen vorgestellt. Diese wurden allerdings nicht beliebig, sondern auf Basis extensiver Experimente ausgewählt. Im Rahmen der Experimente wurden unterschiedliche Verfahren untersucht. Dabei wurde darauf geachtet, dass möglichst ein breites Spektrum an Forschungsfeldern abgedeckt wird. In Abbildung D.1 sind die im Rahmen der Arbeit [Strauß, 2008] untersuchten Methoden und die Einordnung in ihre Forschungsfelder dargestellt. Ergebnisse dieser Arbeit und weiterführende Untersuchungen wurden in [Hellbach et al., 2008; Hellbach et al., 2009b] publiziert. Die im Folgenden weiter diskutierten Verfahren sind hervorgehoben. Da in Kapitel 4 Echo State Netzwerke und in Kapitel 5 Local Models bereits vorgestellt wurden, werden in diesem Anhang nur noch die fehlenden Verfahren kurz erläutert.

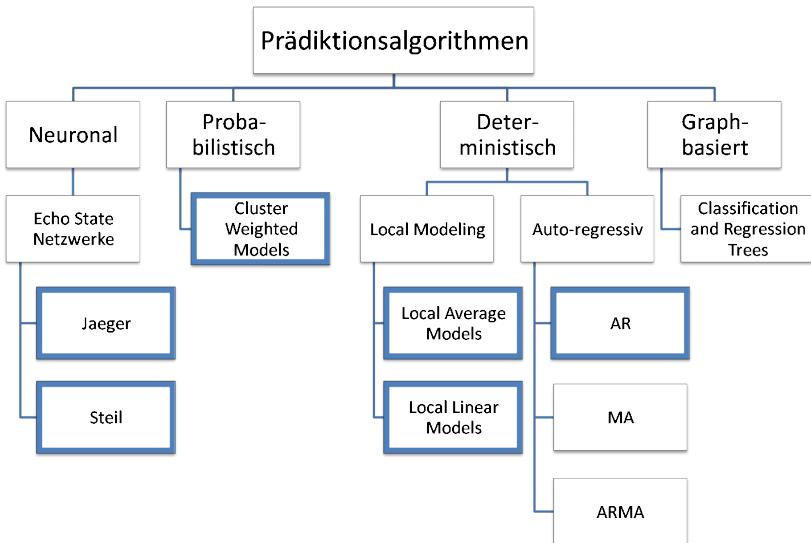


Abbildung D.1.: Untersuchte Prädiktionsalgorithmen:

Die Abbildung zeigt die in [Strauß, 2008] untersuchten Prädiktionsalgorithmen. Für die hervorgehobenen werden die Experimente in diesem Anhang vorgestellt, um die Auswahl für Kapitel 4 und 5 zu begründen.

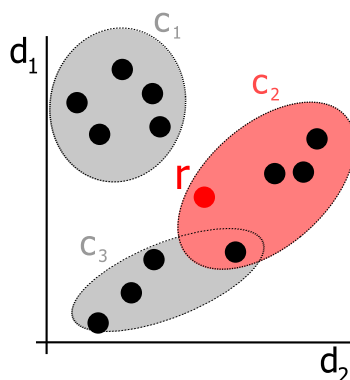


Abbildung D.2.: Grundprinzip des Cluster Weighted Modeling: Im Gegensatz zum Local Modeling wird beim Cluster Weighted Modeling der Einbettungsraum (im Bild 2-dimensional (d_1, d_2)) mit Gaußfunktionen abgedeckt. Zum aktuellen Punkt r werden dann die Cluster bestimmt, deren Gaußfunktionen diesem am nächsten sind. Im Beispiel wird r durch das Cluster c_2 repräsentiert. (Entnommen aus [Strauß, 2008])

D.1. Cluster Weighted Modeling

Ein ausführlichere Beschreibung zu Cluster Weighted Models, als dies hier geschehen kann, findet sich in [Engster und Parlitz, 2006]. Zur Anwendung der Cluster Weighted Models wird ebenfalls die in Abschnitt 5.1 geschilderte Einbettung benötigt. Jeder Punkt der Trajektorie muss also zuerst, wie auch bei Local Models, in den Einbettungsraum überführt werden. Anders als bei Local Models beruht die Prädiktion nicht auf einzelnen Einbettungspunkten. Vielmehr wird die Trajektorie im Einbettungsraum geclustert und für jeden Cluster, wie in Abbildung D.2 angedeutet, mit einer Gaussfunktion repräsentiert. Dabei wird jeder Cluster c_m durch sowohl im Einbettungsraum $P(\mathbf{e}|c_m)$, als auch im Ausgaberaum $P(\mathbf{o}|\mathbf{e}, c_m)$ beschrieben. Die Ausgabe \mathbf{o} bezeichnet, wie auch schon bei den Local Models den weiteren Verlauf für einen bestimmten Punkt im Einbettungsraum.

Die als Input Domain bezeichnete Verteilung $P(\mathbf{e}|c_m)$ kennzeichnet die Zugehörigkeit der zu prädizierenden Trajektorieneinbettung $\mathbf{e} = (x_1, \dots, x_D)$ zu den jeweiligen Clustern c_m , die über den Mittelwertvektor $\mu_m = (\mu_{m,1}, \dots, \mu_{m,D})$ und Varianzvektor $\sigma_m = (\sigma_{m,1}, \dots, \sigma_{m,D})$ beschrieben sind:

$$P(\mathbf{e}|c_m) = \prod_{i=1}^D \frac{1}{\sqrt{2\pi\sigma_{m,i}^2}} \cdot e^{-\frac{(z_i - \mu_{m,i})^2}{2\sigma_{m,i}^2}} \quad (\text{D.1})$$

Dabei fällt auf, dass die Dimensionen des Einbettungsraumes als unabhängig angenommen werden, was die Berechnung insbesondere mit hochdimensionalen Einbettungsräumen wesentlich vereinfacht.

Mit dem Ausgabeterm $P(\mathbf{o}|\mathbf{e}, c_m)$ wird die Wahrscheinlichkeit beschrieben, dass ein bestimmter weiterer Verlauf \mathbf{o} auftritt, wenn bereits die Trajektorie \mathbf{e} beobachtet wurde und der Cluster c_m ausgewählt ist:

$$P(\mathbf{o}|\mathbf{e}, c_m) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \cdot e^{-\frac{(\mathbf{o} - f(\mathbf{e}, \beta_m))^2}{2\sigma_m^2}} \quad (\text{D.2})$$

Der Mittelwert der Gaussfunktion wird durch die sogenannte Clusterfunktion $f(\mathbf{e}, \beta_m)$ festgelegt. Diese Funktion ist vergleichbar mit der Polynomannahme des Local Models. Die Wahrscheinlichkeit $P(\mathbf{o}|\mathbf{e}, c_m)$ wird maximal, wenn gilt $\mathbf{o} = f(\mathbf{e}, \beta_m)$, also die Modellannahme zutrifft.

Um mit Hilfe des Cluster Weighted Models zu präzisieren, wird ein gewichtetes Mittel über alle möglichen weiteren Verläufe berechnet. Die Gewichtung dabei erfolgt entsprechend der Wahrscheinlichkeiten für die weiteren Verläufe:

$$z_{n+1}^{pred} = \int \mathbf{o} P(\mathbf{o}|\mathbf{e}_n) d\mathbf{o} \quad (\text{D.3})$$

$$= \int \mathbf{o} \sum_{m=1}^M P(\mathbf{o}|\mathbf{e}, c_m) P(\mathbf{e}|c_m) P(c_m) d\mathbf{o} \quad (\text{D.4})$$

$$= \frac{\sum_{m=1}^M f(\mathbf{e}_n, \beta_m) P(\mathbf{e}_n|c_m) P(c_m)}{\sum_{m=1}^M P(\mathbf{e}_n|c_m) P(c_m)} \quad (\text{D.5})$$

Wie in den Gleichungen D.1 und D.2 zu erkennen ist, müssen eine Reihe von Parametern bestimmt werden, um das Cluster Weighted Model anwenden zu können. Sinnvollerweise sollten die Parameter entsprechend der beobachteten Daten festgelegt werden. Dazu wird in der Regel der EM-Algorithmus¹ verwendet. Dabei wird im Expectation-Schritt die Posteriori-Verteilung $P(c_m|\mathbf{o}, \mathbf{e})$ für die einzelnen Cluster bestimmt. Im anschließenden MMaximization-Schritt werden auf Basis der berechneten Verteilung die Parameter angepasst. Details dazu finden sich sowohl in [Engster und Parlitz, 2006], als auch etwas kompakter in [Strauß, 2008].

D.2. Autoregressive Models

Der nächste Algorithmus aus dem Bereich Zeitreihenanalyse, der vorgestellt werden soll, sind die Autoregressive Models (AR). Sie treffen als Annahme einen linearen Zusammenhang der Daten in der beobach-

¹Expectation-Maximization-Algorithmus

teten Zeitreihe. Dabei wird davon ausgegangen, dass jeder Punkt der Zeitreihe als Linearkombination seiner p direkten Vorgänger dargestellt werden kann:

$$\mathbf{s}_n = \sum_{i=n-p}^{n-1} v_i \cdot \mathbf{s}_i \quad (\text{D.6})$$

Anders als bei Local Models und Cluster Weighted Models wird nicht die Einbettung aus Abschnitt 5.1 benutzt, um die Vorgänger geeignet auszuwählen zu können. Die AR-Koeffizienten v_i werden auf Basis der Beobachtungen trainiert, um sie anschließend für die Prädiktion nutzen zu können. Dazu existieren unterschiedliche Möglichkeiten, wobei die wichtigsten im Folgenden kurz vorgestellt werden sollen.

D.2.1. Wiener Filter

Der Wiener Filter [Wiener, 1949] macht nichts anderes, als eine lineare Regression über alle Punkte in der beobachteten Zeitreihe durchzuführen. Dazu wird zum Einen eine Matrix \mathbf{E} aufgestellt, die alle beobachteten Punkte der Trajektorie enthält. Zum Anderen werden in den Vektor \mathbf{o} die jeweils weiteren Verläufe der Punkte in \mathbf{E} aufgenommen. Danach kann man die AR-Koeffizienten wie folgt berechnen:

$$v = (\mathbf{E}^T \mathbf{E})^{-1} \mathbf{E}^T \mathbf{o}^T \quad (\text{D.7})$$

Die Verwendung des Wiener Filters verläuft ähnlich zur Berechnung der Local Models aus Abschnitt 5.2. Allerdings wird auf die Auswahl ähnlicher Nachbarn zur Bestimmung der Koeffizienten verzichtet, sondern die gesamte Beobachtung genutzt. Der Standard-Ansatz der Autoregressive Models verzichtet auf die Nutzung der Einbettung. Durch Experimente konnte in [Strauß, 2008] nachgewiesen werden, dass die Verwendung der Einbettung anstelle der Auswahl der letzten p Werte zur deutlichen Verbesserung der Ergebnisse führt.

D.2.2. Durbin-Levinson

Der Algorithmus nach Durbin-Levinson (siehe [Shumway und Stoffer, 2000]) basiert auf der Autokorrelationsfunktion (ACF) $\rho(h)$. Sie muss, bevor die eigentliche Berechnung erfolgen kann, für die beobachtete Trajektorie bestimmt werden. Mit der Autokorrelationsfunktion wird im gewissen Sinne die Selbstähnlichkeit der beobachteten Bewegung ausgedrückt, indem die Korrelationen zwischen den Punkten der Trajektorie unter allen möglichen zeitlichen Verschiebungen betrachtet werden.

Die AR-Koeffizienten werden rekursiv bestimmt. Dabei definiert die Rekursionstiefe die Anzahl der Koeffizienten und somit die Anzahl der Punkte auf der Trajektorie, die zur Prädiktion in Betracht gezogen werden. Im n -ten Rekursionsschritt wird der n -te Koeffizient v_n^n bestimmt. Dabei bezeichnet der obere Index den Rekursionsschritt und der untere den eigentlichen Koeffizienten.

$$v_n^n = \frac{\rho(n) - \sum_{k=1}^{n-1} v_k^{n-1} \rho(n-k)}{1 - \sum_{k=1}^{n-1} v_k^{n-1} \rho(k)} \quad (\text{D.8})$$

Alle übrigen Koeffizienten v_1^n, \dots, v_{n-1}^n wurden bereits in den vorangegangenen Schritten berechnet und müssen noch wie folgt adaptiert werden:

$$v_k^n = v_k^{n-1} - v_n^n v_{n-k}^{n-1} \quad (\text{D.9})$$

D.2.3. Yule-Walker

Die letzte in dieser Arbeit beschriebene Methode stellen die Yule-Walker-Gleichungen. Sie werden ebenfalls ausführlicher in [Shumway und Stoffer, 2000] diskutiert. Auch hier wird wieder die Autokorrelationsfunktion benötigt. Diesmal allerdings in ihrer nicht normalisierten Version als Autokovarianzfunktion $\gamma(h)$. Ist die Funktion für die beobachtete Zeitreihe berechnet, wird folgendes Gleichungssystem gelöst, um die AR-Koeffizienten zu bestimmen:

$$\begin{aligned} \gamma(0) &= v_1\gamma(1) + \cdots + v_p\gamma(p) + \sigma_w^2 \\ \gamma(h) &= v_1\gamma(h-1) + \cdots + v_p\gamma(p) \end{aligned} \quad h = 1, \dots, p \quad (\text{D.10})$$

Anders als beim Durbin-Levinson-Algorithmus werden so alle Koeffizienten auf einmal bestimmt. In der ersten der Yule-Walker Gleichungen für $\gamma(0)$ taucht zusätzlich ein σ_w^2 auf. Dieses stammt aus der Prozessannahme, die für das AR-Modell getroffen wurde, und beschreibt das Messrauschen.

D.3. Experimenteller Vergleich

Die experimentelle Auswertung basiert auf denselben Annahmen und Daten, wie bereits in Abschnitt 4.3 diskutiert. Ebenso wurden dieselben Maße für die Prädiktionsgüte verwendet. Deshalb ist beim Interpretieren der Grafiken darauf zu achten, dass die Skala der Ordinatenachse logarithmisch ist.

Zusätzlich wurden die Experimente auf den in Abbildung D.3 bzw. D.4 dargestellten Sinusdaten bzw. dem Lorenzattraktor durchgeführt. Dies geschah zum Einen, um die Korrektheit der Implementierungen überprüfen zu können, da im Bereich der Zeitreihenanalyse solche Zeitreihen häufig als Benchmark eingesetzt werden. Zum Anderen kann man mit diesen Tests die prinzipielle Leistungsfähigkeit der Algorithmen einschätzen. Die beiden Zeitreihen stellen sozusagen die äußeren Extrema von Zeitreihen dar. Während die Sinusschwingungen ein absolut periodisches Signal darstellen, bei dem zusätzlich noch die einzelnen Dimensionen unabhängig sind, steht der Lorenzattraktor für ein klassisches Beispiel chaotischer Zeitreihen. Darüberhinaus geben die Daten eine Orientierung für die verwendeten Bewegungsdaten, die in ihrem Anspruch zwischen den beiden Extremen einzuordnen sind.

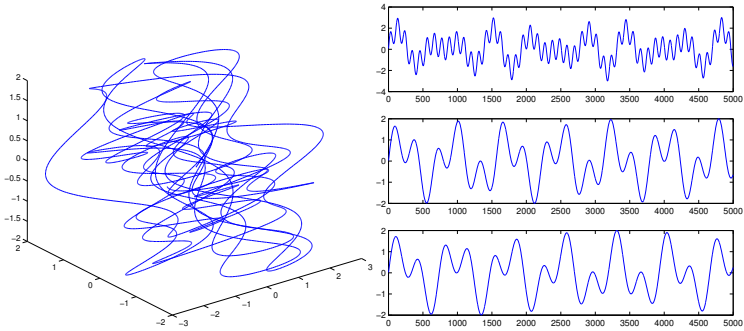


Abbildung D.3.: Sinustestdaten:

Jede der 3 Dimensionen ist eine Funktion überlagerter Sinusfunktionen. Auch wenn das 3-dimensionale Bild der Prädiktion chaotisch erscheint, so besteht die Trajektorie ausschließlich aus voneinander unabhängigen und periodischen Bewegungen. (Entnommen aus [Strauß, 2008])

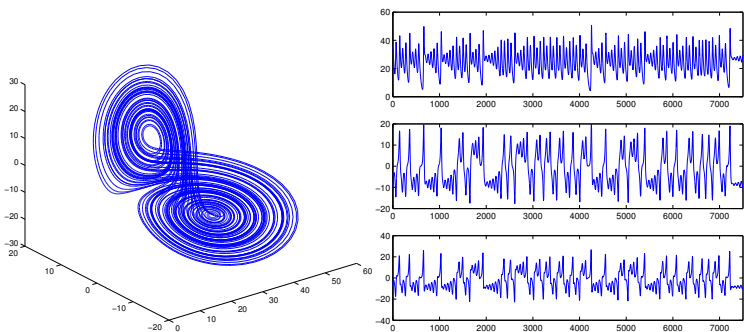


Abbildung D.4.: Lorenz-Attraktor Testdaten:

Neben Sinus- und Bewegungsdaten wurde auch Experimente mit Lorenz-Attraktordaten, als Beispiel chaotischer Zeitreihen, durchgeführt. Die einzelnen Dimensionen sind nicht unabhängig voneinander, so dass eine Prädiktion die Abhängigkeiten unter Umständen nutzen könnte. (Entnommen aus [Strauß, 2008])

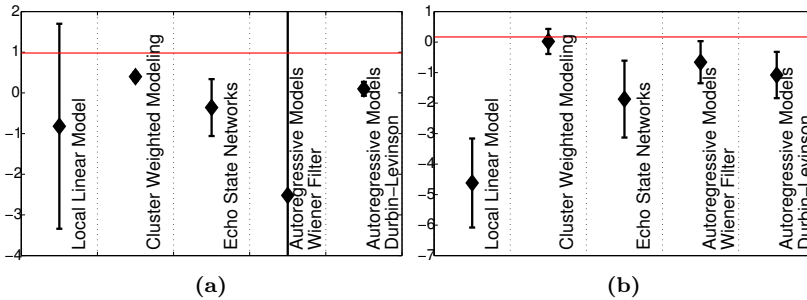


Abbildung D.5.: Fehler $K_{F_{log}}$ auf Sinusdaten und $L_{F_{log}}$ auf dem Lorenz-Attraktor: Links (a) dargestellt ist der Fehler von Kurzschrittprädiktionen auf Sinusfunktionen. Alle Algorithmen schaffen es hier den Referenzalgorithmus zu unterbieten. Auf der rechten Seite (b) sind die Fehlerwerte des LF dargestellt, welche bei den Lorenztests auftraten. Die LLMs liefern die mit Abstand besten Ergebnisse, gefolgt von den ESNs (Steil, Jaeger) und den AR-Modellen.

D.3.1. Sinus und Lorenzattraktor

Beim Prädizieren der Sinustrajektorien weist der Wiener Filter der autoregressiven Modelle im Mittel die besten Ergebnisse auf (vergleiche Abbildung D.5(a)). Die Varianz ist dabei allerdings außergewöhnlich hoch. Dies liegt darin begründet, dass autoregressive Modelle die Tendenz zeigen sich schnell aufzuschaukeln. Als Alternative erhält man mit Local Linear Models bzw. Echo State Netzen im Durchschnitt zwar schlechtere Ergebnisse, dafür liefern die Verfahren stabilere Ergebnisse. Zu beachten ist weiterhin, dass die alle getesteten Verfahren die Referenzalgorithmen klar übertreffen. In den Abbildungen wurde jeweils nur der mittlere Fehler des Referenzalgorithmus eingezeichnet. Die Abweichung wurde nicht dargestellt.

Bei der Prädiktion des chaotischen Lorenzattraktors in Abbildung D.5(b), liefern die Local Linear Models die besten Ergebnisse, während die autoregressiven Modelle ihre gute mittlere Prädiktionsgüte aus Abbildung D.5(a) nicht wiederholen können. Auch die Echo State Netz-

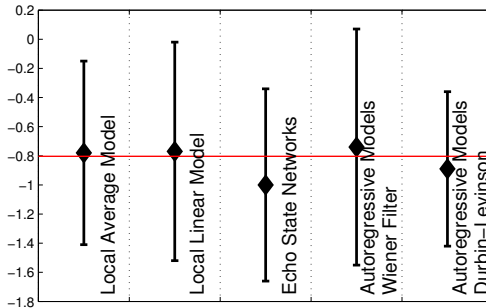


Abbildung D.6.: Fehler $K F_{log}$ auf Bewegungsdaten:

Im Kurzfehler schaffen nur die ESNs eine wirkliche Verbesserung gegenüber dem Referenzalgorithmus. Dahinter liegt die Durbin-Levinson-Variante der AR-Modelle. Die Local Models schaffen es zumindest an den Referenzalgorithmus heranzukommen.

werke liefern brauchbare Ergebnisse bei der Prädiktion. Wie auch auf den Sinustrajektorien schneiden alle Verfahren besser als die Referenzalgorithmen ab.

D.3.2. Realwelt Bewegungsdaten

In den Abbildung D.6 und D.7 sind der Kurz- bzw. Langschrittfehler ausgewählter Verfahren, getestet auf den in Abschnitt 3.1.1 beschriebenen Bewegungsdaten, dargestellt. Dabei sind für die einzelnen Verfahren die jeweils besten Varianten ausgewählt wurden. In [Strauß, 2008] findet sich zu den einzelnen getesteten Parameterkombinationen eine vollständige Übersicht.

Interessant an Abbildung D.6 ist, dass die Referenzalgorithmen im Vergleich zu den anderen Abbildungen in diesem Kapitel recht gute Ergebnisse erzielt. Daraus lässt sich schlussfolgern, dass zumindest für einen kurzen Prädiktionshorizont im Falle von Bewegungsdaten eine simple Annahme völlig ausreichend ist. Damit sind die üblicherweise verwendeten Bewegungsmodelle in Partikel- bzw. Kalmanfilter, wie sie in Trackingsystemen eingesetzt werden, durchaus gerechtfertigt. Benö-

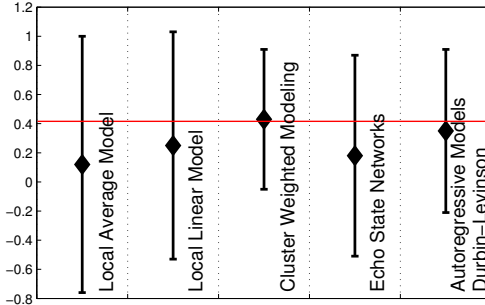


Abbildung D.7.: Fehler LF_{log} auf Bewegungsdaten:

Im Gegensatz zum Kurzfehler schaffen es die Local Models im Langfehler die höchste Prädiktionsgenauigkeit zu erzeugen. Knapp dahinter liegen die Jaeger-ESNs.

tigt man allerdings einen größeren Prädiktionshorizont, werden auch komplexere Annahmen notwendig.

Im Falle des Kurzschrittfehlers liefern die Echos State Netzwerke die besten Ergebnisse, während bei der Prädiktion über einen längeren Zeitraum die Local Linear Model geringfügig besser abschneiden. Die Cluster Weighted Models, die nur in Abbildung D.7 dargestellt sind, schaffen es bei Bewegungsdaten nicht mit dem besten Referenzalgorithmus mitzuhalten. Aufgrund dieser Ergebnisse ergibt sich eine klare Präferenz für Echo State Netzwerke und für Local Models in weiterführenden Untersuchungen.

Im allgemeinen ist die Abweichung der Algorithmen von den Referenzalgorithmen wesentlich kleiner als bei den Experimenten mit Sinus- und Lorenzattraktortrajektorien. Nichtsdestotrotz schaffen es die besten Algorithmen, die Referenz klar zu übertreffen. Vor allem sind sie in der Lage, die Bewegungsabläufe für einige Zeitschritte sinnvoll vorherzusagen.

Aufgrund der Ergebnisse ist anzunehmen, dass es sich bei der Prädiktion von Bewegungsabläufen um eine schwierigere Aufgabe handelt, als es bei den Sinus- und Lorenzattraktordaten der Fall ist. Dies liegt vor allem an dem spontanen bzw. unerwarteten Verhalten, welches bei den

Bewegungsabläufen beobachtet werden kann. Aus diesem Grund spielt beispielsweise auch die Wahl der Anzahl der Neuronen bei den Echo State Netzwerken eine untergeordnete Rolle. Die Ergebnisse, die bei Echo State Netzwerken erzielt wurden, unterscheiden sich nicht signifikant bei der Wahl von 25 bis 250 Neuronen. Anders als bei anderen chaotischen Zeitreihen kann die Prädiktionsgenauigkeit, wie in [Jaeger und Haas, 2004] beschrieben, durch Erhöhen der Neuronenanzahl nicht weiter verbessert werden.

D.3.3. Berechnungszeit

Um Anwendungen auch im Realwelteinsatz benutzen zu können, spielt die Rechenzeit eine wesentliche Rolle. Insbesondere soll die Berechnung der Prädiktion nicht länger dauern, als die Länge des Vorhersagehorizontes. Für die Implementierung der einzelnen Algorithmen wurde auf Matlab zurückgegriffen. Dadurch sollen die in Abbildung D.8 veranschaulichten Rechenzeiten nur zur Orientierung dienen. In der Grafik sind jeweils für unterschiedlich lange Prädiktionshorizonte die Rechenzeiten dargestellt. Die Rechenzeit wird dabei in den für das Training nötigen Teil und in die eigentliche Prädiktion unterschieden.

Autoregressive Modelle und Echo State Netzwerke mit geringer Neuronenanzahl benötigen eine Rechenzeit von 0.7 ms bis 10 ms pro Prädiktionsschritt. Von seiten der Rechenzeiten können sie daher durchaus als realwelttauglich eingeschätzt werden.

Local Models und Cluster Weighted Models benötigen wesentlich länger. Bei den Local Models ist die Nachbarschaftssuche in der großen Anzahl an Trainingsdaten der wesentliche Grund für die hohe Rechenzeit. An dieser Stelle kann durch eine geeignetere Strategie noch recht viel Zeit eingespart werden. Bei den Cluster Weighted Models spielt der Optimierungsablauf mit Hilfe des EM-Algorithmus eine große Rolle.

D.4. Fazit

In den vergleichenden Untersuchungen wurden vier unterschiedliche Algorithmen gegenübergestellt. Untersucht wurden dabei sowohl die An-

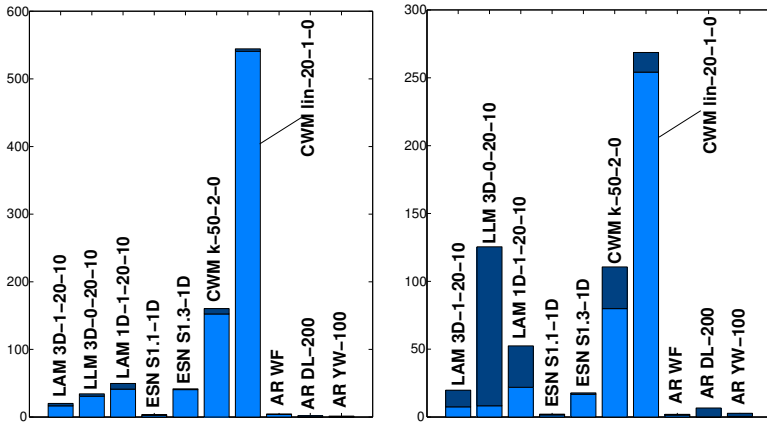
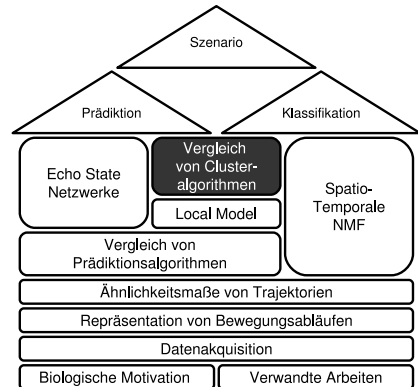


Abbildung D.8.: Trainings- und Prädiktionszeiten der Algorithmen: Auf der linken Seite erfolgte die Berechnung von 100 Prädiktionsschritten auf Basis einer vorhandenen Lernzeitreihe der Länge 900, während auf der rechten Seite beide Größen jeweils 500 betragen. Im Bild hellgrau dargestellt ist die Trainingszeit, die hauptsächlich von der Länge der Lernzeitreihe abhängig ist und daher auf dem linken Diagramm dominiert. Dunkelgrau gefärbt ist die Zeit, welche für die Prädiktion benötigt wurde. Alle Zeiten sind in Sekunden angegeben und resultierten aus MatLab-Berechnungen. Auffällig sind die teilweise sehr hohen Berechnungszeiten bei LM und CWM, welche beim LM in hohen Prädiktionszeiten begründet sind, wenn viele Schritte zu präzisieren sind (rechtes Diagramm). Besonders wenn beim LLM viele Parameterneuberechnungen durchzuführen sind, steigt diese Zeit stark an (jeweils 2.Säule). (Entnommen aus [Strauß, 2008])

forderungen an die Rechenzeit als auch die Prädiktionsgüte. Bei der Prädiktionsgüte wurden sowohl Kurzzeit-, als auch Langzeitprädiktionen betrachtet. Bei allen Tests haben die Echo State Netzwerke sehr gute Ergebnisse erzielt. Kurz hinter den Echo State Netzwerken sind die Local Models einzuordnen. Beide Verfahren wurden für weitere Untersuchungen ausgewählt, die in den Kapiteln 4 und 5 aufgezeigt sind.

Anhang E

Vergleich von Cluster- algorithmen



Für die Erweiterung der Local Models wurde nach einer Möglichkeit gesucht, eine beliebig große Wissensbasis zur Verfügung zu stellen. Da mit dem Anwachsen der Wissensbasis aufgrund der Nachbarsuche die Rechenzeit ebenfalls anwächst, ist es notwendig, die Wissensbasis sinnvoll zu repräsentieren. Am geeignetsten scheint es, die Wissensbasis zu unterteilen und für die Prädiktion nur den entsprechenden Teil auszuwählen. Im Folgenden sollen daher die in Abbildung E.1 hervorgehobenen Clusteralgorithmen untersucht werden, wie gut sie in der Lage sind, den Einbettungsraum zu unterteilen. Aufgrund der Vielzahl der in [Matthias, 2008]¹ untersuchten Algorithmen sollen jeweils nur die grundlegenden Ideen erwähnt werden. Die weiteren Details sind den entsprechenden Literaturstellen zu entnehmen.

¹Die Masterarbeit wurde vom Autor im Rahmen des Projektes betreut.

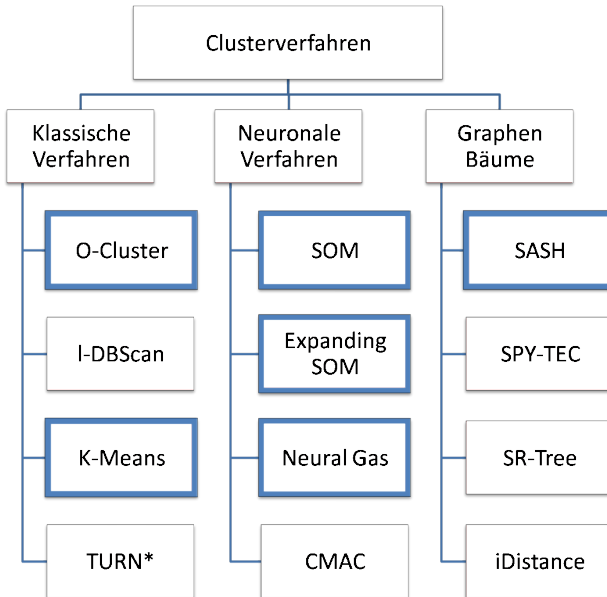


Abbildung E.1.: Untersuchte Clusteralgorithmen:

Die Abbildung zeigt die in [Matthias, 2008] untersuchten Prädiktionsalgorithmen. Für die hervorgehobenen werden die Experimente in diesem Anhang vorgestellt, um die Auswahl für die Erweiterung der Local Models (Kapitel 5) zu begründen.

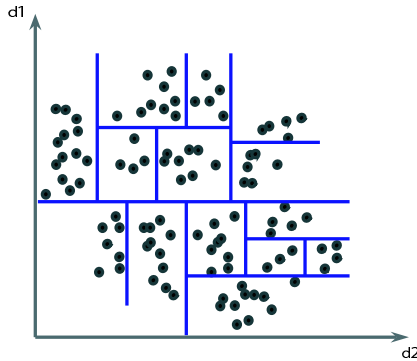


Abbildung E.2.: Unterteilung des Raumes auf Basis des O-Cluster-Verfahrens: *Durch das sukzessive Festlegen von Schnittebenen wird der Raum achsparallel unterteilt. (Entnommen aus [Matthias, 2008])*

E.1. O-Cluster

Die Idee des in [Milenova und Campos, 2002] beschriebenen O-Cluster Algorithmus besteht in der sukzessiven Unterteilung des zu clusterenden Raumes mit Hilfe von Schnittebenen². Der Algorithmus stellt eine Weiterentwicklung des OptiGrid-Verfahrens von [Hinneburg und Keim, 1999] dar.

Bei beiden Verfahren werden die Datenpunkte auf die jeweiligen Raumachsen projiziert. Auf Basis der Häufungen der Punkte bezüglich der jeweiligen Raumachsen werden Kandidaten zum Unterteilen des Raumes bestimmt. Die Raumachse bzw. die entsprechenden Hypothesen, die die beste Unterteilung liefern, wird gewählt und der Raum in zwei Halbräume unterteilt. Für diese beiden Halbräume wird rekursiv weiterverfahren, um eine Unterteilung des gesamten Raumes, wie in Abbildung E.2 zu erreichen.

Das Festlegen, ob eine Unterteilung als gute bzw. sogar beste Unterteilung angesehen werden kann, erfolgt durch einen χ^2 -Test. Dazu wird ein Histogramm der Datenverteilung entlang der projizierten Raumach-

²Cutting Planes

se erstellt. Das Ziel ist es, eine Schnittebene in eine Region mit geringer Dichte zu platzieren, wobei sich auf beiden Seiten der Ebene hohe Dichtendichten befinden sollen. Mit Hilfe des χ^2 -Tests wird entschieden, ob der Unterschied zwischen den vermuteten Regionen hoher und niedriger Dichte statistisch signifikant ist.

E.2. Neuronale Ansätze

In dem Gebiet der Neuroinformatik existieren auch eine gewisse Anzahl an Verfahren, die durch die rezeptiven Felder der Neuronen in der Lage sind, einen Raum zu segmentieren. Aus diesen Verfahren sind drei für die Experimente herausgegriffen worden.

E.2.1. Neuronales Gas

Das Neuronale Gas [Kohonen, 1995] enthält eine feste Anzahl von Neuronen, die zufällig im Eingaberaum initialisiert werden. Zum Anpassen ihrer Position werden die Trainingsdatenpunkte sequentiell präsentiert. Für jeden Datenpunkt wird das Neuron bestimmt, welches den geringsten Euklidischen Abstand³ aufweist. Dieses Neuron, welches man als Best-Matching-Neuron bezeichnet, wird entsprechend einer definierten Lernrate in Richtung des gegebenen Inputs verschoben.

Zusätzlich zur Adaption des Best-Matching-Neurons werden alle anderen Neuronen auch trainiert. Dazu wird eine Reihenfolge entsprechend ihrer Abstände zum aktuell präsentierten Datenpunkt festgelegt. Auf Basis dieser Reihenfolge wird mit Hilfe einer sogenannten Nachbarschaftsfunktion die Stärke der Anpassung festgelegt. Die Nachbarschaftsfunktion wird so festgelegt, dass Neuronen, die ohnehin schon recht nah am Datenpunkt lagen, wesentlich stärker zu ihm hin verschoben werden als Neuronen in weiter entfernten Bereichen.

³Aus formaler Sicht sind auch andere Distanzmaße geeignet.

E.2.2. Self Organizing Feature Map

Die Self Organizing Feature Maps (SOM bzw. SOFM) [Kohonen, 1990] funktionieren nach demselben Grundprinzip wie das Neuronale Gas. Auch bei ihnen werden die Neuronen in der Nachbarschaft des Best-Matching-Neurons in abgeschwächter Form adaptiert. Die Nachbarschaft wird allerdings nicht mehr durch den Abstand zum präsentierten Input definiert. Vielmehr werden die Neuronen über eine Graphenstruktur verknüpft. Die durch den Graphen ausgedrückte Topologie wird durch den Designer vorgegeben und soll der tatsächlichen Anordnung der Daten im Raum entsprechen. Die Nachbarschaft wird bei den Self Organizing Feature Maps über das Vorhandensein einer Kante im Graphen definiert.

E.2.3. Expanding Self Organizing Feature Maps

Mit den Expanding Self Organizing Feature Maps (ESOM) [Shum et al., 2002] wurde eine aktuelle Weiterentwicklung der SOM in die experimentellen Untersuchungen aufgenommen. Im Unterschied zu den Self Organizing Feature Maps werden alle Trainingsdaten in das Innere einer Kugel⁴ transformiert.

Das Training läuft nach demselben Prinzip, wie in Abschnitt E.2.2 beschrieben, ab. Allerdings wird nach dem Anpassen der Neuronenposition mit einem sogenannten Expansionskoeffizienten skaliert. Dadurch wandern die Neuronen zur Oberfläche der Kugel bzw. sie expandieren weg vom Zentrum der Trainingsdaten. Auf Grund dieser expandierenden Kraft, die auf die Neuronen ausgewirkt wird, sollen Topologiedefekte, die häufig bei Self Organizing Feature Maps auftreten, vermieden werden.

E.3. K-Means

Beim k-Means-Clusterverfahren geht man von einer Menge von Datenpunkten in einem beliebigen Raum aus. Das Problem besteht darin, aus

⁴genaugenommen einer Einheitshyperkugel

dieser Punktmenge k sogenannte Center- oder Seed-Punkte auszuwählen, so dass der mittlere quadratische Abstand jedes Datenpunktes in der Gesamtmenge zu seinem jeweils nächsten Center minimal ist. Allerdings existiert kein Algorithmus, der in polynomieller Zeit zu einer exakten Lösung kommt. Daher existieren eine Menge approximative bzw. heuristische Varianten des k -Means. Ein guter Überblick dazu kann in [Kanungo et al., 2003] nachgelesen werden.

E.4. SASH

Beim in [Houle und Sakuma, 2005] vorgestellten SASH-Algorithmus handelt es sich nicht um ein Clusterverfahren im eigentlichen Sinne, mit dem eine Segmentierung des Raumes vorgenommen wird. Vielmehr werden die Datenpunkte der Trainingsmenge so sortiert, dass eine schnelle Suche von benachbarten Punkten möglich wird. Da für das Local Model die Nachbarschaftssuche eine entscheidende Rolle spielt, wurde es in den Vergleich aufgenommen.

Die Abkürzung SASH steht für den Begriff Spatial Approximation Sample Hierarchie. In dem Namen wird durch den Begriff Hierarchie angedeutet, dass eine baumartige Struktur aufgebaut wird. Tatsächlich handelt es sich nicht um einen Baum im mathematischen Sinne, da Knoten in einer tieferen Schicht mehrere Vorgängerknoten haben können. Jeder Knoten der Struktur bzw. des Graphen steht für einen Datenpunkt. Welcher Datenpunkt welche Position im Grafen erhält, spielt keine Rolle. Es existiert lediglich die Forderung, dass von Hierarchiestufe zu Hierarchiestufe die Anzahl der Datenpunkte bzw. Knoten verdoppelt wird. Erst durch die Verknüpfung der Knoten wird die räumliche Beziehung der Datenpunkte repräsentiert. Dabei wird vom Wurzelknoten zu den Blättern vorgegangen. Ein Knoten wird dann mit den Knoten in der darunterliegenden Schicht verknüpft, wenn dieser zu einer Menge von nächsten Nachbarn gehört. Die Mächtigkeit der Menge dient als Parameter des Verfahrens. Innerhalb einer Hierarchiestufe erfolgt keine Verknüpfung.

Auf diese Weise kann für einen Datenpunkt der SASH-Graph von der Wurzel zu den Blättern durchlaufen und in jeder Hierarchiestufe eine definierte Anzahl potenzieller nächster Nachbarn zurückgeliefert wer-

den. Aus diesen Nachbarschaftskandidaten werden als letztes die nächsten ausgewählt. Somit wird der Suchaufwand erheblich reduziert.

E.5. Experimenteller Vergleich

Als erstes soll untersucht werden, in wie weit eine Clusterung die Prädiktionsergebnisse beeinflusst. Dazu wird als erstes eine Prädiktion mit der gesamten Wissensbasis durchgeführt. Die Prädiktion dient für die weiteren Vergleiche als Referenz. Durch die Auswahl eines Clusters für die Prädiktion, wird die Datenbasis, aus der die Nachbarn bestimmt werden, stark reduziert. Eine Clusterung, und damit ein gutes Clusterverfahren, sollte sich dadurch auszeichnen, dass in den nicht betrachteten Clustern möglichst nur Punkte enthalten sind, die ohnehin für Nachbarschaftsauswahl nicht in Frage gekommen wären.

Zur Evaluierung wurden die in Abschnitt 3.1.1 vorgestellten Bewegungsdaten verwendet. Insgesamt wurden 236 repräsentative Trajektorien ausgewählt. Damit ergibt sich zur Clusterung ein Datenvolumen von ca. 200 000 Positionen, die in den Einbettungsraum überführt und anschließend geclustert werden.

Das verwendete Fehlermaß orientiert sich an dem Langschrittfehler aus Abschnitt 4.3.2. Allerdings wird nicht mit dem originalen Verlauf, sondern mit den auf Basis der gesamten Wissensbasis prädizierten Punkten \mathbf{s}_i^{GT} verglichen:

$$LF_{GT} = \frac{1}{N \cdot \sigma^2} \cdot \frac{\sum_{p=1}^N \frac{1}{\sqrt{p}} (\mathbf{s}_i^{pred} - \mathbf{s}_i^{GT})^2}{\sum_{p=1}^N \frac{1}{\sqrt{p}}} \quad (\text{E.1})$$

In Abbildung E.3 sind die entsprechenden Ergebnisse dargestellt. Auch für diese Darstellung wurde der Fehler logarithmiert abgetragen. Dabei fällt auf, dass sämtliche Clusterverfahren Prädiktionsergebnisse ermöglichen, die sehr nah an die Referenz herankommen. Insgesamt schneiden beide Varianten der Self Organizing Feature Map am besten ab. Dabei ist besonders interessant, dass signifikante Unterschiede zwischen der Prädiktion mit Local Average Models und Local Linear Models existieren.

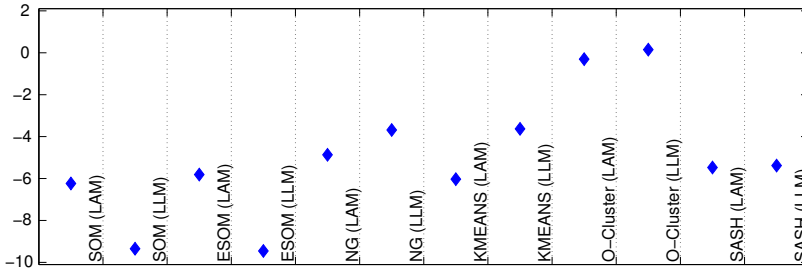


Abbildung E.3.: Fehler zur Ground-Truth-Referenz:

Für jedes Clusterverfahren wurde eine Prädiktion auf Basis der Local Average Models (LAM) und der Local Linear Models (LLM) errechnet. Diese Prädiktion wurde mit einer Prädiktion verglichen, die auf der gesamten Wissensbasis beruht.

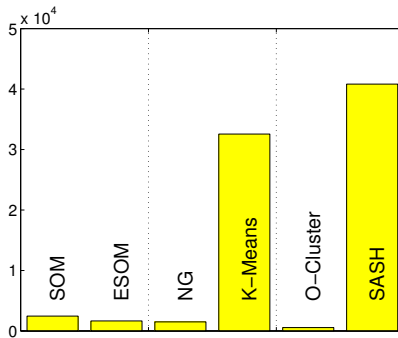


Abbildung E.4.: Rechenzeiten für das Ermitteln der Cluster:

Mit jedem der Clusterverfahren wurde eine Segmentierung der Daten bestimmt und damit die Rechenzeit ermittelt. Auch wenn diese Berechnung offline durchgeführt werden kann, ist aufgrund der enormen Unterschiede auf die Verwendung von K-Means oder SASH zu verzichten.

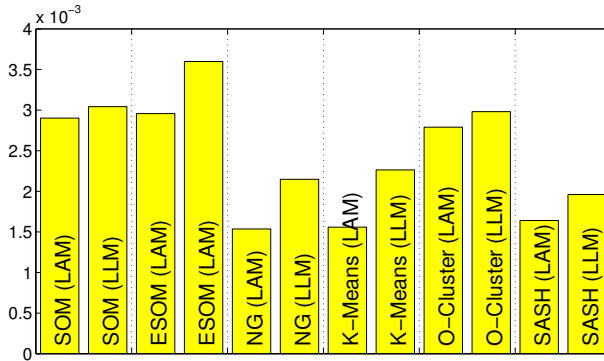


Abbildung E.5.: Rechenzeiten für einen Prädiktions-schritt: Für die Prädiktion wurde jeweils der am besten passende Cluster für die zu prädizierende Beobachtung bestimmt. Nur die Daten aus dem entsprechenden Cluster wurden für die Nachbarschaftsuche der Local Models verwendet.

Neben den Prädiktionsergebnissen ist natürlich noch interessant, wie schnell die einzelnen Algorithmen arbeiten. Dabei werden zwei Berechnungszeiten betrachtet. In Abbildung E.4 wird die Zeit, die zum Erstellen der Clusterung benötigt wird, verglichen. Theoretisch kann das Erstellen der Cluster im Vorfeld erfolgen und spielt damit im Sinne eines echtzeitfähigen Einsatzes keine Rolle. Perspektivisch sollte man allerdings die Clusterung auch dynamisch anpassen können. In einem solchen Fall wird die Rechenzeit auch für den Echtzeitbetrieb interessant.

Abbildung E.5 stellt die Zeit, die für einen Prädiktions-schritt benötigt wird, dar. Diese Zeit ist allerdings nicht nur abhängig von dem verwendeten Local Model. Es spielt auch die Zeit für die Auswahl des entsprechenden Cluster eine Rolle. Außerdem schlägt sich die Anzahl der Datenpunkte in einem Cluster auf die Rechenzeit nieder. Wie in der Abbildung zu erkennen ist, schneidet das Neuronale Gas (NG) am besten ab.

E.6. Fazit

Wird die Clusterung mit einer Self Organizing Feature Map durchgeführt, werden in Kombination mit Local Linear Models die besten Prädiktionsergebnisse erzielt. Auch SASH und K-Means liefern brauchbare Prädiktionsergebnisse. Allerdings benötigen beide Verfahren viel Rechenzeit, um die Clusterung durchzuführen. Bei der Zeit für die eigentliche Prädiktion, die die Auswahl des aktuellen Clusters beinhaltet, schneidet das Neuronale Gas am besten ab. Daher wird für weitere Untersuchungen die Self Organizing Feature Map bzw. das recht ähnliche Neuronale Gas gewählt.

Literaturverzeichnis

- [Agrawal et al., 1993] Agrawal, R., Faloutsos, C., und Swami, A. N. (1993). Efficient Similarity Search In Sequence Databases. In *FO-DO '93: Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, Seiten 69–84, London, UK. Springer-Verlag.
- [Bäck et al., 1991] Bäck, T., Hoffmeister, F., und Schwefel, H.-P. (1991). A survey of evolution strategies. In Belew, R. und Booker, L., editors, *Proceedings of the fourth International Conference on Genetic Algorithms*, Seiten 2–9.
- [Baier, 2006] Baier, V. (2006). *Motion Perception and Prediction*. PhD thesis, Technische Universität München.
- [Beauchamp et al., 2002] Beauchamp, M., Lee, K., Haxby, J., und Martin, A. (2002). Parallel visual motion processing streams for manipulable objects and human movements. *Neuron*, 34:149–159.
- [Bennewitz et al., 2005] Bennewitz, M., Burgard, W., Cielniak, G., und Thrun, S. (2005). Learning Motion Patterns of People for Compliant Robot Motion. *The International Journal of Robotics Research*, 24:31–48.
- [Bennewitz et al., 2002] Bennewitz, M., Burgard, W., und Thrun, S. (2002). Learning Motion Patterns of Persons for Mobile Service Robots. In *ICRA, Proc. of the VDI-Conference Robotik 2002 (Robotik)*, Seiten 3601–3606. Proc. of the IEEE International Conference on Robotics & Automation (ICRA).

- [Berry et al., 2007] Berry, M. W., Browne, M., Langville, A. N., Pauca, P. V., und Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155–173.
- [Besl und McKay, 1992] Besl, P. J. und McKay, N. D. (1992). A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256.
- [Bolder et al., 2007] Bolder, B., Dunn, M., Gienger, M., Janssen, H., Sugiura, H., und Goerick, C. (2007). Visually Guided Whole Body Interaction. In *ICRA*, Seiten 3054–3061.
- [Bonda et al., 1996] Bonda, E., Petrides, M., Ostry, D., und Evans, A. (1996). Specific involvement of human parietal systems and the amygdala in the perception of biological motion. *Journal of Neuroscience*, 16:3737–3744.
- [Brakensiek, 2002] Brakensiek, A. (2002). *Modellierungstechniken und Adaptionsverfahren für die On- und Off-Line Schrifterkennung*. Dissertation, Technische Universität München, München.
- [Bresenham, 1965] Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30.
- [Brogan und Johnson, 2003] Brogan, D. C. und Johnson, N. L. (2003). Realistic Human Walking Paths. In *CASA*, Seiten 94–.
- [Bruce und Gordon, 2004] Bruce, A. und Gordon, G. (2004). Better Motion Prediction for People-tracking. In *In Proceedings of ICRA 2004*.
- [Bui et al., 2001] Bui, H. H., Venkatesh, S., und West, G. (2001). Tracking and Surveillance in Wide-Area Spatial Environments Using the Abstract Hidden Markov Model. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 15(1):177–195.
- [Cai und Ng, 2004] Cai, Y. und Ng, R. (2004). Indexing Spatio-Temporal Trajectories with Chebyshev Polynomials. In *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, Seiten 599–610, New York, NY, USA. ACM.
-

- [Cemgil et al., 2006] Cemgil, A., Kappen, B., und Barber, D. (2006). A generative model for music transcription. *IEEE Transactions on Speech and Audio Processing*, 14:679–694.
- [Chakrabarti et al., 2001] Chakrabarti, K., Keogh, E., Mehrotra, S., und Pazzani, M. (2001). Locally adaptive dimensionality reduction for indexing large time series databases. In *SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, volume 27. No.2, Seiten 151 – 162, New York, NY, USA. ACM.
- [Chakrabarti et al., 2002] Chakrabarti, K., Keogh, E., Mehrotra, S., und Pazzani, M. (2002). Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Trans. Database Syst.*, 27(2):188–228.
- [Chappell und Taylor, 1993] Chappell, G. und Taylor, J. (1993). The temporal Kohonen map. *Neural Networks*, 6:441–445.
- [Cutting, 1978] Cutting, J. E. (1978). A program to generate synthetic walkers as dynamic point-light displays. *Behavior Research Methods & Instrumentation*, 10(1):91–94.
- [Dixon und Khosla, 2004] Dixon, K. R. und Khosla, P. K. (2004). Trajectory Representation Using Sequenced Linear Dynamical Systems. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*.
- [Downing et al., 2001] Downing, P., Jiang, Y., Shuman, M., und Kanwisher, N. (2001). A cortical area selective for visual processing of the human body. *Science*, 293:2470–2473.
- [Eggert et al., 2009] Eggert, J., Hellbach, S., Kolarow, A., Körner, E., und Groß, H.-M. (2009). Prediction and classification of motion trajectories using spatio-temporal nmf. In *32nd Annual Conference on Artificial Intelligence Paderborn (GER)*, LNCS 5803, Seiten 597–606. Springer.
- [Eggert und Körner, 2004] Eggert, J. und Körner, E. (2004). Sparse Coding and NMF. In *IJCNN*, Seiten 2529 – 2533.
- [Eggert et al., 2004] Eggert, J., Wersing, H., und Körner, E. (2004). Transformation-invariant representation and NMF. In *IJCNN*, Sei-
-

ten 2535 – 2539.

- [Elfes, 1989] Elfes, A. (1989). Using Occupancy Grids for Mobile Robot Perception and Navigation. *Computer*, 12(6):46–57.
- [Elman, 1990] Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.
- [Engster und Parlitz, 2006] Engster, D. und Parlitz, U. (2006). *Handbook of Time Series Analysis*, chapter Local and Cluster Weighted Modeling for Time Serie Prediction, Seiten 38 – 65. WILEY-VCH, Weinheim.
- [Eurich et al., 1995] Eurich, C., Roth, G., Schwegler, H., und Wiggers, W. (1995). Simulander: A neural network model for the orientation movement of salamanders. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 176(3):379–389.
- [Eurich et al., 1997] Eurich, C., Schwegler, H., und Woesler, R. (1997). Coarse coding: applications to the visual system of salamanders. *Biological Cybernetics*, 77:41–47.
- [Faloutsos et al., 1997] Faloutsos, C., Jagadish, H., Mendelzon, A., und Milo, T. (1997). A Signature Technique for Similarity-Based Queries. In *SEQUENCES '97: Proceedings of the Compression and Complexity of Sequences 1997*, Seite 2, Washington, DC, USA. IEEE Computer Society.
- [Govea et al., 2004] Govea, V., Alejandro, D., und Fraichard, T. (2004). Motion Prediction for Moving Objects: a Statistical Approach. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Seiten 3931–3936, New Orleans, LA (US).
- [Graf et al., 2000] Graf, B., Baum, W., Traub, A., und Schraft, R. (2000). Konzeption dreier Roboter zur Unterhaltung der Besucher eines Museums. In *Robotik 2000. Leistungsstand, Anwendungen, Visionen, Trends : Tagung Berlin, 29. und 30. Juni 2000*, Seiten 529–536.
- [Gross und Böhme, 2000] Gross, H.-M. und Böhme, H.-J. (2000). PERSES - a Vision-based Interactive Mobile Shopping Assistant. In *in: Proc. IEEE Int. Conference on Systems, Man and Cybernetics*
-

- (*IEEE-SMC 2000*), Seiten 80–85, Nashville. IEEE Omnipress.
- [Gross et al., 2008] Gross, H.-M., Böhme, H.-J., Schröter, C., Müller, S., König, A., Martin, C., Merten, M., und Bley, A. (2008). Shop-Bot: Progress in Developing an Interactive Mobile Shopping Assistant for Everyday Use. In *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics (IEEE-SMC)*, Seiten 3471–3478.
- [Grossman und Blake, 2002] Grossman, E. und Blake, R. (2002). Brain areas active during visual perception of biological motion. *Neuron*, 35:1167–1175.
- [Grossman et al., 2000] Grossman, E., Donnelly, M., Price, R., Pickens, D., Morgan, V., Neighbor, G., und Blake, R. (2000). Brain areas involved in perception of biological motion. *Journal of Cognitive Neuroscience*, 12:711–720.
- [Guyon et al., 2006] Guyon, I., Gunn, S., Nikravesh, M., und Zadeh, L., editors (2006). *Feature Extraction, Foundations and Applications*. Springer.
- [Hahn et al., 2008] Hahn, M., Krüger, L., und Wöhler, C. (2008). 3D Action Recognition and Long-Term Prediction of Human Motion. In *International Conference in Computer Vision Systems (ICVS)*, Seiten 23–32.
- [Hellbach et al., 2009a] Hellbach, S., Eggert, J., Körner, E., und Gross, H.-M. (2009a). Basis Decomposition of Motion Trajectories using Spatio-Temporal NMF. In *Int. Conf. on Artificial Neural Networks (ICANN) 2009, Limassol, (CY)*, LNCS 5769, Seiten 804–814. Springer.
- [Hellbach et al., 2009b] Hellbach, S., Eggert, J., Körner, E., und Gross, H.-M. (2009b). Time Series Analysis for Long Term Prediction of Human Movement Trajectories. In *International Conference on Neural Information Processing (ICONIP 2008), Auckland (NZ)*, LNCS 5507, Seiten 567–574. Springer.
- [Hellbach et al., 2007] Hellbach, S., Lau, B., Eggert, J. P., Körner, E., und Gross, H.-M. (2007). Optimization-Based Motion Segmentation using Additional Visual Cues. In *Proc. 52. International Scientific Colloquium*, Seiten 327–332. Universitätsverlag Ilmenau.
-

- [Hellbach et al., 2008] Hellbach, S., Strauss, S., Eggert, J., Körner, E., und Gross, H.-M. (2008). Echo State Networks for Online Prediction of Movement Data - Comparing Investigations. In *Int. Conf. on Artificial Neural Networks (ICANN) 2008, Prague (CR)*, LNCS 5163, Seiten 710–719. Springer.
- [Hellbach und Wollenschläger, 2009] Hellbach, S. und Wollenschläger, J. (2009). Evaluierung unterschiedlicher Möglichkeiten zur Repräsentation von Bewegungsabläufen. Technical Report S457/09, Technische Universität Ilmenau, Fachgebiet Neuroinformatik und Kognitive Robotik.
- [Hellwig, 2009] Hellwig, S. (2009). Vergleich verschiedener Ähnlichkeitsmaße für Funktionsverläufe am FG Neuroinformatik und Kognitive Robotik. Studienarbeit, Technische Universität Ilmenau. Betreuer: Sven Hellbach und Erik Schaffernicht; Betreuender Hochschullehrer: Horst-Michael Groß.
- [Helsper, 2008] Helsper, S. (2008). Untersuchungen zur Online-Fähigkeit von Algorithmen zur Prädiktion von Trajektorien. Studienarbeit, Technische Universität Ilmenau. Betreuer: Sven Hellbach; Betreuender Hochschullehrer: Horst-Michael Groß.
- [Hesse, 2009] Hesse, W. (2009). *Untersuchung gerichteter Interaktion multivariater biomedizinischer Signale auf Basis des Verhersageprinzips nach Granger*. PhD thesis, Technische Universität Ilmenau.
- [Hinneburg und Keim, 1999] Hinneburg, A. und Keim, D. A. (1999). Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering. In *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, Seiten 506–517. ACM Digital Library, Morgan Kaufmann Publishers Inc. VLDB1999.
- [Houle und Sakuma, 2005] Houle, M. E. und Sakuma, J. (2005). Fast Approximate Similarity Search in Extremely High-Dimensional Data Sets. In *Proceedings of the 21st International Conference on Data Engineering*, Seiten 619–630. IEEE Computer Society. ICDE2005.
- [Hu et al., 2004] Hu, W., Tan, T., Wang, L., und Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on systems, man and cybernetics. Part C : Applications*
-

- and reviews*, 34(3):334–352.
- [Ilg und Giese, 2002] Ilg, W. und Giese, M. A. (2002). Modeling of Movement Sequences Based on Hierarchical Spatial-Temporal Correspondence of Movement Primitives. In *BMCV '02: Proceedings of the Second International Workshop on Biologically Motivated Computer Vision*, Seiten 528–537, London, UK. Springer-Verlag.
- [Jaeger und Haas, 2004] Jaeger, H. und Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. *Science*, April:78 – 80.
- [Jin et al., 2004] Jin, X., Wang, J., und Sun, J.-G. (2004). Symbolization of Mobile Object Trajectories with the Support to Motion Data Mining. In *ER (Workshops)*, Seiten 103–113.
- [Johansson, 1973] Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Visual perception of biological motion and a model for its analysis.*, 14:201–211.
- [Johansson, 1974] Johansson, G. (1974). Vector Analysis in Visual Perception of Rolling Motion - A Quantitative Approach. *Psychologische Forschung*, 36:311–319.
- [Johnson und Hogg, 1996] Johnson, N. und Hogg, D. (1996). Learning the Distribution of Object Trajectories for Event Recognition. *Image and Vision Computing*, 14:609–615.
- [Jordan, 1986] Jordan, M. I. (1986). Serial Order: A Parallel Distributed Processing Approach. Technical Report ICS Report 8604, Institute for Cognitive Science, University of California at San Diego, La Jolla, CA.
- [Junejo und Foroosh, 2007] Junejo, I. N. und Foroosh, H. (2007). Trajectory Rectification and Path Modeling for Video Surveillance. In *Proc. IEEE 11th International Conference on Computer Vision IC-CV 2007*, Seiten 1–7.
- [Kalman, 1960] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45.
-

- [Kanungo et al., 2003] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., und Wu, A. Y. (2003). A Local Search Approximation Algorithm for k-Means Clustering.
- [Keogh et al., 2000] Keogh, E. J., Chakrabarti, K., Pazzani, M. J., und Mehrotra, S. (2000). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286.
- [Khalid und Naftel, 2005] Khalid, S. und Naftel, A. (2005). Classifying Spatiotemporal Object Trajectories using Unsupervised Learning of Basis Function Coefficients. In *International Multimedia Conference*.
- [Kohonen, 1990] Kohonen, T. (1990). The Self-Organizing Map. In *Proceedings of the IEEE*, volume 78-3, Seiten 1464–1480. IEEE Computer Society.
- [Kohonen, 1995] Kohonen, T. (1995). *Self-Organizing Maps*. Springer-Verlag, Berlin.
- [Koskela et al., 1998] Koskela, T., Varsta, M., Heikkonen, J., und Kaski, K. (1998). Temporal sequence processing using recurrent SOM. In *Proc. Second International Conference on Knowledge-Based Intelligent Electronic Systems KES '98*, volume 1, Seiten 290–297.
- [Kuipers, 1983] Kuipers, B. (1983). Modeling Human Knowledge of Routes: Partial Knowledge and Individual Variation. In *AAAI*, Seiten 216–219.
- [Lange und Lappe, 2006] Lange, J. und Lappe, M. (2006). A Model of Biological Motion Perception from Configurational Form Cues. *The Journal of Neuroscience*, 26(11):2894–2906.
- [Lau, 2007] Lau, B. (2007). Optimization-Based Motion Segmentation Using Additional Visual Cues. Diplomarbeit, Technische Universität Ilmenau. Betreuer: Sven Hellbach; Betreuender Hochschullehrer: Horst-Michael Groß.
- [Lee und Nakamura, 2007] Lee, D. und Nakamura, Y. (2007). Mimesis Scheme using a Monocular Vision System on a Humanoid Robot. In *ICRA*, Seiten 2162–2168.
-

- [Lee und Seung, 2001] Lee, D. D. und Seung, H. S. (2001). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing*, 13:556–562.
- [Lee et al., 2003] Lee, K. K., Yu, M., und Xu, Y. (2003). Modeling of human walking trajectories for surveillance. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 2, Seiten 1554–1559.
- [Li et al., 1996] Li, C.-S., Yu, P. S., und Castelli, V. (1996). HierarchyScan: A Hierarchical Similarity Search Algorithm for Databases of Long Sequences. In *ICDE '96: Proceedings of the Twelfth International Conference on Data Engineering*, Seiten 546–553, Washington, DC, USA. IEEE Computer Society.
- [Li, 2001] Li, S. Z. (2001). *Markov Random Field Modeling in Image Analysis*. Springer-Verlag.
- [Loganathan und Huber, 2008] Loganathan, A. P. und Huber, M. (2008). An Approach for Behavior Discovery using Clustering of Dynamics. IEEE International Conference on Systems, Man and Cybernetics (SMC2008).
- [Matthias, 2008] Matthias, R. (2008). Einsatz von Cluster-Algorithmen zur Verbesserung der Prädiktionsgüte von Bewegungsabläufen. Masterarbeit, Fachhochschule Erfurt. Betreuer: Sven Hellbach; Betreuender Hochschullehrer: Gunnar Schorch und Horst-Michael Groß.
- [Méary et al., 2007] Méary, D., Kitromilides, E., Mazens, K., Graff, C., und Gentaz, E. (2007). Four-Day-Old Human Neonates Look Longer at Non-Biological Motions of a Single Point-of-Light. *PLoS ONE*, 2(1):1–5.
- [Milenova und Campos, 2002] Milenova, B. L. und Campos, M. M. (2002). O-Cluster: Scalable Clustering of Large High Dimensional Data Sets. In *Proceedings 2002 IEEE International Conference on Data Mining*, Seiten 290–297. IEEE Computer Society. ICDM2002.
- [Muehlig et al., 2009] Muehlig, M., Gienger, M., Hellbach, S., Steil, J. J., und Goerick, C. (2009). Interactive Learning of Movement Tasks through Imitation. In *2009 IEEE International Conference on Robotics and Automation (ICRA 2009), Kobe (JP)*, Seiten 1177–1184.
-

IEEE Omnipress.

- [Mühlig, 2007] Mühlig, M. (2007). Echtzeitfähige Umsetzung der visuellbasierten Bestimmung objektspezifischer Flussfelder. Studienarbeit, Technische Universität Ilmenau. Betreuer: Sven Hellbach; Betreuender Hochschullehrer: Horst-Michael Groß.
- [Müller et al., 2008] Müller, S., Hellbach, S., Schaffernicht, E., Ober, A., Scheidig, A., und Gross, H.-M. (2008). Whom to talk to? Estimating user interest from movement trajectories. In *17th IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN 2008)*, München, Seiten 532–538. IEEE Omnipress.
- [Naftel und Khalid, 2006] Naftel, A. und Khalid, S. (2006). Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. *Multimedia Syst.*, 12(3):227–238.
- [Neundorf, 2002] Neundorf, W. (2002). *Numerische Mathematik. Vorlesungen, Übungen, Algorithmen und Programme*. Shaker Verlag GmbH Aachen.
- [Nunziati et al., 2005] Nunziati, W., Sclaroff, S., und Del Bimbo, A. (2005). An Invariant Representation for Matching Trajectories Across Uncalibrated Video Streams. In *4th Int'l Conf. on Image and Video Retrieval*, volume 3568/2005, Seiten 318–327.
- [Ober, 2007] Ober, A. (2007). Analyse von Bewegungstrajektorien zur nutzerangepassten Dialoginitiierung. Diplomarbeit, Technische Universität Ilmenau. Betreuer: Steffen Müller, Andrea Scheidig und Sven Hellbach; Betreuender Hochschullehrer: Horst-Michael Groß.
- [Ogawara et al., 2007] Ogawara, K., Li, X., und Ikeuchi, K. (2007). Marker-less Human Motion Estimation using Articulated Deformable Model. In *ICRA*, Seiten 46–51.
- [Okada und Hasegawa, 2007] Okada, S. und Hasegawa, O. (2007). Classification of Temporal Data Based on Self-organizing Incremental Neural Network. In *ICANN (2)*, Seiten 465–475.
- [Omlor und Giese, 2006] Omlor, L. und Giese, M. A. (2006). Unsupervised Learning of Spatio-temporal Primitives of Emotional Gait. In *PIT*, Seiten 188–192.
-

- [Oram und Perrett, 1996] Oram, M. und Perrett, D. (1996). Integration of form and motion in the anterior superior temporal polysensory area (STPa) of the macaque monkey. *Journal on Neurophysiology*, 76:109–129.
- [Otero et al., 2006] Otero, N., Knoop, S., Nehaniv, C., Syrdal, D., Dautenhahn, K., und Dillmann, R. (2006). Distribution and Recognition of Gestures in Human-Robot Interaction. *The 15th IEEE International Symposium on Robot and Human Interactive Communication, 2006. ROMAN 2006.*, Seiten 103–110.
- [Oyama und Uno, 2006] Oyama, T. und Uno, Y. (2006). Estimation of a human planned trajectory from a measured trajectory. *Syst. Comput. Japan*, 37(9):1–11.
- [Parlitz, 2006] Parlitz, H. D. I. A. U. (2006). Nonlinear Analysis of Time Series Data. In *Handbook of Time Series Analysis*, number Kapitel 2, Seiten 5–37.
- [Perng et al., 2000] Perng, C.-s., Wand, H., Zhang, S. R., und Parker, D. S. (2000). Landmarks: A New Model for Similarity-Based Pattern Querying in Time Series Databases. In *ICDE '00: Proceedings of the 16th International Conference on Data Engineering*, Seite 33, Washington, DC, USA. IEEE Computer Society.
- [Pollick, 2004] Pollick, F. E. (2004). The features people use to recognize human movement style. *Lecture Notes in Artificial Intelligence*, 2915:10–19.
- [Pollick et al., 2003] Pollick, F. E., Hale, J. G., und McAleer, P. (2003). Visual Perception of Humanoid Movement. In *Third International Workshop on Epigenetic Robotics Boston, MA, USA*.
- [Ptito et al., 2003] Ptito, M., Faubert, J., Gjedde, A., und Kupers, R. (2003). Separate pathways for contour and biological-motion cues in motion-defined animal shapes. *NeuroImage*, 19:246–252.
- [Puce et al., 1998] Puce, A., Allison, T., Bentin, S., Gore, J., und McCarthy, G. (1998). Temporal cortex activation in humans viewing eye and mouth movements. *Journal of Neuroscience*, 18:2188–2199.
- [Rajapakse und Wyse, 2003] Rajapakse, M. und Wyse, L. (2003). NMF vs ICA for face recognition. In *ISPA 2003*, volume 2, Seiten 605–610.
-

- [Rao et al., 2002] Rao, C., Yilmaz, A., und Shah, M. (2002). View-Invariant Representation and Recognition of Actions. *International Journal of Computer Vision*, Volume 50, Number 2:Seiten 203–226.
- [Rechenberg, 1973] Rechenberg, I. (1973). *Evolutionsstrategie*. Friedrich Fromman Stuttgart.
- [Rhodes et al., 2007] Rhodes, B. J., Bomberger, N. A., Zandipour, M., Waxman, A. M., und Seibert, M. (2007). Cognitively-Inspired Motion Pattern Learning & Analysis Algorithms for Higher-Level Fusion and Automated Scene Understanding. In *Proc. IEEE Military Communications Conference MILCOM 2007*, Seiten 1–6.
- [Rosenstein und Cohen, 1997] Rosenstein, M. T. und Cohen, P. R. (1997). Concepts From Time Series. In *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, Seiten 739–745. American Association for Artificial Intelligence.
- [Sakoe und Chiba, 1990] Sakoe, H. und Chiba, S. (1990). Dynamic programming algorithm optimization for spoken word recognition. Seiten 159–165.
- [Santi et al., 2003] Santi, A., Servos, P., Vatikiotis-Bateson, E., Kuratate, T., und Munhall (2003). Perceiving biological motion: dissociating visible speech from walking. *Journal of Cognitive Neuroscience*, 15:800–809.
- [Schaal et al., 2003] Schaal, S., Peters, J., Nakanishi, J., und Ijspeert, A. (2003). Control, Planning, Learning, and Imitation with Dynamic Movement Primitives. In *Workshop on Bilateral Paradigms on Humans and Humanoids, IEEE International Conference on Intelligent Robots and Systems*, Seiten 1–21.
- [Scheidig et al., 2006] Scheidig, A., Müller, S., Martin, C., und Gross, H.-M. (2006). Generating person's movement trajectories on a mobile robot. In *Proc. of RO-MAN*, Seiten 747 – 752.
- [Schenk, 2009] Schenk, K. (2009). Kompensation von fehlerhaften Bewegungsprädiktionen nach Erweiterung der Wissensbasis. Bachelorarbeit, Technische Universität Ilmenau. Betreuer: Sven Hellbach; Betreuender Hochschullehrer: Horst-Michael Groß.
-

- [Schröter et al., 2007] Schröter, C., Höchemer, M., und Gross, H.-M. (2007). A Particle Filter for the Dynamic Window Approach to Mobile Robot Control. In *Proc. 52nd Int. Scientific Colloquium Ilmenau, Germany*, volume I, Seiten 425–430. Universitätsverlag.
- [Scott, 1992] Scott, D. W. (1992). *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons.
- [Shepard, 1987] Shepard, R. (1987). Toward a Universal Law of Generalization for Psychological Science. *Science*, 237:1317–1323.
- [Shimosaka et al., 2007] Shimosaka, M., Mori, T., und Sato, T. (2007). Robust Action Recognition and Segmentation with Multi-Task Conditional Random Fields. In *ICRA*, Seiten 3780–3786.
- [Shin und Singh, 1990] Shin, D. H. und Singh, S. (1990). Path Generation for Robot Vehicles Using Composite Clothoid Segments. Technical Report CMU-RI-TR-90-31, Robotics Institute, Pittsburgh, PA.
- [Shum et al., 2002] Shum, W.-H., Jin, H.-D., Leung, K.-S., und Wong, M.-L. (2002). A Self-Organizing Map with Expanding Force for Data Clustering and Visualization. In *Proceedings 2002 IEEE International Conference on Data Mining*, Seiten 434–441. IEEE Computer Society. ICDM2002.
- [Shumway und Stoffer, 2000] Shumway, R. H. und Stoffer, D. S. (2000). *Time Series Analysis and Its Applications*. Springer Texts in Statistics.
- [Small, 2005] Small, M. (2005). *Applied Nonlinear Time Series Analysis*, volume 52 of *World Scientific Series on Nonlinear Science A*. World Scientific.
- [Snoek und Worring, 2005] Snoek, C. G. und Worring, M. (2005). Multimodal Video Indexing: A Review of the State-of-the-art. *Multimedia Tools and Applications*, 25(1):5–35.
- [Steil, 2007] Steil, J. J. (2007). Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning. *Neural Networks*, 20:353 – 364.
- [Stein, 2003] Stein, K. (2003). *Qualitative Repraesentation und Generalisierung von Bewegungsverlaeufen*. PhD thesis, Institut für Infor-
-

matik der Technischen Universität München.

- [Stein et al., 2009] Stein, T., Fischer, A., Boesnach, I., Gehrig, D., Köhler, H., und Schwameder, H. (2009). Kinematische Analyse menschlicher Alltagsbewegungen für die Mensch-Maschine-Interaktion. In *Sporttechnologie zwischen Theorie und Praxis*. J. Edelmann-Nusser and K. Witte and E.F. Moritz.
- [Strauß, 2008] Strauß, S. (2008). Zeitreihenanalyse zur Online-Prädiktion von Bewegungsabläufen. Diplomarbeit, Technische Universität Ilmenau. Betreuer: Sven Hellbach; Betreuender Hochschullehrer: Horst-Michael Groß.
- [Suzuki et al., 2007] Suzuki, N., Hirasawa, K., Tanaka, K., Kobayashi, Y., Sato, Y., und Fujino, Y. (2007). Learning motion patterns and anomaly detection by Human trajectory analysis. In *IEEE International Conference on Systems, Man and Cybernetics, 2007. ISIC.*, Seiten 498–503. IEEE.
- [Takano und Nakamura, 2006] Takano, W. und Nakamura, Y. (2006). Humanoid Robot’s Autonomous Acquisition of Proto-Symbols Through Motion Segmentation. In *Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots*, Seiten 425–431, Genoa, Italy.
- [Takeda et al., 2007] Takeda, T., Hirata, Y., und Kosuge, K. (2007). HMM-based Error Recovery of Dance Step Selection for Dance Partner Robot. In *ICRA*, Seiten 1768–1773.
- [Tani und Nolfi, 1999] Tani, J. und Nolfi, S. (1999). Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. volume 12, Seiten 1131–1141.
- [Thompson et al., 2005] Thompson, J., Clarke, M., Stewart, T., und Puce, A. (2005). Configural processing of biological motion in human superior temporal sulcus. *Journal of Neuroscience*, 25:9059–9066.
- [Topp und Christensen, 2005] Topp, E. A. und Christensen, H. I. (2005). Tracking for following and passing persons. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, Seiten 2321–2327.
-

- [Torres et al., 1996] Torres, L., Kunt, M., und Pereira, F. (1996). Second Generation Video Coding Schemes and Their Role in MPEG-4. In *European Conf. on Multimedia Applications, Services, and Techniques*, Seiten 799–824.
- [Troje, 2002] Troje, N. F. (2002). Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *J. Vis.*, 2(5):371–387.
- [Vaina et al., 2001] Vaina, L., Solomon, J., Chowdhury, S., Sinha, P., und Belliveau, J. (2001). Functional neuroanatomy of biological motion perception in humans. In *Proceedings of National Academic Science USA*, volume 98, Seiten 11656–11661.
- [Vallortigara et al., 2005] Vallortigara, G., Regolin, L., und Maronato, F. (2005). Visually Inexperienced Chicks Exhibit Spontaneous Preference for Biological Motion Patterns. *PLoS ONE*, 3(7):1312–1316.
- [Vlachos et al., 2002] Vlachos, M., Gunopoulos, D., und Kollios, G. (2002). Discovering Similar Multidimensional Trajectories. In *IC-DE '02: Proceedings of the 18th International Conference on Data Engineering*, Seite 673, Washington, DC, USA. IEEE Computer Society.
- [Weser et al., 2006] Weser, M., Westhoff, D., Westhoff, D., und Huser, M. (2006). Multimodal People Tracking and Trajectory Prediction based on Learned Generalized Motion Patterns. In *Proc. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seiten 541–546.
- [Wiener, 1949] Wiener, N. (1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Wiley.
- [Williams et al., 2007] Williams, B., Toussaint, M., und Storkey, A. (2007). A Primitive Based Generative Model to Infer Timing Information in Unpartitioned Handwriting Data. In *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Seiten 1119–1124.
- [Williams et al., 2006] Williams, B. H., Toussaint, M., und Storkey, A. J. (2006). Extracting Motion Primitives from Natural Handwriting Data. In *ICANN (2)*, Seiten 634–643.
-

- [Yu und Kim, 2006] Yu, B. und Kim, S. H. (2006). Interpolating and Using Most Likely Trajectories in Moving-Objects Databases. In *DEXA*, Seiten 718–727.
- [Zajdel et al., 2007] Zajdel, W., Krijnders, J. D., Andringa, T., und Gavrilu, D. M. (2007). CASSANDRA: audio-video sensor fusion for aggression detection. In *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance AVSS 2007*, Seiten 200–205.
- [Zajdel und Kröse, 2005] Zajdel, W. und Kröse, B. J. A. (2005). A sequential Bayesian algorithm for surveillance with nonoverlapping cameras. *IJPRAI*, 19(8):977–996.
- [Zell, 1994] Zell, A. (1994). *Simulation Neuronaler Netze*. Oldenbourg.
-

