Forschungsberichte aus der
Industriellen Informationstechnik | 20

Kristine Back

Erkennung menschlicher Aktivitäten durch Erfassung und Analyse von Bewegungstrajektorien

KIT Scientific Publishing

Kristine Back

Erkennung menschlicher Aktivitäten durch Erfassung und Analyse von Bewegungstrajektorien

Forschungsberichte aus der Industriellen Informationstechnik
Band 20

Institut für Industrielle Informationstechnik
Karlsruher Institut für Technologie
Hrsg. Prof. Dr.-Ing. Fernando Puente León

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

Erkennung menschlicher Aktivitäten durch Erfassung und Analyse von Bewegungstrajektorien

von
Kristine Back

KIT Scientific Publishing

Karlsruher Institut für Technologie
Institut für Industrielle Informationstechnik

Erkennung menschlicher Aktivitäten durch Erfassung
und Analyse von Bewegungstrajektorien

Zur Erlangung des akademischen Grades einer Doktor-Ingenieurin
von der KIT-Fakultät für Elektrotechnik und Informationstechnik des
Karlsruher Instituts für Technologie (KIT) genehmigte Dissertation

von Dipl.-Ing. Kristine Back, geb. in Heidelberg

Tag der mündlichen Prüfung: 27. November 2018
Hauptreferent: Prof. Dr.-Ing. F. Puente León, KIT
Korreferent: Prof. Dr.-Ing. G. Rigoll, TUM

Impressum

Scientific Publishing

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

www.ksp.kit.edu

Print on Demand 2019 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 2190-6629
ISBN 978-3-7315-0909-7
DOI 10.5445/KSP/1000091818

Vorwort

Diese Dissertation entstand während meiner Zeit als wissenschaftliche Mitarbeiterin am Institut für Industrielle Informationstechnik (IIIT) des Karlsruher Instituts für Technologie (KIT). Ich möchte ganz herzlich allen danken, die zu ihrem Gelingen beigetragen haben.

An erster Stelle danke ich meinem Doktorvater Prof. Dr.-Ing. Fernando Puente León für seine Betreuung und für die Möglichkeit, diese Arbeit am IIIT anfertigen zu können. Mein besonderer Dank gilt auch Prof. Dr.-Ing. Gerhard Rigoll von der TU München für die Übernahme des Korreferats.

Weiterhin bedanke ich mich bei meinen Studenten, die durch ihren Einsatz wertvolle Beiträge geleistet haben. Meinen Kollegen am Institut danke ich für die gute Zusammenarbeit und die vielen schönen Stunden während und außerhalb der Arbeitszeit. Meinem Bruder Michael danke ich für die vielen guten Ratschläge und zu guter Letzt gilt mein ganz großer Dank meinen Eltern für ihre Hilfe und Unterstützung, ohne die diese Arbeit nicht möglich gewesen wäre.

Stuttgart, im Juni 2019 Kristine Back

Inhaltsverzeichnis

Abkürzungen und Symbole

Allgemeine Abkürzungen

Akronym	Bedeutung
APF	*Annealed* Partikel-Filter
BoF	*„Bag of Features“*
BoW	*„Bag of Words“*
CRF	*Conditional Random Field*
EA	Evolutionärer Algorithmus
EVP	Evolutionäres Posen-Tracking
ES	Evolutionsstrategie
GA	Genetischer Algorithmus
GFTT	*Good Features to Track*
GGM	gerichtetes graphisches Modell
GM	Graphisches Modell
HIK	*Histogram Intersection*-Kern
HMM	*Hidden Markov*-Modell
HOF	Histogramme von optischem Fluss
HOG	Histogramme orientierter Gradienten
ISA	*Interacting Simulated Annealing*
KLT	Kanade-Lucas-Tomasi-Tracker
LBP	*Local Binary Patterns*
LOO	*Leave One Out*
MBH	*Motion Boundary*-Histogramme
MBS	*Motion Boundary Sum*
MRF	*Markov Random Field*
MS	*Mean Shift*
OF	Optischer Fluss
PD	Periodischer Detektor

Akronym	Bedeutung
SG	*Sum of Gradients*
SIFT	*Scale Invariant Feature Transform*
SOF	*Sum of Optical Flow*
STIP	*Spatio-Temporal Interest Points*
SURF	*Speeded-Up Robust Features*
SVM	*Support Vector Machine*
UGM	ungerichtetes graphisches Modell

Mathematische Symbole

Notation	Bedeutung
*	Prädiktion
$\mathbf{1}_{\{a=b\}}$	Indikatorfunktion $\mathbf{1}_{\{a=b\}} = \begin{cases} 1 & \text{falls } a = b \\ 0 & \text{falls } a \neq b \end{cases}$
a	Skalar
$\mathbf{a}$	Vektor
$\mathbf{A}$	Matrix
$\mathcal{A}$	Menge
acc	Klassifikationsgenauigkeit
c	Index Kamera, Index Bildzelle
C	Anzahl Kameras, Anzahl Bildzellen
cb, $\mathcal{CB}$	(Index) *Codebook*
ch, ch	(Index) Merkmalskanal
d	Index Zustandskomponente, Distanz
D	Anzahl Zustandskomponenten
$\mathbf{d}$	Trajektorien-Deskriptor
$\tilde{\mathbf{d}}$	Deskriptor-Prototyp
$\mathcal{D}$	Datenmenge
$e(\cdot)$	Energiefunktion, Abweichung
$f(\cdot)$	Gewichtungs- bzw. Fitnessfunktion, Merkmalsfunktion bei GMs
$\mathbf{f} = \left[f^x, f^y\right]$	Optischer Fluss
g	Gaußfunktion
$G_{i,j}$	Bildraster
h	Filterfunktion, Bandbreite bei *Mean Shift*-Verfahren

Notation	Bedeutung
$\mathbf{h}$	Histogramm
i, $\mathbf{i}$	(Grauwert-) Bild
IS	Indexselektion
K	Kern
k	Kernprofil, gefiltertes Bild
L	Länge einer Trajektorie in Zeitschritten
LBP	*Local Binary Pattern* (LBP)-Deskriptor
m	Generation
$\mathbf{m}$	Mittelpunkt einer Trajektorie, *Mean Shift*-Vektor
M	Anzahl Generationen
mut, Mut	Index bzw. Operator Mutation
n_σ	Anzahl örtliche Zellen bei Trajektorien-Deskriptoren
$\mathcal{N}(\cdot)$	Dichte einer normalverteilten Zufallsvariable
N	Anzahl
o, $\mathbf{o}$	Diskretes Beobachtungssymbol
O	Anzahl diskreter Beobachtungssymbole
p	Dichtefunktion
p, $\mathbf{p}$	Kandidatenmodell bei *Mean Shift*-Tracking
p	Index Eltern
P	Anzahl Stützstellen zur Bestimmung von LBP-Deskriptoren
pos	Index Position
q, $\mathbf{Q}$	Rauschkovarianz(-matrix)
q, $\mathbf{q}$	Zielmodell bei *Mean Shift*-Tracking
r	Merkmalsdetektor, Bewegungsbild
R	Radius zur Bestimmung von LBP-Deskriptoren
rec , Rec	Index bzw. Operator Rekombination
s	Diskretes Zustandssymbol, Klasse
S	Anzahl diskreter Zustandssymbole bzw. Klassen
S	Suchfenster
sel , Sel	Index bzw. Operator Selektion
t	Diskreter Zeitpunkt
T	Anzahl Zeitpunkte
train	Index Training
T	transponiert
$\mathcal{U}(\cdot)$	Dichte einer gleichverteilten Zufallsvariable
v, $\mathbf{v}$	Prädiktionsrauschen, Verschiebung

Notation	Bedeutung
w	Wort-Index
w, $\mathbf{w}$	Mutationsrauschen
W	Anzahl an Worten
$\mathcal{W}$	Suchfenster bei Tracking mit optischem Fluss
$\mathbf{x} = [x, y]$	Koordinaten eines Bildpunktes
x, $\mathbf{x}$	Zustand
X	Individuum
$\mathcal{X}$	Population, Menge an Partikeln
${}^{\gamma}\mathbf{x}$	Genotyp
${}^{\phi}\mathbf{x}$	Phänotyp
y, $\mathbf{y}$	Beobachtung, Aktionsdeskriptor
Z	Merkmalstrajektorie
$\mathcal{Z}$	Menge an Trajektorien
α^{arm}, α^{leg}	Seitlicher Öffnungswinkel der Arme bzw. Beine
α^{lb}, α^{ub}	Seitliche Neigung des Unter- bzw. Oberkörpers
β	Inverse Temperatur
β^{arm}, β^{leg}	Öffnungswinkel der Arme bzw. Beine nach vorne und hinten
β^{lb}, β^{ub}	Neigung des Unter- bzw. Oberkörpers nach vorne und hinten
γ^{arm}, γ^{leg}	Beugungswinkel der Ellbogen bzw. Knie
γ^{lb}, γ^{ub}	Rotation des Unter- bzw. Oberkörpers
δ_{ij}	Kronecker-Delta $\delta_{ij} = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j \end{cases}$
δ^{arm}, δ^{leg}	Seitliche Bewegung der Hände bzw. Füße
Δ_{σ}	Größe Bildbereich zur Berechnung von Trajektorien-Deskriptoren
ϵ^{arm}, ϵ^{leg}	Bewegung der Hände bzw. Füße nach oben und unten
Γ	Raum Genotyp
λ	Strategieparameter, Gewicht bei graphischen Modellen
Λ	Raum Strategieparameter
π	Fitness, Gewicht
$\boldsymbol{\pi}$	Index gewichtet
ρ	Bhattacharyya-Koeffizient
σ	Örtlicher Skalenfaktor

Notation	Bedeutung
τ	Zeitlicher Skalenfaktor
Φ	Raum Phänotyp
ψ	Potential

1. Einleitung

1.1. Motivation und Überblick

Für die Entwicklung intelligenter technischer Systeme, die selbstständig in menschlicher Umgebung agieren sollen, ist die Fähigkeit der Wahrnehmung des Verhaltens von Menschen von entscheidender Bedeutung. Die Anwendungsgebiete für solche Systeme sind vielfältig: Zum einen ist die Mensch-Maschine-Interaktion zu nennen. Dazu zählen Roboter, die mit Menschen interagieren und kommunizieren, sowie intelligente Umgebungen, die sich beispielsweise durch Gesten steuern lassen. Andere Einsatzmöglichkeiten finden sich in der Sicherheitstechnik und der verhaltensbasierten Biometrie, wozu die Identifikation von Personen durch die Analyse ihres Verhaltens, z. B. ihres Ganges, zählt [96]. Auch können die Animationstechnik, die Sportwissenschaft und die Medizintechnik von der Modellierung und Analyse menschlicher Bewegungen profitieren. Schließlich sei die automatische Annotation von Videodaten genannt, welche eine inhaltsbasierte Durchsuchung von Video-Datenbanken ermöglicht.

Der allgemeine Ablauf der Aktivitätsanalyse lässt sich folgendermaßen zusammenfassen: Am Anfang steht die *Datenerfassung* durch Einsatz von Sensoren. Der nächste Arbeitsschritt ist die *Merkmalsextraktion*. Diese zielt darauf ab, relevante Informationen aus den vorhandenen Sensordaten zu gewinnen und in einer bestimmten Form zu repräsentieren. Diese Darstellung dient anschließend der *Modellierung* und *Analyse* der beobachteten Bewegungen.

Diese Arbeit befasst sich ausschließlich mit der Aktivitätserkennung auf Basis von Videoaufnahmen. Die Interpretation von Bewegungen durch maschinelles Sehen funktioniert bereits gut in einfachen, kontrollierten Umgebungen. Komplexe Aktivitäten in unkontrollierten Szenarien bleiben jedoch ein ungelöstes Problem. Je nach Anwendungsfall

stehen unterschiedliche Aspekte im Vordergrund, von der Klassifikation von Aktivitäten, Szenen, Körpersprache und Mimik bis hin zur detaillierten Analyse und Rekonstruktion der Bewegungen einzelner Körperbereiche. Diese verschiedenen Aufgaben sind mit spezifischen Anforderungen verbunden, und so gibt es eine Vielzahl von Ansätzen, sie zu meistern. Die existierenden Herangehensweisen lassen sich in zwei große Klassen einteilen. Die erste basiert auf der Extraktion von Informationen über die genaue Pose von Menschen, meist mit Hilfe eines Modells des menschlichen Körpers. Die zweite Klasse ist die modellfreie Verarbeitung, welche Merkmale über den Bild- und Bewegungsinhalt von Videos zur Interpretation heranzieht.

Das *Körper-Tracking* befasst sich damit, die genaue Pose einer Person zu ermitteln und über der Zeit zu verfolgen. Das Ziel besteht darin, eine detaillierte Beschreibung menschlicher Bewegungsabläufe zu erhalten. Bei der Ermittlung der Pose eines einzelnen Zeitpunktes spricht man von Posenschätzung, während die Verfolgung über mehrere Zeitschritte als Posen-Tracking bezeichnet wird. Kommerzielle Systeme – sog. *Motion Capturing*-Systeme – verwenden häufig spezielle Marker, die am menschlichen Körper angebracht werden, oder benötigen sehr viele Kameras. Technische Systeme, die sich nahtlos in ein menschliches Umfeld eingliedern sollen, müssen sich Methoden bedienen, die ohne Sensoren oder Marker auskommen, welche direkt am Menschen angebracht werden. Hierzu sind Verfahren von Interesse, die zum Posen-Tracking alleine auf Basis der Aufnahmen einer oder mehrerer Kameras in der Lage sind. Das *markerlose* Körper-Tracking mit wenigen Kameras und in unkontrollierten Umgebungen wird jedoch bisher nicht gut beherrscht [63], und es besteht noch großer Forschungsbedarf, um diese Technologien handhabbarer zu machen und neue Einsatzmöglichkeiten außerhalb des Labors zu erschließen.

Modellfreie Verfahren der Bewegungsanalyse extrahieren bestimmte Merkmale aus Videodaten, welche verschiedene Informationen repräsentieren. Solche Methoden kommen mit geringerem Aufwand, weniger Vorwissen und weniger Aufnahmeperspektiven als das Körper-Tracking aus und sind somit flexibler in ihren Einsatzmöglichkeiten, vor allem im Hinblick auf die Verarbeitung von beliebigen Aufnahmen in unkontrollierten Szenarien. Bei der Aktivitätserkennung besteht das Ziel

außerdem nicht in der Gewinnung einer genauen Beschreibung des Posenverlaufs, sondern darin, eine Aussage darüber zu treffen, welche Aktivitäten in einer Szene auftreten. Daher wird in diesen Fällen in der Regel auf merkmalsbasierte Verfahren zurückgegriffen [48]. Durch Bild- und Videomerkmale können vielfältige Informationen repräsentiert werden. Hierzu zählen *Bildinformationen*, welche das Erscheinungsbild beteiligter Personen bzw. Objekte codieren. Beispiele solcher Merkmale sind Bildgradienten oder andere Bildmerkmale. *Bewegungsmerkmale* repräsentieren explizit die in einer Sequenz enthaltene Bewegung. Hierzu ist beispielsweise der optische Fluss eine beliebte Informationsquelle. Neben isolierten Merkmalen können auch Informationen über den Zusammenhang zwischen den Merkmalen betrachtet werden. Solche *Strukturinformationen* machen eine Aussage über die örtliche bzw. zeitliche Konfiguration auftretender Merkmale. Schließlich können *Kontextinformationen* hinzugezogen werden, um beispielsweise die Umgebung zu beschreiben, in der sich eine Aktion abspielt, oder um anderes vorhandenes Zusatzwissen einzubringen.

Zur Aktivitätsanalyse müssen zunächst eine geeignete Modellform gewählt und die Modellparameter durch einen Lernprozess ermittelt werden. Dies erfolgt meist überwacht, wobei Trainingsdaten mit bekanntem Bewegungsinhalt zum Lernen des Modells herangezogen werden. Die Erkennung von Aktivitäten kann als Klassifikationsproblem angesehen werden. Die Aufgabe dabei besteht darin, das Verhalten von Menschen in verschiedene Kategorien einzuteilen. Eine der Herausforderungen bei der Modellierung von Aktivitäten ist die große Variabilität in der Ausführung. Bewegungsarten, welche die gleiche semantische Bedeutung besitzen, können sehr unterschiedliche Merkmale hervorbringen. Anforderungen an ein Aktivitätsmodell sind, einerseits über alle möglichen Varianten einer Aktivitätsklasse *verallgemeinern* zu können und andererseits zwischen den verschiedenen Klassen zu *diskriminieren*. Ein weiterer Aspekt ist die Formulierung von geeigneten Kategorien. Komplexe Aktivitäten setzen sich aus einfacheren Aktionen zusammen. Dabei stellt sich die Frage, ob eine direkte Modellierung der gesamten Aktivität oder eine hierarchische Darstellung auf Basis atomarer Aktionen zu bevorzugen ist. Für reale Anwendungen ist außerdem nicht nur eine Klassifikation isolierter Aktivitäten gefragt, sondern auch deren Segmentierung. Diese

hat die Aufgabe, fortlaufende Beobachtungen in einzelne Sequenzen zu unterteilen, welche jeweils eine zusammenhängende Aktion enthalten. In vielen verwandten Arbeiten wird die Thematik der Segmentierung bzw. Detektion von Aktionen vernachlässigt. Auch zielen die meisten etablierten Datensätze auf die Evaluation von Methoden der Aktivitätserkennung ab und beinhalten kurze Videosequenzen mit bereits segmentierten Aktionen.

1.2. Zielsetzung und Struktur dieser Arbeit

Im Rahmen dieser Arbeit wird zunächst die Extraktion von Merkmalen zur Repräsentation von Aktivitäten behandelt. Hierzu wird sowohl das markerlose Körper-Tracking als auch ein merkmalsbasierter Ansatz betrachtet. Der Fokus liegt dabei auf der Extraktion und Repräsentation des *dynamischen Inhaltes* von Aktivitäten. Anschließend wird auf Basis der gewonnenen Merkmale eine Aktivitätserkennung durchgeführt. Explizite Informationen über beteiligte Objekte, den Kontext der Szene oder Interaktionen zwischen Personen werden dabei nicht betrachtet.

Zunächst wird in Kapitel 2 auf den Stand der Technik der für diese Arbeit relevanten Themengebiete eingegangen. In Kapitel 3 wird eine Methode des dreidimensionalen, markerlosen Körper-Trackings vorgestellt. Dabei wird die Pose einer Person auf Basis eines Körpermodells durch Gelenkwinkel repräsentiert. Das Tracking der Pose erfolgt mit einem evolutionären Algorithmus. Als Beobachtungen werden Kameraaufnahmen mehrerer Perspektiven verwendet. Die ausgeführte Bewegung der Person wird somit durch die geschätzten Gelenkwinkel-Verläufe repräsentiert, welche reichhaltige Informationen über die Bewegungsdynamik enthalten. Kapitel 4 behandelt einen merkmalsbasierten Ansatz der Bewegungserfassung. Hierbei besteht die Zielsetzung darin, Merkmale über den dynamischen Bewegungsablauf zu extrahieren, welche auch komplexe Aktivitäten repräsentieren können. Dazu werden lokale Merkmale detektiert, die besonders charakteristisch für die in einer Sequenz vorhandene Bewegung sind, und ein Tracking dieser Merkmale über einen längeren Zeitraum ausgeführt. Die resultierenden Merkmalstrajektorien werden durch ihren Verlauf sowie Informationen über Textur und Bewegung in ihrer lokalen Umgebung dargestellt. In

Kapitel 5 werden die gewonnenen Merkmale zur Modellierung und Erkennung von Aktivitäten eingesetzt. Neben der reinen Klassifikation von Aktionen wird auch der Aspekt der Segmentierung betrachtet. Hierbei wird ein Ansatz verfolgt, die Erkennung und Segmentierung gemeinsam durchzuführen, indem eine Abfolge von Aktionen durch ein sequenzielles Modell repräsentiert wird. Schließlich wird in Kapitel 6 diese Arbeit zusammengefasst und ein Ausblick auf weiterführende Forschungen gegeben.

1.3. Eigener Beitrag

In diesem Abschnitt wird eine Übersicht über die eigenen Beiträge dieser Arbeit zum Stand der Technik gegeben.

Hierzu gehört das Evolutionäre Posen-Tracking (EVP) – eine neue Methode des dreidimensionalen, markerlosen Körper-Tracking. Dieses baut auf verwandten, auf Partikelfiltern basierenden Verfahren auf, verwendet stattdessen jedoch einen evolutionären Algorithmus zur Posenschätzung. Der wesentliche Vorzug dieser Methode ist das Zusammenspiel der gewählten Evolutionsfaktoren um eine effiziente Durchsuchung des Zustandsraumes zu erzielen. In Versuchen wird gezeigt, dass hiermit ein erfolgreiches Tracking mit deutlich geringerem Aufwand als bei verwandten Ansätzen gelingt. Zusätzlich wird ein erweitertes Dynamikmodell vorgeschlagen und am Beispiel von Geh-Bewegungen evaluiert, welches Trackingfehler aufgrund von Mehrdeutigkeiten in Kamerabildern, v. a. im Falle weniger Aufnahmeperspektiven, reduziert.

Für die merkmalsbasierte Aktivitätserfassung wird eine neue Methode zur Gewinnung von Merkmalstrajektorien entwickelt. Diese zeichnet sich durch die Selektion besonders relevanter Merkmale und ein robustes Tracking aus. Dazu wird eine Initialisierungsmethode vorgeschlagen, welche nur dynamische und saliente Merkmale in den Trackingprozess aufnimmt. In verwandten Arbeiten wird zum Merkmalstracking häufig der optische Fluss verwendet. Dieser eignet sich jedoch v. a. zur Gewinnung von Trajektorien kurzer Dauer, da das Merkmalstracking schnell abdriftet. Um dies zu verhindern, werden in dieser Arbeit Texturmerkmale auf Basis von *Local Binary Patterns* zum Tracking hinzugezogen. In Experimenten wird gezeigt, dass das in dieser Arbeit entwickelte

Verfahren Langzeit-Trajektorien von hoher Qualität liefert, die besser in der Lage sind, komplexe Ereignisse in Videos zu modellieren, als Trajektorien, die nur sehr kurze Merkmalsverläufe enthalten.

Zur Repräsentation der gewonnenen Trajektorien werden außerdem durch das SURF (*Speeded-Up Robust Features*) Verfahren inspirierte Deskriptoren vorgestellt, welche mit geringerem Aufwand als die häufig verwendeten Histogramm-basierten Deskriptoren berechnet werden können und vergleichbare Ergebnisse liefern.

Die Aktivitätsmodellierung auf Basis von Merkmalstrajektorien zielt in verwandten Arbeiten auf die Klassifikation bereits segmentierte Aktionsabschnitte ab. Im Gegensatz dazu wird in dieser Arbeit die Aufgabe der Segmentierung mitberücksichtigt. Dazu werden die Merkmalstrajektorien in ein probabilistisches Sequenzmodell eingebunden, welches eine gemeinsame Segmentierung und Klassifikation aufeinanderfolgender Aktionen umsetzt.

2. Stand der Technik

Wie bereits in Kapitel 1 erläutert wurde, lassen sich die Herangehensweisen der Merkmalsextraktion zur Repräsentation menschlicher Bewegungen einteilen in das Posen-Tracking und die merkmalsbasierte Bewegungserfassung. Diese beiden Ansätze werden in den nächsten beiden Abschnitten diskutiert. Anschließend wird auf die Aktivitätserkennung eingegangen. Darauf folgend werden verwandte Methoden im Detail besprochen, die eine Aktionserkennung basierend auf Merkmalstrajektorien durchführen.

2.1. Bildbasiertes markerloses Posen-Tracking

Ausführliche Übersichten über das markerlose Körper-Tracking sind beispielsweise in [63–65, 73, 87] zu finden. Zu den Herausforderungen dabei zählt die Komplexität der menschlichen Anatomie und der möglichen Posen. Andere Schwierigkeiten ergeben sich aus Unterschieden im Aussehen und der Kleidung von Personen, sowie verschiedenen Umgebungsbedingungen. Die Posenschätzung ist schwieriger als das Posen-Tracking, da bei Letzterem bereits eine Initialpose aus dem vorigen Zeitpunkt vorliegt, und lediglich in einer gewissen Umgebung dieses Startwertes nach der aktuellen Pose gesucht werden muss. In der jüngeren Forschung verschmelzen die beiden Themengebiete zunehmend [86]. Besonders schwierig gestaltet sich das Posen-Tracking bei der Verwendung weniger oder sogar nur einer Kamera. Die meisten Mehrdeutigkeiten bei der Zuordnung zu verschiedenen Posen rühren vom Verlust dreidimensionaler Information bei der Bildaufnahme [86], welcher durch die Verwendung mehrerer Kameras teilweise ausgeglichen werden kann. Das robuste Körper-Tracking mit weniger als vier Kameras zählt zu den künftigen Hauptaufgaben der Forschung [87].

Die verschiedenen existierenden Ansätze der Posenschätzung bzw. des Posen-Trackings können in *modellfreie* und *modellbasierte* Ansätze eingeteilt werden, je nachdem ob eine explizite Modellierung des menschlichen Körpers durchgeführt wird oder nicht.

2.1.1. Modellfreie Ansätze

Die modellfreie bzw. direkte Posenerkennung kann durch die Detektion einzelner Körperteile erfolgen. Dabei wird der menschliche Körper als Menge von Körperteilen modelliert, die durch statistische Bedingungen zusammenhängen [86]. Anstelle eines Körper-Trackings erfolgt hier eine Detektion der unterschiedlichen Körperteile. Vorteile sind, dass jedes Bild individuell verarbeitet werden kann, was z. B. bei schnellen Bewegungen nützlich ist, sowie die Fähigkeit, mit Verdeckungen umzugehen [65]. Außerdem wird die Problematik umgangen, eine Inferenz in einem hochdimensionalen Raum durchzuführen, da diese durch mehrfache Inferenz in niederdimensionalen Räumen ersetzt wird [86].

In [88] wird der Körper durch eine lose Ansammlung an Körperteilen in Form eines ungerichteten graphischen Modells repräsentiert. Die Gelenke sind die Knoten des Modells und die Kanten stellen kinematische Zusammenhänge zwischen den Gelenken und weitere Randbedingungen dar. Die Posenschätzung und das Tracking lassen sich damit als Inferenz im Modell formulieren. Es werden sowohl die Posenschätzung in Einzelbildern als auch das Posen-Tracking behandelt, wobei beim Tracking deutlich bessere Ergebnisse erzielt werden. Die Gewichtungsfunktionen, welche zum Vergleich mit den vorliegenden Beobachtungen dienen, setzen sich aus Vordergrundsilhouetten, Kanten und Detektoren einzelner Körperteile zusammen. Bei *Pictorial Structures* [4, 50, 107] werden diskriminative Modelle der zweidimensionalen Erscheinung von Posen gelernt. Sie werden weitläufig bei der 2D-Posenschätzung eingesetzt und können durch gemeinsame Inferenz über mehrere 2D-Projektionen auf die dreidimensionale Posenschätzung erweitert werden [3]. Die in [3] verwendeten Bildmerkmale basieren auf Farbe oder Grauwert-Histogrammen. Die Modellierung erfolgt mit einem *Conditional Random Field* (CRF), welches in Potentiale bzgl. einzelner Körperteile zerlegt wird. In [34] werden nicht nur Bildinformationen genutzt, sondern auch

Merkmale, die spezifische Bewegungsinformationen von Körperteilen bei unterschiedlichen Posen beschreiben.

Laut [87] sind teilbasierte Ansätze bisher zu ungenau und eignen sich v. a. für die Initialisierung generativer Methoden, die ein kinematisches Körpermodell verwenden. Eine andere Herangehensweise ist die direkte Codierung des Erscheinungsbildes verschiedener Posen in den vorhandenen Beobachtungen, z. B. in Form von Silhouetten oder Bildmerkmalen [10]. Das Training erfolgt häufig mit synthetisch erzeugten Daten [65]. Diese Methoden können allerdings schlecht mit nicht im Training beobachteten Eingangsdaten umgehen, und große Trainingsdatensätze führen zu mehr Ambiguitäten bei der Zuordnung [65].

2.1.2. Modellbasierte Ansätze

Bei der modellbasierten Erkennung wird ein Modell des menschlichen Körpers angenommen, welches die kinematischen Eigenschaften und das Erscheinungsbild von Personen explizit repräsentiert. Diese Herangehensweise ist in einigen der frühesten Arbeiten des Themengebietes zu finden und stellt nach wie vor eine der dominanten Forschungsrichtungen dar [65]. Häufig werden bei der modellbasierten Posenschätzung generative Ansätze verfolgt. Die grundsätzliche Vorgehensweise dabei ist „Synthetisieren und Testen“, d. h. es werden Posenhypothesen gebildet, welche dann mit Hilfe der vorliegenden Beobachtungen bewertet werden.

Häufig wird der menschliche Körper als kinematische Kette im dreidimensionalen Raum repräsentiert. Die einzelnen Körperteile können beispielsweise durch Zylinder angenähert werden, welche über die Gelenke miteinander verbunden sind. Die Konfiguration des Modells bzw. der Zustandsvektor gibt die Pose der Person an. In [37] wird gezeigt, dass durch Verwenden eines hochgenauen Gitter-Modells, welches speziell an eine Person angepasst wird, starke Verbesserungen in der Genauigkeit des Körper-Trackings erreicht werden können. Eine geläufige Formulierung des Körper-Trackings ist als Bayes'sches Schätzproblem [27, 83]. Das Ziel dabei ist die Schätzung der A-posteriori-Dichte des Zustandsvektors bei vorhandenen Beobachtungen. Die Bewertung von Zustandshypothesen mit den vorliegenden Beobachtungen erfolgt mit der Gewichtungsfunktion, häufig auch *Likelihood* genannt. Häufig ver-

wendete Gewichtungsfunktionen basieren auf Körpersilhouetten und -kanten [27, 28, 85].

Eine große Herausforderung stellt die hohe Dimensionalität des Zustandsraumes dar. Frühe Algorithmen des Körper-Trackings waren meist deterministischer Natur [65]. Da diese nur eine Zustandshypothese verfolgen, scheitern sie häufig an Mehrdeutigkeiten. Ein robusteres Tracking kann mit stochastischen Methoden wie dem Partikel-Filter erreicht werden. Jedoch ist auch dieses unzureichend für das Posen-Tracking [38, 65]. Dies liegt u. a. daran, dass aufgrund des hochdimensionalen Zustandsraumes eine sehr große Partikelanzahl benötigt wird. Um dieses Problem zu lösen wird beim *Annealed* Partikel-Filter (APF) [27, 28] der Partikel-Filter-Ansatz mit simuliertem *Annealing* – einem globalen Optimierungsverfahren – kombiniert. Eine eng verwandte Methode ist das *Interacting Simulated Annealing*-Partikel-Filter (ISA) [38]. Auch andere Optimierungsverfahren wurden bereits für das Körper-Tracking vorgeschlagen. In [49] wird eine hierarchische Partikelschwarm-Optimierung durchgeführt. [37] kombiniert lokale und globale Optimierungsverfahren in einem mehrstufigen Algorithmus. In der ersten Stufe erfolgt die Posenschätzung mittels ISA, welches ein robustes Tracking und eine automatische Initialisierung der Pose ermöglichst. Nachfolgend erfolgt eine Verfeinerung der Pose durch lokale Optimierung und Filterung.

Viele generative Modelle liefern gute Ergebnisse, wenn viele Kamerablickwinkel vorhanden sind. In Szenarien mit einer oder wenigen Kameras besteht jedoch noch großer Forschungsbedarf [85, 86]. In diesen Fällen ist ein aussagekräftiges A-priori-Modell für die beobachteten Bewegungsmuster sinnvoll bzw. notwendig. Häufig werden diese Modelle aus Trainingsdaten gelernt (z. B. *Motion Capture*-Daten oder manuell annotierte Trainingssequenzen) [1, 84]. Solche Modelle sind effektiv, aber nur begrenzt einsetzbar in allgemeinen Szenarien, da sie nur wenige Bewegungsarten und nur eine geringe Variation innerhalb der Bewegungsarten erlauben [65, 86]. Der Versuch, ein möglichst allgemeingültiges Bewegungsmodell zu lernen, wird beispielsweise in [83] unternommen. Andere Ansätze zur Erzeugung allgemeingültiger Dynamikmodelle streben eine physikalische Modellierung menschlicher Bewegungen und der Interaktion zwischen Personen und der Umgebung an [13, 14]. Bisher reichen diese Methoden aber nicht an gelernte (personenspezifische)

Modelle heran [87]. Anstelle der Verwendung allgemeingültiger Dynamikmodelle wird in [108] die Posenschätzung durch eine gekoppelte Aktionserkennung unterstützt. Zunächst wird eine Aktionserkennung vorgenommen, deren Ergebnis als A-priori-Verteilung der Aktionsklassen in die nachfolgende Posenschätzung einfließt. Dabei wird für jede Aktion eine niederdimensionale Repräsentation des Zustandsraumes gelernt und eine Optimierung mit *Interacting Simulated Annealing* (ISA) über die aktionsspezifischen Zustandsräume durchgeführt.

2.2. Merkmalsbasierte Bewegungserfassung

Die Methoden der *Merkmalsextraktion* werden hier ähnlich wie in [72] in *globale* und *lokale* Bildrepräsentationen eingeteilt. Ausführliche Zusammenfassungen der Thematik sind beispielsweise in [72, 96, 101] zu finden.

2.2.1. Globale Merkmale

Globale Bewegungsmerkmale repräsentieren Informationen über den gesamten Bereich einer Bildfolge, in der sich die Bewegung abspielt. Die Interessensregion des Akteurs kann z. B. durch Personendetektion und -tracking oder Hintergrundsubtraktion detektiert werden. Aus dieser Interessensregion werden *Templates* für verschiedene Aktionen gebildet. Beim *Motion Energy Image (MEI)* [11] und bei *Space-Time-Shapes* [9] werden Körpersilhouetten aufeinanderfolgender Zeitschritte zusammengefügt und daraus Merkmale gebildet. Das *Motion History Image (MHI)* [11] enthält durch unterschiedliche Gewichtung vergangener Silhouetten zusätzlich Informationen über die zeitliche Abfolge. Anstelle von Silhouetten kann auch Bewegungsinformation verwendet werden, welche beispielsweise durch den optischen Fluss in der Interessensregion beschrieben werden kann [32]. In [2] werden unterschiedliche Varianten und Anwendungen dieser Art von Merkmalen diskutiert.

Globale Merkmale beinhalten eine Fülle an Informationen, da sie die komplette Interessensregion beschreiben. Jedoch sind sie anfällig für Störungen, hängen vom Blickwinkel ab und beinhalten Informationen über das Erscheinungsbild der betrachteten Person. Im Falle von Sil-

houetten ist eine gute Hintergrundsubtraktion nötig. Dieses Problem entfällt bei Methoden mit optischem Fluss oder Gradienten. Allerdings sind diese empfindlich gegenüber Beleuchtungsschwankungen. Globale Merkmale liefern gute Ergebnisse in kontrollierten Umgebungen, haben allerdings Schwierigkeiten bei Verdeckungen [72]. Gute Ergebnisse werden berichtet, wenn sie mit Grammatikmodellen [101] oder diskriminativen Dynamikmodellen kombiniert werden [72]. Globale Merkmale werden als besonders geeignet angesehen für Aufnahmen aus großer Entfernung oder sehr geringer Auflösung [101], wie beispielsweise in Überwachungsanwendungen.

2.2.2. Lokale Merkmale

Lokale Bewegungsmerkmale versuchen, nur die relevanten Informationen von Bewegungen zu repräsentieren. Anstatt die komplette Interessensregion zu beschreiben, werden hier nur interessante Bereiche einer Bildsequenz betrachtet und durch bestimmte Merkmale repräsentiert. Diese Interessensregionen werden keinen bestimmten Körperteilen zugeordnet. Die vorliegende Aktion wird basierend auf der Statistik dieser lokalen Merkmale dargestellt.

In jüngerer Zeit erfreuen sich sog. *Space-Time Interest Points (STIPs)* großer Beliebtheit. Die Motivation dabei besteht darin, die Bereiche einer Bildsequenz aufzufinden, welche die wichtigsten Informationen über die vorliegende Bewegung beinhalten und die Bewegung nur basierend auf diesen spärlichen Merkmalen zu interpretieren. STIPs werden an Bildpunkten detektiert, welche sowohl in örtlicher als auch zeitlicher Richtung Variationen aufweisen, da diese häufig interessante Ereignisse repräsentieren. In [55] wird der Harris-Eckendetektor [42] auf den dreidimensionalen Fall erweitert. In [103] wird die Determinante einer Hesse-Matrix in örtlicher und zeitlicher Richtung verwendet, um dichte und skaleninvariante Merkmale zu detektieren. Dollár et al. [29] vertreten die Ansicht, dass direkte dreidimensionale Erweiterungen von vielen 2D-Merkmalsdetektoren ungeeignet für die STIP-Detektion sind, da die zeitliche Dimension gesondert betrachtet werden muss, und schlagen stattdessen einen Detektor vor, der Gauß-Filter in örtlicher und Gabor-Filter in zeitlicher Richtung verwendet und darauf ausgelegt ist, eher zu viele als zu wenige STIPs zu detektieren.

Nach der STIP-Detektion werden Bildvolumen um die ermittelten STIPs mit einer gewissen Ausdehnung in örtlicher und zeitlicher Richtung durch bestimmte Merkmale beschrieben und in einen Merkmalsvektor – den STIP-Deskriptor – überführt. Die Merkmale, die zur Erzeugung der STIP-Deskriptoren verwendet werden, können zum einen zweidimensionale Bildmerkmale sein, welche in dreidimensionalen Volumen einer Bildsequenz ermittelt werden. Beispiele hierfür sind Grauwert-Histogramme oder Histogramme örtlicher Bildgradienten [29, 57]. Eine Vielzahl von Merkmalsdeskriptoren, welche erfolgreich für Problemstellungen wie Objekterkennung oder Texturklassifikation verwendet werden, wurden bereits für die Aktivitätserkennung untersucht. Es gibt allerdings auch viele Ansätze, welche der dreidimensionalen Natur von Bildsequenzen direkt Rechnung tragen. Hierzu zählen Methoden, die Bewegungsinformationen in den STIP-Volumen beschreiben, wie Histogramme von Vektoren optischen Flusses [29, 58]. In [53] werden Histogramme orientierter Gradienten (HOG) [25] auf den dreidimensionalen Fall erweitert. In [80] wird ein dreidimensionaler SIFT-Deskriptor [59] und in [103] ein dreidimensionaler SURF-Deskriptor [7] verwendet. Die in [109] vorgestellten *Local Trinary Patterns* basieren auf dem *Local Binary Pattern*-Deskriptor [69]. In [97] werden verschiedene lokale Orts-Zeit-Merkmale für die Aktionserkennung ausgewertet. Die besten Ergebnisse werden dabei durch Kombination von Merkmalen erzielt, die auf Gradienten und optischem Fluss basieren.

Vorteile lokaler Merkmale sind die einfache Extraktion ohne aufwändige Vorverarbeitungsschritte. Außerdem sind sie robuster gegenüber Verdeckungen als andere Ansätze. Auch bezüglich anderer Umgebungsbedingungen, wie dem Blickwinkel etc., weisen sie eine gewisse Robustheit auf [72]. Eine zuverlässige Merkmalsdetektion ist vonnöten, weshalb diese Verfahren sich besonders für schnelle und periodische Bewegungen eignen [101], welche starke Detektorantworten bewirken. Ein Nachteil bei der Verwendung lokaler Merkmale ist das Fehlen von Informationen über die örtliche und zeitliche Struktur von Aktionen. Um Informationen über den zeitlichen Ablauf von Bewegungen zu erhalten, wird in einigen Ansätzen anstelle der Betrachtung isolierter Merkmale ein Merkmalstracking durchgeführt, welches in Merkmalstrajektorien resultiert. Da diese Herangehensweise auch in dieser Arbeit verwen-

det wird, werden verwandte Methoden in Abschnitt 2.4 ausführlich diskutiert.

2.3. Aktionserkennung

Die Methoden der Aktionserkennung können in zwei Klassen unterteilt werden [72]: die *direkte Klassifikation* und die *sequenzielle Modellierung*. Bei der direkten Klassifikation wird die zeitliche Struktur von Aktionen nicht explizit berücksichtigt. Sequenzielle Modelle dagegen repräsentieren den zeitlichen Ablauf von Aktionen durch Zustandsmodelle.

2.3.1. Direkte Klassifikation

Bei der direkten Klassifikation wird die gesamte Aktion durch einen Aktionsdeskriptor dargestellt und es erfolgt keine zeitliche Modellierung des Ablaufs. Der Aktionsdeskriptor kann sowohl aus globalen als auch lokalen Bewegungsrepräsentationen gebildet werden. Die Aktionserkennung kann z. B. durch *Template-Matching* oder diskriminative Klassifikatoren erfolgen. Bei der Verwendung lokaler Merkmale liegen zunächst einzelne Merkmalsdeskriptoren vor, die lokale Ereignisse in Videos darstellen. Zur Beschreibung einer gesamten Beobachtungssequenz müssen die lokalen Merkmale in einen Videodeskriptor bzw. Aktionsdeskriptor überführt werden. Dieser kann aus einer unstrukturierten Ansammlung der STIP-Deskriptoren bestehen. Dies wird als „*Bag of Features*“ (BoF)- oder „*Bag of Words*“ (BoW)-Ansatz bezeichnet. Hierbei wird keinerlei örtlicher oder zeitlicher Zusammenhang zwischen Merkmalen modelliert. Die zugrundeliegende Idee ist, dass eine Aussage darüber, welche Merkmale auftreten, unabhängig davon wann oder wo sie auftreten, bereits sehr aussagekräftige Informationen für die Aktionserkennung liefert. Die Klassifikation erfolgt meist mit diskriminativen Klassifikatoren, wie z. B. *Support Vector Machines* (SVMs). Vorteile dieses Ansatzes sind eine gewisse Invarianz gegenüber dem Blickwinkel und der Variabilität von Aktionen. Allerdings sind Informationen über die Struktur der Merkmale für die Aktivitätserkennung durchaus wichtig, weshalb immer mehr Verfahren versuchen, diese in geeigneter Weise zu integrieren. Eine einfache Methode dazu ist die Unterteilung der

Einzelbilder in mehrere Bildraster und die Auswertung der Merkmalsvorkommen in den einzelnen Zellen dieser Raster. Die Unterteilung kann in gleichmäßige Raster erfolgen oder abhängig vom Auftreten der Person. Eine andere Herangehensweise ist die Betrachtung von Korrelationen zwischen Merkmalen, wie es z. B. in [92] in Form einer Merkmals-*Co-Occurence*-Matrix erfolgt. Einige Methoden verwenden graphische Modelle zur Darstellung verschiedener Bildbereiche [101].

2.3.2. Sequenzielle Modelle

Zur sequenziellen Modellierung wird anstelle eines gesamten Merkmalsvektors für eine Aktion eine Merkmalssequenz benötigt. Aktionen können durch Zustandsmodelle repräsentiert werden, wobei Zustandsübergänge und Zusammenhänge von Zuständen und Beobachtungen modelliert werden. Hierzu können generative oder diskriminative Modelle verwendet werden.

Zu den generativen Modellen zählen *Hidden Markov Modelle* (HMM), welche häufig für die Aktionserkennung eingesetzt werden [72]. Gewöhnlich wird jede Aktion durch je ein HMM repräsentiert [101]. Die Aktionserkennung erfolgt durch Prüfen, welches Modell mit der größten Wahrscheinlichkeit die vorhandene Beobachtungssequenz hervorgebracht hat. Yamato et al. [106] bilden Beobachtungen durch Vektorquantisierung aus Körpersilhouetten. In [104] werden HMMs zur Erkennung von Handtrajektorien verwendet.

Bei der Verwendung von generativen Modellen müssen vereinfachende statistische Annahmen getroffen werden, z. B., dass der aktuelle Zustand nur von der aktuellen und nicht von vergangenen Beobachtungen abhängt. Dies ist bei menschlichen Bewegungen meist nicht der Fall, woraus sich Nachteile von generativen Modellen ergeben. Durch diskriminative Modelle kann dies umgangen werden. *Conditional Random Fields* (CRF) sind diskriminative Modelle, die zur Aktivitätsmodellierung eingesetzt werden können [72, 101]. Sie sind in der Lage, komplexe Zusammenhänge zwischen Zuständen und Beobachtungen verschiedener Zeitpunkte zu modellieren. In [89] werden CRFs zur Aktivitätserkennung eingesetzt und mit HMMs verglichen. Dort wird gezeigt, dass durch die Verwendung von diskriminativen Modellen und die Modellierung von Zusammenhängen zwischen Zuständen und Beobachtungen

über einen längeren Zeitraum eine deutlich bessere Aktionserkennung als mit HMMs ermöglicht wird.

2.4. Aktionserkennung mittels Merkmalstrajektorien

Die Gewinnung von Merkmalstrajektorien aus Videosequenzen wird seit kurzem immer beliebter. Dabei werden Merkmale nach gewissen Kriterien detektiert und über mehrere Zeitschritte verfolgt. Deskriptoren dieser Trajektorien können aus ihrem Verlauf selbst oder aus Bild- oder Bewegungsinformationen um die Trajektorien gebildet werden. Diese Herangehensweise ist v. a. für komplexe Aktivitäten bei hochauflösenden Aufnahmen geeignet [62].

Häufig wird der Kanade-Lucas-Tomasi (KLT) Merkmals-Tracker eingesetzt [61, 62, 74]. Sun et al. [92] tracken SIFT-Merkmale [59]. Die Wahl ist motiviert durch die Robustheit und Skaleninvarianz von SIFT-Merkmalen. Das Tracking erfolgt durch Ermitteln von Mermalskorrespondenzen (*Matching*). In einer nachfolgenden Arbeit [91] fokussieren sich die Autoren auf die Gewinnung dichter, lang-andauernder Trajektorien. Dazu verfolgen sie zusätzlich zu den SIFT-Punkten weitere Merkmale sowie zufällig ini-tialisierte Punkte mit dem KLT-Tracker. Wang et al. [98, 99] extrahieren *dichte* Trajektorien. Sie motivieren dies damit, dass dichtes Sampeln von Merkmalen bei der Bildklassifikation spärlichen Merkmalen überlegen ist. Punkte werden zu jedem Zeitpunkt in einem dichten Raster gewählt und mit Hilfe von dichtem optischem Fluss verfolgt. Die Autoren berichten von robusteren und kohärenteren Trajektorien als mit dem KLT-Tracker oder dem Tracking von SIFT-Merkmalen. Die Trajektorien werden auf eine Länge von $L = 15$ Zeitschritte begrenzt, um das Abdriften der Punkte zu verhindern. Die gesamte Merkmalsextraktion erfolgt in einem Multiskalenansatz, d. h. es werden Trajektorien aus 8 örtlichen Skalen extrahiert. Yi et al. [110] extrahieren „hervorstechende“ (*saliente*) Trajektorien. Die Extraktion der Trajektorien erfolgt zunächst wie Wangs dichte Trajektorien. Im Gegensatz dazu sollen irrelevante Trajektorien entfernt werden, indem ein Salienzmaß betrachtet wird. Saliente Merkmale stechen entweder aufgrund von Bildmerkmalen

oder Bewegung hervor. Diese beiden Maße werden verknüpft, um spärliche, aussagekräftige Trajektorien zu erhalten. Im Vergleich zu Wang erreichten Yi et al. eine weitaus bessere Wahl von Trajektorien in dem Bereich, in dem sich Aktivitäten abspielen.

Als Deskriptoren der Trajektorien steht zunächst deren Verlauf selbst zur Verfügung [61, 62, 91, 92, 98, 99]. Zusätzlich werden häufig weitere Merkmale in einer lokalen Umgebung der Trajektorien extrahiert. Dazu zählen Textur- und Farbmerkmale [61, 62, 68], SIFT- [92] oder SURF-Deskriptoren [68]. In mehreren Arbeiten werden Histogramme orientierter Gradienten (HOG) und optischen Flusses (HOF) [25] eingesetzt [68, 74]. Die sehr erfolgreichen dichten Trajektorien [98, 99] verwenden neben HOG- und HOF-Merkmalen auch Histogramme von differenziertem optischem Fluss, sog. *Motion Boundary*-Histogramme (MBH), welche in [26] zur Personendetektion vorgestellt wurden. Letztere erweisen sich v. a. in der Gegenwart von Kamerabewegung als überlegen, da sie die relative Bewegung beschreiben und gleichförmige Kamerabewegung unterdrücken können.

Die Modellierung erfolgt in den meisten Fällen nach dem BoW-Prinzip. Zur Klassifikation der daraus resultierenden Deskriptoren eignen sich (Mehrkanal-)*Support Vector Machine*s (SVMs) [61, 74, 98, 99]. Die Fusion verschiedener Merkmale erfolgt in einigen Arbeiten auch mittels *Multiple Kernel Learning* [68, 91, 92].

Neben der isolierten Betrachtung der lokalen Trajektorienmerkmale wird auch teilweise angestrebt, holistische Merkmale bzw. strukturelle Informationen über auftretende Merkmale zu berücksichtigen [62, 68, 92].

2.4.1. Diskussion

Bei komplexen Aktivitäten sind Merkmalstrajektorien den STIPs klar überlegen. Allerdings haben sie bislang zu wenig Aufmerksamkeit erhalten [62, 91]. Herausforderungen liegen zum einen in der verlässlichen und effizienten Trajektoriengewinnung [91] sowie der Formulierung geeigneter Deskriptoren zur Repräsentation der Trajektorien.

Die verschiedenen Herangehensweisen lassen sich einerseits in langandauernde und kurze Trajektorien einteilen. Während von Vertretern ersterer Ansätze argumentiert wird, dass lange Trajektorien aussage-

kräftigere Informationen beinhalten, liegt der Grund für die Extraktion lokaler Merkmalsverläufe in der Schwierigkeit der Trajektorienextraktion. Während es einige frühe Arbeiten gibt, die Bewegungsverläufe über eine längere Zeit verfolgen, sind diese in jüngerer Zeit etwas aus der Mode geraten. Bei Methoden wie dem KLT-Tracker oder dem Tracking mit optischem Fluss driften die Trajektorien nach kurzer Zeit ab und werden fehlerhaft. Daher werden in aktuellen Arbeiten meist kurze Trajektorien mit einer Länge von etwa einer halben Sekunde eingesetzt.

Ein weiterer Aspekt ist die Frage, ob die Verfolgung dichter oder spärlicher Merkmale zu bevorzugen ist. In [91] werden mehrere Merkmalsdetektoren kombiniert, um eine große Anzahl an Merkmalen zu erhalten. Wangs dichte Trajektorien schneiden in einigen anspruchsvollen Datensätzen mit am besten von allen Methoden ab. Es ist allerdings unklar, ob dies durch die Menge oder die Qualität der Trajektorien begründet ist. Außerdem wird im Gegensatz zu anderen Methoden ein Multiskalenanansatz verfolgt. In [48] resultieren bessere Ergebnisse, wenn spärliche Merkmale verwendet werden, welche besonders informativ sind. Auch die Arbeit von Jain et al. [46] legt nahe, dass es von Vorteil ist, irrelevante Informationen, wie etwa über den Hintergrund oder aufgrund von Kamerabewegung, zu entfernen bzw. gesondert zu betrachten. Yi und Lin [110] modifizieren Wangs Methode durch die Eliminierung irrelevanter (nicht-salienter) Trajektorien und erreichen damit bessere Ergebnisse.

Bezüglich der Deskription der Trajektorien hat es sich besonders in fordernden Szenarien als notwendig erwiesen, nicht nur den Verlauf selbst, sondern auch Informationen über die lokale Nachbarschaft der Trajektorien zu verwenden. Merkmale basierend auf optischem Fluss liefern hierbei besonders gute Ergebnisse. Ein vielversprechender Ansatz ist die Kombination komplementärer Informationen.

3. Aktivitätserfassung durch modellbasiertes Körper-Tracking

3.1. Einleitung

Dieses Kapitel befasst sich mit der Bewegungserfassung durch markerloses dreidimensionales Körper-Tracking. In dieser Arbeit wird ein modellbasierter, generativer Ansatz verfolgt. Dabei wird die Pose eines Menschen durch einen Zustandsvektor repräsentiert, welcher die Konfiguration des verwendeten Körpermodells darstellt. Der Verlauf der Pose wird basierend auf Videoaufnahmen mehrerer Perspektiven mit dem vorgestellten Tracking-Algorithmus ermittelt. Die entwickelte Methode soll allgemein für den Einsatz in verschiedenen Szenarien geeignet sein und mit wenig Vorwissen über das Erscheinungsbild der Personen sowie der ausgeführten Bewegungsarten auskommen.

Eine der Herausforderungen beim Posen-Tracking ist die hohe Dimensionalität des Zustandsraumes, da der menschliche Körper eine große Anzahl an Freiheitsgraden besitzt. In diesem Kapitel wird eine Methode vorgestellt, die einen evolutionären Algorithmus zum Körper-Tracking einsetzt. Evolutionäre Algorithmen sind heuristische Optimierungsverfahren, welche sich besonders für Problemstellungen mit Herausforderungen eignen, wie sie beim Körper-Tracking vorliegen. Der entwickelte Algorithmus ähnelt und ist inspiriert durch das *„Interacting Simulated Annealing“*-Partikelfilter (ISA) [38] und setzt ebenfalls das simulierte *Annealing* ein.

Der Informationsverlust bei der Projektion einer dreidimensionalen Pose auf zweidimensionale Bildaufnahmen führt häufig zu Mehrdeutigkeiten bei der Lokalisierung von Körperbereichen. Dies kann teilweise

durch die Verwendung mehrerer Aufnahmeperspektiven ausgeglichen werden. Auch der Einsatz von Dynamikmodellen kann hierfür Abhilfe schaffen. Allerdings weisen menschliche Bewegungen einen hohen Grad an Nichtlinearität auf, weshalb die Formulierung geeigneter Dynamikmodelle zur Unterstützung des Trackings schwierig ist. Hier wird für Geh-Bewegungen ein Dynamikmodell formuliert, welches insbesondere im Falle weniger Aufnahmeperspektiven das Tracking unterstützen soll.

In Abschnitt 3.2 werden zunächst die benötigten Grundlagen gegeben. Anschließend wird in Abschnitt 3.3 das Posen-Tracking mit evolutionärem Algorithmus erläutert. In Abschnitt 3.4 wird auf durchgeführte Versuche und Ergebnisse eingegangen.

3.2. Grundlagen

In diesem Abschnitt werden die benötigten Grundlagen vorgestellt, auf denen das evolutionäre Posen-Tracking aufbaut. Zunächst wird das simulierte *Annealing* erläutert. Anschließend werden Methoden des markerlosen Körper-Trackings besprochen, die interagierende Partikelsysteme verwenden. Zuletzt werden einige Grundlagen evolutionärer Algorithmen gegeben.

3.2.1. Simuliertes Annealing

Das simulierte *Annealing* [52, 76], auch als simulierte Abkühlung bezeichnet, ist ein heuristisches Verfahren zum Lösen kombinatorischer Optimierungsprobleme. Es handelt sich um ein globales Optimierungsverfahren. Exakte Lösungen können für viele reale Probleme nicht in annehmbarer Zeit bestimmt werden. Um solche Optimierungsprobleme dennoch zu lösen, wurden Heuristiken entwickelt. Das simulierte *Annealing* wurde 1983 von Kirkpatrick et al. vorgestellt [52]. Inspiriert wurde das Verfahren durch die Beobachtung, dass es viele Gemeinsamkeiten zwischen kombinatorischen Optimierungsproblemen und der statistischen Mechanik gibt. Die Grundidee besteht in der Nachahmung der kontrollierten Abkühlung von Festkörpern.

Um Grundzustände, d. h. Zustände minimaler Energie, in Festkörpern zu finden, beispielsweise um eine Substanz in eine kristalline Struktur

zu überführen, wird das sog. *Annealing* eingesetzt. Dies ist ein kontrollierter Abkühlprozess, bei dem ein Stoff zunächst geschmolzen und dann durch langsame Absenkung der Temperatur zum Erstarren gebracht wird. Dabei muss dem System bei jeder Temperatur ermöglicht werden, ins Gleichgewicht zu kommen, um Defekte zu vermeiden. Kirkpatrick stellte fest, dass die Aufgabe, viele Teilchen in eine Konfiguration minimaler Energie zu bringen, Ähnlichkeiten aufweist mit der Suche einer Kombination von Parametern einer Optimierungsaufgabe, die eine Zielfunktion minimiert oder maximiert.

Im Folgenden wird der Ablauf des simulierten *Annealings* am Beispiel einer *Minimierungsaufgabe* erläutert. Die Energiefunktion $e(x)$ gibt die Energie eines Zustandes x an. Gesucht ist der Zustand minimaler Energie. Das simulierte *Annealing* läuft iterativ ab. Dabei soll verhindert werden, dass die Optimierung in einem lokalen Minimum endet, ohne eine sehr große Menge an Anfangszuständen auszuprobieren. Dazu sollen zu Beginn des Abkühlens auch Verschlechterungen des Zustandes zugelassen werden, um den Zustandsraum weiträumig durchsuchen zu können. Wenn die ungefähre Position des Optimums gefunden wurde, sollen Verschlechterungen immer unwahrscheinlicher werden, um nur noch eine Feinabstimmung der Lösung durchzuführen.

Die praktische Realisierung des simulierten *Annealings* kann z. B. mittels des *Metropolis-Algorithmus* erfolgen [52]. In jedem Schritt wird die aktuelle Konfiguration zufällig variiert und die Energieänderung Δe betrachtet. Wurde eine bessere Konfiguration gefunden, d. h. $\Delta e < 0$, ersetzt die neue Konfiguration die bisherige. Falls eine Verschlechterung stattgefunden hat, wird diese mit einer bestimmten Wahrscheinlichkeit akzeptiert. Andernfalls wird die alte Konfiguration beibehalten. Diese Wahrscheinlichkeit hängt von der aktuellen Temperatur T ab und gehorcht einer Boltzmann-Verteilung. Die Wahrscheinlichkeit, dass eine Verschlechterung um Δe bei der Temperatur T erlaubt wird, lautet [76]

$$P_{\mathrm{acc},T}(\Delta e) = \mathrm{e}^{-\Delta e / T}, \quad \Delta e > 0\,. \tag{3.1}$$

Die Optimierung beginnt mit einer hohen Temperatur, welche im Laufe der Iterationen gemäß einem vorgegebenen *Annealing*-Schema verringert wird. Je höher die Temperatur, desto wahrscheinlicher werden also Verschlechterungen zugelassen. Bei sinkender Temperatur wird dies

immer unwahrscheinlicher. Der prinzipielle Ablauf ist in Algorithmus 1 dargestellt (siehe auch [76]).

Algorithmus 1 Metropolis-Algorithmus.

Anfangszustand x_0, M Iterationen.
$T(m)$: Temperatur in Iteration m.
von $m = 1$ bis M **wiederhole**
  Erzeuge neuen Zustandsvorschlag z
  Werte Energiedifferenz aus $\Delta e_m = e(z) - e(x_{m-1})$
  falls $\Delta e_m < 0$ **dann** ▷ Verbesserung
    $x_m \leftarrow z$ ▷ akzeptieren
  sonst
    Akzeptiere z mit Wahrscheinlichkeit $P_{\mathrm{acc},T} = \mathrm{e}^{-\Delta e_m / T(m)}$.
Sonst behalte alten Zustand.

Der Verlauf der Temperatur $T(m)$ in den Iterationen $m = 1, \ldots, M$ wird als *Annealing*- oder Abkühlschema bezeichnet. Es können hierbei mehrere Versuche bei einer bestimmten Temperatur erfolgen, um ein „Gleichgewicht“ zu erreichen, bevor die Temperatur weiter abgesenkt wird. Die Starttemperatur kann so gewählt werden, dass zu Beginn ein bestimmter Prozentsatz an Zügen akzeptiert wird, z. B. 95 % [76]. Damit werden die meisten schlechten Züge erlaubt, was in einer aggressiven, zufälligen Durchsuchung des Zustandsraumes resultiert. Anschließend wird T sukzessive verringert, so dass der Zustand allmählich in der Endkonfiguration „einfriert“.

Ein weiterer Aspekt, der bei der praktischen Realisierung des Verfahrens berücksichtigt werden kann, ist die Begrenzung der Streuung bei der Erzeugung neuer Zustandsvorschläge bei sinkender Temperatur [76]. Dies ist sinnvoll, um die Optimierung zu beschleunigen. Bei hohen Temperaturen möchte man möglichst den gesamten Zustandsraum durchsuchen. Hierbei ist demnach eine starke Variation der Zustandskonfigurationen erwünscht. Befindet man sich jedoch in der Phase der Feinabstimmung, soll nur noch in einem lokalen Bereich nach dem besten Zustand gesucht werden.

Das Simulierte *Annealing* ist ein sehr gut geeignetes Verfahren für Optimierungsprobleme mit hochdimensionalen Suchräumen und Energiefunktionen mit vielen lokalen Optima.

3.2.2. Markerloses Körper-Tracking mit interagierenden Partikelsystemen

Das markerlose Körper-Tracking wird häufig als Bayes'sches Schätzproblem formuliert [27, 85]. Das Ziel dabei ist die Schätzung der A-posteriori-Dichte $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ eines Zustands $\mathbf{x}_t$ zum Zeitpunkt t bei gegebenen Beobachtungen der vergangenen und des aktuellen Zeitpunktes $\mathbf{y}_{1:t} = [\mathbf{y}_1, \ldots, \mathbf{y}_t]$. Für die Zustandsübergänge wird ein Markov-Modell erster Ordnung angenommen, d. h. der aktuelle Zustand hängt nur vom vorangehenden Zustand ab:

$$p(\mathbf{x}_t|\mathbf{x}_{1:t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})\,. \tag{3.2}$$

Außerdem wird angenommen, dass die aktuelle Beobachtung $\mathbf{y}_t$ nur vom aktuellen Zustand abhängt:

$$p(\mathbf{y}_t|\mathbf{x}_{1:t-1}) = p(\mathbf{y}_t|\mathbf{x}_t)\,. \tag{3.3}$$

Damit ergibt sich folgende rekursive Formel zur Bestimmung der A-posteriori-Dichte [5, 30, 85]:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) \cdot p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\ \mathrm{d}\mathbf{x}_{t-1}\,. \tag{3.4}$$

Dabei beschreibt $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ die zeitliche Zustandspropagation von $t-1$ nach t. Das Integral in Gleichung (3.4) entspricht der Prädiktion des Zustandes basierend auf der A-posteriori-Dichte des vergangenen Zustandes und des Modells der zeitlichen Zustandsentwicklung [5]

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) \cdot p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\ \mathrm{d}\mathbf{x}_{t-1}\,. \tag{3.5}$$

Basierend auf der Messung $\mathbf{y}_t$ erfolgt die Korrektur des prädizierten Zustandes mit der *Likelihood*-Dichte $p(\mathbf{y}_t|\mathbf{x}_t)$.

Eine große Herausforderung des modellbasierten Körper-Trackings ist die hohe Dimensionalität des Zustandsraumes. Mögliche Abhilfe bieten stark einschränkende Annahmen, beispielsweise bezüglich der ausgeführten Bewegungsarten oder des Blickwinkels. Für ein möglichst allgemeingültiges Körper-Tracking haben sich herkömmliche Methoden der Bayes'schen sequenziellen Schätzung, wie das Kalman- oder Partikel-Filter, als unzureichend erwiesen [38]. Stattdessen haben sich statistische Optimierungsverfahren durchgesetzt. Zwei dieser Methoden, die eng miteinander verwandt sind, werden im Folgenden dargestellt.

Annealed Partikel-Filter (APF)

In [28] wird das *Annealed* Partikel-Filter (APF) vorgestellt. Dieses kombiniert einen Partikel-Filter-Ansatz mit simuliertem *Annealing*. Die Motivation besteht in der Formulierung eines möglichst allgemeingültigen Algorithmus zum Körper-Tracking. Die Autoren wenden sich vom Bayes'schen Tracking ab und verzichten auf die Schätzung der A-posteriori-Dichte $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. Stattdessen wird der Zustand $\mathbf{x}_t$ gesucht, der eine Gewichtungsfunktion $f(\mathbf{y}_t, \mathbf{x})$ maximiert. Die Gewichtungsfunktion muss an weniger Stützstellen ausgewertet werden, da keine Verteilungsdichte mehr approximiert werden muss. Außerdem kann eine intuitive Gewichtung verwendet werden, wodurch auch schwer zu modellierende Informationen berücksichtigt werden können. Zur Lösung der Optimierungsaufgabe wird ein partikelbasierter Ansatz gewählt, da dieser sich bei multimodalen Verteilungen eignet und mehrere Hypothesen am Leben erhält, die sich möglicherweise in der Zukunft als vorteilhaft erweisen. Da die Gewichtungsfunktion beim Körper-Tracking häufig viele lokale Maxima enthält, wird eine stochastische Optimierung mit einem Partikelschwarm vorgeschlagen, die ähnlich dem simulierten *Annealing* abläuft [28]. Der Partikelschwarm $\mathcal{X}$ besteht aus N Partikeln

$$\mathcal{X} = \{(\mathbf{x}_i, \pi_i)\}, \quad i = 1, \ldots, N. \tag{3.6}$$

Dabei repräsentiert $\mathbf{x}_i$ den Zustandsvektor und π_i das Gewicht des i-ten Partikels. In jedem zu verarbeitenden Zeitschritt t werden M *Annealing*-Stufen durchlaufen. Abbildung 3.1 zeigt den Ablauf des APF. In jeder Stufe m werden jeweils die typischen Arbeitsschritte eines Partikel-Filters – Gewichtung, Selektion und Diffusion – ausgeführt.

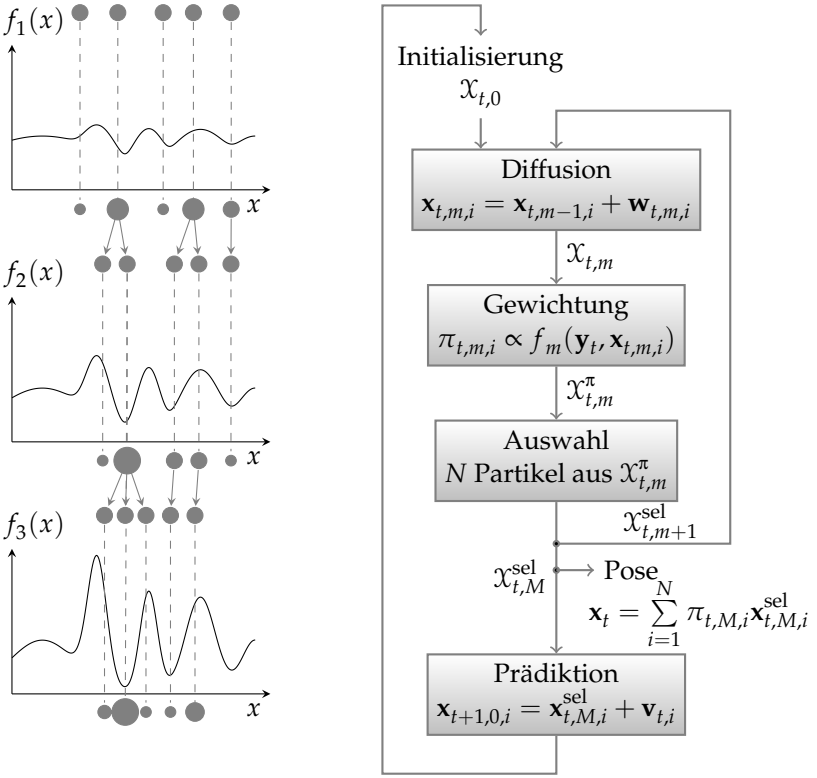


Abbildung 3.1. Veranschaulichung des simulierten *Annealings* für $M = 3$ (links) und Ablauf des *Annealed* Partikel-Filters (rechts).

Im Gewichtungsschritt werden dabei geglättete Versionen

$$f_m(\mathbf{y}_t, \mathbf{x}_t) = f(\mathbf{y}_t, \mathbf{x}_t)^{\beta_m}, \quad m = 1, \ldots, M, \tag{3.7}$$

der Optimierungsfunktion f verwendet. Durch eine geschickte Glättung soll wie beim simulierten *Annealing* verhindert werden, dass der Zustand in einem lokalen Optimum konvergiert. Die Parameter β_m können als inverse Temperaturen oder als Überlebensraten der Partikel aufgefasst werden. Sie werden so gewählt, dass f_1 sehr stark geglättet wird, damit in frühen Iterationen auch schlechtere Zustände erhalten werden und

somit der Zustandsraum durchsucht wird. f_M enthält dagegen stark ausgeprägte Extrema und eignet sich für die lokale Feinabstimmung der Lösung.

Die Partikelstreuung zwischen Zeitschritten t und *Annealing*-Stufen m erfolgt jeweils durch Addition eines weißen Rauschprozesses $\mathbf{v}_t$ bzw. $\mathbf{w}_{t,m}$, $m = 0, \ldots, M$, mit spezifischen Varianzen $\mathbf{Q}_v$ und $\mathbf{Q}_{w,m}$. Für die Prädiktion des Partikelschwarms $\mathcal{X}_{t,M}^{\text{sel}}$ in den nächsten Zeitschritt wird $\mathbf{Q}_v$ so gewählt, dass die Varianzen der einzelnen Zustandskomponenten jeweils der Hälfte ihrer maximal erwarteten Bewegungen entsprechen [28]. Im Laufe des *Annealings* sollte die Stärke der Streuung sukzessiv verringert werden. In einem ersten Ansatz erfolgt dies nach einem festen Schema [28]:

$$\mathbf{Q}_{w,m} = \alpha^m \mathbf{Q}_{w,0}\,, \quad \text{mit } \alpha = 0{,}5\,. \tag{3.8}$$

In [27] wird jedoch ein adaptives Varianzschema vorgeschlagen, welches durch eine hierarchische Suche Verbesserungen mit sich bringt. Das Ziel dabei ist, eine Partitionierung des Suchraumes vorzunehmen, so dass bereits lokalisierte Körperbereiche die weitere Suche einschränken. Dies soll ohne eine fest vorgegebene Hierarchie, sondern durch eine weiche Aufteilung erfolgen. Um dieses Verhalten zu erreichen, werden die Diffusionsvarianzen der einzelnen Zustandskomponenten proportional zu deren Kovarianzen innerhalb des Partikelschwarms gewählt. In Schritt m ergibt sich

$$\mathbf{Q}_{w,t,m} \propto \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}_{t,m,i} - \bar{\mathbf{x}}_{t,m}\right) \cdot \left(\mathbf{x}_{t,m,i} - \bar{\mathbf{x}}_{t,m}\right)^{\mathrm{T}}\,, \tag{3.9}$$

mit

$$\bar{\mathbf{x}}_{t,m} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{t,m,i}\,. \tag{3.10}$$

Dabei ist $\bar{\mathbf{x}}_{t,m}$ der mittlere Zustand in Zeitschritt t und Stufe m. Durch dieses Vorgehen wird verhindert, dass einem bereits mit hohem Vertrauen bestimmten Körperbereich zu viel Rauschen aufaddiert wird, wodurch die Lokalisierung zunichte gemacht würde. In [27] wird damit

eine Effizienzsteigerung um den Faktor 2 erzielt, d. h. ein erfolgreiches Tracking ist mit der Hälfte der Partikel im Vergleich zu einem festen Varianzschema möglich.

Interacting Simulated Annealing (ISA)

Ein sehr eng mit dem APF verwandtes Verfahren ist das *Interacting Simulated Annealing*, welches in [38] vorgestellt wird. Dort wird argumentiert, dass bessere Ergebnisse beim markerlosen Körper-Tracking erzielt werden können, wenn man es als Optimierungsproblem betrachtet statt als Filter- bzw. Schätzproblem. Dies ist unter anderem dadurch begründet, dass die am Körper-Tracking beteiligten stochastischen Prozesse, beispielsweise der Beobachtungsprozess, sehr schwer zu modellieren sind. Dagegen können intuitive Gewichtungsfunktionen, die die Qualität eines Partikels beschreiben, leicht modelliert werden. Daher wird ein Algorithmus vorgestellt, dessen Ziel darin besteht, das Maximum einer gewählten Gewichtungsfunktion zu finden. Auch hier wird ein interagierendes Partikelsystem mit *Annealing*-Eigenschaften verwendet.

Die Abweichung eines Partikels von den Beobachtungen wird durch die Energiefunktion $e(\mathbf{x})$ quantifiziert. Die Sequenz der Gewichtungsfunktionen in den *Annealing*-Iterationen $m = 1, \ldots, M$ ergibt sich aus

$$f_m(\mathbf{x}) = \exp\left(-\beta_m e(\mathbf{x})\right) \tag{3.11}$$

mit dem *Annealing*-Schema $\{\beta_m\}$, $m = 1, \ldots, M$. Der Ablauf des ISA ist in Alg. 2 gegeben.

Je nach Wahl der Parameter $\epsilon_{t,m}$ ergeben sich unterschiedliche Selektionsmechanismen. Für $\epsilon_{t,m} = 0\ \forall m$ ergibt sich gerade das APF. In [38] wird gezeigt, dass mit

$$\epsilon_{t,m} = \frac{1}{\sum_{k=1}^{N} \pi_{t,m,k}} \tag{3.12}$$

eine bessere Approximation der wahren Zustandsverteilung durch den Partikelschwarm erreicht werden kann, was darauf hindeutet, dass diese Selektionsmethode zu bevorzugen ist.

Algorithmus 2 *Interacting Simulated Annealing*-Algorithmus.

Anzahl an Partikeln N, Transitionskernel K
Zeitschritt t: Startpartikel $\mathcal{X}_{t,1}$
von $m = 1$ bis M **wiederhole**
  Selektion
    Gewichtung: $\pi_{t,m,i} \leftarrow f_m(\mathbf{x}_{t,m,i})$ für alle i
    von $i = 1$ bis N **wiederhole**
      Ziehe Zufallszahl κ aus $\mathcal{U}[0,1]$
      falls $\kappa \leq \epsilon_{t,m}\pi_{t,m,i}$ **dann**
        $\mathbf{x}^{\text{sel}}_{t,m,i} \leftarrow \mathbf{x}_{t,m,i}$
      sonst
        $\mathbf{x}^{\text{sel}}_{t,m,i} \leftarrow \mathbf{x}_{t,m,j}$ mit Wahrscheinlichkeit $\frac{\pi_{t,m,j}}{\sum_{k=1}^{N} \pi_{t,m,k}}$
  Mutation
    Ziehe $\mathbf{x}_{t,m+1,i}$ aus $K_m(\mathbf{x}^{\text{sel}}_{t,m,i})$ für alle i
Prädiktion
  Ziehe $\mathbf{x}_{t+1,1,i}$ aus $K_t(\mathbf{x}^{\text{sel}}_{t,M,i})$ für alle i

Der Transitionskernel $K_m(\mathbf{x})$ besteht in einem Gauß-Kern mit der Kovarianzmatrix $\mathbf{Q}_m$. Wie auch beim APF wird ein dynamisches Varianzschema gemäß Gl. (3.9) einer deterministischen Kovarianzentwicklung vorgezogen.

3.2.3. Evolutionäre Algorithmen

Nun werden Grundlagen evolutionärer Algorithmen behandelt, welche für das in Abschnitt 3.3 vorgestellte Verfahren relevant sind. Evolutionäre Algorithmen (EAs) [40, 100] sind durch die natürliche Evolution inspirierte heuristische Optimierungsverfahren. Sie sind universell einsetzbar, und ihre Stärken kommen vor allem dann zum Tragen, wenn andere Verfahren versagen, ein Problem mit akzeptablem Aufwand zu lösen [100], z. B. im Falle hochdimensionaler Zustandsräume und komplexer Optimierungsfunktionen mit vielen lokalen Optima.

Evolutionäre Algorithmen ahmen Vorgänge der natürlichen Evolution in vereinfachter Weise nach. Die gängigen Begrifflichkeiten sind ebenfalls der Biologie entlehnt. Ein Lösungskandidat für das vorliegende Optimierungsproblem wird als *Individuum* bezeichnet. Das Grundprinzip evolutionärer Algorithmen besteht darin, dass eine *Population* aus Individuen nach einer optimalen Lösung sucht, indem sie einem simulierten Evolutionsprozess unterzogen wird. Dieser Prozess wird durch die Evolutionsfaktoren *Variation* und *Selektion* vorangetrieben. Im Gegensatz zur natürlichen Evolution stützen sich evolutionäre Algorithmen auf ein klar definiertes Optimalitätskriterium, welches die Qualität eines Individuums bewertet. Man spricht hierbei von der Bewertungs- oder auch *Fitnessfunktion*. Sie kann als Analogie zum Selektionsdruck aufgefasst werden. Je nach Variante des EA kann sich die Güte eines Individuums auf seine Fortpflanzungs- oder Überlebenschancen auswirken.

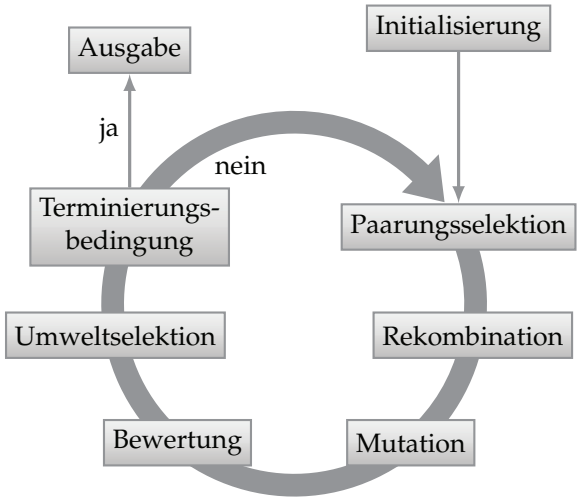


Abbildung 3.2. Schematischer Ablauf eines evolutionären Algorithmus.

Anhand Abbildung 3.2 wird der schematische Ablauf eines evolutionären Algorithmus kurz erläutert [100]. Zunächst wird die Population – meist zufällig – initialisiert. Danach laufen mehrere Evolutionszyklen

ab, welche als *Generationen* bezeichnet werden, in denen nacheinander bestimmte Evolutionsoperatoren auf die aktuelle Generation angewandt werden. Die *Paarungsselektion* wählt Elternindividuen aus, die sich fortpflanzen. Dies erfolgt durch die *Rekombination*. Hierbei werden die Gene mehrerer Eltern neu kombiniert und somit neue Kindsindividuen erzeugt. Durch die Durchmischung des Genmaterials der Eltern können komplett neue Erscheinungsformen der Individuen (sog. Phänotypen) erzeugt werden. In die resultierende Population bringt die *Mutation* weitere Diversität ein. Als nächster Schritt folgt die *Umweltselektion*, bei der die neuen Individuen bewertet werden und die Elternpopulation für die nächste Generation gebildet wird. Hierbei kann eine Kombination der Kinder und der Eltern erfolgen oder die Kinder können die Elterngeneration ersetzen. Als Terminierungsbedingung kann beispielsweise das Erreichen einer Grenze bezüglich der Güte der Population oder das Durchlaufen einer bestimmten Anzahl an Generationen verwendet werden.

Formale Einführung evolutionärer Algorithmen

Die formale Beschreibung von evolutionären Algorithmen orientiert sich an [100]. Ein *Individuum* X wird durch einen Tupel

$$X = ({}^{\gamma}\mathbf{x}, \lambda, \pi) \tag{3.13}$$

repräsentiert. Dabei bezeichnet ${}^{\gamma}\mathbf{x} \in \Gamma$ den *Genotyp* von X. Der Genotyp beschreibt die Repräsentation des Lösungskandidaten, d. h. die Codierung bzw. das genetische Material. $\lambda \in \Lambda$ sind optionale Zusatzinformationen, sog. *Strategieparameter*. Diese können beispielsweise Parametereinstellungen von Operatoren bezüglich dieses Individuums sein. $\pi \in \mathbb{R}$ ist die Güte bzw. *Fitness* von X. Der *Phänotyp* ${}^{\phi}\mathbf{x} \in \Phi$ lässt sich mittels der *Decodierungsfunktion*

$$\begin{aligned} &dec : \Gamma \rightarrow \Phi, \\ &{}^{\phi}\mathbf{x} = dec({}^{\gamma}\mathbf{x}) \end{aligned} \tag{3.14}$$

aus dem Genotyp berechnen. Wird keine Codierung verwendet, ist die Decodierungsfunktion gerade der Identitätsoperator.

Die Güte eines Individuums wird durch die *Fitnessfunktion* bestimmt:

$$f : \Phi \rightarrow \mathbb{R},$$
$$\pi = f(^{\phi}\mathbf{x}) \tag{3.15}$$

Diese ist auf dem Phänotyp definiert. Die Fitnessfunktion mit integrierter Decodierung wird als

$$g : \Gamma \rightarrow \mathbb{R},$$
$$\pi = g(^{\gamma}\mathbf{x}) = f(dec(^{\gamma}\mathbf{x})) \tag{3.16}$$

bezeichnet. Eine Population setzt sich aus mehreren Individuen zusammen:

$$\mathcal{X} = \{X_i\}, \quad i = 1 \ldots N. \tag{3.17}$$

Der Genotyp, die Strategieparameter sowie die Fitness eines Individuums können durch die Evolutionsoperatoren modifiziert werden. Der *Mutationsoperator*, definiert auf einer Population der Größe N, wird bezeichnet durch:

$$\mathsf{Mut} : (\Gamma \times \Lambda)^N \rightarrow (\Gamma \times \Lambda)^N \tag{3.18}$$

Die *Rekombination* wird durch die Abbildung

$$\mathsf{Rec} : (\Gamma \times \Lambda)^N \rightarrow (\Gamma \times \Lambda)^K \tag{3.19}$$

dargestellt. Dabei ist N die Größe der Elternpopulation und K die Größe der Kindergeneration. Der *Selektionsoperator* wird zunächst als Indexselektion dargestellt:

$$\mathsf{IS} : \mathbb{R}^N \rightarrow \{1, \ldots, N\}^K,$$
$$\mathbf{i}^{\mathrm{sel}} = \mathsf{IS}\{\boldsymbol{\pi}\}, \tag{3.20}$$

mit $\boldsymbol{\pi} = \{\pi_i\}$, $i = 1, \ldots, N$, und $\mathbf{i}^{\mathrm{sel}} = \{i_k^{\mathrm{sel}}\}$, $k = 1, \ldots, K$. Dabei werden K Individuen aus einer Population der Größe N ausgewählt. Damit ergibt sich die Selektion durch folgende Abbildung:

$$\mathsf{Sel} : (\Gamma \times \Lambda \times \mathbb{R})^N \rightarrow (\Gamma \times \Lambda \times \mathbb{R})^K,$$
$$\mathcal{X} = \{X_i\}, i = 1, \ldots, N \mapsto \mathcal{X}^{\mathrm{sel}} = \left\{X_{i_k^{\mathrm{sel}}}\right\}, k = 1, \ldots, K. \tag{3.21}$$

Evolutionsfaktoren

Je nach Kombination verschiedener Evolutionsoperatoren kann eine Vielzahl von Algorithmen mit sehr unterschiedlichen Verhaltensweisen entworfen werden. Häufig wird entweder die Rekombination oder die Mutation als primärer Operator für die Variation verwendet.

Die Rekombination kann durch *kombinierende* Operatoren umgesetzt werden. Die kombinierende Rekombination setzt das Genmaterial verschiedener Individuen neu zusammen. Dadurch werden keine neuen Gene erzeugt, jedoch können aus einer neuen Anordnung der Gene vielfältige und neuartige Individuen (Phänotypen) hervorgebracht werden. Diese Herangehensweise ist für die Erhaltung der Vielfalt einer Population und damit die Erforschung des Suchraumes geeignet. Beispiele für Rekombinationsoperatoren sind der uniforme *Crossover* oder der 1-Punkt-*Crossover* [100]. Die *interpolierende* Rekombination hingegen erzeugt Kinder, deren genetische Eigenschaften zwischen denen der Eltern liegen. Ein Beispiel hierfür ist der arithmetische *Crossover* [100]. Hiermit kann eine Feinabstimmung der Lösung erreicht werden. Des Weiteren können auch extrapolierende Operatoren herangezogen werden, welche eine Prognose über interessante Regionen des Zustandsraumes anstellen.

Die Mutation kann sowohl für die Erforschung als auch die Feinabstimmung eingesetzt werden. Als Beispiel wird hier die Gauß-Mutation vorgestellt, welche für reellwertige Genotypen ${}^{\gamma}\mathbf{x} \in \mathbb{R}^D$ geeignet ist.

Algorithmus 3 Gauß-Mutation.

Eingabe: Individuum X, Schrittweite σ
von $d = 1$ bis D **wiederhole**
Wähle w^d zufällig gemäß $\mathcal{N}(0, \sigma)$
${}^{\gamma}x^{\mathrm{mut},d} \leftarrow {}^{\gamma}x^{d} + w^{d}$
${}^{\gamma}x^{\mathrm{mut},d} \leftarrow \min\left(\max\left({}^{\gamma}x^{\mathrm{mut},d}, {}^{\gamma}x^{d}_{\min}\right), {}^{\gamma}x^{d}_{\max}\right)$
Ausgabe: Individuum X^{mut}

Die Schrittweite σ bestimmt die Stärke der Mutation. Wählt man die Schrittweite groß, erhält man eine erforschende Mutation, während mittels kleinem σ in einer lokalen Umgebung des aktuellen Individu-

ums gesucht wird. Zur Begrenzung des mutierten Genotyps auf den zulässigen Suchraum kann alternativ zum Vorgehen in Algorithmus 3 auch solange mutiert werden, bis sich der Zustand ${}^{\gamma}x^{\mathrm{mut},d}$ innerhalb der zulässigen Grenzen $\left[{}^{\gamma}x^{d}_{\min}, {}^{\gamma}x^{d}_{\max}\right]$ befindet.

Für evolutionäre Algorithmen sind zwei Selektionsmechanismen relevant: die Elternselektion und die Umweltselektion. Die Selektion kann sowohl mit als auch ohne Selektionsdruck erfolgen. In der Regel konzentriert sich dieser nur auf einen der beiden Selektionsschritte.

Die Umweltselektion hat die Aufgabe, die Eltern für die nächste Generation zu bestimmen. Dies erfolgt meist mit Selektionsdruck, damit sich die nachfolgende Generation aus den besseren Individuen zusammensetzt. Häufig wird Duplikatfreiheit gefordert, um eine ausreichend große Vielfalt zu erhalten. Die Selektion kann deterministisch erfolgen, z. B. mit der *Besten-Selektion* [100], bei der aus einer Population der Größe N die K Individuen mit der besten Fitness gewählt werden.

Bei der Elternselektion für die Rekombination sollte jedes Individuum eine Chance haben, gewählt zu werden. Auch hier kann eine deterministische Methode angewandt werden, bei der jedes Individuum eine feste Anzahl an Kindern erzeugt. Eine probabilistische Elternselektion ohne Selektionsdruck stellt beispielsweise die *Uniforme Selektion* dar [100], bei der jedes Individuum die gleiche Chance besitzt, selektiert zu werden. Ein Selektionsdruck kann beispielsweise mit der *Fitnessproportionalen Selektion* aufgebaut werden, bei der Individuen mit einer Wahrscheinlichkeit zur Rekombination gewählt werden, die von ihrer Gewichtung abhängt gemäß

$$P^{\mathrm{sel}}(X_i) = \frac{\pi_i}{\sum_{i=1}^{N} \pi_i} \,. \tag{3.22}$$

Diese Methode besitzt einige Nachteile [100]. Wenn beispielsweise ein Individuum eine wesentlich höhere Fitness als die anderen besitzt, wird dieses die Selektion stark dominieren, so dass eine zu geringe Artenvielfalt resultiert. Außerdem kann es vorkommen, dass das beste Individuum gar nicht gewählt wird. Abhilfe kann z. B. eine rangbasierte Selektion schaffen. Zu diesen Methoden zählen die *q-fache Turnierselektion* und das *Stochastisch-universelle Sampling* [100].

Varianten evolutionärer Algorithmen

Im Folgenden wird kurz auf verbreitete Standardalgorithmen eingegangen. Nach [100] lassen sich EAs in drei Teilgebiete einteilen – den genetischen Algorithmen, den Evolutionsstrategien und dem genetischen Programmieren. Die ersten beiden werden im Folgenden kurz beschrieben.

Beim *genetischen Algorithmus* (GA) in der klassischen Form wird ein Individuum durch eine binäre Zeichenkette codiert. Der Suchraum des Genotyps hat damit die Form

$$\Gamma = \mathfrak{B}^D = \{0,1\}^D, \tag{3.23}$$

wobei D die Dimension des Genotyps ist. Es existieren jedoch auch reellwertige GAs. Die Mutation spielt beim genetischen Algorithmus nur eine untergeordnete Rolle. Die treibenden Kräfte sind die Elternselektion und die Rekombination. Ebenso ist die Umweltselektion kaum relevant. Beim sog. Standard-GA wird die gesamte Elternpopulation am Ende einer Generation von den Kindern ersetzt. Meist wird jedoch eine Kombination der Populationen angewandt, wobei die Kinder nur einen gewissen Anteil der neuen Population ausmachen.

Bei *Evolutionsstrategien* (ES) wird generell ein reellwertiger Genotyp verwendet. Meist wird keine Codierung verwendet und es gilt $\Gamma = \Phi$. Im Gegensatz zum GA werden bei der Paarungsselektion alle Elternindividuen mit gleicher Wahrscheinlichkeit ausgewählt. Der Selektionsdruck ist hier in der Umweltselektion zu finden, bei der nur die besten Individuen gewählt werden. Die Mutation ist hier der entscheidende Operator; dafür eignet sich beispielsweise die Gauß-Mutation. Auf die Rekombination wird teilweise komplett verzichtet. Die Mutation muss in diesem Fall die Erforschung sowie die Feinabstimmung übernehmen [100].

3.3. Evolutionäres Posen-Tracking

Zu den Herausforderungen des markerlosen Körper-Trackings zählt die hohe Dimensionalität des Zustandsraumes und die Tatsache, dass die dafür verwendeten Gewichtungsfunktionen häufig viele lokale Optima besitzen. Wie in Abschnitt 3.2.3 bemerkt wurde, kommen die Stärken

evolutionärer Algorithmen genau bei solchen Optimierungsproblemen zum Tragen. EAs laufen zyklisch ab und die Evolutionsfaktoren *Mutation* und *Selektion* ähneln der Vorgehensweise von Partikelfiltern. Aufgrund dieser Punkte sind evolutionäre Verfahren prädestiniert für eine Kombination mit simuliertem *Annealing*, ähnlich dem Vorgehen bei APF und ISA. Zusätzlich kann die Optimierung bei EAs durch die *Rekombination* vorangetrieben werden. In [27] wurde bereits gezeigt, dass durch einen einfachen *Crossover*-Operator starke Verbesserungen beim APF erzielt werden können. In diesem Abschnitt wird eine Methode zum markerlosen Körper-Tracking vorgestellt, welche sich einer Evolutionsstrategie in Kombination mit simuliertem *Annealing* bedient. Die Generationen des EA entsprechen hierbei den *Annealing*-Stufen des ISA.

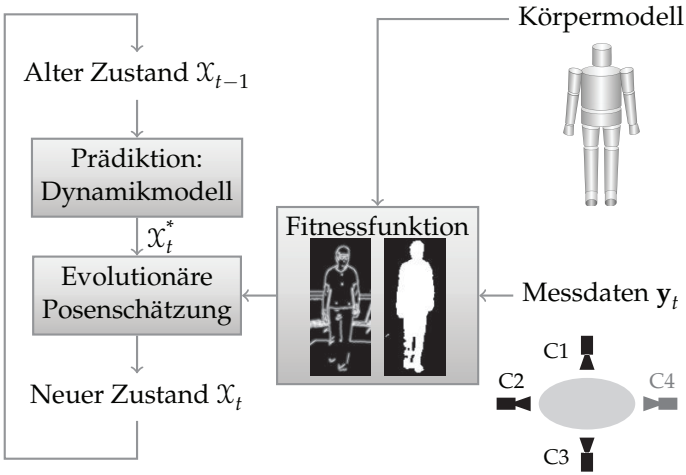


Abbildung 3.3. Übersicht über den Ablauf des Evolutionären Posen-Trackings (EVP).

3.3.1. Übersicht

Abbildung. 3.3 zeigt eine schematische Übersicht über den Ablauf des in dieser Arbeit beschriebenen Evolutionären Posen-Trackings (EVP). Die Grundlage bildet ein Körpermodell, welches die Pose der beobachteten Person in einem Zustandsvektor $\mathbf{x}$ codiert. Auf den Aufbau des

verwendeten Modells wird in Abschnitt 3.3.2 eingegangen. Gemäß den gängigen Begrifflichkeiten evolutionärer Algorithmen wird der Zustand nun auch als *Individuum* bezeichnet. Zur Bildung des Genotyps wird keine Codierung des Zustandsvektors $\mathbf{x}$ verwendet und somit gilt

$$^{\gamma}\mathbf{x} = {}^{\phi}\mathbf{x} := \mathbf{x}\,. \tag{3.24}$$

Eine *Population* setzt sich aus einer bestimmten Anzahl N an Individuen zusammen:

$$\mathcal{X} = \{X_i\}\,, \quad i = 1, \ldots, N\,. \tag{3.25}$$

Das i-te Individuum wird durch $X_i = (\mathbf{x}_i, \pi_i)$ repräsentiert. Vor der Gewichtung besitzt jedes Individuum die gleiche Fitness $\pi_i = \frac{1}{N}$. In diesem Fall wird für die Population auch die alternative Notation

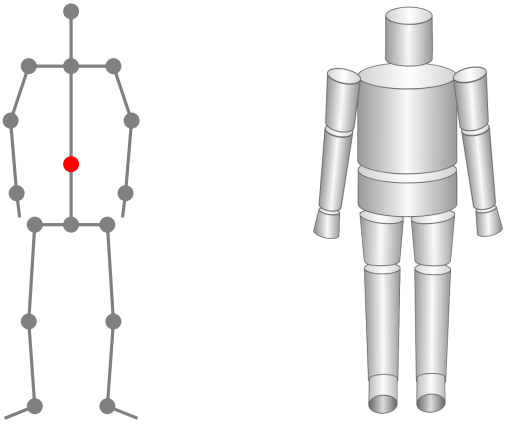
$$\mathcal{X} = \{\mathbf{x}_i\}\,, \quad i = 1, \ldots, N \tag{3.26}$$

verwendet.

Der Ausgangspunkt des Trackings zum Zeitpunkt t ist die Population $\mathcal{X}_{t-1}$ des vorigen Zeitschrittes. Ausgehend vom alten Zustand wird zunächst mit einem Dynamikmodell eine Prädiktion durchgeführt (Abschnitt 3.3.5). Daraus resultiert der prädizierte Zustand $\mathcal{X}_t^*$, welcher als Eingabe für die evolutionäre Posenschätzung verwendet wird, worauf in Abschnitt 3.3.3 eingegangen wird. Im Rahmen der Posenschätzung erfolgt die Bewertung der Güte der Zustandshypothesen durch Vergleich mit den vorliegenden Messwerten $\mathbf{y}_t$ gemäß der *Fitnessfunktion* $f(\mathbf{x}, \mathbf{y}_t)$, welche in Abschnitt 3.3.6 beschrieben wird. Der resultierende Zustand $\mathcal{X}_t$ dient schließlich der Initialisierung im nächsten Zeitschritt.

3.3.2. Körpermodell

Das verwendete Körpermodell ist in Abbildung 3.4 schematisch abgebildet. Der menschliche Körper wird als kinematische Kette im dreidimensionalen Raum dargestellt. Der Bauchnabel bildet den Ursprung des Modells. Die Körperglieder werden durch Zylinder angenähert, welche über die Gelenke miteinander verbunden sind. Die Modellkonfiguration wird durch die Gelenkwinkel und die absolute Position im Raum repräsentiert.



(a) Kinematische Kette **(b)** Zylindermodell

Abbildung 3.4. Schematische Abbildung des verwendeten Körpermodells.

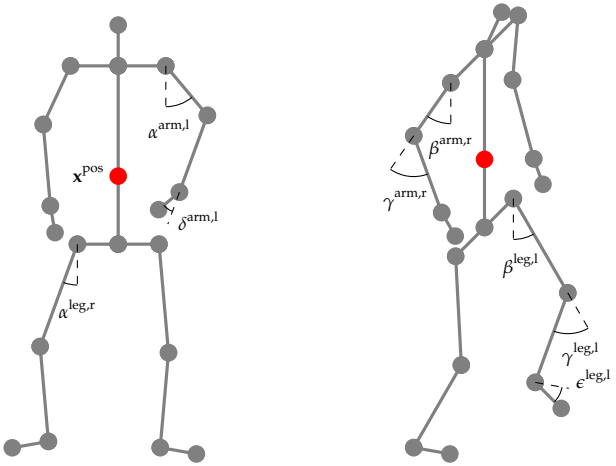
Die Position im Raum setzt sich aus den x-, y- und z-Koordinaten des Modellursprungs zusammen:

$$\mathbf{x}^{\text{pos}} = \begin{bmatrix} x^{\text{pos}} & y^{\text{pos}} & z^{\text{pos}} \end{bmatrix} . \tag{3.27}$$

Abbildung 3.5 zeigt die Gelenkwinkel der Arme und Beine. α beschreibt den seitlichen Öffnungswinkel der Arme bzw. Beine. β ist die Bewegung der Arme bzw. Beine nach vorne und hinten. γ repräsentiert die Beugungswinkel der Ellbogen und Knie. ϵ ist die Bewegung der Hände bzw. Füße nach oben und unten und δ ist die seitliche Bewegung der Hände. Damit ergeben sich für Arme und Beine die Zustandsvektoren für die rechte bzw. linke Körperhälfte:

$$\mathbf{x}^{\text{arm,r/l}} = \begin{bmatrix} \alpha^{\text{arm,r/l}} & \beta^{\text{arm,r/l}} & \gamma^{\text{arm,r/l}} & \epsilon^{\text{arm,r/l}} & \delta^{\text{arm,r/l}} \end{bmatrix} , \tag{3.28}$$

$$\mathbf{x}^{\text{leg,r/l}} = \begin{bmatrix} \alpha^{\text{leg,r/l}} & \beta^{\text{leg,r/l}} & \gamma^{\text{leg,r/l}} & \epsilon^{\text{leg,r/l}} \end{bmatrix} . \tag{3.29}$$



(a) Körpermodell frontale Ansicht **(b)** Körpermodell seitliche Ansicht

Abbildung 3.5. Zustandskomponenten des Körpermodells: Modellursprung und Gelenkwinkel der Arme und Beine.

Die Pose des Ober- und Unterkörpers wird jeweils durch drei Winkel beschrieben. α^{ub} und α^{lb} sind die Neigung von Ober- bzw. Unterkörper zur Seite. β^{ub} und β^{lb} sind die Neigungswinkel nach vorne und hinten. Schließlich sind γ^{ub} und γ^{lb} die Rotationen bezüglich einer festen, senkrecht zum Boden verlaufenden Ebene. Damit ergeben sich die Vektoren

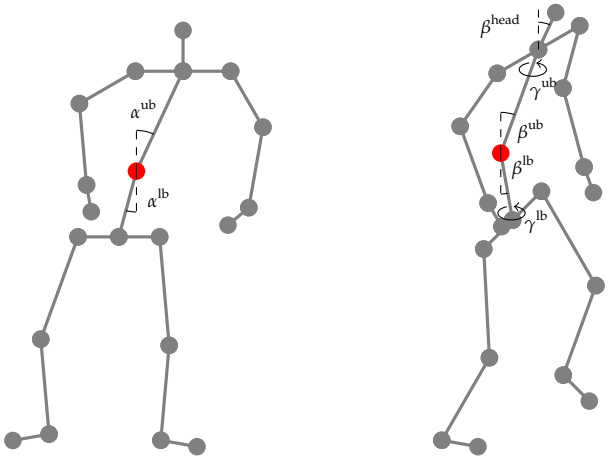
$$\mathbf{x}^{\text{ub}} = \begin{bmatrix} \alpha^{\text{ub}} & \beta^{\text{ub}} & \gamma^{\text{ub}} \end{bmatrix}, \tag{3.30}$$

$$\mathbf{x}^{\text{lb}} = \begin{bmatrix} \alpha^{\text{lb}} & \beta^{\text{lb}} & \gamma^{\text{lb}} \end{bmatrix}. \tag{3.31}$$

Der Kopf hat einen Neigungswinkel nach vorn $x^{\text{head}} = \beta^{\text{head}}$. Der gesamte Zustandsvektor hat schließlich die Form

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^{\text{pos}} & \mathbf{x}^{\text{arm,l}} & \mathbf{x}^{\text{arm,r}} & \mathbf{x}^{\text{leg,l}} & \mathbf{x}^{\text{leg,r}} & \mathbf{x}^{\text{ub}} & \mathbf{x}^{\text{lb}} & x^{\text{head}} \end{bmatrix} \tag{3.32}$$

und besitzt 28 Freiheitsgrade.



(a) Körpermodell frontale Ansicht **(b)** Körpermodell seitliche Ansicht

Abbildung 3.6. Zustandskomponenten des Körpermodells: Gelenkwinkel des Kopfes, des Ober- und des Unterkörpers.

3.3.3. Evolutionäre Posenschätzung

Der Ablauf der evolutionären Posenschätzung wird nun erläutert und ist in Abb. 3.7 anhand einer Generation veranschaulicht. In jedem Zeitschritt wird eine feste Anzahl M an Generationen durchlaufen.

Ausgangspunkt der Posenschätzung im Zeitschritt t ist die Elternpopulation $\mathcal{X}^{\text{p}}_{t,1}$ der ersten Generation. In der m-ten Generation liegt die Elternpopulation $\mathcal{X}^{\text{p}}_{t,m}$ der Größe N_{p} vor:

$$\mathcal{X}^{\text{p}}_{t,m} = \{\mathbf{x}^{\text{p}}_{t,m,i}\}\,, \quad i = 1, \ldots, N_{\text{p}}\,. \tag{3.33}$$

Elternselektion und Rekombination

Zur Bildung der neuen Population $\mathcal{X}^{\text{rec}}_{t,m}$ werden durch Rekombination der Eltern neue Individuen erzeugt:

$$\mathcal{X}^{\text{rec}}_{t,m} = \text{Rec}\left\{\mathcal{X}^{\text{p}}_{t,m}, N\right\}\,. \tag{3.34}$$

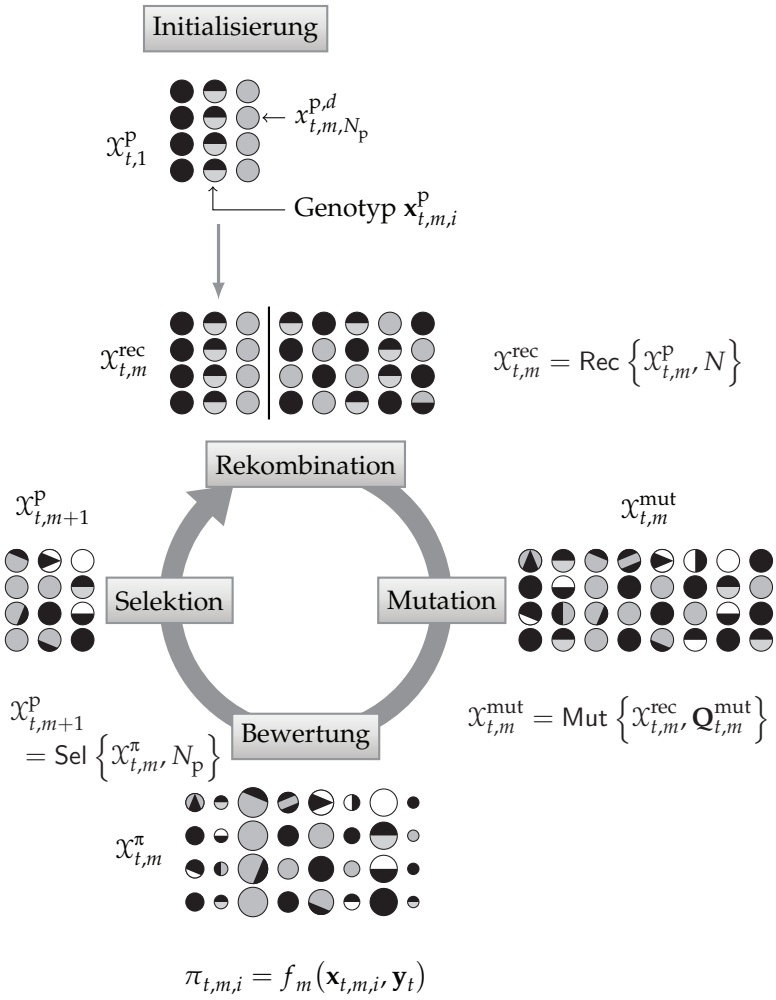


Abbildung 3.7. Ablauf der evolutionären Posenschätzung anhand einer Generation.

Dabei werden die N_p Eltern direkt übernommen und $N - N_\mathrm{p}$ weitere Kinder erzeugt. Somit setzt sich die rekombinierte Population aus N Individuen zusammen. Bei der Elternselektion wird hier kein Selektionsdruck aufgebaut; stattdessen pflanzen sich alle Elternindividuen mit gleicher Wahrscheinlichkeit fort. Dieses Vorgehen sorgt für eine hohe Vielfalt der Kinderpopulation und somit einer guten Erforschung des Suchraumes. Für die Erzeugung der $N - N_\mathrm{p}$ neuen Kinder wird eine Form der kombinierenden Rekombination gewählt, da der Rekombinationsoperator einen erforschenden Charakter besitzen soll. Dabei wird jede einzelne Zustandskomponente von einem zufällig ausgewählten Elternindividuum übernommen Der hier verwendete Operator, angewandt auf die gesamte Population, ist in Algorithmus 4 gegeben.

Algorithmus 4 EVP-Rekombination $\mathcal{X}^\mathrm{rec} = \mathsf{Rec}\left\{\mathcal{X}^\mathrm{p}, N\right\}$.

Eingabe: Elternpopulation $\mathcal{X}^\mathrm{p} = \{\mathbf{x}_1^\mathrm{p}, \ldots, \mathbf{x}_{N_\mathrm{p}}^\mathrm{p}\}$, Größe der Ausgabepopulation N
Übernehmen der Eltern: $\mathcal{X}^\mathrm{rec} = \mathcal{X}^\mathrm{p}$
von $i = N_\mathrm{p} + 1$ bis N **wiederhole** ▷ Kinder erzeugen
  von $d = 1$ bis D **wiederhole**
    Wähle j zufällig aus $\mathcal{U}\left([1, \ldots, N_\mathrm{p}]\right)$
    $x_i^{\mathrm{rec},d} \leftarrow x_j^{\mathrm{p},d}$
  $\mathcal{X}^\mathrm{rec} \leftarrow \{\mathcal{X}^\mathrm{rec}, \mathbf{x}_i^\mathrm{rec}\}$
Ausgabe: $\mathcal{X}^\mathrm{rec}$

Mutation

Als Nächstes wird auf die durch Rekombination entstandene Population $\mathcal{X}_{t,m}^\mathrm{rec}$ eine *Mutation* angewandt. Daraus resultiert die mutierte Population

$$\mathcal{X}_{t,m}^\mathrm{mut} = \mathsf{Mut}\left\{\mathcal{X}_{t,m}^\mathrm{rec}, \mathbf{Q}_{t,m}^\mathrm{mut}\right\}. \tag{3.35}$$

Die angewandte Mutationsmethode ist in Algorithmus 5 dargestellt. Es werden nicht alle Individuen, sondern nur eine zufällige Anzahl N^mut mutiert. Bei jedem dieser Individuen wird im Gegensatz zu vielen verbreiteten Ansätzen lediglich eine zufällige Anzahl D^mut an Zu-

standskomponenten mutiert. Dabei werden die zufällig ausgewählten Zustandsgrößen durch Gauß-Mutation (siehe Alg. 3) verrauscht. Für ein Individuum i und eine Zustandskomponente d ergibt sich beispielsweise:

$$\mathbf{x}_{t,m,i}^{\mathrm{mut},d} = \mathbf{x}_{t,m,i}^{\mathrm{rec},d} + w_{t,m,i}^{d} \,. \tag{3.36}$$

Dabei ist $w_{t,m}^{d}$ ein mittelwertfreier normalverteilter Rauschprozess, dessen Varianz $q_{t,m}^{\mathrm{mut},d}$ aus der Population der vorhergehenden Generation bestimmt wurde. Auf die Berechnung der Varianz wird in Abschnitt 3.3.4 eingegangen.

Algorithmus 5 EVP-Mutation $\mathcal{X}^{\mathrm{mut}} = \mathsf{Mut}\left\{\mathcal{X}, \mathbf{Q}^{\mathrm{mut}}\right\}$.

Eingabe: Population $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, Mutationsvarianz $\mathbf{Q}^{\mathrm{mut}}$
Wähle N^{mut} zufällig aus $\mathcal{U}([1, N])$
Wähle $\mathbf{i}^{\mathrm{mut}}$ zufällig aus $\mathcal{U}\left([1, N]^{N^{\mathrm{mut}}}\right)$
$\mathcal{X}^{\mathrm{mut}} \leftarrow \mathcal{X}$ ▷ initialisieren
für alle $i \in \mathbf{i}^{\mathrm{mut}}$ **wiederhole** ▷ mutieren
  Wähle D^{mut} zufällig aus $\mathcal{U}([1, D])$
  Wähle $\mathbf{d}^{\mathrm{mut}}$ zufällig aus $\mathcal{U}\left([1, D]^{D^{\mathrm{mut}}}\right)$
  für alle $d \in \mathbf{d}^{\mathrm{mut}}$ **wiederhole**
    wiederhole
      Wähle w_i^d zufällig gemäß $\mathcal{N}\left(0, \sqrt{q^{\mathrm{mut},d}}\right)$
      $x_i^{\mathrm{mut},d} \leftarrow x_i^{\mathrm{mut},d} + w_i^d$
    bis $x_i^{\mathrm{mut},d} \in \left[x_{\min}^d, x_{\max}^d\right]$
Ausgabe: $\mathcal{X}^{\mathrm{mut}}$

Gewichtung und Umweltselektion

Nun folgt die Bewertung der mutierten Population $\mathcal{X}_{t,m}^{\mathrm{mut}}$ bezüglich der Fitnessfunktion f_m. Die Fitness eines Individuums lautet

$$\pi_{t,m,i} = f_m\left(\mathbf{x}_{t,m,i}, \mathbf{y}_t\right) . \tag{3.37}$$

Da die evolutionäre Posenschätzung über *Annealing*-Eigenschaften verfügen soll, wird in jeder Generation m eine gemäß dem *Annealing*-Schema $\{\beta_m\}$ geglättete Version

$$f_m(\mathbf{x}, \mathbf{y}) = \exp\left(-\beta_m \, e(\mathbf{x}, \mathbf{y})\right) \tag{3.38}$$

der Fitnessfunktion herangezogen, die gemäß Abschnitt 3.3.6 aus einer Energiefunktion $e(\mathbf{x}, \mathbf{y})$ berechnet wird.

Aus der gewichteten Population $\mathcal{X}^{\pi}_{t,m}$ werden schließlich N_p Individuen für die nächste Elterngeneration $\mathcal{X}^{\mathrm{p}}_{t,m+1}$ mittels einer stark zielgerichteten Selektionsmethode ausgewählt:

$$\mathcal{X}^{\mathrm{p}}_{t,m+1} = \mathsf{Sel}\left\{\mathcal{X}^{\pi}_{t,m}, N_\mathrm{p}\right\}. \tag{3.39}$$

Dies erfolgt mit einer deterministischen Selektionsmethode, die ähnlich der Besten-Selektion abläuft, jedoch nicht duplikatfrei ist. Der entwickelte Selektionsoperator ist in Algorithmus 6 als Index-Selektion gegeben. Es werden N_p aus N Individuen so ausgewählt, dass die Verteilung der Anteile ihrer Gewichte gleich bleibt. Dazu werden zunächst die relativen Häufigkeiten aller Gewichte berechnet:

$$k_i = \frac{\pi_i}{\sum_{j=1}^{N} \pi_j}, \quad i = 1, \ldots, N. \tag{3.40}$$

Die absolute Häufigkeit, mit der das i-te Individuum selektiert werden soll, wird zu $K_i = \lfloor k_i \cdot N_\mathrm{p} + 0{,}5 \rfloor$ berechnet. Anschließend werden nacheinander die stärksten Individuen jeweils K_i-mal gezogen, solange bis N_p Individuen selektiert worden sind.

Nach dem Durchlaufen von M Generationen wird aus der zuletzt bestimmten Generation das Individuum mit der höchsten Gewichtung $\mathbf{x}_t \coloneqq \mathbf{x}^{\mathrm{mut}}_{t,M,i_{\max}}$ als die endgültige Pose des Zeitschrittes t ausgewählt:

$$i_{\max} = \arg\max_i \pi_{t,M,i}. \tag{3.41}$$

3.3.4. Bestimmen der Mutationsvarianzen

Die Varianzen der Rauschprozesse für die Mutation müssen sehr sorgfältig gewählt werden. Zu Beginn der Evolution sollen die Individuen

weit gestreut werden, um neue Bereiche des Suchraums zu erschließen (explorative Mutation). Im Laufe der Generationen soll die Varianz sinken, so dass die Mutation am Ende für die Feinabstimmung sorgt. Die Varianz darf jedoch nicht zu schnell sinken, damit die Suche nicht in einem lokalen Optimum endet.

Ein dynamisches Varianzschema ist gegenüber einer festen Vorgabe zu bevorzugen, da hiermit eine adaptive hierarchische Suche umgesetzt wird, wie bereits in Abschnitt 3.2.2 diskutiert wurde. Eine selbstständige Varianzeinschränkung gemäß Gleichung (3.9) ist gut für dominante Körperglieder, wie den Torso, geeignet, da diese meist robust erkannt werden und somit eine rasche Varianzreduktion erfolgt. Bei Gliedmaßen, die einen geringeren Einfluss auf die Fitnessfunktion ausüben und schwieriger zu lokalisieren sind, wie z. B. Arme und Hände, kann dieses Vorgehen jedoch problematisch sein, da die entsprechenden Varianzen sich nicht ausreichend verringern. In dieser Arbeit wird eine Variante gewählt, welche die Vorzüge der autonomen Varianzentwicklung umsetzt, jedoch in den o. g. Fällen Abhilfe schafft, indem ein dynamisches mit einem erzwungenen Varianzschema kombiniert wird.

Algorithmus 6 EVP-Umweltselektion $\mathbf{i}^{\mathrm{sel}} = \mathsf{IS}\left\{\boldsymbol{\pi}, N_{\mathrm{p}}\right\}$.

Eingabe: Gewichte $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_N\}$, N_{p} zu selektierende Individuen
Annahme: Gewichte sind absteigend sortiert $\pi_1 \geq \pi_2 \geq \ldots \geq \pi_N$
von $i = 1$ bis N **wiederhole** ▷ Häufigkeiten berechnen
  Berechne $K_i = \left\lfloor \frac{\pi_i}{\sum_{j=1}^{N} \pi_j} N_{\mathrm{p}} + 0{,}5 \right\rfloor$
Initialisierung $\mathbf{i}^{\mathrm{sel}} = \{\ \}$, $i = 1$, $\kappa = 1$
wiederhole
  $\mathbf{i}^{\mathrm{sel}} \leftarrow \{\mathbf{i}^{\mathrm{sel}}, i\}$
  $\kappa \leftarrow \kappa + 1$
  falls $\kappa > K_i$ **dann**
    $i \leftarrow i + 1$, $\kappa \leftarrow 1$
bis N_{p} Individuen selektiert sind
Ausgabe: $\mathbf{i}^{\mathrm{sel}}$

$\mathbf{Q}_{t,m}^{\text{mut}} = \mathrm{E}\{\mathbf{w}_{t,m,i}^{\mathrm{T}} \mathbf{w}_{t,m,i}\}$ ist die Kovarianzmatrix des Mutationsrauschens zum Zeitpunkt t und in der Generation m. Dabei sind $\mathbf{Q}_{t,m}^{\text{mut}}$ positiv semidefinite Diagonalmatrizen mit den Diagonalelementen $q_{t,m}^{\text{mut},d}$, $d = 1, \ldots, D$. In der ersten Generation erfolgt die Mutation mit fest vorgegebenen Varianzen $\mathbf{Q}_{t,1}^{\text{mut}} = \mathbf{Q}_{0}^{\text{mut}}$. In der m-ten Generation wird die Population $\mathcal{X}_{t,m-1}^{\pi}$ zur Bestimmung von $\mathbf{Q}_{t,m}^{\text{mut}}$ herangezogen. Dazu erfolgt zunächst eine erneute Selektion basierend auf der Population $\mathcal{X}_{t,m-1}^{\pi}$. Hierzu wird mit dem Stochastisch-universellen Sampling [100] ein weniger stark zielgerichteter Selektionsoperator als bei der Elternselektion verwendet:

$$\breve{\mathcal{X}}_{t,m} = \mathsf{Sel}_{\text{sus}} \left\{ \mathcal{X}_{t,m-1}^{\pi} \right\} . \tag{3.42}$$

Aus dieser Population wird eine Abschätzung der Varianz vorgenommen gemäß

$$\breve{q}_{t,m}^{\text{mut},d} = \frac{1}{N-1} \sum_{i=1}^{N} \left(\breve{x}_{t,m,i}^{d} - \breve{\bar{x}}_{t,m}^{d} \right)^2 , \quad d = 1, \ldots, D , \tag{3.43}$$

mit dem gewichteten Mittelwert

$$\breve{\bar{\mathbf{x}}}_{t,m} = \frac{1}{N} \sum_{i=1}^{N} \pi_{t,m,i} \, \breve{\mathbf{x}}_{t,m,i} . \tag{3.44}$$

Nun wird für jede Zustandskomponente geprüft, ob der so ermittelte Wert $\breve{q}_{t,m}^{\text{mut},d}$ unterhalb einer für die aktuelle Generation maximal zulässigen Varianz $q_{m,\max}^{\text{mut},d}$ liegt. Ist dies gegeben, wird der geschätzte Wert übernommen, ansonsten der Maximalwert

$$q_{t,m}^{\text{mut},d} = \min \left\{ \breve{q}_{t,m}^{\text{mut},d}, q_{m,\max}^{\text{mut},d} \right\} . \tag{3.45}$$

3.3.5. Prädiktion

Liegt ein gutes Bewegungsmodell zur Prädiktion der Posen vor und erfolgt die Initialisierung der Population in einem Zeitschritt damit nahe an der tatsächlichen Pose, wird die nachfolgende Posenschätzung

stark erleichtert, vor allem im Falle weniger Kameraperspektiven. Allerdings stellt es eine große Herausforderung dar, dynamische Modelle menschlicher Bewegungen zu formulieren, die einerseits informativ genug sind, um eine genaue Prädiktion zu erreichen, jedoch gleichzeitig eine möglichst große Fülle an verschiedenen Bewegungsarten beschreiben können.

Hier wird zunächst ein sehr allgemeingültiges Modell betrachtet, um der Spontaneität menschlicher Bewegungen gerecht zu werden. Ein sehr einfaches Dynamikmodell stellt die Addition eines mittelwertfreien, normalverteilten Rauschprozesses $\mathbf{v}_t$ mit spezifischen Varianzen dar [27, 85]:

$$\mathbf{x}_{t,i}^{*} = \mathbf{x}_{t-1} + \mathbf{v}_{t,i}\,. \tag{3.46}$$

Dabei ist $\mathbf{Q}_v = \mathrm{E}\{\mathbf{v}_t^{\mathrm{T}}\mathbf{v}_t\}$ die Kovarianzmatrix des Prädiktionsrauschens mit $\mathrm{E}\{v_t^i, v_t^j\} = \delta_{ij}$. Weiterhin wird gefordert, dass sich die Körperwinkel in anatomisch plausiblen Bereichen befinden (siehe [133]) und es nicht zu Durchkreuzungen zwischen Körperteilen kommt. Das Dynamikmodell, welches aus $\mathbf{Q}_v$ und den Winkelbegrenzungen besteht, kann für bestimmte Bewegungsarten oder Personen aus Trainingsdaten gelernt werden, wie in [85]. Aktionsspezifische Dynamikmodelle sind hilfreich, wenn sie auf die entsprechenden Aktionen angewandt werden, im Falle anderer Aktionen sind sie jedoch nachteilig [85]. Hier wird daher ein gemeinsames Modell für alle Bewegungsarten verwendet.

In einzelnen Kamerabildern kommt es häufig zu Überlappungen verschiedener Körperteile. Je weniger Aufnahmeperspektiven zur Verfügung stehen, desto schwieriger ist es, solche Mehrdeutigkeiten aufzulösen. Um hier dennoch eine korrekte Zuordnung zu erreichen, kann ein stärkeres Dynamikmodell wertvolle Zusatzinformationen liefern. Daher wird anhand von Geh-Bewegungen untersucht, wie das einfache Dynamikmodell erweitert werden kann, um im Falle von Überlappungen dennoch ein erfolgreiches Tracking zu erreichen. Die Vorteile des Modells (3.46) sollen dabei weitgehend erhalten bleiben. Zunächst wird geprüft, ob sich die Arme oder Beine der rechten und linken Körperhälfte in einzelnen Bildern gegenseitig überdecken. Tritt eine Überlappung auf, werden die Geschwindigkeiten der betroffenen Gelenkwinkel aus den Schätzwerten mehrerer vergangener Zeitschritte bestimmt. Dann wird

für den Gelenkwinkel x_t^d derjenigen Körperhälfte, die sich schneller bewegt, seine geschätzte Geschwindigkeit v_t^d zur Prädiktion hinzugezogen:

$$x_{t,i}^{*d} = x_{t-1}^d + v_t^d \Delta t + v_{t,i}^d \,. \tag{3.47}$$

Die übrigen Zustandskomponenten werden weiterhin wie zuvor bestimmt. Details zur Umsetzung dieses Vorgehens können in [133] nachgelesen werden.

3.3.6. Gewichtung

Die Gewichtung besitzt die Aufgabe, Posenhypothesen basierend auf den vorhandenen Kameraaufnahmen zu bewerten. Die Qualität der Gewichtungsfunktion ist somit entscheidend für ein erfolgreiches Körper-Tracking. Hier wird eine sehr allgemeingültige Gewichtungsmethode basierend auf [27] und [85, 90] eingesetzt, welche auf Kantenbildern und Körpersilhouetten basiert. Hierzu wird kein Vorwissen über das Erscheinungsbild oder die Kleidung der beobachteten Person benötigt.

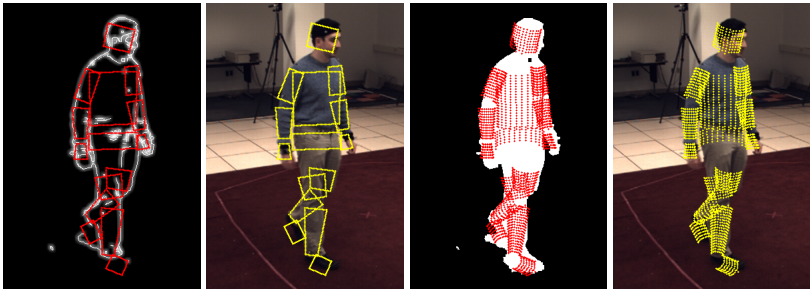


Abbildung 3.8. Beispiel für die Bestimmung der Kanten- und Silhouettengewichtung (Bildquelle [85]). Links ist das geglättete Kantenbild zu sehen, auf das die Kanten einer Posenhypothese projiziert sind. Daneben sind die projizierten Modellkanten auf dem Eingangsbild zu sehen. Die beiden rechten Bilder zeigen die projizierten Modellsilhouetten auf dem Bildvordergrund bzw. dem Eingangsbild.

Als Beobachtungen liegen Aufnahmen von C Kameras vor: $\mathbf{y} = \{\mathbf{i}^c\}$, $c = 1, \ldots, C$. Zunächst wird die auf Bildkanten und anschließend die auf

Körpersilhouetten basierte Gewichtung erläutert, bevor auf die Kombination der einzelnen Gewichtungen eingegangen wird. Aus Gründen der Übersichtlichkeit wird der Index des aktuellen Zeitschritts hier weggelassen.

Kantengewichtung

Bilder 3.8(a) und 3.8(b) veranschaulichen die Bestimmung der Kantengewichtung. Aus dem Eingangsbild jeder Kamera wird zunächst das Kantenbild mit dem Canny-Algorithmus [16] bestimmt. Kanten des Hintergrundes werden entfernt. Das resultierende Bild der Vordergrundkanten wird mit einem Gauß-Filter geglättet und auf den Wertebereich $[0, 1]$ normiert. Das geglättete und normierte Kantenbild der c-ten Kamera wird als $k_{\mathrm{e}}^{c}(\mathbf{x})$ bezeichnet.

Das Oberflächenmodell einer Pose $\mathbf{x}$ wird auf das Koordinatensystem der Kamera c projiziert. Es werden Bildpunkte $\mathbf{x}_{\mathrm{e},r}^{c}$, $r = 1, \ldots, R_{\mathrm{e}}^{c}$ bestimmt, an denen sich Modellkanten befinden, indem die Kanten des Oberflächenmodells in einem gleichmäßigen Raster abgetastet werden. An vertikalen Kanten zwischen einzelnen Körperteilen werden keine Punkte ermittelt.

Die Abweichung zwischen dem Modell und den Bilddaten der Kamera c wird bestimmt zu

$$e_{\mathrm{e}}(\mathbf{x}, \mathbf{i}^{c}) = \frac{1}{R_{\mathrm{e}}^{c}} \sum_{r=1}^{R_{\mathrm{e}}^{c}} (1 - k_{\mathrm{e}}^{c}(\mathbf{x}_{\mathrm{e},r}^{c}))^{2} \,. \tag{3.48}$$

Die Gesamtabweichung für alle vorhandenen Kameras lautet

$$e_{\mathrm{e}}(\mathbf{x}, \mathbf{y}) = \frac{1}{C} \sum_{c=1}^{C} e_{\mathrm{e}}^{c}(\mathbf{x}, \mathbf{i}^{c}) \,. \tag{3.49}$$

Silhouettengewichtung

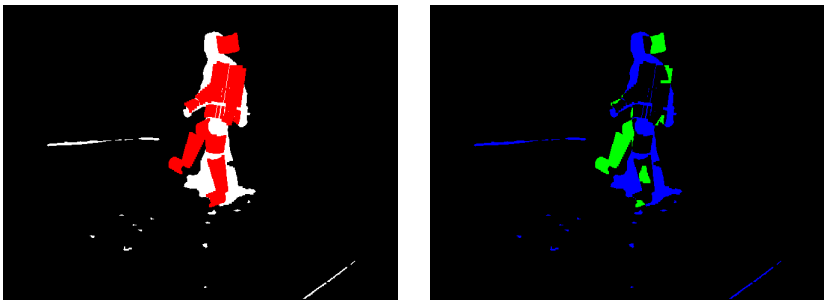
Die Silhouettengewichtung misst die Abweichung zwischen einer Pose und den aus den Eingangsbildern bestimmten Körpersilhouetten. Dazu werden zunächst die binären Silhouettenbilder $k_{\mathrm{s}}^{c}(\mathbf{x})$, $c = 1, \ldots, C$ durch Hintergrundsubtraktion wie in [85] bestimmt. Für die vorliegende

Pose werden Silhouettenmasken für alle Kameras ermittelt. Analog zu oben werden Bildpunkte $\mathbf{x}_{\mathrm{s},r}^{c}$, $r = 1, \ldots, R_{\mathrm{s}}^{c}$ bestimmt, an denen sich die projizierte Modellsilhouette befindet. Abbildungen 3.8(c) und 3.8(d) zeigen eine projizierte Modellsilhouette auf der Bildsilhouette und dem Originalbild. Zur Verringerung des Rechenaufwandes werden nicht alle Punkte der Modellsilhouette ausgewertet, sondern sie wird ebenfalls auf einem gleichmäßigen Raster abgetastet. Die Abweichung ergibt sich damit zu

$$e_{\mathrm{s}}(\mathbf{x}, \mathbf{y}) = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{R_{\mathrm{s}}^{c}} \sum_{r=1}^{R_{\mathrm{s}}^{c}} (1 - k_{\mathrm{s}}^{c}(\mathbf{x}_{\mathrm{s},r}^{c}))^{2} \,. \tag{3.50}$$

Bidirektionale Silhouettengewichtung

Die Silhouettengewichtung kann mit geringem Aufwand berechnet werden, jedoch ermöglicht sie kein Posen-Tracking hoher Qualität. Der Grund dafür ist ihre Asymmetrie: Es wird gefordert, dass sich die projizierte Pose innerhalb des Silhouettenbildes befindet, aber nicht umgekehrt. Dies hat zur Folge, dass häufig Posen geschätzt werden, welche nicht die gesamte Vordergrundsilhouette überdecken, da beispielsweise beide Beine auf den selben Bereich oder die Arme im Bereich des Torsos platziert werden.



(a) (b)

Abbildung 3.9. Beispiel der bidirektionalen Silhouettengewichtung. (a): Projektion einer Modellsilhouette (rot) auf das Silhouettenbild (weiß). (b): Bereich der Bildsilhouette, der nicht durch das Modell bedeckt ist (blau), Bereich des Modells, der außerhalb der Bildsilhouette liegt (grün).

Um dieses Problem zu umgehen, wurde die bidirektionale Silhouettengewichtung vorgeschlagen [85, 90], welche Bereiche der Bildsilhouette bestraft, die nicht durch die Modellsilhouette abgedeckt werden. Dazu werden die projizierten Modellkanten und -silhouetten als binäre Bildmasken M_{e}^{c} bzw. M_{s}^{c} ausgedrückt. Zunächst werden die Größen G_{io}^{c}, G_{mo}^{c} und G_{ov}^{c} dreier Bildbereiche ausgewertet:

1. Der Bereich M_{io} der Bildsilhouette, der nicht durch das Modell bedeckt ist:

$$G_{\mathrm{io}}^{c} = \sum_{r=1}^{R} k_{\mathrm{s}}^{c}(\mathbf{x}_r)\left(1 - M_{\mathrm{s}}^{c}(\mathbf{x}_r)\right) . \tag{3.51}$$

2. Der Bereich M_{mo} des Modells, der sich außerhalb der Bildsilhouette befindet:

$$G_{\mathrm{mo}}^{c} = \sum_{r=1}^{R} M_{\mathrm{s}}^{c}(\mathbf{x}_r)\left(1 - k_{\mathrm{s}}^{c}(\mathbf{x}_r)\right) . \tag{3.52}$$

3. Der überlappende Bereich M_{ov}:

$$G_{\mathrm{ov}}^{c} = \sum_{r=1}^{R} k_{\mathrm{s}}^{c}(\mathbf{x}_r) M_{\mathrm{s}}^{c}(\mathbf{x}_r) . \tag{3.53}$$

Hierbei wird über den gesamten Bildbereich summiert, R ist die Gesamtzahl an Bildpunkten. Der überlappende Bereich G_{ov}^{c} soll maximal werden, während die beiden anderen Bereiche minimal sein sollen. Die unterschiedlichen Bereiche sind in Bild 3.9 dargestellt. Bild 3.9(a) zeigt eine projizierte Modellsilhouette auf einem Silhouettenbild. In Bild 3.9(b) sind die beiden fehlerhaften Bereiche M_{io} und M_{mo} dargestellt.

Die Anzahl der Punkte der Modellmaske außerhalb der Vordergrundsilhouette G_{mo}^{c} wird auf die Gesamtgröße der Modellsilhouette

$$\sum_{r=1}^{R} M_{\mathrm{s}}^{c}(\mathbf{x}_r) = G_{\mathrm{mo}}^{c} + G_{\mathrm{ov}}^{c} \tag{3.54}$$

normiert.

Analog wird G^c_{io} auf die Größe der Bildsilhouette

$$\sum_{r=1}^{R} k^c_{\text{s}}(\mathbf{x}_r) = G^c_{\text{io}} + G^c_{\text{ov}} \tag{3.55}$$

normiert. Die Energiefunktion bestraft die beiden nicht-überlappenden, normierten Regionen:

$$e^c_{\text{bis}}(\mathbf{x}, \mathbf{i}^c) = (1-a)\frac{G^c_{\text{mo}}}{G^c_{\text{mo}} + G^c_{\text{ov}}} + a\frac{G^c_{\text{io}}}{G^c_{\text{io}} + G^c_{\text{ov}}}\,. \tag{3.56}$$

Für $a = \frac{1}{2}$ ergibt sich eine symmetrische Energiefunktion, für $a = 0$ verhält sich (3.56) gerade wie die einseitige Silhouettenabweichung (3.50).

Gesamte Gewichtungsfunktion

In dieser Arbeit werden die Kanten- und bidirektionale Silhouettengewichtung fusioniert. Die Gesamtabweichung ergibt sich als gewichtete Summe der Einzelabweichungen

$$e_{\text{e,bis}}(\mathbf{x}, \mathbf{y}) = \sum_{c=1}^{C} \alpha_{\text{e}} e_{\text{e}}(\mathbf{x}, \mathbf{i}^c) + (1-\alpha_{\text{e}}) e_{\text{bis}}(\mathbf{x}, \mathbf{i}^c)\,. \tag{3.57}$$

Die Gewichtungs- bzw. Fitnessfunktion ergibt sich schließlich zu

$$f_{\text{e,bis}}(\mathbf{x}, \mathbf{y}) = \exp\left(-e_{\text{e,bis}}(\mathbf{x}, \mathbf{y})\right)\,. \tag{3.58}$$

3.4. Ergebnisse

Im Folgenden wird das evolutionäre Posen-Tracking anhand von Experimenten evaluiert. Zunächst werden das betrachtete Szenario und die verwendeten Evaluationsmethoden beschrieben. Anschließend werden die durchgeführten Versuche und die erzielten Ergebnisse dargestellt.

3.4.1. Szenario

Das Posen-Tracking wird auf den HumanEva-Datensatz [85] angewandt. Dieser Datensatz wurde speziell für die Entwicklung markerloser

Tracking-Methoden entwickelt, um einen quantitativen Vergleich verschiedener Methoden zu ermöglichen. Dazu wurden synchronisierte Videoaufnahmen und markerbasierte dreidimensionale *Motion Capture*-Daten aufgenommen. Letztere dienen als Grundwahrheit für die Beurteilung der Tracking-Verfahren. Dazu schlagen die Autoren ein Fehlermaß vor, um die Ergebnisse markerloser Verfahren mit den Markerverläufen zu vergleichen. Anstelle von speziellen eng anliegenden Anzügen, wie sie im *Motion Capturing* häufig verwendet werden, tragen die Testpersonen normale Kleidung, auf der reflektierende Marker angebracht sind. Dies führt allerdings zu einer verringerten Genauigkeit der *Motion Capture*-Daten [85]. Die Testpersonen führen einige vordefinierte Aktionen in mehreren Wiederholungen aus.

Es existieren zwei Versionen des Datensatzes. Bei der zweiten Version (HumanEva-II) kam ein verbessertes Hardware-System zum Einsatz mit besseren Grundwahrheitsdaten und besserer Synchronisation. Testergebnisse werden primär für HumanEva-II berichtet [85], die Daten des ersten Datensatzes (HumanEva-I) dienen eher als Trainingsdaten. Die Testsequenzen aus HumanEva-II bestehen in Aufnahmen zweier Personen, die eine Sequenz von Geh-, Lauf- und stehenden Balancier-Bewegungen ausführen. Es stehen Farbvideos aus vier Perspektiven mit einer Bildrate von 60 Hz zur Verfügung. Die Grundwahrheit für diese Sequenzen werden nicht direkt zur Verfügung gestellt, mit Ausnahme einiger Zeitschritte, die zur Initialisierung von Tracking-Methoden verwendet werden können. Die Fehlermaße können stattdessen auf der Internetseite des Datensatzes ermittelt werden, indem die Tracking-Ergebnisse in einer vordefinierten Form hochgeladen werden [44]. Auf die Fehlerberechnung wird im folgenden Abschnitt eingegangen.

3.4.2. Evaluationsmethoden

In der Literatur finden sich unterschiedliche Methoden, das markerlose Körper-Tracking zu bewerten. In [85] wird eine Übersicht über Evaluationsmethoden verschiedener Publikationen gegeben. Die Bewertung von Tracking-Ergebnissen kann qualitativ oder quantitativ erfolgen. Bei der qualitativen Auswertung wird die Passgenauigkeit einer Pose durch Betrachten ihrer Projektion auf die vorliegenden Bildaufnahmen beurteilt. Für eine quantitative Auswertung wird ein Fehlermaß basierend auf als

wahr angenommenen Posen benötigt. Diese können durch manuelle Annotation oder ein markerbasiertes *Motion Capturing* gewonnen werden. Eine weitere Alternative ist die Evaluation mittels synthetischer Daten.

In [85] wird ein Fehlermaß zwischen zwei Posen vorgeschlagen, welches, im Gegensatz zu Gelenkwinkeldifferenzen, unabhängig vom gewählten Körpermodell universell einsetzbar ist. Dieses basiert auf mehreren virtuellen Markern, welche die Positionen von Gelenken und Gliedmaßen darstellen. Um dieses Maß für ein bestimmtes Körpermodell einsetzen zu können, müssen aus diesem lediglich die Positionen der virtuellen Marker bestimmt werden.

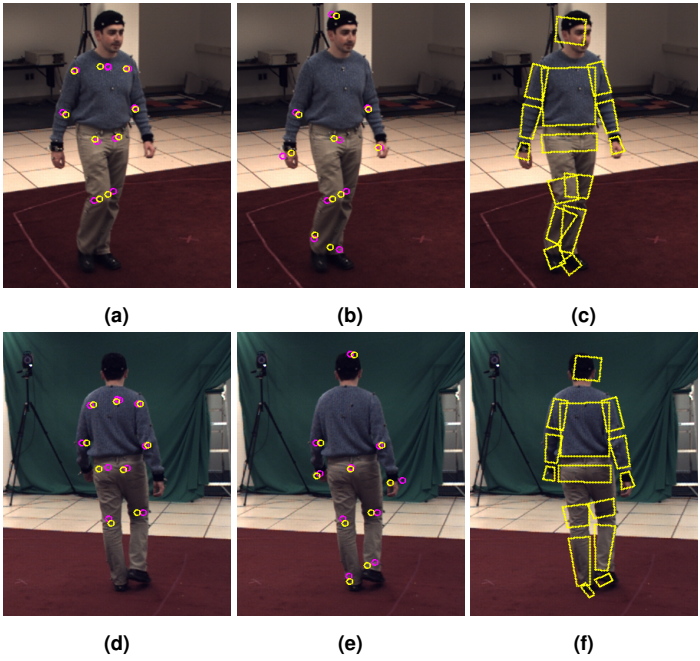


Abbildung 3.10. Virtuelle Marker für die Fehlerberechnung für ein Beispiel des HumanEva-II-Datensatzes. In den ersten beiden Spalten sind die Marker der aus *Motion Capturing* gewonnenen Grundwahrheit in Magenta und die der geschätzten Pose in Gelb dargestellt. Rechts ist das projizierte Modell der geschätzten Pose zu sehen. Die virtuellen Marker sind unterteilt in die anatomischen Richtungen *proximal* (zum Körperzentrum hin gelegen, linke Spalte) und *distal* (vom Körperzentrum entfernt, mittlere Spalte).

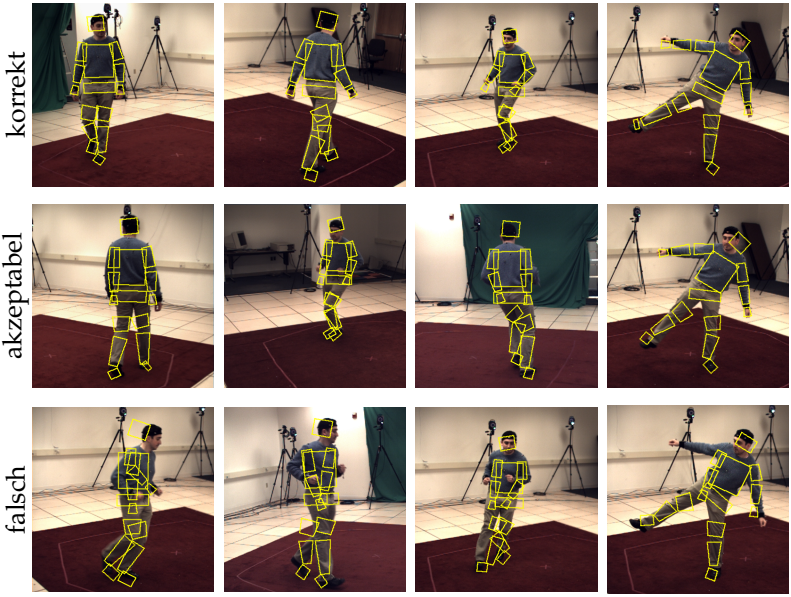


Abbildung 3.11. Veranschaulichung des Fehlermaßes: Beispiele für korrekte, akzeptable und falsche Posen nach dem Bewertungsschema aus [88].

Eine Pose $\mathbf{x}$ wird dabei durch $M = 15$ virtuelle Marker dargestellt als $\{\mathbf{m}_i(\mathbf{x})\}, i = 1, \ldots, M$. Dabei gibt $\mathbf{m}_i(\mathbf{x}) \in \mathbb{R}^3$ die Position des i-ten virtuellen Markers im dreidimensionalen Raum an. Die Abweichung zwischen zwei Posen $\mathbf{x}$ und $\tilde{\mathbf{x}}$ ist der mittlere absolute Abstand zwischen den einzelnen Markern:

$$d(\mathbf{x}, \tilde{\mathbf{x}}) = \frac{1}{M} \sum_{i=1}^{M} \|\mathbf{m}_i(\mathbf{x}) - \mathbf{m}_i(\tilde{\mathbf{x}})\| . \tag{3.59}$$

Abbildung 3.10 zeigt ein Beispiel für die virtuellen Marker einer geschätzten Pose und der Grundwahrheit.

Für eine Sequenz der Länge T wird die mittlere Abweichung betrachtet:

$$\mu_d = \frac{1}{T} \sum_{t=1}^{T} d(\mathbf{x}_t, \tilde{\mathbf{x}}_t) \,. \tag{3.60}$$

Beim Fehlermaß (3.60) wird davon ausgegangen, dass das Tracking-Ergebnis in Form einer einzigen Pose vorliegt [85]. Bei Algorithmen, die die A-posteriori-Dichte durch multimodale Dichten annähern, wie beispielsweise APF oder ISA, wird empfohlen, das Ergebnis durch die wahrscheinlichste Pose zu repräsentieren.

In [88] werden die gemäß Gleichung (3.59) ermittelten Fehler qualitativ folgendermaßen bewertet: Fehler unter 80 mm werden als korrekte, unter 120 mm als akzeptable bzw. größtenteils richtige und über 120 mm als falsche Posen angesehen. Korrekt bedeutet hierbei, dass alle Körperteile richtig lokalisiert, jedoch kleine Ungenauigkeiten erlaubt sind. Abbildung 3.11 zeigt Beispiele für korrekte, akzeptable und falsche Posen.

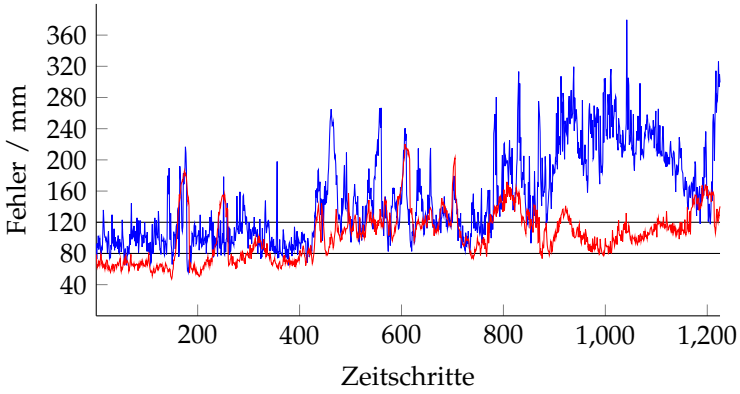
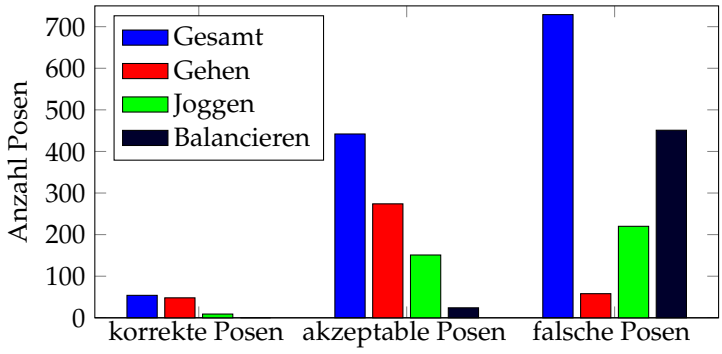


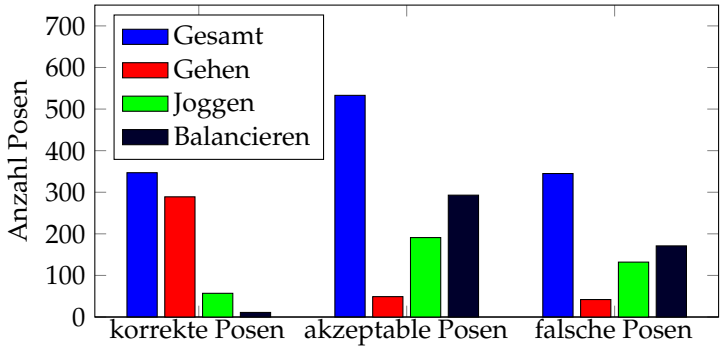
Abbildung 3.12. Verläufe des Fehlers für ISA (blau) und EVP (rot).

Tabelle 3.1. Mittelwert und Standardabweichung der Fehler in mm von ISA und EVP.

Tracker	Gehen	Joggen	Balancieren	Gesamt
μ_d^{ISA}	$96{,}05 \pm 22{,}80$	$128{,}57 \pm 41{,}59$	$192{,}70 \pm 51{,}85$	$143{,}53 \pm 58{,}38$
$\mu_d^{\text{ISA,MAP}}$	$102 \pm 24{,}54$	$134{,}19 \pm 40{,}56$	$202{,}87 \pm 50{,}53$	$151{,}25 \pm 59{,}36$
μ_d^{EVP}	$78{,}94 \pm 29{,}16$	$113{,}45 \pm 30{,}81$	$115{,}70 \pm 22{,}66$	$103{,}94 \pm 31{,}98$



(a) ISA



(b) EVP

Abbildung 3.13. Fehlerhistogramme aufgeteilt nach Aktionen von ISA und EVP.

3.4.3. Evolutionäres Posentracking

Nachdem die Vorgehensweise zur Auswertung vorgestellt wurde, werden nun die Resultate des Körper-Trackings mittels evolutionärer Posenschätzung (EVP) diskutiert. Dazu wird dieser Algorithmus mit dem *„Interacting Simulated Annealing“*-Partikelfilter (ISA) bei gleichen Rahmenbedingungen verglichen. Für beide Methoden werden eine sehr geringe Populationsgröße von $N = 50$ Individuen bzw. Partikeln verwendet und $M = 10$ Generationen bzw. *Annealing*-Stufen durchlaufen.

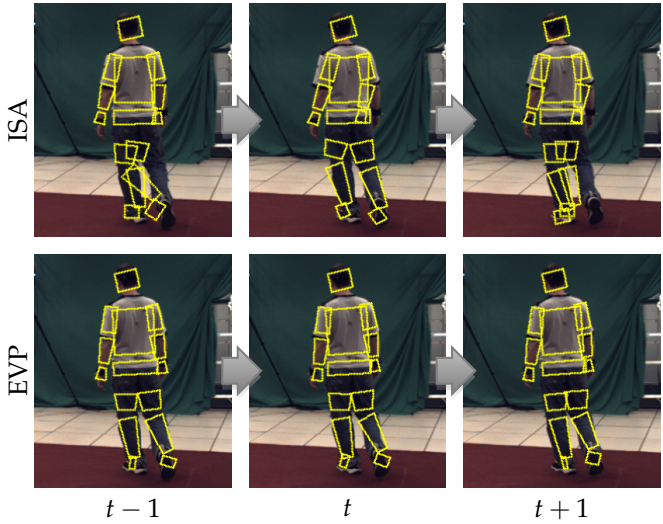


Abbildung 3.14. Beispiel für Fluktuationen der ISA-Posen in drei aufeinanderfolgenden Zeitschritten.

Die Größe der Elternpopulation beträgt $N_\mathrm{p} = 10$. Als *Annealing*-Schema wird $\beta_m = \beta_0^m$ mit $\beta_0 = 0{,}7$ gewählt. Die Prädiktion erfolgt mit dem schwachen Dynamikmodell gemäß Abschnitt 3.3.5. Im Folgenden wird die erste Testsequenz des HumanEva-II-Datensatzes betrachtet.

Im ersten Zeitschritt wird die Startpose als unbekannt angenommen. Es wird lediglich die ungefähre Position der Person im Raum vorgegeben. Beide Verfahren zeigen sich in der implementierten Version dazu imstande, automatisch die Initialpose für das Körper-Tracking zu ermitteln. Dies wird dadurch ermöglicht, dass es sich in beiden Fällen um globale Optimierungsverfahren handelt. Das Tracking kann innerhalb von ca. drei Zeitschritten die korrekte Pose ermitteln.

Abbildung 3.12 zeigt die Verläufe der Fehler in mm gemäß Gleichung (3.59) für ISA und EVP. Es wurde jeweils die gesamte Sequenz von 1225 Zeitschritten verarbeitet. Beide Verfahren sind in der Lage, sich von Fehlern zu erholen und die Person ohne Tracking-Verlust über die komplette Sequenz zu verfolgen. In Tabelle 3.1 sind Mittelwerte und

Standardabweichungen der Fehler zu sehen. Dabei werden zunächst die Abschnitte der Sequenz, die die einzelnen Aktionen beinhalten, separat betrachtet und anschließend die mittleren Fehler der gesamten Sequenz bestimmt. Der ISA-Algorithmus repräsentiert die Pose eines Zeitschrittes durch den Mittelwert des gesamten Partikelschwarms. Beim EVP wird das Individuum mit der besten Fitness herangezogen. Zum Vergleich damit wurde für das Ergebnis des ISA der Fehler der Pose mit der höchsten Gewichtung (MAP) berechnet. Abbildung 3.13 zeigt die Histogramme der korrekten, akzeptablen und falschen Posen für die einzelnen Aktionen sowie die gesamte Sequenz.

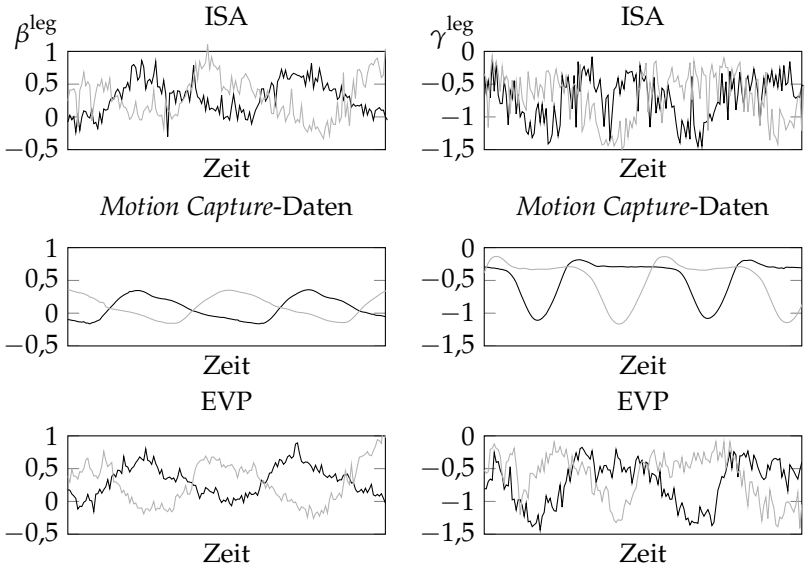


Abbildung 3.15. Ausschnitt der Verläufe einiger Zustandsgrößen des linken (grau) und rechten Beins (schwarz) für ISA, EVP und *Motion Capture*-Daten für zwei Schritte einer gehenden Person.

Die mittels ISA geschätzten Posen weisen über die gesamte Sequenz höhere Fehler als die Ergebnisse des EVP auf. Nur sehr wenige Posen des ISA werden als korrekt klassifiziert. Beim Gehen sind die Posen

überwiegend akzeptabel, während beim Joggen und Balancieren die meisten Posen fehlerhaft sind. Insgesamt liegt die Mehrzahl der Abweichungen oberhalb von 120 mm und somit im als falsch angesehenen Bereich. Beim EVP gelingt während der Geh-Phase überwiegend die Schätzung der korrekten Pose. Zwischenzeitlich kommt es zu größeren Abweichungen, was dadurch verursacht wird, dass die beiden Beine während eines Schrittes verwechselt werden. Bei den beiden anderen Aktionen verschlechtert sich die Genauigkeit. Insgesamt liegen die meisten Fehler im akzeptablen Bereich von 80 bis 120 mm.

Auch die Ergebnisse in [85] zeigen, dass für ISA deutlich mehr Partikel für ein erfolgreiches Tracking benötigt werden. Die dort durchgeführten Versuche für den ISA-Algorithmus mit lediglich 50 Partikeln ergeben Fehler in einem sehr ähnlichen Bereich wie in den hiesigen Experimenten. Um mittels ISA ähnliche Ergebnisse zu erreichen, wie sie in Tabelle 3.1 durch EVP erzielt werden, sind in [85] mit 100 Partikeln doppelt so viele erforderlich. Durch Verwendung von 200 Partikeln für ISA ergeben sich etwas bessere Ergebnisse als für EVP mit 50 Individuen.

Anhand der Fehlerverläufe ist zu erkennen, dass bei den ISA-Schätzungen häufige Fluktuationen (sog. *Jitter*) auftreten. Diese sind in der Literatur als typisches Problem des ISA bekannt [37]. In Abbildung 3.14 sind diese charakteristischen Schwankungen anhand dreier aufeinanderfolgender Zeitschritte veranschaulicht. Während beim ISA starke Fluktuationen im Verlauf der Posen zu sehen ist, weist der evolutionäre Algorithmus einen wesentlich glatteren Verlauf auf. In Abbildung 3.15 sind Ausschnitte der mittels ISA sowie EVP geschätzten Verläufe einiger Zustandsgrößen gezeigt. Es sind die Bewegungen der Oberschenkel nach vorne und hinten, β^{leg}, und die Beugungswinkel der Knie, γ^{leg}, abgebildet. Zum qualitativen Vergleich ist ein typischer Verlauf dieser Winkel, berechnet aus *Motion Capture*-Daten des HumanEva-I-Datensatzes, in der Mitte dargestellt. Auch anhand dieser Verläufe werden die wesentlich höhere Genauigkeit und die stark reduzierten Schwankungen der EVP-Ergebnisse ersichtlich.

In den folgenden Bildern werden geschätzte Posen von ISA und EVP einiger Zeitpunkte einander gegenübergestellt. Für ISA ist jeweils die mittlere Pose abgebildet, für EVP das Individuum mit der höchsten Gewichtung.

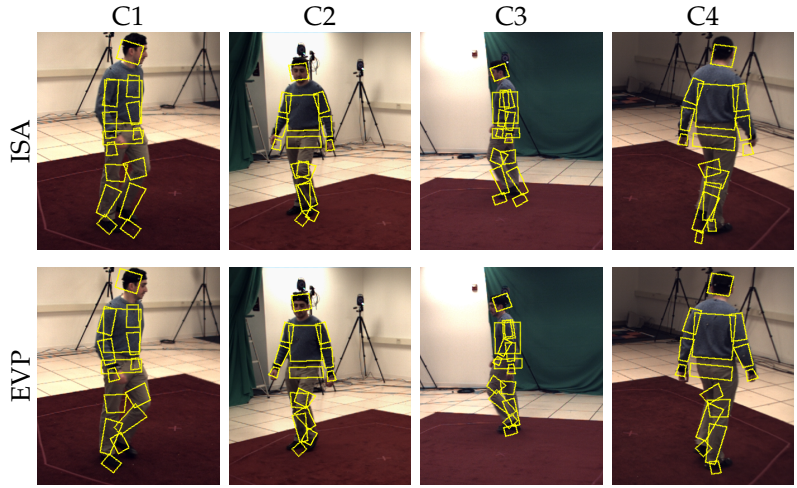


Abbildung 3.16. Tracking-Ergebnisse des ISA (oben) und EVP (unten) in einem Zeitschritt, während die Person geht. Der Fehler des ISA beträgt 95,62 mm, der des EVP 60,55 mm. Die Pose des EVP ist korrekt, die des ISA akzeptabel.

In den Abbildungen 3.16 und 3.17 sind zwei Beispiele während der Geh-Phase zu sehen. Der Fehler des ISA beträgt im ersten Fall 95,62 mm, die Pose wird demnach als größtenteils richtig akzeptiert. Anhand Abbildung 3.16 ist zu erkennen, dass beide Verfahren alle Körperteile überwiegend korrekt lokalisieren. ISA hat jedoch Schwierigkeiten, die Beine richtig zu schätzen. Da sich die Beine gerade kreuzen und einen geringen Abstand voneinander haben, kommt es in den Bildern zu Überlappungen und Mehrdeutigkeiten, die ISA Probleme bereiten. Dem EVP gelingt die korrekte Zuordnung der Beine. Die Arme werden vom EVP ebenso mit einer höheren Genauigkeit geschätzt, und es ergibt sich eine korrekte Pose mit einem Fehler von 60,55 mm. Im zweiten Beispiel, wie in Abbildung 3.17 zu sehen ist, platziert ISA die Beine richtig, jedoch treten große Abweichungen beim Schätzen der Arme auf. Der Fehler beträgt hier 121,47 mm, die Pose wird knapp als falsch klassifiziert. Beim EVP werden Arme und Beine korrekt geschätzt und ein sehr geringer Fehler von 52,87 mm erreicht.

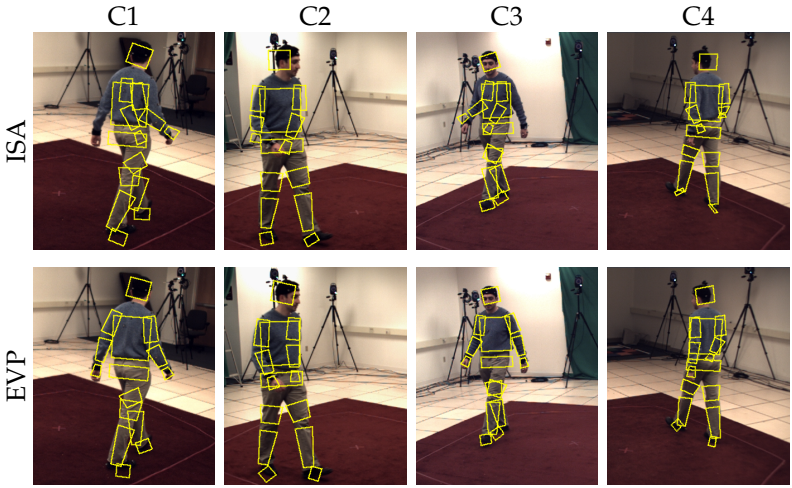


Abbildung 3.17. Tracking-Ergebnisse des ISA (oben) und EVP (unten) in einem Zeitschritt, während die Person geht. Der Fehler des ISA beträgt 121,47 mm, der des EVP 52,87 mm. Bei ISA wird der Körper in der Kurve zu spät gedreht.

In den nächsten beiden Beispielen joggt die Person. Die Posen beider Verfahren weisen beim Joggen größere Fehler auf als beim Gehen, in beiden Fällen werden überwiegend akzeptable Posen geschätzt (siehe Abbildung 3.13). Im Beispiel aus Abbildung 3.18 befindet sich die Person gerade in der Kurve. Beim ISA wird die Rotation des Körpers spät erkannt, weshalb in diesem Zeitbereich große Fehler auftreten. Auch für das Beispiel aus Abbildung 3.19 schätzt das ISA eine falsche Pose. Die Posen des EVP sind in beiden Beispielen akzeptabel mit Fehlern von 89,79 mm und 103,93 mm.

Abbildungen 3.20 und 3.21 zeigen Posen während des Balancierens. Bei dieser Aktion ergeben sich mittels ISA fast ausschließlich falsche Posen, während die Posen des EVP überwiegend akzeptabel sind. Im ersten Beispiel ergibt sich beim EVP eine knapp fehlerhafte Pose, bei welcher der rechte Arm der Person nicht korrekt platziert wird. Im zweiten Fall ist die EVP-Schätzung akzeptabel mit einer Abweichung von 102,11 mm.

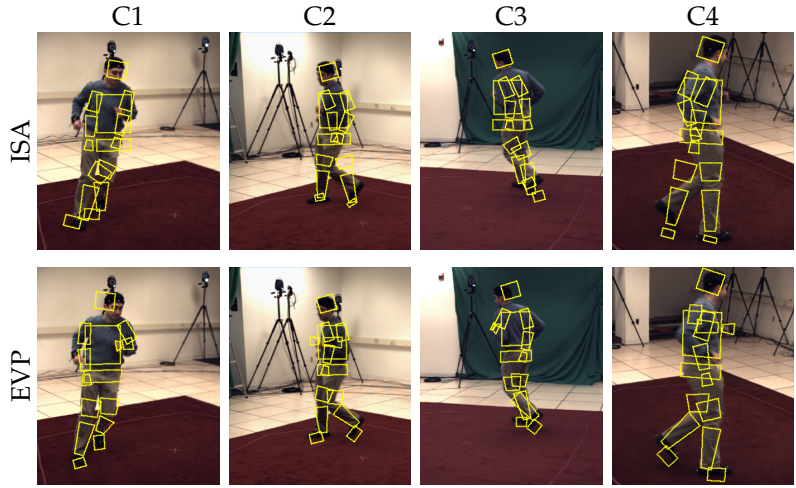


Abbildung 3.18. Tracking-Ergebnisse des ISA (oben) und EVP (unten) in einem Zeitschritt, während die Person joggt. Der Fehler des ISA beträgt 234,86 mm, der des EVP 89,79 mm.

Weitere typische Beispiele sind in Abbildung 3.22 zusammengefasst. Wie man erkennen kann, nimmt das ISA häufig eine falsche Zuordnung der Beine und Arme vor – beide Beine werden auf denselben Bildbereich platziert. Bei den Posen des EVP tritt dieses Verhalten wesentlich seltener auf. Auch bei den prinzipiell richtig lokalisierten Körperbereichen liegt beim EVP eine höhere Genauigkeit vor.

3.4.4. Dynamikmodell

Im vorigen Abschnitt sind im Fehlerverlauf des EVP temporär große Abweichungen der geschätzten von der korrekten Pose zu sehen. Wie bereits erwähnt wurde, kann dies in Fällen auftreten, wenn während eines Schrittes das Schwung- mit dem Standbein verwechselt wird. Dieser Sachverhalt ist in Abbildung 3.23 oben für den 176-ten Zeitschritt zu sehen. Auf den ersten Blick erscheint die Pose akzeptabel. In den Bildern der zweiten und vierten Kamera ist jedoch die Verwechslung der Beine sichtbar. Dennoch ergibt sich bei den Projektionen der Pose auf die ein-

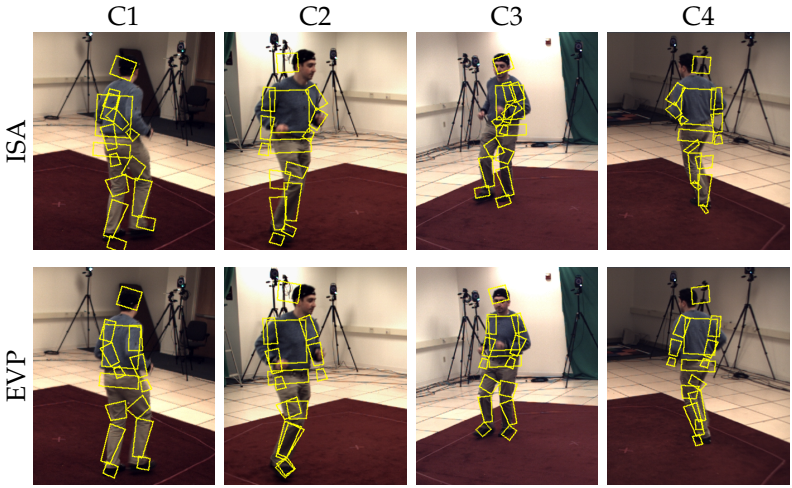


Abbildung 3.19. Tracking-Ergebnisse des ISA (oben) und EVP (unten) in einem Zeitschritt, während die Person joggt. Der Fehler des ISA beträgt 202,53 mm, der des EVP 103,93 mm.

zelnen Kameraperspektiven eine gute Abdeckung der Person und somit eine hohe Gewichtung der geschätzten Pose. Die Verwechslung beginnt gerade dann, wenn sich die Beine kreuzen, da ab diesem Zeitpunkt durch die Variation der Individuen beide Hypothesen erreicht und hoch gewichtet werden. Alleine basierend auf dem vorliegenden Bildmaterial lässt sich diese Mehrdeutigkeit somit nur schwer korrekt auflösen. Anhand dieses Beispiels wird ersichtlich, warum in solchen Fällen die Posenschätzung misslingen kann. Um hierfür Abhilfe zu schaffen, wird das erweiterte Dynamikmodell gemäß Abschnitt 3.3.5 herangezogen, bei dem im Falle von Überlappungen vergangene Schätzwerte zur Prädiktion einiger Zustandskomponenten verwendet werden. In Abbildung 3.24 sind die Verläufe der Fehler mit schwachem und erweitertem Dynamikmodell während des Geh-Phase der im vorigen Abschnitt betrachteten Sequenz abgebildet. Bei Verwendung des schwachen Modells ergibt sich ein mittlerer Fehler von 78,94 mm $\pm$ 29,16 mm, mit dem erweiterten 74,90 mm $\pm$ 13,40 mm. Im Bereich des 176-ten Zeitschrittes ist eine große Abweichung zwischen den beiden Fehlern zu sehen. Der Fehler

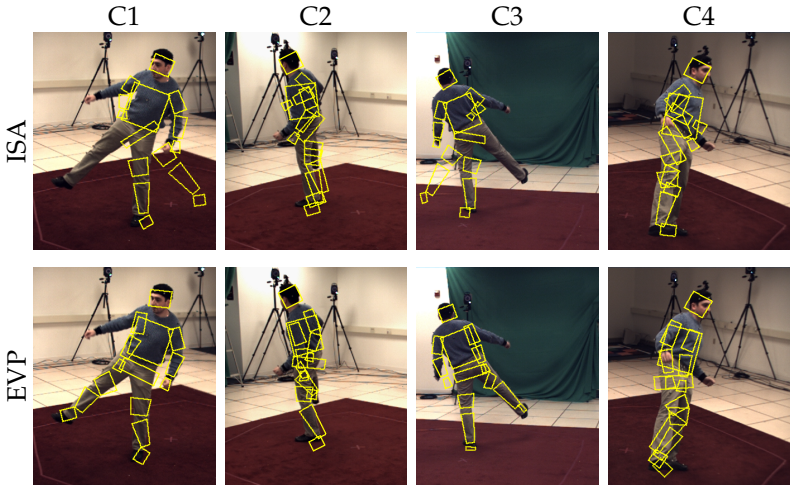


Abbildung 3.20. Tracking-Ergebnisse des ISA (oben) und EVP (unten) in einem Zeitschritt, während die Person balanciert. Der Fehler des ISA beträgt 267,12 mm, der des EVP 127,31 mm.

des schwachen Dynamikmodells wird für eine gewisse Zeit sehr groß, während mit dem erweiterten Modell durchgehend die korrekte Pose geschätzt wird. Die entsprechenden Kamerabilder beider Verfahren zu diesem Zeitpunkt sind in Abbildung 3.23 zu sehen. Betrachtet man lediglich die Bilder der ersten und dritten Kamera, sehen die Posen ähnlich aus. Ohne das erweiterte Dynamikmodell ergibt sich allerdings ein Fehler von 182,70 mm, während es mit ihm lediglich 61,77 mm sind. Dies liegt daran, dass die Abweichung von der wahren dreidimensionalen Pose im ersten Fall aufgrund der Verwechslung der Beine groß ist, während mit dem erweiterten Modell eine korrekte Zuordnung der Beine erfolgt. Im Bereich um den 250-ten Zeitschritt haben beide Verfahren Schwierigkeiten, wobei das erweiterte Modell innerhalb des akzeptablen Bereichs bleibt, während das einfache falsche Posen schätzt.

Schließlich wird eine Sequenz mit lediglich drei Kameraperspektiven aus dem HumanEva-I-Datensatz betrachtet. Ein Ausschnitt daraus ist in Abbildung 3.25 zu sehen, wobei der Verlauf eines Geh-Schrittes durch vier Zeitpunkte am Beispiel einer Kameraperspektive veranschaulicht

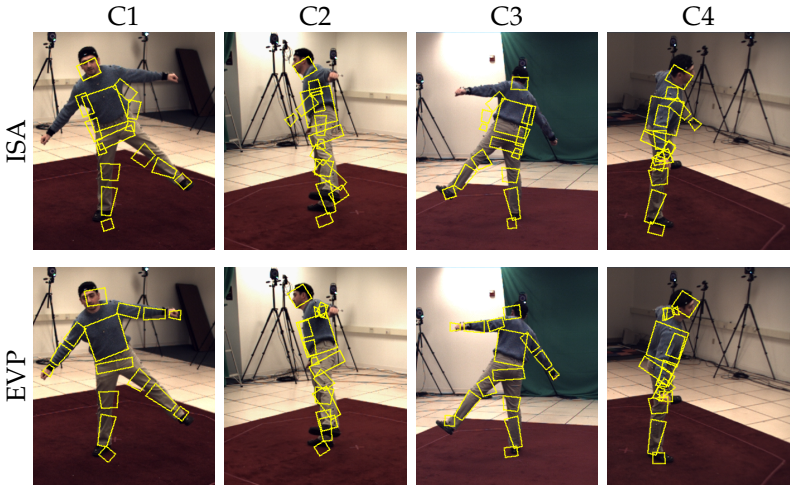


Abbildung 3.21. Tracking-Ergebnisse des ISA (oben) und EVP (unten) in einem Zeitschritt, während die Person balanciert. Der Fehler des ISA beträgt 217,86 mm, der des EVP 102,11 mm.

wird. Dabei sind zwei der drei Kameras direkt gegenüber voneinander positioniert, so dass beide sehr ähnliche Informationen bezüglich der Kanten- und Silhouettenbilder liefern. Dadurch wird die Auflösung von Mehrdeutigkeiten bei sich überdeckenden Körperteilen weiter erschwert. Auch hier kommt es mit dem einfachen Dynamikmodell beim Kreuzen der Beine zu Fehlern bei der Zuordnung. In Abbildung 3.26 sind die zugehörigen Verläufe einiger Beinwinkel in einem größeren Zeitfenster dargestellt. Daraus wird ersichtlich, dass es beim schwachen Modell nach dem Kreuzen der Beine zu einer Verwechslung von Schwung- und Standbein kommt, so dass sich zweimal hintereinander das selbe Bein nach vorne bewegt. Durch Verwendung des erweiterten Dynamikmodells gelingt es hingegen, die Mehrdeutigkeiten aufgrund der sich überlappenden Körperbereiche aufzulösen und eine korrekte Zuordnung von Stand- und Schwungbein vorzunehmen. Es ergibt sich eine sehr ähnliche Form der Verläufe wie bei den zum Vergleich dargestellten *Motion Capture*-Daten.

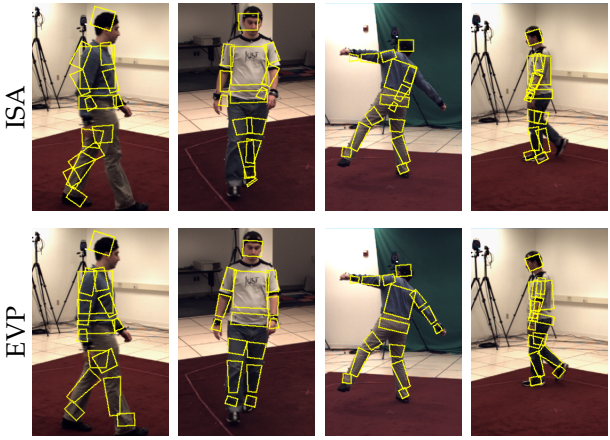


Abbildung 3.22. Tracking-Ergebnisse des ISA (oben) und EVP (unten) in verschiedenen Sequenzen.

3.4.5. Diskussion

Sowohl mit dem *Interacting Simulated Annealing*-Partikelfilter als auch dem Evolutionären Posen-Tracking gelingt ein robustes Körper-Tracking über einen längeren Zeitraum. Beide Verfahren sind in der Lage, selbstständig die Pose zu initialisieren und sich von Fehlern zu erholen.

Mittels EVP wird jedoch ein erfolgreiches Körper-Tracking mit deutlich weniger Individuen erreicht als es das ISA für vergleichbare Ergebnisse benötigt. Die für das ISA typischen Fluktuationen, bei denen die geschätzten Posen von einem Zeitpunkt zum nächsten große Sprünge aufweisen, werden beim EVP weitgehend vermieden.

Die Stärken des EVP sind hauptsächlich durch die verwendeten Evolutionsfaktoren Variation und Selektion begründet. Das Zusammenspiel von Rekombination und Mutation resultiert in einer großen Artenvielfalt und einer effizienten Durchsuchung des Zustandsraumes. Bereits in [27] wurde gezeigt, dass die Einführung eines *Crossover*-Operators deutliche Verbesserungen bei der Posenschätzung bewirkt. In dieser Arbeit ist die Rekombination die Hauptverantwortliche für die Erforschung des Zustandsraumes. Es werden dabei nicht nur zwei, sondern mehrere

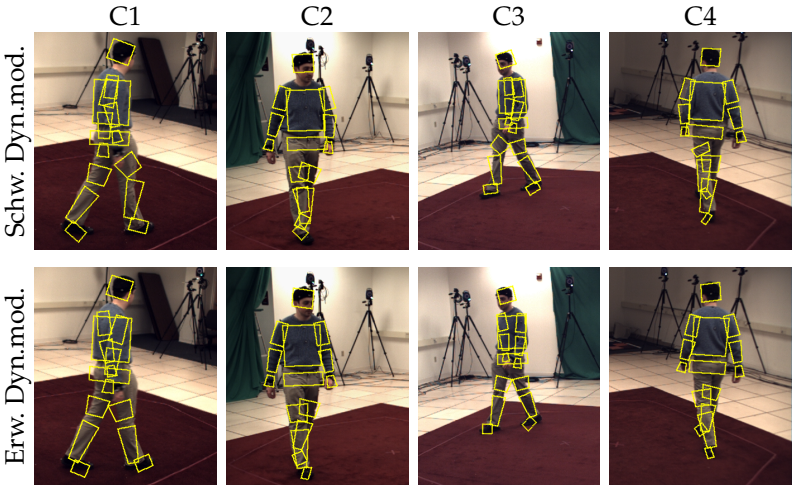


Abbildung 3.23. Tracking-Ergebnisse mit schwachem (oben) und erweitertem Dynamikmodell (unten) in einem Zeitschritt. Beim schwachen Dynamikmodell werden rechtes und linkes Bein vertauscht, was anhand der Bilder von C2 und C4 ersichtlich wird. Der Fehler beim einfachen Dynamikmodell beträgt 182,70 mm, beim erweiterten 61,77 mm .

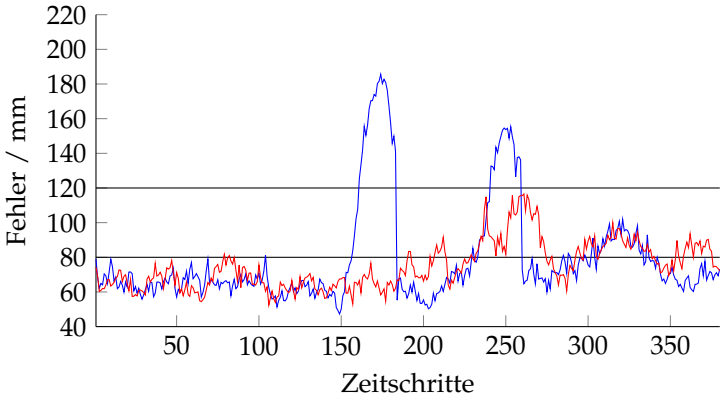


Abbildung 3.24. Verläufe des Fehlers des schwachen (blau) und des erweiterten Dynamikmodells (rot) während der Geh-Phase der Sequenz.

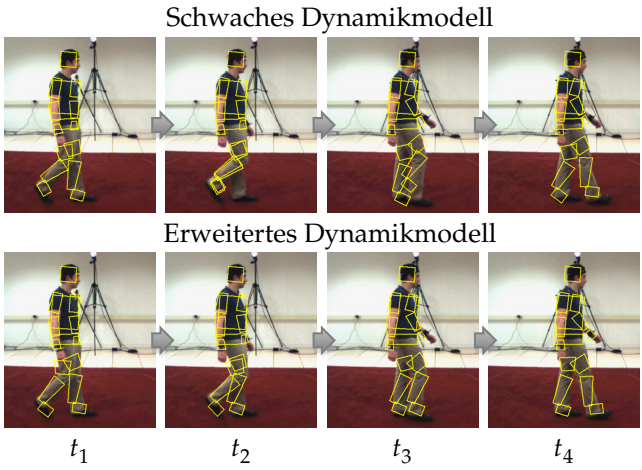


Abbildung 3.25. EVP mit einfachem (oben) und erweitertem (unten) Dynamikmodell in vier Zeitschritten t_1 bis t_4.

Eltern durchmischt. Durch die Kombination von Zustandsgrößen verschiedener Eltern können komplett neuartige Posen entstehen. Dadurch wird eine wesentlich größere Artenvielfalt erreicht, woraus auch bei schwacher Streuung bei Mutation und Prädiktion große Variationen der Kinderpopulation resultieren. Während die Mutation beim EVP eine geringere Rolle spielt, ist das ISA auf die Partikelstreuung in den einzelnen *Annealing*-Stufen angewiesen, um Variation in den Partikelschwarm zu bringen. Beim EVP werden dagegen nur zufällig einige Zustandskomponenten mutiert, da die Mutation hier eher die Rolle der Feinabstimmung übernimmt. Durch die Durchmischung der vorteilhaften Eigenschaften der Eltern bei der Rekombination werden gerade jene neuen Bereiche des Zustandsraumes erschlossen, die vielversprechend erscheinen. Damit gelingt es, große Bereiche des Zustandsraumes auf intelligente Art zu durchsuchen: Ohne den Rekombinationsoperator wären mehr Individuen bei stärkerer Mutation nötig, um denselben Bereich des Zustandsraumes abzudecken. Mit diesem Vorgehen können daher auch mit einer kleinen Population sehr gute Ergebnisse erzielt werden.

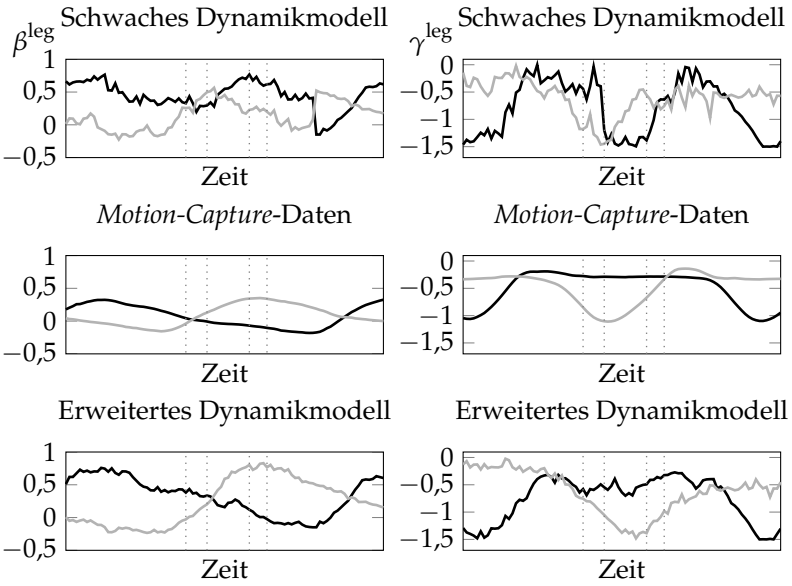


Abbildung 3.26. Verläufe von Beinwinkeln (links und rechts) für schwaches Dynamikmodell, erweitertes Dynamikmodell und *Motion Capture*-Daten. Die Zeitpunkte t_1 bis t_4 sind als gepunktete Linien angedeutet.

Dies erklärt die Stärke des evolutionären Suchalgorithmus und wird aus dem Vergleich mit dem ISA ersichtlich.

Eine Schwierigkeit beim Körper-Tracking ist das Auftreten von Mehrdeutigkeiten durch sich überdeckende Körperteile in einzelnen Kameraperspektiven. Diese erschweren die Posenschätzung vor allem im Falle weniger Kameras stark. Dies kann beispielsweise dazu führen, dass bei Geh- oder Lauf-Bewegungen die Beine nach dem Kreuzen miteinander verwechselt werden. Um hierfür Abhilfe zu schaffen, wurde ein erweitertes Dynamikmodell vorgeschlagen und für Geh-Bewegungen untersucht. Hiermit konnte im Falle von Ambiguitäten eine deutliche Verringerung der Fehler erreicht werden.

4. Aktivitätserfassung basierend auf Merkmalstrajektorien

4.1. Einleitung

Das modellbasierte Körper-Tracking liefert mit Posenverläufen eine sehr reichhaltige Bewegungsrepräsentation. Kenntnisse über die Pose eines Menschen und der Bewegung einzelner Körperteile erlauben eine detaillierte Analyse der ausgeführten Aktivitäten. Das Körper-Tracking ist jedoch sehr aufwändig und bisher nicht in beliebigen, unkontrollierten Szenarien einsetzbar. Gerade bei Aufnahmen einzelner Kameras besteht noch großer Forschungsbedarf. Häufig ist man außerdem gar nicht an den Details der Ausführung von Bewegungen interessiert, sondern möchte lediglich erkennen, welche Aktionen in einer Szene auftreten.

Für solche Anwendungen eignen sich Methoden der Aktivitätserkennung, die auf der Extraktion bestimmter Bild- bzw. Videomerkmale aus Bildsequenzen basieren. Der Aufwand der Merkmalsextraktion dieser Ansätze ist wesentlich geringer als bei der Posenschätzung. Es wird deutlich weniger Vorwissen vorausgesetzt, was eine höhere Vielseitigkeit bezüglich der Einsatzmöglichkeiten mit sich bringt. Diese Verfahren sind außerdem sehr flexibel was die Anzahl und Anordnung der Sensoren sowie die Umgebung betrifft. Gerade in Szenarien, bei denen nur Aufnahmen einer Kamera mit dynamischen Hintergründen, Kamerabewegungen und Interaktionen zwischen Personen vorliegen, sind Methoden der merkmalsbasierten Aktionserkennung geeignet, da sie ohne Weiteres zum Einsatz kommen können.

Das Ziel in diesem Kapitel besteht in der Gewinnung von Informationen über den dynamischen Verlauf lokaler Merkmale in Form von Merkmals-trajektorien. Dies wird durch ein Tracking von Merkmalen realisiert, die besonders charakteristisch für die in einer Sequenz auf-

tretenden Bewegungen sind. Die Merkmale sollen über einen längeren Zeitraum verfolgt werden, um auch komplexe Ereignisse in Videos beschreiben zu können. Die resultierenden Trajektorien werden durch verschiedene Deskriptoren repräsentiert. Diese beschreiben Textur und Bewegung in lokalen Umgebungen der Trajektorien. Die entwickelte Methodik zielt auf den Einsatz in Szenarien ab, in dem komplizierte Aktivitäten auftreten, bei Aufnahmen, die eine ausreichend hohe Auflösung besitzen, um ein robustes Tracking zu ermöglichen.

In Abschnitt 4.2 werden zunächst die benötigten Grundlagen dargestellt. Dabei wird auf für diese Arbeit relevante Methoden der Detektion und Repräsentation von Merkmalen sowie des bildbasierten Trackings eingegangen. Anschließend wird in Abschnitt 4.3 der entwickelte Algorithmus zur Gewinnung der Merkmalstrajektorien erläutert, welcher schließlich in Abschnitt 4.4 in Versuchen angewandt wird. Eine quantitative Bewertung des Trackings wird bezüglich der Eignung zur Aktionserkennung in Kapitel 5 durchgeführt.

4.2. Grundlagen

Im Folgenden werden die benötigten Grundlagen für die merkmalsbasierte Bewegungserfassung gegeben. Zunächst wird auf die Detektion und Repräsentation von Merkmalen in Bildfolgen eingegangen. Anschließend werden einige Methoden der Merkmalsextraktion genauer vorgestellt, die in dieser Arbeit zum Einsatz kommen und schließlich wird auf das bildbasierte Tracking mittels *Mean Shift* eingegangen.

4.2.1. Detektion und Deskription von Interessenspunkten in Bildfolgen

Methoden der Videoanalyse, die auf lokalen Merkmalen basieren, erfordern geeignete Merkmalsdetektoren. Dabei sollen Punkte in einer Bildfolge gefunden werden, die interessante Ereignisse repräsentieren. Diese werden als *Spatio-Temporal Interest Points* (STIP) bezeichnet und sollen an Punkten (x, y, t) detektiert werden, die in örtlicher und zeitlicher Dimension Variationen des Bildinhaltes aufweisen. Im Bereich der merkmalsbasierten Videoanalyse wurde bereits eine Vielzahl von

Ansätzen zur Detektion solcher STIPs vorgeschlagen. Viele davon sind Erweiterungen von Merkmalsdetektoren in Einzelbildern auf Bildsequenzen.

In einer sehr frühen Arbeit in diesem Bereich haben Laptev und Lindeberg [56] den Harris-Eckendetektor [42] auf den dreidimensionalen Fall erweitert. Damit werden Punkte detektiert, welche „Ecken" in Orts- und Zeitrichtung darstellen, da diese häufig interessante Ereignisse repräsentieren. Die Detektion erfolgt im Orts-Zeit-Skalenraum. Für den örtlichen Skalenfaktor σ und den zeitlichen Skalenfaktor τ wird das Eingangsbild $i(\mathbf{x}, t)$ mit einem Gauß-Kern

$$g(\mathbf{x}, t, \sigma, \tau) = \frac{1}{\sqrt{(2\pi)^3 \sigma^4 \tau^2}} \cdot \exp\left(-\frac{x^2 + y^2}{2\sigma^2} - \frac{t^2}{2\tau^2}\right) \tag{4.1}$$

gefiltert. Die Skalenraum-Darstellung lautet

$$k(\mathbf{x}, t, \sigma, \tau) = g(\mathbf{x}, t, \sigma, \tau) * i(\mathbf{x}, t)\,, \tag{4.2}$$

wobei $*$ den dreidimensionalen Faltungsoperator darstellt. Die Detektion erfolgt mit Hilfe der Matrix der zweiten Momente, die in einem lokalen Bereich der Bildfolge, gewichtet mit einem Gauß-Fenster, gemittelt wird

$$\mathbf{H}(\mathbf{x}, t, \sigma, \tau) = g(\mathbf{x}, t, \sigma, \tau) * \begin{bmatrix} k_x^2(\cdot) & k_x(\cdot)k_y(\cdot) & k_x(\cdot)k_t(\cdot) \\ k_x(\cdot)k_y(\cdot) & k_y^2(\cdot) & k_y(\cdot)k_t(\cdot) \\ k_x(\cdot)k_t(\cdot) & k_y(\cdot)k_t(\cdot) & k_t^2(\cdot) \end{bmatrix}. \tag{4.3}$$

Die partiellen Ableitungen erster Ordnung von $k(\mathbf{x}, t, \sigma, \tau)$ lauten dabei $k_x(\cdot) = \frac{\partial}{\partial x} k(\mathbf{x}, t, \sigma, \tau)$ etc.

Es sollen Punkte detektiert werden, an denen $\mathbf{H}$ große Eigenwerte besitzt. Die Bildung des Detektors erfolgt mittels Spur und Determinante von $\mathbf{H}$

$$r^{\text{Harris3D}}(\mathbf{x}, t, \sigma, \tau) = \det(\mathbf{H}(\mathbf{x}, t, \sigma, \tau)) - \kappa \text{spur}^3(\mathbf{H}(\mathbf{x}, t, \sigma, \tau)) \tag{4.4}$$

und die STIPs werden an lokalen Maxima von $r^{\text{Harris3D}}(\mathbf{x}, t, \sigma, \tau)$ lokalisiert. In [56] werden σ und τ durch automatische Skalenselektion gewählt.

Dollár [29] konstatiert, dass direkte dreidimensionale Erweiterungen von zweidimensionalen Merkmalsdetektoren ungeeignet für die Detektion von STIPs sind, da die Zeit als dritte Dimension andere Eigenschaften als die örtlichen Dimensionen besitzt und somit gesondert behandelt werden muss. Gemäß den Beobachtungen in [29] sind dreidimensionale Ecken, wie sie mit Laptevs Detektor aus Gleichung (4.4) gefunden werden, für Bewegungsabläufe geeignet, die durch eine Umkehrung der Bewegungsrichtung charakterisiert sind. Dies trifft v. a. auf einfache Aktionen wie „Winken“ oder „Gehen“ zu, wie sie auch in [56] betrachtet werden. Aktionen, die eher durch allmähliche Bewegungsabläufe gekennzeichnet sind, enthalten allerdings kaum solche „Ecken“, was zur Folge hat, dass zu wenige Merkmale detektiert werden. In [29] werden u. a. Gesichtsausdrücke und das Verhalten von Ratten untersucht. Diese gehören zu Bewegungsarten, bei denen der Detektor (4.4) ungeeignet ist.

Dollár stellt stattdessen einen Detektor vor, der eher zur Detektion von zu vielen als von zu wenigen Interessenspunkten neigt. Als Detektorfunktion wird eine Filterung der Bildfolge mit einem Gauß-Filter $g(\mathbf{x}, \sigma)$ in örtlicher und einem komplexen Gabor-Wavelet in zeitlicher Richtung durchgeführt. Die Detektorfunktion ergibt sich damit zu

$$\begin{aligned} r^{\text{periodic}}(\mathbf{x}, t, \sigma, \tau) = &\left(i(\mathbf{x}, t) * g(\mathbf{x}, \sigma) * h_{\text{ev}}(t, \tau, \omega)\right)^2 \\ &+ \left(i(\mathbf{x}, t) * g(\mathbf{x}, \sigma) * h_{\text{od}}(t, \tau, \omega)\right)^2 , \end{aligned} \tag{4.5}$$

mit

$$h_{\text{ev}} = -\cos(2\pi t\omega)\, \mathrm{e}^{-t^2/\tau^2} , \tag{4.6}$$

$$h_{\text{od}} = -\sin(2\pi t\omega)\, \mathrm{e}^{-t^2/\tau^2} , \tag{4.7}$$

wobei $\omega = \frac{4}{\tau}$ verwendet wird. Der Detektor reagiert am stärksten auf periodische Bewegungen, ist aber nicht auf diese beschränkt. Alle Bereiche, die örtliche Strukturen aufweisen und komplexe Bewegungen beinhalten, resultieren in einer Antwort des Detektors. Auf rein translatorische Bewegungen erfolgt dagegen keine oder nur eine sehr schwache Reaktion.

Es existieren eine Menge weiterer STIP-Detektoren. Willems et al. [103] verwenden die Determinante der dreidimensionalen Hesse-Matrix, um

einen Detektor zu erhalten, welcher skaleninvariant ist und mehr Merkmale als Laptevs Detektor liefert. Diese Methode ähnelt einer dreidimensionalen Variante des *Speeded-Up Robust Features* (SURF)-Detektors [7] (siehe Abschnitt 4.2.3) und ist effizient aufgrund der Verwendung von Rechteckfiltern.

In [97] werden verschiedene STIP-Detektoren für die Aktionserkennung untersucht und in Kombination mit mehreren Deskriptoren miteinander verglichen. Betrachtet werden der periodische Detektor (4.5), die dreidimensionalen Harris- und Hesse-Detektoren sowie die Verwendung von Punkten, die auf einem dichten, gleichmäßigen Raster ausgewählt werden. Der periodische Detektor übertrifft die beiden anderen diskutierten Methoden, mit Ausnahme bei sehr einfachen Bewegungsarten, bei denen der 3D-Harris-Detektor etwas bessere Ergebnisse liefert. Es sei angemerkt, dass die dichten Punkte insgesamt am besten abschneiden, jedoch in einer sehr großen Anzahl an Merkmalen und somit einem hohen Verarbeitungsaufwand resultieren. Auch in [81] werden verschiedene Detektoren verglichen: der periodische Detektor, der 3D-Harris-Detektor, dreidimensionale Gabor-Filter und DoG-Filter (*Difference of Gaussians*) in Orts- und Zeitrichtung [19]. Die Detektoren werden als Baustein eines Algorithmus zur Aktionserkennung evaluiert. Der periodische Detektor übertrifft die anderen deutlich bezüglich der Erkennungsrate, obwohl er nur für einen Skalenfaktor angewandt wird. Er ist außerdem mit einem wesentlich geringeren Rechenaufwand verbunden.

Die bisher vorgestellten STIP-Detektoren wurden zum Auffinden isolierter Merkmale entwickelt. Die Deskriptoren der Merkmale werden in Volumen um diese Punkte berechnet. Diese Volumen, auch als Kuboide bezeichnet, haben eine bestimmte Ausdehnung, abhängig von den zugehörigen Skalenfaktoren σ und τ. Deskriptoren werden häufig aus Bildgradienten oder optischem Fluss ermittelt. In [29] werden außerdem Deskriptoren direkt aus normierten Grauwerten gebildet. In [58] werden Histogramme orientierter Gradienten (HOG) und des optischen Flusses (HOF) zur Deskription lokaler STIPs vorgeschlagen. Der HOG-Deskriptor wurde ursprünglich als globales Merkmal zur Objekterkennung entwickelt und wird im folgenden Abschnitt erläutert. In [97] erweist sich v. a. die Kombination aus Gradienten- und Flussde-

skriptoren als sehr gutes Merkmal zur Aktionserkennung. Weiterhin existieren Erweiterungen etablierter Deskriptoren für Bildmerkmale, wie dreidimensionale SIFT- [80] oder SURF-Deskriptoren [103].

4.2.2. Orientierungshistogramme von Gradienten und optischem Fluss

In [25] schlagen Dalal und Triggs sog. *Histogramme orientierter Gradienten (HOG)* als robuste Merkmale zur Objekterkennung vor und evaluieren sie am Beispiel der Personendetektion. Der HOG-Deskriptor beschreibt die lokale Verteilung von Bildgradienten und repräsentiert Form und Aussehen im Bild enthaltener Objekte [25].

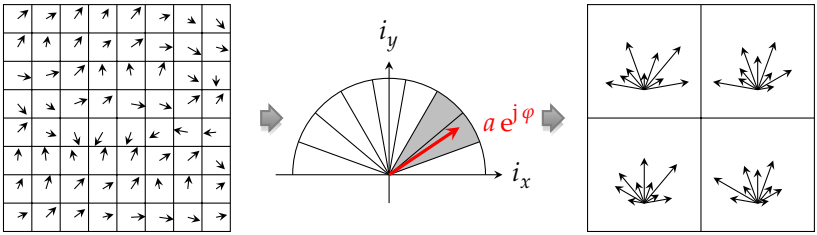


Abbildung 4.1. Veranschaulichung der Bestimmung und des Aufbaus des vorzeichenlosen (*unsigned*) HOG-Deskriptors.

Die Merkmale werden im gesamten Bildbereich ermittelt. Dazu wird das Bild in Zellen eingeteilt und für jede Zelle ein Histogramm der Gradientenorientierungen bestimmt. Um Kontrastunterschiede auszugleichen, erfolgt eine lokale Normierung größerer Bildbereiche, indem mehrere Zellen zu Blöcken zusammengefasst und diese lokal normiert werden.

Zur Bestimmung der Bildgradienten in horizontaler und vertikaler Richtung erfolgt eine Differentiation mit den Filtern $[-1,0,1]$ bzw. $[-1,0,1]^{\mathrm{T}}$ ohne Glättung. HOG-Deskriptoren können mit Beachtung des Vorzeichens der Gradientenrichtung (*signed HOG*) oder ohne (*unsigned* HOG) bestimmt werden. Bei der Personendetektion eignet sich laut [25] die vorzeichenlose Variante besser, dies gilt aber nicht für alle Objektarten. Abbildung 4.1 veranschaulicht die Bestimmung und

den Aufbau der HOG-Merkmale beispielhaft ohne Beachtung des Vorzeichens der Gradientenrichtung. Die Standardumsetzung verwendet 9 vorzeichenlose Winkelabschnitte.

In [26] wird im Speziellen die Personendetektion in Videos behandelt, und es wird zusätzlich zu den Gradientenmerkmalen Information über die in Videos auftretende Bewegung hinzugezogen. Dadurch erreichen die Autoren eine bessere Personendetektion, da neben dem Erscheinungsbild von Menschen auch typische Bewegungen berücksichtigt werden. Es werden Merkmale gesucht, die für Personen charakteristische Bewegungen modellieren können und auch bei Kamera- und Hintergrundbewegung robust funktionieren. Ein Problem bei Kamera- und Hintergrundbewegung ist, dass die meisten Bewegungsmerkmale absolute Bewegung modellieren. In [26] wird als Lösung vorgeschlagen, differenzierten optischen Fluss zu betrachten, um diese Einflüsse zu verringern. Dazu werden im Wesentlichen zwei neue Merkmale entwickelt – jeweils in mehreren Varianten.

Sei $f^x(\mathbf{x})$ der optische Fluss eines Bildpaares in horizontaler und $f^y(\mathbf{x})$ in vertikaler Richtung. Beim ersten Merkmal werden die beiden Fluss-Komponenten separat voneinander behandelt und jeweils nach x und y differenziert. Für die x-Komponente ergibt sich

$$\frac{\partial}{\partial x} f^x(\mathbf{x}) \coloneqq f^x_x(\mathbf{x}) \,, \qquad \frac{\partial}{\partial y} f^x(\mathbf{x}) \coloneqq f^x_y(\mathbf{x}) \,. \tag{4.8}$$

Aus den beiden Ableitungen in Gleichung (4.8) werden Gradientenbetrag und -orientierung berechnet und nach dem Schema des HOG-Deskriptors ein Merkmalsvektor $\mathbf{h}^{\mathrm{MBH}_x}$ gebildet. Das gleiche Vorgehen wird auf das horizontale Flussfeld $f^y(\mathbf{x})$ angewandt, um das Merkmal $\mathbf{h}^{\mathrm{MBH}_y}$ zu erhalten. Diese Merkmale werden von den Autoren in [26] als *Motion Boundary*-Histogramme (MBH) bezeichnet.

Kombiniert man die obigen Ableitungen auf andere Weise und fügt je die nach x bzw. y abgeleiteten Flusskomponenten zusammen, erhält man sog. *Internal Motion*-Histogramme (IMH).

Die hier erläuterten Histogramm-Merkmale werden auch für die Deskription *lokaler* Merkmale eingesetzt [58]. In diesem Fall werden die Deskriptoren nicht global für einen gesamten Bildbereich, sondern lediglich in lokalen Nachbarschaften detektierter Merkmale berechnet.

HOG-Deskriptoren und Histogramme von optischem Fluss (HOF) haben sich im Bereich der Aktionserkennung bereits als sehr nützlich erwiesen [58]. Laptev et al. [58] bestimmen HOG- und HOF-Deskriptoren in Kuboiden um detektierte STIPs. Die Kuboide werden in $n_x \times n_y \times n_\tau$ Zellen unterteilt und in jeder Zelle werden Histogramme der Orientierungen der Gradienten und des optischen Flusses gebildet. Die normierten Histogramme der einzelnen Zellen werden zu HOG- und HOF-Merkmalsvektoren zusammengefügt. Eine übliche Zelleneinteilung ist z. B. $n_x = n_y = 3$, $n_\tau = 2$ [58, 98]. In [98, 99] werden zur Deskription von Merkmalstrajektorien HOG-, HOF- und MBH-Deskriptoren mit großem Erfolg eingesetzt, worauf in Abschnitt 4.3.4 eingegangen wird.

4.2.3. Speeded-Up Robust Features – SURF

Der SURF-Algorithmus [7] beschreibt eine Methode zur Detektion und Deskription lokaler, skalierungs- und rotationsinvarianter Bildmerkmale. Diese ähneln den *Scale Invariant Feature Transform* (SIFT)-Merkmalen [59], können allerdings mit deutlich geringerem Aufwand berechnet werden.

Der SURF-Algorithmus detektiert „*Blob*"-artige Strukturen mit Hilfe der Hesse-Matrix. Die Hesse-Matrix für einen Punkt $\mathbf{x}$ und den Skalenfaktor σ lautet

$$\mathbf{H}(\mathbf{x}, \sigma) = \begin{bmatrix} k_{xx}(\mathbf{x}, \sigma) & k_{xy}(\mathbf{x}, \sigma) \\ k_{xy}(\mathbf{x}, \sigma) & k_{yy}(\mathbf{x}, \sigma) \end{bmatrix}, \tag{4.9}$$

wobei das Grauwertbild $i(\mathbf{x})$ mit Gauß-Filtern gemäß

$$k_{xx}(\mathbf{x}, \sigma) = i(\mathbf{x}) * \frac{\partial^2}{\partial x^2} g(\mathbf{x}, \sigma) \tag{4.10}$$

usw. gefiltert wird. Bei der praktischen Umsetzung werden die Gauß-Filter durch Rechteckfilter (sog. *Box*-Filter) angenähert [7]. Die so gefilterten Bilder werden als $\tilde{k}(\mathbf{x}, \sigma)$ bezeichnet. Die Detektorantwort ergibt sich als Determinante der approximierten Hesse-Matrix $\tilde{\mathbf{H}}(\mathbf{x}, \sigma)$ zu

$$r^{\text{SURF}}(\mathbf{x}, \sigma) = \det(\tilde{\mathbf{H}}(\mathbf{x}, \sigma)) = \tilde{k}_{xx}(\mathbf{x}, \sigma)\tilde{k}_{yy}(\mathbf{x}, \sigma) - (w\tilde{k}_{xy}(\mathbf{x}, \sigma))^2 . \tag{4.11}$$

Das Gewicht w wird für die Energieerhaltung der angenäherten Gauß-Kerne benötigt.

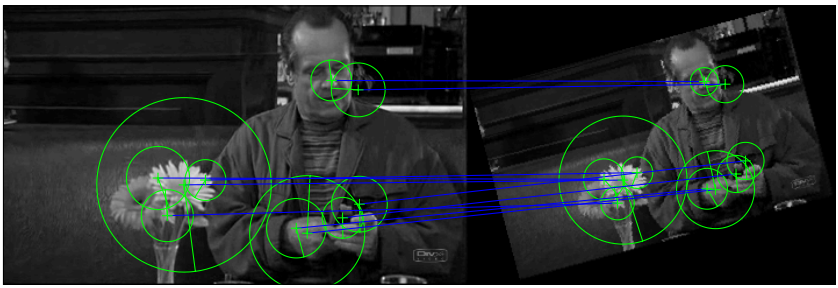


Abbildung 4.2. Beispiel: Korrespondenzen von SURF-Merkmalen anhand eines Bildes aus dem *Hollywood Human Actions* (HOHA)-Datensatz [58].

Die Rechteckfilterung entspricht der Summe der Bildpunkte, an deren Positionen das Filter den Wert eins annimmt. Die Faltung kann dadurch mittels Integralbildern realisiert werden, was dazu führt, dass der Berechnungsaufwand unabhängig von der Filtergröße ist. Das Integralbild eines Bildes i an den Punktkoordinaten (x, y) berechnet sich gemäß

$$i_\Sigma(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} i(x, y) \tag{4.12}$$

und gibt die Fläche des Rechtecks zwischen dem Punkt (x, y) und dem Ursprung des Bildkoordinatensystems an. Mit Integralbildern kann die Fläche eines beliebigen Rechtecks mit lediglich vier Additionen berechnet werden. Laut [7] lassen sich mit dieser Vereinfachung vergleichbare oder sogar bessere Ergebnisse erzielen als mit diskretisierten Gauß-Filtern. Mit diesem Vorgehen kann die Skalenraumdarstellung mit sehr geringem Aufwand konstruiert werden. Es wird kein Downsampling des Bildes benötigt, stattdessen werden die Filter sukzessive vergrößert und auf das unveränderte Eingangsbild angewandt.

Der Skalenraum wird in *Oktaven* unterteilt. Jede Oktave gliedert sich wiederum in eine feste Anzahl an Skalierungsstufen. Pro Oktave muss die Filtergröße mehr als verdoppelt werden. Je nach Bildgröße werden bis zu vier Oktaven durchlaufen. Interessenspunkte werden an Extrema der Detektorantwort (4.11) im Multiskalenraum durch Nicht-Maximum-Unterdrückung detektiert. Anschließend erfolgt eine genaue Lokalisierung der örtlichen Merkmalsposition und des Skalenfaktors

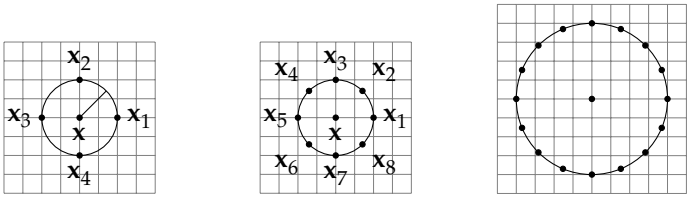
durch Interpolation im Orts-Skalenraum. Die genaue Lokalisierung ist hier besonders wichtig, da die einzelnen Skalierungsebenen recht weit auseinander liegen.

Der *Deskriptor* für die SURF-Merkmale repräsentiert die Verteilung von Intensitätsänderungen in der Umgebung eines Merkmalspunktes. Er basiert auf Filterantworten von Haar-Wavelets erster Ordnung in horizontaler und vertikaler Richtung, wodurch auch hier die Vorzüge der Integralbilder zum Tragen kommen. Zunächst wird die Hauptorientierung eines Punktes ermittelt. Der Deskriptor wird in einer quadratischen Region um den Merkmalspunkt bestimmt, die um die Hauptorientierung rotiert wird und deren Größe vom Skalenfaktor des Punktes abhängt. Diese Region wird in Subregionen unterteilt und in jeder Region werden die Haar-Wavelet-Antworten an bestimmten Abtastpunkten ausgewertet. Der Deskriptor einer Subregion ergibt sich aus der Summe der Filterantworten in horizontaler und vertikaler Richtung und den Summen ihrer Beträge, um Informationen über die Polaritäten der Intensitätsänderungen in den Deskriptor zu integrieren. Daraus ergibt sich ein vierdimensionaler Vektor für jede Subregion. In der Standardimplementierung ergibt sich insgesamt ein 64-dimensionaler Deskriptor, welcher als SURF-64 bezeichnet wird.

Der SURF-Deskriptor besitzt neben der Rotations- und Skalierungsinvarianz eine Reihe weiterer vorteilhafter Eigenschaften. Die Wavelet-Antworten sind aufgrund des Bandpass-Charakters von Wavelet-Filtern invariant gegenüber dem Offset der Beleuchtung. Durch Normierung des Merkmalsvektors wird Kontrastinvarianz erreicht. Wird keine Rotationsinvarianz benötigt, kann auf die Bestimmung der Hauptorientierung und der Rotation der Merkmalsregion verzichtet werden, wodurch einerseits eine Ersparnis der Rechenzeit erreicht wird und sich andererseits aussagekräftigere Deskriptoren ergeben, welche sich dennoch als robust gegenüber geringen Rotationen erweisen. Eine Reduktion der Rechenzeit kann auch durch die Verwendung von weniger Subregionen und somit eines kürzeren Merkmalsvektors erzielt werden. Diese als SURF-36 bezeichnete Variante erreicht zwar etwas schlechtere Ergebnisse, stellt aber einen guten Kompromiss dar, wenn die Rechenzeit im Vordergrund steht.

In Experimenten zur Objekterkennung [7] werden bessere Ergebnisse als bei SIFT berichtet. Bay et al. führen dies darauf zurück, dass bei

SURF die Filterantworten der Subregionen aufsummiert werden, während SIFT Histogramme verwendet und damit Informationen einzelner Orientierungsabschnitte der Filterantworten betrachtet. Dies macht SURF robuster und weniger rauschempfindlich. Abbildung 4.2 zeigt ein Beispiel für das *Matching* von SURF-Merkmalen zwischen einem Originalbild und einer rotierten und skalierten Variante des Bildes.



(a) $R = 1, P = 4$ **(b)** $R = 1, P = 8$ **(c)** $R = 2, P = 16$

Abbildung 4.3. Verwendete Bildpunkte zur Berechnung der LBPs für verschiedene Nachbarschaften.

4.2.4. Local Binary Patterns

Local Binary Patterns (LBP) (lokale Binärmuster) [69] sind eine Methode zur Beschreibung von Bildtexturen. Sie sind invariant bezüglich Rotation und monotonen Grauwert-Transformationen.

Der Deskriptor an einer Position $\mathbf{x}$ wird in einer kreisförmigen Nachbarschaft um $\mathbf{x}$ berechnet. Es werden die Grauwerte von P Bildpunkten betrachtet, die äquidistant auf einem Kreis mit Radius R und Mittelpunkt $\mathbf{x}$ liegen, und mit dem Grauwert des mittleren Punktes verglichen. Die Grauwerte an Positionen, die nicht mit den Pixelpositionen übereinstimmen, werden dabei interpoliert. Abbildung 4.3 zeigt die Punkte zur Berechnung von LBPs für verschiedene Werte von R und P. i sei der Grauwert des Mittelpunktes $i := i(\mathbf{x})$. Die Grauwerte der Nachbarpunkte werden als $i_p = i(\mathbf{x}_p)$, $p = 0, \ldots, P-1$, bezeichnet.

Die lokale Textur T in der Nachbarschaft von $\mathbf{x}$ wird als Verbunddichte der Grauwerte der $P+1$ Stützpunkte definiert:

$$T := p(i, i_0, \ldots, i_{P-1}) \,. \tag{4.13}$$

Um Invarianz bezüglich des Grauwertes zu erhalten, wird der Grauwert des Mittelpunktes von allen i_p subtrahiert und angenommen, dass die Differenzen $i_p - i$ unabhängig von i sind, woraus folgende Näherung resultiert:

$$T \approx p(i)p(i_0 - i, \ldots, i_{P-1} - i)\,. \tag{4.14}$$

Die Verteilung $p(i)$ beschreibt die gesamte Helligkeit des Bildes und enthält somit keine relevanten Informationen über lokale Bildtexturen, weshalb dieser Term weggelassen wird:

$$T \approx p(i_0 - i, \ldots, i_{P-1} - i)\,. \tag{4.15}$$

Weiterhin wird Invarianz bezüglich der Skalierung der Grauwerte erreicht, indem nur die Vorzeichen der Differenzen $i_p - i$ betrachtet werden

$$T \approx p(s(i_0 - i), \ldots, s(i_{P-1} - i))\,, \tag{4.16}$$

mit

$$s(x) = \begin{cases} 1\,, & x \geq 0 \\ 0\,, & x < 0 \end{cases}\,. \tag{4.17}$$

Von dieser Binarisierung der Grauwertdifferenzen kommt der Name „Binärmuster“. Der LBP-Texturdeskriptor ergibt sich, indem der Grauwertverteilung aus Gleichung (4.16) ein skalarer Wert aus der Menge $\{0, \ldots, 2^P - 1\}$ zugeordnet wird gemäß

$$LBP_{P,R}(\mathbf{x}) = \sum_{p=0}^{P} s(i_p - i)2^p\,. \tag{4.18}$$

In [69] wird aus dem Deskriptor (4.18) ein diskriminationsstarkes rotationsinvariantes Merkmal gebildet. Dieses basiert auf der Erkenntnis, dass sog. *homogene* Binärmuster (engl. *uniform patterns*) grundlegende Eigenschaften lokaler Bildtexturen wiedergeben. Homogene Muster erlauben nur eine bestimmte Anzahl an Diskontinuiäten, d. h. Vorzeichenwechsel, in der kreisförmigen Musterdarstellung. Solche homogenen Muster beschreiben beispielsweise Kanten, Ecken oder Punkte. Als

Homogenitätsmaß $U_{P,R}(\mathbf{x})$ wird die Anzahl an Bit-Änderungen benachbarter Punkte von $LBP_{P,R}(\mathbf{x})$ definiert. Ein Muster wird als homogen bezeichnet, wenn $U_{P,R}(\mathbf{x}) \leq 2$ gilt. Damit wird ein Texturmerkmal definiert zu

$$LBP^{\text{riu2}}_{P,R}(\mathbf{x}) = \begin{cases} \sum_{p=0}^{P-1} s(i_p - i)\,, & \text{falls } U_{P,R}(\mathbf{x}) \leq 2 \\ P+1 & \text{sonst} \end{cases}. \tag{4.19}$$

Es können $P+1$ homogene Binärmuster auftreten. $LBP^{\text{riu2}}_{P,R}(\mathbf{x})$ kann $P+2$ Werte annehmen aus $\{0,\ldots,P+1\}$. Der Wert $P+1$ beinhaltet hierbei alle inhomogenen Muster. Da hier im Gegensatz zu (4.18) nicht jedem Muster durch Multiplikation mit 2^p ein eindeutiger Wert zugeordnet wird, sondern lediglich die positiven Vorzeichen der Grauwertdifferenzen gezählt werden, ist $LBP^{\text{riu2}}_{P,R}(\mathbf{x})$ invariant gegenüber Rotationen. Abbildung 4.4 zeigt die homogenen Muster für $P = 8$.

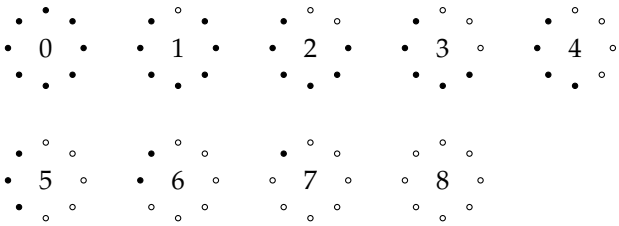
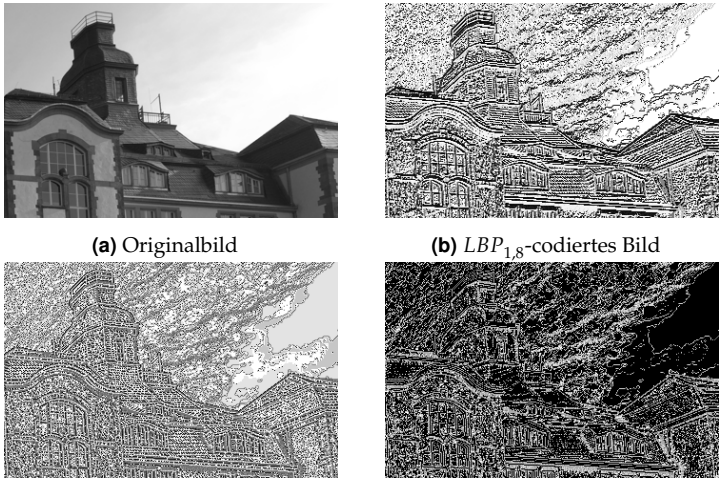


Abbildung 4.4. Homogene Muster einer Nachbarschaft mit $P = 8$. Schwarze Kreise entsprechen Bit-Werten von null, weiße Kreise Werten von eins.

Durch die Verwendung von $LBP^{\text{riu2}}_{P,R}(\mathbf{x})$ können Texturmerkmale gebildet werden, welche eine bessere Diskriminationsfähigkeit besitzen als aus $LBP_{P,R}(\mathbf{x})$ gebildete Merkmale [69]. Der Grund hierfür ist, dass inhomogene Muster wesentlich seltener auftreten als homogene und somit ihre Verteilungen schlecht geschätzt werden können. Die Textur eines Bildbereiches wird schließlich als Histogramm von LBP-Werten mehrerer Bildpunkte dargestellt, und die Ähnlichkeit zweier Texturen erfolgt durch Vergleich ihrer Histogramme.



(a) Originalbild

(b) $LBP_{1,8}$-codiertes Bild

(c) $LBP^{\text{riu2}}_{1,8}$-codiertes Bild

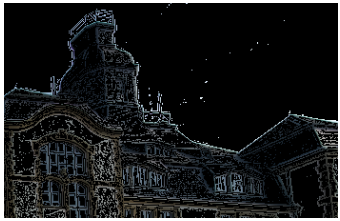
(d) $LBP^{\text{major}}_{1,8}$-codiertes Bild

Abbildung 4.5. Originalbild und codierte Bilder verschiedener LBP-Varianten.

Es gibt einige Variationen von *Local Binary Patterns*. Heikkilä et al. [43] stellen eine auf LBPs basierende Methode zur Deskription lokaler Merkmale vor, welche dem SIFT-Verfahren ähnelt, jedoch einen geringeren Rechenaufwand besitzt. Die Methode wird als *Center-Symmetric Local Binary Patterns* (CS-LBP) bezeichnet, da gegenüberliegende Paare von Bildpunkten miteinander verglichen werden. Dort wird außerdem zur Binarisierung der Grauwertdifferenzen ein Schwellwert verwendet, welcher eine höhere Robustheit bei flachen Bildregionen bewirkt. Weiterhin existieren einige dreidimensionale Erweiterungen von LBPs zur Verarbeitung von Bildfolgen. In [109] werden *Local Trinary Patterns* als Bewegungs-Deskriptoren für die Aktionserkennung vorgeschlagen, welche weniger die Bildinformation, sondern hauptsächlich die Bewegungsrichtung codieren. In [112] wird die LBP-Methodik auf dreidimensionale Volumen angewandt. Die entwickelten VLBP/LBP-TOP-Deskriptoren werden zur Klassifikation dynamischer Texturen eingesetzt mit dem Anwendungsbeispiel von Gesichtsausdrücken.

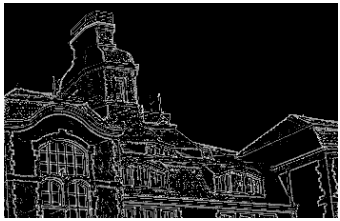
In [67] wird eine weitere Unterscheidung der homogenen Merkmale vorgenommen. Dort werden sog. Hauptmuster (*Major Patterns*) zum

(a) Bild mit Maske aus homogenen Mustern ($th > 0$)



(b) Bild mit Maske aus Hauptmustern ($th > 0$)

(c) $LBP^{\text{major}}_{1,8,0}$-codiertes Bild



(d) $LBP^{\text{major}}_{1,8,5}$-codiertes Bild

Abbildung 4.6. Oben: Vergleich homogener Muster und Hauptmuster. (a): Eingangsbild überlagert mit einer Maske, welche für homogene Muster Werte von eins und für inhomogene Muster Werte von null annimmt. (b): Eingangsbild entsprechend maskiert mit Hauptmustern. Unten: Vergleich LBP^{major}-codierter Bilder mit Schwellwerten $th = 0$ (c) und $th = 5$ (d).

Tracking von Objekten verwendet. Diese Hauptmuster sind z. B. Ecken und Kanten. Flächen zählen dagegen zu sog. *Minor Patterns* (Nebenmuster). Die Hauptmuster für eine Anzahl von $P = 8$ Nachbarn sind die homogenen Muster mit den Indices zwei bis sechs in Abbildung 4.4. Die Indices der Nebenmuster werden bei diesem Operator zu null gesetzt, woraus der $LBP^{\text{major}}_{1,8}$-Deskriptor resultiert. Wird wie bei CS-LBP zur Binarisierung der Grauwertdifferenzen in Gleichung (4.17) ein Schwellwert $th > 0$ verwendet, wird der Deskriptor mit $LBP^{\text{major}}_{1,8,th}$ bezeichnet. Abbildung 4.5 zeigt ein Bild und die entsprechenden codierten Bilder $LBP_{1,8}$, $LBP^{\text{riu2}}_{1,8}$ und $LBP^{\text{major}}_{1,8}$. In Abbildung 4.6 werden die Hauptmuster mit den homogenen Mustern verglichen.

Für den Einsatz der Hauptmuster zum Tracking werden Objekte als Verbund-Histogramme des $LBP^{\text{major}}_{1,8,th}$-Deskriptors mit $th > 0$ und der

Farbverteilung im RGB-Raum repräsentiert und mittels des *Mean-Shift*-Verfahrens (siehe Abschnitt 4.2.6) verfolgt.

Der Grund für die Konzentration auf die Hauptmuster besteht darin, dass der Hintergrund häufig aus Nebenmustern, z. B. Flächen, besteht. Dadurch verhält sich der Tracker robuster, da das Objektmodell weniger durch den Hintergrund verfälscht wird. Die Verwendung eines Schwellwertes $th > 0$ sorgt ebenso für eine geringere Empfindlichkeit gegenüber leichten Ungleichmäßigkeiten des Hintergrundes, wie auch anhand Abbildung 4.6 ersichtlich wird. Diese Methode zeigt sich in [67] vor allem dann gegenüber alleiniger Verwendung der Farbverteilung als stark überlegen, wenn sich die Objekte farblich nicht stark vom Hintergrund abheben oder wenn das Objektfenster viele Hintergrundpunkte enthält.

4.2.5. Optischer Fluss

Der optische Fluss wird verwendet, um Bewegung in Bildfolgen zu beschreiben. Er ist ein Vektorfeld, welcher die Verschiebung jedes Bildpunktes zwischen zwei Bildern angibt. Wird die Bewegung einer dreidimensionalen Szene auf die Bildebene projiziert, resultiert ein zweidimensionales Vektorfeld, welches die Verschiebung jedes Bildpunktes zwischen zwei Zeitpunkten angibt [35]. Dieses Bewegungsfeld wird als optischer Fluss bezeichnet. Die Berechnung des optischen Flusses hat zum Ziel, dieses Feld aus dem Helligkeitsverlauf einer Bildfolge zu schätzen [35]. Da es hierzu keine eindeutige Lösung gibt, existieren verschiedene Ansätze. Die klassische Herangehensweise sind die differentiellen Methoden [45].

Dabei wird angenommen, dass lediglich translatorische Bewegung auftritt und eine konstante Beleuchtung vorliegt. Dann kann das Verhältnis der Bildintensitäten an einem Punkt $\mathbf{x}$ zweier aufeinanderfolgender Zeitschritte mit der Abtastzeit Δt durch

$$i(\mathbf{x}, t) = i(\mathbf{x} + \mathbf{f}, t + \Delta t) \tag{4.20}$$

beschrieben werden [35]. $\mathbf{f}$ stellt dabei die Verschiebung des Punktes $\mathbf{x}$ zwischen den Zeitpunkten t und $t + \Delta t$ dar. Diese Annahmen sind i. d. R. zwar nicht erfüllt, haben sich in der Praxis allerdings als geeignet

erwiesen. Die rechte Seite des obigen Ausdrucks wird nun durch eine Taylor-Reihe erster Ordnung angenähert

$$i(\mathbf{x}+\mathbf{f}, t+\Delta t) \approx i(\mathbf{x}, t) + \mathbf{f}\nabla^{\mathrm{T}} i(\mathbf{x}, t) + i_t(\mathbf{x}, t) \tag{4.21}$$

mit $\nabla i = [i_x, i_y]$. Setzt man (4.21) in (4.20) ein, so ergibt sich

$$i_t(\mathbf{x}, t) + \mathbf{f}\nabla^{\mathrm{T}} i(\mathbf{x}, t) = 0\,. \tag{4.22}$$

Diese Gleichung wird auch als Kontinuitätsgleichung bezeichnet. Der Begriff wurde dem entsprechenden Konzept aus der Hydrodynamik entlehnt [45]. Für die Kontinuitätsgleichung des optischen Flusses existiert keine eindeutige Lösung. Es sind daher weitere Einschränkungen bezüglich der Verschiebung $\mathbf{f}$ nötig.

Einen Lösungsansatz stellen die differentiellen Methoden erster Ordnung dar. Es wird die Gradientenbedingung (4.22) für Punkte in einer Nachbarschaft um $\mathbf{x}$ hinzugenommen. Dabei wird angenommen, dass diese die gleiche Bewegung ausführen. $\mathbf{f}$ kann dann z. B. mittels *Least-Squares*-Schätzung ermittelt werden. Die zu minimierende Funktion ist dabei

$$e(\mathbf{f}) = \sum_{\mathbf{x}} w(\mathbf{x}) \left(\mathbf{f}\nabla^{\mathrm{T}} i(\mathbf{x}, t) + i_t(\mathbf{x}, t)\right)^2 \rightarrow \min\,. \tag{4.23}$$

Durch Differenzieren von (4.23) nach f^x und f^y ergibt sich das folgende Gleichungssystem (in kompakter Schreibweise):

$$\underbrace{\begin{bmatrix} \sum w i_x^2 & \sum w i_x i_y \\ \sum w i_x i_y & \sum w i_y^2 \end{bmatrix}}_{\mathbf{\Phi}} \mathbf{f}^{\mathrm{T}} = -\underbrace{\begin{bmatrix} \sum w i_x i_t \\ \sum w i_y i_t \end{bmatrix}}_{\mathbf{b}}\,. \tag{4.24}$$

Wenn (4.24) vollen Rang besitzt, ergibt sich die geschätzte Verschiebung zu $\hat{\mathbf{f}}^{\mathrm{T}} = -\mathbf{\Phi}^{-1}\mathbf{b}$.

Es gibt eine Vielzahl an Methoden, diese Schätzung zu verbessern. Eine Möglichkeit zur Erhöhung der Genauigkeit, v. a. für große Verschiebungen, ist eine iterative Schätzung, da in Gleichung (4.24) Terme der Taylor-Entwicklung höherer Ordnung vernachlässigt werden [35]. Zu anderen Erweiterungen zählen bessere Bewegungsmodelle anstatt der Annahme gleichförmiger Bewegung in einer Nachbarschaft, Multiskalenansätze sowie probabilistische Formulierungen [35].

4.2.6. Bildbasiertes Tracking mit Mean Shift-Verfahren

Mean Shift-Verfahren

Das *Mean Shift*-Verfahren [18, 22] ist eine Methode zur Bestimmung lokaler Maxima einer Verteilungsdichte $p(\mathbf{x})$. Das Verfahren wurde von Fukunaga und Hosteler [36] entwickelt und wird in der Bildverarbeitung für unterschiedliche Aufgaben verwendet, z. B. zur Segmentierung, zum Clustering oder zum bildbasierten Tracking. Das Verfahren entspricht einem Gradientenaufstieg mit adaptiver Schrittweite und beruht auf der Verwendung von Kern-Dichteschätzern.

Da der Gradient von $p(\mathbf{x})$ i. Allg. nicht explizit bestimmt werden kann, wird mittels Kern-Dichteschätzung eine nicht-parametrische Schätzung basierend auf einer bestimmten Anzahl an Abtastwerten $\mathbf{x}_i$ der Dichte p durchgeführt. Die Kern-Dichteschätzung von p an einem Punkt $\mathbf{x} \in \mathbb{R}^D$ bei N_h Beobachtungen $\{\mathbf{x}_i\}$, $i \in \{1, \ldots, N_h\}$, in einem Suchfenster $S_h(\mathbf{x})$ lautet

$$\hat{p}_{h,K}(\mathbf{x}) = \frac{c_k^D}{N_h h^D} \sum_{i=1}^{N_h} k\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right). \tag{4.25}$$

Dabei ist $k(x)$ das Profil des verwendeten Kerns $K(\mathbf{x})$ mit

$$K(\mathbf{x}) = c_k^D \, k(\|\mathbf{x}\|^2). \tag{4.26}$$

Der Parameter h wird als Bandbreite bezeichnet und c_k^D ist eine Normierungskonstante. Bei Verwendung eines differenzierbaren Kerns ergibt sich die Schätzung des Dichtegradienten durch den Gradienten des Dichteschätzers [21, 36]:

$$\begin{aligned} \widehat{\nabla} p_{h,K}(\mathbf{x}) &= \nabla \hat{p}_{h,K}(\mathbf{x}) \\ &= \frac{2c_k^D}{N_h \, h^{D+2}} \sum_{i=1}^{N_h} (\mathbf{x} - \mathbf{x}_i) \, k'\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right). \end{aligned} \tag{4.27}$$

Wird eine neue Funktion

$$g(x) = -k'(x) \tag{4.28}$$

definiert, welche das Profil eines Kerns

$$G(\mathbf{x}) = c_g^D \, g(\|\mathbf{x}\|) \tag{4.29}$$

darstellt, ergibt sich [22]

$$\widehat{\nabla} p_{h,K}(\mathbf{x}) = \frac{2c_k^D}{h^2 c_g^D} \, \hat{p}_{h,G}(\mathbf{x}) \, \mathbf{m}_{h,G}(\mathbf{x}) \, . \tag{4.30}$$

Dabei ist

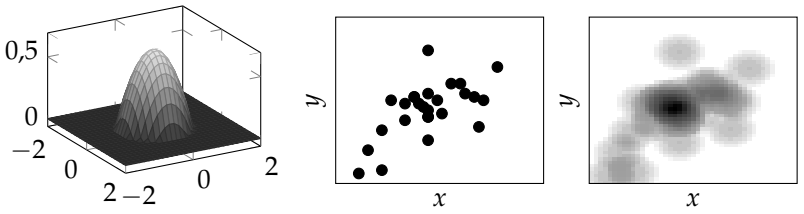
$$\mathbf{m}_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{N_h} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{N_h} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \tag{4.31}$$

der D-dimensionale *Mean Shift*-Vektor [22]. Dieser entspricht der Differenz zwischen dem Mittelpunkt $\mathbf{x}$ und dem gewichteten Mittel der Punkte $\mathbf{x}_i$ im Suchfenster $S_h(\mathbf{x})$. Aus (4.30) folgt:

$$\mathbf{m}_{h,G}(\mathbf{x}) = \frac{h^2}{2} \frac{c_{g,D}}{c_{k,D}} \cdot \frac{\widehat{\nabla} p_{h,K}(\mathbf{x})}{\hat{p}_{h,G}(\mathbf{x})} \tag{4.32}$$

Der *Mean Shift*-Vektor zeigt in Richtung des Dichtegradienten [18, 22]. Er stellt somit einen nicht-parametrischen Gradientenschätzer dar und kann verwendet werden, um zu einem lokalen Maximum der geschätzten Verteilungsdichte aufzusteigen. Der Gradientenaufstieg erfolgt iterativ. In jedem Iterationsschritt wird der *Mean Shift*-Vektor an der aktuellen Position berechnet und anschließend das Suchfenster solange um diesen Vektor verschoben, bis das Verfahren an einem lokalen Maximum von $\hat{p}(\mathbf{x})$ konvergiert. Die aufeinanderfolgenden Mittelpunkte des Suchbereichs in den Iterationen $j = 0, 1, \ldots$ werden mit $\{\tilde{\mathbf{x}}_j\}$, $j \in \{0, 1, \ldots\}$, bezeichnet. Die Iterationsvorschrift lautet

$$\tilde{\mathbf{x}}_{j+1} = \tilde{\mathbf{x}}_j + \mathbf{m}_{h,G}(\tilde{\mathbf{x}}_j) = \frac{\sum_{i=1}^{N_h} \mathbf{x}_i \, g\left(\left\|\frac{\tilde{\mathbf{x}}_j-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{N_h} g\left(\left\|\frac{\tilde{\mathbf{x}}_j-\mathbf{x}_i}{h}\right\|^2\right)} \, . \tag{4.33}$$



(a) Epanechnikov-Kern **(b)** Abtastpunkte **(c)** Geschätzte Dichte

Abbildung 4.7. Kern-Dichteschätzer: gegebene Abtastwerte einer unbekannten Dichte und Kern-Dichteschätzung mit Epanechnikov-Kern.

Wenn der Kern K ein konvexes und monoton abnehmendes Profil besitzt, ist das Verfahren konvergent [22]. Häufig wird der Epanechnikov-Kern eingesetzt, der das Profil

$$k_{\mathrm{E}}(x) = \begin{cases} 1 - x & \text{für } 0 \leq x \leq 1 \\ 0 & \text{sonst} \end{cases} \tag{4.34}$$

besitzt. Damit ergibt sich für $g_{\mathrm{E}}(x)$ das Einheitsprofil und Gl. (4.33) vereinfacht sich zu [21]

$$\tilde{\mathbf{x}}_{j+1} = \frac{1}{N_h} \sum_{i=1}^{N_h} \mathbf{x}_i \,, \tag{4.35}$$

was dem Mittelwert der Punkte $\mathbf{x}_i$ in $S_h(\tilde{\mathbf{x}}_j)$ entspricht. Abbildung 4.7 zeigt ein Beispiel für eine Kern-Dichteschätzung mit dem Epanechnikov-Kern.

Mean Shift-Tracking

Für die Anwendung des *Mean Shift*-Verfahrens zum Tracken von Objekten [23] werden ein Objektmodell und ein Ähnlichkeitsmaß zum Vergleich von Modellen benötigt. Das *Zielmodell* wird bei der Initialisierung eines Objektes ermittelt und dient als Referenz für die Suche der Objektposition in nachfolgenden Schritten.

In einem neuen Zeitschritt wird das Objekt in einem Suchfenster um die alte Position gesucht. Dabei wird angenommen, dass sich das Objekt

innerhalb eines Schrittes nicht aus dem Suchfenster heraus bewegt. Es werden *Kandidatenmodelle* im Suchfenster mit dem Zielmodell durch ein geeignetes Ähnlichkeitsmaß verglichen um die Position zu finden, an der die Distanz zwischen Ziel- und Kandidatenmodell minimal ist. Das Ähnlichkeitsmaß wird mittels des *Mean Shift*-Algorithmus maximiert und somit die neue Objektposition ermittelt. Im nächsten Schritt wird das Suchfenster um die neue Position zentriert.

Das Ziel wird durch seine Wahrscheinlichkeitsdichte q in einem bestimmten Merkmalsraum dargestellt. Häufig wird die Farbverteilung des Objekts verwendet. Die Verteilungen des Ziel- und Kandidatenmodells müssen aus den vorliegenden Daten geschätzt werden. Dies erfolgt häufig durch Verwenden von Histogramm-Modellen, um den Rechenaufwand gering zu halten.

Das Zielmodell wird im eindimensionalen Fall als Histogramm mit m Abschnitten dargestellt [24]:

$$\hat{\mathbf{q}} = \{\hat{q}_u\}\,, \quad u = 1, \ldots, m\,, \quad \text{mit} \sum_{u=1}^{m} \hat{q}_u = 1\,. \tag{4.36}$$

Analog dazu lautet das Kandidatenmodell an der Position $\mathbf{x}$

$$\hat{\mathbf{p}}(\mathbf{x}) = \{\hat{p}_u(\mathbf{x})\}\,, \quad u = 1, \ldots, m\,, \quad \text{mit} \sum_{u=1}^{m} \hat{p}_u(\mathbf{x}) = 1\,. \tag{4.37}$$

Die Modelle werden in elliptischen Regionen geschätzt, welche auf den Einheitskreis normalisiert werden.

Für die Schätzung des Zielmodells ergibt sich [24]

$$\hat{q}_u = C \sum_{i=1}^{N} k\left(\|\mathbf{x}_i\|^2\right) \delta(b(\mathbf{x}_i) - u)\,, \tag{4.38}$$

wobei $\{\mathbf{x}_i\}, i = 1, \ldots, N$, die normalisierten Koordinaten der Bildpunkte im Bereich des Zielmodells sind. $b(\mathbf{x}_i)$ gibt den Histogrammindex an der Stelle $\mathbf{x}_i$ an. $\delta(u)$ ist die Delta-Funktion und C eine Normierungskonstante, damit $\sum_{u=1}^{m} \hat{q} = 1$ gilt. Die einzelnen Bildpunkte werden mit dem Kern K, zentriert um $\mathbf{x} = 0$, gewichtet. Dadurch werden Punkte am Rande des Objektbereichs schwächer gewertet.

Das Kandidatenmodell an der Stelle $\mathbf{x}$ wird aus den Bildpunkten $\{\mathbf{x}_i\}$, $i = 1, \ldots, N_h$, in einem Suchfenster $S_h(\mathbf{x})$ berechnet und ergibt sich entsprechend Gl. (4.38) zu

$$\hat{p}_u(\mathbf{x}) = C_h \sum_{i=1}^{N_h} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \delta(b(\mathbf{x}_i) - u)\,. \tag{4.39}$$

h ist ein Skalierungsparameter und bestimmt die Anzahl von Bildpunkten, die zur Berechnung des Kandidatenmodells herangezogen wird.

Gesucht ist die Position $\mathbf{x}^{\mathrm{tg}}$, an der das Ziel- und Kandidatenmodell einander am ähnlichsten sind. Als Ähnlichkeitsmaß wird der *Bhattacharyya-Koeffizient* verwendet. Dessen diskrete Schätzung basierend auf den Histogramm-Modellen lautet [23]

$$\rho(\mathbf{x}) \equiv \rho\left[\hat{p}(\mathbf{x}), \hat{q}\right] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{x})\hat{q}_u}\,. \tag{4.40}$$

Geometrisch kann dies interpretiert werden als der Kosinus des Winkels zwischen den m-dimensionalen Einheitsvektoren $\left(\sqrt{\hat{p}_1}, \ldots, \sqrt{\hat{p}_m}\right)^{\mathrm{T}}$ und $\left(\sqrt{\hat{q}_1}, \ldots, \sqrt{\hat{q}_m}\right)^{\mathrm{T}}$. Mit (4.40) kann die Distanz zwischen den beiden Modellen angegeben werden als

$$d(\mathbf{x}) = \sqrt{1 - \rho(\mathbf{x})}\,. \tag{4.41}$$

Die Minimierung der Distanz zwischen dem Ziel- und dem Kandidatenmodell entspricht der Maximierung des Bhattacharyya-Koeffizienten bezüglich der Position $\mathbf{x}$. Diese Maximierung erfolgt durch *Mean Shift*-Iterationen. Die resultierende Iterationsvorschrift für die Berechnung der Zielposition $\hat{\mathbf{x}}^{\mathrm{tg}}_{j+1}$ in Iterationsschritt $j+1$ lautet

$$\hat{\mathbf{x}}^{\mathrm{tg}}_{j+1} = \frac{\sum_{i=1}^{N_h} \mathbf{x}_i w_{ij} g\left(\left\|\frac{\hat{\mathbf{x}}^{\mathrm{tg}}_j - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{N_h} w_{ij} g\left(\left\|\frac{\hat{\mathbf{x}}^{\mathrm{tg}}_j - \mathbf{x}_i}{h}\right\|^2\right)} \tag{4.42}$$

mit den Gewichtungsfaktoren

$$w_{ij} = w_i(\hat{\mathbf{x}}_j^{\mathrm{tg}}) = \sum_{u=1}^{m} \delta(b(\mathbf{x}_i) - u) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{x}}_j^{\mathrm{tg}})}} \,. \tag{4.43}$$

Gleichung (4.42) entspricht der Iterationsvorschrift aus Gleichung (4.33) mit den zusätzlichen Gewichten w_{ij}, welche den Vergleich der beiden Modelle beinhalten. Eine Herleitung der Gleichungen (4.42) und (4.43) ist in Anhang A.1 gegeben.

Mittels des *Mean Shift*-Verfahrens kann somit in jedem Zeitschritt iterativ das Maximum von ρ gefunden werden. Bei Verwendung des Epanechnikov-Kerns stellt die Ermittlung der neuen Zielpositionen gerade eine gewichtete Mittelung der Punkte im Suchfenster dar. Einzelheiten zur praktischen Realisierung können beispielsweise in [24] nachgelesen werden.

4.3. Bewegungserfassung durch Merkmalstracking

In dieser Arbeit erfolgt die Extraktion von Bewegungsinformationen durch das Tracking von Merkmalspunkten. Die Vorgehensweise dabei ist, zunächst bestimmte Merkmale in einer Videosequenz zu detektieren und diese anschließend über der Zeit zu verfolgen. Dazu werden Merkmale verwendet, die zum einen robust zu verfolgen sind. Es wird außerdem eine spärliche Repräsentation angestrebt, daher sollen die Merkmale möglichst charakteristisch für die in einer Sequenz vorhandenen Aktivitäten sein. Dies wird erreicht, indem markante Merkmale verfolgt werden, welche sich bewegen. Diese werden im Folgenden auch als *Aktionspunkte* bezeichnet. Das Tracking erfolgt mittels optischem Fluss und *Mean Shift*-Tracking. Das Ergebnis sind Merkmalstrajektorien, welche Informationen über die Bewegungsdynamik der vorliegenden Bildsequenz beinhalten. Zusätzlich werden im Verlauf des Trackings Merkmale extrahiert, welche Textur und Bewegung in einer lokalen Umgebung der Trajektorien repräsentieren.

4.3.1. Übersicht und Ablauf

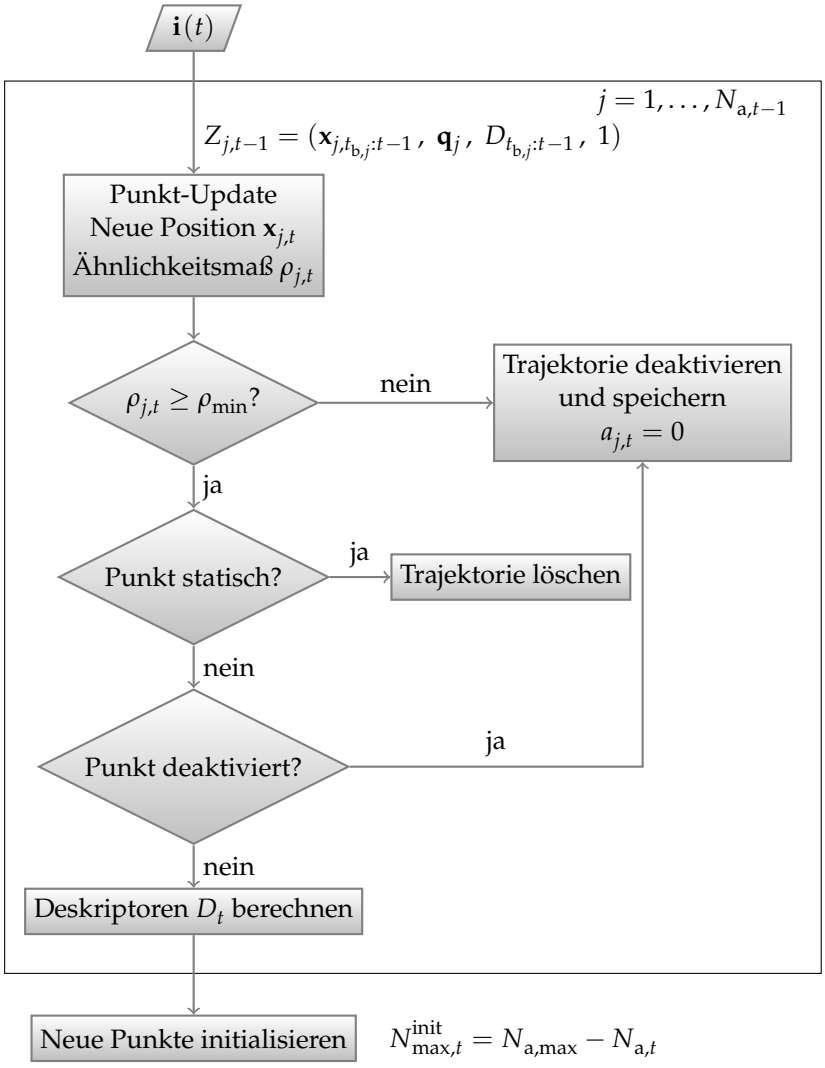


Abbildung 4.8. Übersicht über den Ablauf des Aktionspunkt-Trackings.

Bild 4.8 zeigt einen Überblick über den Ablauf des Merkmalstrackings. Eine Trajektorie wird zum Zeitpunkt t repräsentiert durch

$$Z_t = \left(\mathbf{x}_{t_\mathrm{b}:t} \,,\; \mathbf{q} \,,\; D_{t_\mathrm{b}:t} \,,\; a_t \right) . \tag{4.44}$$

Dabei ist

$$\mathbf{x}_{t_\mathrm{b}:t} = \left[\mathbf{x}_{t_\mathrm{b}}, \mathbf{x}_{t_\mathrm{b}+1}, \ldots, \mathbf{x}_t \right] \tag{4.45}$$

der Positionsverlauf vom Zeitpunkt der Initialisierung t_b der Trajektorie bis zum aktuellen Zeitpunkt. $\mathbf{q}$ ist das Zielmodell des Punktes für das *Mean Shift*-Tracking, welches bei der Initialisierung bestimmt wird. $D_{t_\mathrm{b}:t}$ ist eine Gruppe von Deskriptoren zur Repräsentation der Trajektorie. Der Indikator a_t gibt an, ob die Trajektorie zum Zeitpunkt t aktiv ist.

Zu einem Zeitpunkt t seien zunächst $N_{\mathrm{a},t-1}$ aktive Trajektorien vorhanden

$$\mathcal{Z}_{\mathrm{a},t-1} = \left\{ Z_{j,t-1} \right\} , \quad j = 1, \ldots, N_{\mathrm{a},t-1} . \tag{4.46}$$

Ausgehend von der vorigen Position des j-ten Aktionspunktes $\mathbf{x}_{j,t-1}$ wird mit der in Abschnitt 4.3.3 erläuterten Methode die neue Position $\mathbf{x}_{j,t}$ ermittelt.

Bevor die neu ermittelte Position übernommen wird, erfolgt eine Überprüfung der Trajektorie. Beim Merkmalstracking wird ein Ähnlichkeitsmaß $\rho_{j,t}$ bestimmt, welches die Übereinstimmung des Punktmodells mit den Beobachtungen an der neu ermittelten Position wiedergibt. Falls dieses Ähnlichkeitsmaß eine Schwelle ρ_min unterschreitet, gilt der aktuelle Punkt als verloren. In diesem Fall wird die Trajektorie gespeichert und aus dem Tracking-Prozess entfernt. Andernfalls wird die neu ermittelte Position übernommen.

Bei der Initialisierung kann es vorkommen, dass Punkte in Bildbereichen detektiert werden, die keine relevante Bewegungsinformation enthalten, z. B. im Hintergrund. Solche statische Punkte sollen frühzeitig erkannt und nicht weiter verfolgt werden. Dazu wird für jede Trajektorie einmalig nach einer bestimmten Zeit τ_static die Historie der Verschiebungsvektoren $\mathbf{v}_t = \mathbf{x}_t - \mathbf{x}_{t-1}$ über die gesamte Lebensdauer der Trajektorie überprüft. Unterschreitet der Median des Betrags der Verschiebungsvektoren eine bestimmte Grenze v_min,

$$\text{med}_{i=t_\text{b}:t_\text{b}+\tau_\text{static}} \left(\|\mathbf{v}_i\| \right) < v_\text{min} \,, \tag{4.47}$$

wird die Trajektorie als statisch angesehen und sofort gelöscht.

Wenn ein Aktionspunkt seine Bewegung abgeschlossen hat und sich nicht weiter bewegt, soll er nicht weiter verfolgt werden. Das gleiche gilt für Punkte, die trotz Überwachung des Ähnlichkeitsmaßes in den Hintergrund abdriften. In diesen Fällen soll das Tracking des Punktes beendet und die Trajektorie gespeichert werden. Um zu erkennen wann ein Aktionspunkt inaktiv wird, werden die Verschiebungsvektoren $\mathbf{v}$ während einer bestimmten Anzahl vergangener Schritte τ_d überwacht. Lag der Median des Betrags der Verschiebungsvektoren τ_d Schritte lang unterhalb von v_min,

$$\text{med}_{i=t-\tau_\text{d}:t} \left(\|\mathbf{v}_i\| \right) < v_\text{min} \,, \tag{4.48}$$

wird die Trajektorie deaktiviert.

Für die übriggebliebenen aktiven Punkte werden bestimmte Deskriptoren berechnet, welche Bild- und Bewegungsinformation in einer lokalen Nachbarschaft des Punktes repräsentieren. Die Deskriptor-Berechnung wird in Abschnitt 4.3.4 erläutert.

Wenn alle Punkte $Z_{j,t-1}$ abgearbeitet worden sind, können neue Punkte initialisiert werden, falls die maximal erlaubte Anzahl an aktiven Trajektorien nicht überschritten wurde. Auf die Merkmalsdetektion wird im folgenden Abschnitt eingegangen.

4.3.2. Detektion der Aktionspunkte

Die Kriterien, die in dieser Arbeit zur *Detektion* der Aktionspunkte herangezogen werden, sollen zum einen berücksichtigen, an welchen Stellen einer Bildfolge Bewegung auftritt und zum anderen, welche Merkmale sich zum Tracking eignen.

In verwandten Arbeiten werden zur Initialisierung von Trajektorien bereits STIP-Detektoren verwendet, z. B. der Harris3D-Detektor aus Gleichung (4.4) in [74]. Andere Methoden verwenden lediglich Detektoren von Bildmerkmalen, wie SIFT [92] oder SURF [68]. Wang et al. [98] verwenden „dichte“ Merkmale, die auf einem gleichmäßigen Bildraster initialisiert werden. Sie wollen dadurch einen Informationsverlust

durch zu spärliche Merkmale verhindern. In einer Nachfolgearbeit [99] verwerfen sie Merkmale in homogenen Bildbereichen, indem sie das Akzeptanzkriterium von Shi und Tomasi [82] heranziehen, auf welches später eingegangen wird. In [110] werden „saliente“ Trajektorien verwendet. Es werden zunächst Wangs dichte Trajektorien extrahiert und diese nachträglich gefiltert. Dazu werden zwei Salienzmaße vorgeschlagen, welche Trajektorien starker Bewegung oder auffälliger Bildmerkmale charakterisieren. Die besten Ergebnisse erhalten die Autoren durch Kombination der beiden Salienzmaße.

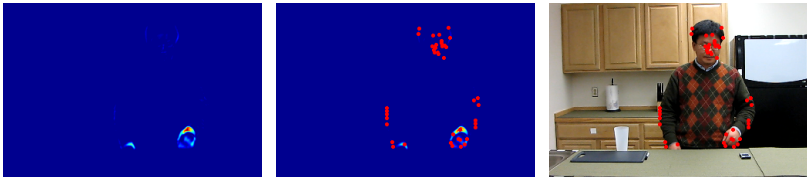
Zur Frage, welche Merkmale sich zum Tracking eignen, schlagen Shi und Tomasi [82] ein Akzeptanzkriterium vor. Dieses dient dazu, dass nur Merkmale in den Tracking-Prozess aufgenommen werden, welche sich gut verfolgen lassen. Ein allgemeines Kriterium dafür zu formulieren, ist schwierig, da dies vom verwendeten Tracker abhängt. Tomasi und Kanade [95] definieren gute Merkmale als solche, die von einem bestimmten Tracker robust verfolgt werden können. Das in [82] vorgeschlagene Kriterium wird mit *Good Features to Track* (GFTT) bezeichnet. Merkmale werden als geeignet erachtet, wenn die Eigenwerte der Matrix

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} i_x^2(\mathbf{x}) & i_x(\mathbf{x})i_y(\mathbf{x}) \\ i_x(\mathbf{x})i_y(\mathbf{x}) & i_y^2(\mathbf{x}) \end{bmatrix} \tag{4.49}$$

eine bestimmte Schwelle überschreiten: $\min(\lambda_1, \lambda_2) > \lambda$. Zur praktischen Umsetzung kann die Determinante der Matrix (4.49) als Detektor verwendet werden. Zur Wahl der Schwelle wird in [99] ein adaptiver Wert vorgeschlagen, welcher vom aktuellen Eingangsbild abhängt:

$$\lambda = 0{,}001 \cdot \max_{\mathbf{x}} \min(\lambda_1(\mathbf{x}), \lambda_2(\mathbf{x}))\,. \tag{4.50}$$

In dieser Arbeit sollen Bildmerkmale mit Bewegungsinformation kombiniert werden. Die Ergebnisse der salienten Trajektorien aus [110] demonstrieren die Wichtigkeit einer guten Initialisierung. Mit Verwendung eines Salienzmaßes, welches Bild- und Bewegungsinformation repräsentiert, erzielen die Autoren deutlich bessere Ergebnisse als mit dichten Trajektorien. Allerdings werden dort die Trajektorien nachträglich gefiltert und die Salienz für die gesamte Trajektorie berechnet. Die Auswahl geeigneter Punkte soll in dieser Arbeit dagegen bereits bei der Initialisierung der Trajektorien erfolgen, um den Aufwand des Trackings



(a) STIP-Detektor

(b) Detektierte Punkte an lokalen Maxima der Detektorantwort

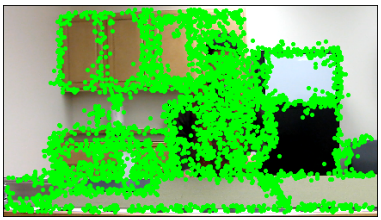
(c) Detektierte Punkte auf Eingangsbild

Abbildung 4.9. Beispiel für die Detektion von Aktionspunkten mit alleiniger Verwendung des STIP-Detektors (Bildquelle [62]). Zunächst wird die Antwort des STIP-Detektors des gesamten Bildes berechnet (a). Als Nächstes werden lokale Maxima der Detektorantwort ermittelt und die Punkte mit der stärksten Bewegung ausgewählt (b). (c) zeigt die resultierenden Punkte auf dem Eingangsbild.

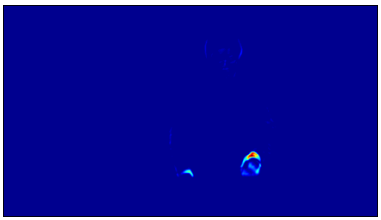
zu reduzieren, indem spärliche, möglichst informative Merkmale verfolgt werden. Daher werden für die Merkmalsdetektion Salienzmaße für Bildmerkmale und Bewegung kombiniert. Im Folgenden werden drei Initialisierungsmethoden vorgestellt.

Die erste ist die Verwendung eines STIP-Detektors. Die Methoden der STIP-Detektion aus Abschnitt 4.2.1 filtern Bildsequenzen in örtlicher und zeitlicher Richtung und beinhalten demnach bereits Bild- und Bewegungsinformationen. In dieser Arbeit hat sich der periodische Detektor aus Gleichung (4.5) als am besten geeignet erwiesen. Er detektiert eine große Anzahl an Punkten und reagiert auch auf schwache, langsame Bewegungen (siehe auch Abschnitt 4.2.1). Andere Detektoren resultieren in zu spärlichen Punkten oder reagieren zu stark auf Rauschen. Außerdem hat der periodische Detektor einen sehr geringen Rechenaufwand.

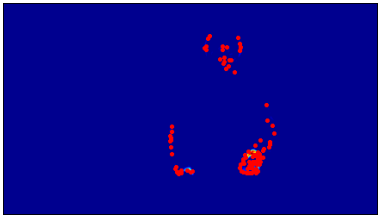
Zur Detektion von Merkmalspunkten wird zunächst das Bewegungsbild $r^{\text{periodic}}(\mathbf{x}, t, \sigma, \tau)$ nach Gleichung (4.5) berechnet. Merkmalskandidaten werden ermittelt, indem lokale Maxima des Bewegungsbildes durch Nicht-Maximum-Unterdrückung in einer 3×3-Nachbarschaft bestimmt werden. Um zur Initialisierung einer Trajektorie akzeptiert zu werden, wird gefordert, dass der Wert des STIP-Detektors eine Schwelle $r_{\text{min}}^{\text{init}}$ überschreitet. Des Weiteren wird geprüft, ob sich in einer bestimmten Nachbarschaft des Kandidaten bereits eine aktive Trajektorie befindet. Falls ja, wird der Punkt verworfen. Insgesamt dürfen zur Zeit t



(a) Detektierte SURF-Merkmale



(b) Bewegungsdetektor



(c) Mit Bewegungsdetektor gefilterte Merkmale

(d) Resultierende Punkte auf Eingangsbild

Abbildung 4.10. Beispiel für die Detektion von Aktionspunkten mit Verwendung von SURF- und Bewegungsdetektion (Bildquelle [62]). Zunächst werden SURF-Merkmale detektiert (a). In (b) ist der Bewegungsdetektor für das gesamte Bild zu sehen. Die SURF-Merkmale werden mittels des Bewegungsdetektors gefiltert und diejenigen Punkte mit der stärksten Bewegung ausgewählt (c). Bild (d) zeigt schließlich die gewählten Punkte auf dem Eingangsbild.

maximal $N_{\text{max},t}^{\text{init}}$ neue Punkte initialisiert werden. Dieser Wert ergibt sich aus der Differenz der maximal erlaubten Anzahl aktiver Punkte und der Anzahl bereits aktiver Punkte $N_{\text{max},t}^{\text{init}} = N_{\text{a,max}} - N_{\text{a},t}$. Aus den lokalen Maxima des Bewegungsdetektors werden schließlich die maximal $N_{\text{max}}^{\text{init}}$ Punkte mit der größten Detektorantwort gewählt. Ein Beispiel der Detektion ist in Bild 4.9 zu sehen.

Ein Nachteil dieses Detektors ist, dass die örtliche Filterung der Eingangsbilder lediglich in einer Glättung mit einem zweidimensionalen Gauß-Filter besteht. Dies alleine ist kein Detektor für besonders robuste Bildmerkmale. Daher wird im Folgenden vorgeschlagen, den periodischen Detektor als Salienzmaß für Bewegung zu betrachten und mit einem weiteren Detektor für Bildmerkmale zu kombinieren. Durch die Kombination soll die Qualität der detektierten Punkte verbessert und

mehr Gewicht auf die Bild-Salienz gelegt werden. Hierzu werden zwei Methoden betrachtet, der GFTT- und der SURF-Detektor.

Dabei werden zunächst mit einer der beiden Varianten N^{ip} Bildmerkmale mit den stärksten Detektorantworten extrahiert. Punkte, in deren unmittelbarer Umgebung sich bereits eine aktive Trajektorie befindet, werden wieder verworfen. Anschließend wird für jeden Kandidaten $\tilde{\mathbf{x}}_{t,j}$ der Wert des Bewegungsdetektors $r^{\text{periodic}}(\tilde{\mathbf{x}}_{t,j}, t, \sigma, \tau)$ geprüft und alle Punkte verworfen, die die Schwelle $r^{\text{init}}_{\text{min}}$ unterschreiten. Aus den restlichen werden die maximal $N^{\text{init}}_{\text{max}}$ Punkte selektiert, welche die stärkste Bewegung aufweisen. Abbildung 4.10 veranschaulicht den Ablauf dieser Methode anhand eines Beispiels mit dem SURF-Detektor.

Schließlich werden in Regionen um die neu initialisierten Punkte jeweils die Tracker-spezifischen Punkt-Deskriptoren initialisiert, worauf im folgenden Abschnitt eingegangen wird.

4.3.3. Tracking der Aktionspunkte

Zum *Tracking* der detektierten Merkmale ist bei verwandten Verfahren die am meisten verbreitete Methode der Kanade-Lucas-Tomasi-Tracker (KLT) [60]. Sun et al. [91, 92] verfolgen SIFT-Merkmale, indem sie Korrespondenzen von SIFT-Deskriptoren suchen. Eine andere Methode ist die Verfolgung von Punkten mit dichtem optischem Fluss [98, 99]. In [99] wird diese Methode mit dem KLT-Tracker sowie dem SIFT-*Matching* verglichen und es wird gezeigt, dass die Verwendung des optischen Flusses den beiden anderen überlegen ist, da sowohl beim KLT-Tracker als auch beim SIFT-Matching häufige Fehler und große Sprünge in den Trajektorien auftreten. Außerdem sind beim KLT- und SIFT-Tracker die Trajektorien meist sehr spärlich bzw. dichtes SIFT-Matching wäre sehr aufwändig [99]. Durch Tracking mit dichtem optischem Fluss werden diese Probleme dagegen vermieden und es ergeben sich glatte Trajektorienverläufe [99].

Eine Schwierigkeit beim Merkmalstracking ist, dass die Trajektorien sehr schnell abdriften und fehlerhaft werden. Daher werden in vielen Arbeiten Punkte nur über eine kurze Dauer von z. B. 15 Bildern verfolgt [47, 98, 99, 110]. Es gibt jedoch auch Ansätze, die es sich zum Ziel setzen, länger andauernde Trajektorien zu gewinnen [62, 78, 91]. Die Motiva-

tion dazu besteht darin, dass die Beschreibung von Bewegungsarten unterschiedlicher Komplexität verschiedene Beschreibungsmethoden erfordert [91]. Dazu zählen auch Trajektorien längerer Dauer und Interaktionen zwischen diesen. Zu Herausforderungen dabei zählt, dass es schwierig ist, Merkmalspunkte über einen langen Zeitraum erfolgreich zu verfolgen sowie die Frage, welche Deskriptoren sich eignen, um Trajektorien variabler Länge darzustellen [91]. Sun et al. [91] setzen daher drei verschiedene Merkmalsdetektoren und -Tracker ein, um eine ausreichend dichte Menge lang andauernder Trajektorien zu erhalten.

Diese Arbeit setzt sich ebenfalls zum Ziel, Trajektorien über einen langen Zeitraum zu verfolgen. Dazu wird die Verwendung verschiedener Informationsquellen in Betracht gezogen und ein zweistufiger Tracker vorgeschlagen. Um Punkte bei schnellen Bewegungen nicht zu verlieren, wird zunächst die Bewegungsrichtung mittels optischem Fluss wie in [98, 99] verfolgt. Der optische Fluss hat sich für die Gewinnung kurzer Trajektorien bereits bewiesen, und das Flussfeld wird für die nachfolgende Deskriptor-Bestimmung ohnehin benötigt. Für die Extraktion von Langzeit-Trajektorien ist diese Methode jedoch nicht geeignet, da sie nur Korrespondenzen zwischen einzelnen Bildern berücksichtigt [78]. Damit die Punkte nicht abdriften, müssen sie korrigiert werden.

Hierzu wird eine Methode angewandt, welcher Referenzmodelle für die zu verfolgenden Merkmale zugrunde liegen. Dies erfolgt mittels *Mean Shift*-Tracking (MST). Dabei werden keine Annahmen über auftretende Bewegungsformen angenommen. Histogramm-Modelle sind außerdem robust gegen leichte Objektdeformationen. Für das Objekt-Tracking mittels *Mean Shift* werden häufig Farbhistogramme verwendet. Hier werden aber keine Objekte großer Ausdehnung, sondern einzelne Bildmerkmale verfolgt. Diese Merkmale stellen Bildregionen markanter Textur dar. Die Farbverteilung alleine ist nicht geeignet, solche lokalen Bereiche zu repräsentieren. Daher ist die Textur der Punktregionen ein wichtiges Merkmal zum Tracking. Der Rechenaufwand des Trackings hängt mit der Dimension des Objektmodells und der Größe des Suchfensters zusammen. Da hier lokale Merkmale verfolgt werden und durch das Tracking mittels optischem Fluss bereits eine erste Schätzung der Punktposition vorliegt, können kleine Suchfenster verwendet werden. Um ein effizientes Tracking zu erreichen, sollen möglichst kompakte Histogramm-Modelle zum Einsatz kommen.

Mittels LBPs gelingt eine kompakte und dennoch diskriminationsstarke Texturbeschreibung. LBPs wurden in einigen Arbeiten bereits zum Tracking von Objekten verwendet. Takala und Pietikainen [94] verwenden Farbe, Textur und Bewegung zum Tracking. In [67] werden Farb-Textur-Histogramme im Rahmen des *Mean Shift* (MS)-Tracking-Algorithmus eingesetzt. Da in dieser Arbeit nicht einige wenige Objekte, sondern eine Vielzahl von Interessenspunkten verfolgt werden sollen, ist der Rechenaufwand ein entscheidender Faktor. Hier werden wie in [67] Histogramme von *Major Patterns* (siehe Abschnitt 4.2.4) als Objektmodell angewandt. Damit ergeben sich Histogramme mit lediglich 5 Bins. Außerdem wird irrelevante Hintergrundinformation unterdrückt, was ein robusteres Tracking ermöglicht.

Hochdimensionale Farb-Textur-Histogramme wie in [67] sind für die hiesige Anwendung jedoch ungeeignet. Dort werden $(8 \times 8 \times 8 \times 5)$-dimensionale Histogramme verwendet (3 Farbkanäle, ein Texturkanal), woraus ein 2560-dimensionales Objektmodell resultiert. Die Nutzung von Farbinformation als zusätzliches Merkmal soll dennoch ermöglicht werden. Dies wird umgesetzt, indem separate Texturmodelle für einzelne Farbkanäle bestimmt werden. Verwendet man beispielsweise zwei Farbkanäle für die Objektdarstellung, ergibt sich ein Zielmodell der Größe 5×5.

Das Zielmodell $\mathbf{q}_j$ für eine Trajektorie Z_j, welches für das *Mean Shift*-Tracking benötigt wird, wird an ihrem Initialisierungszeitpunkt $t_{\mathrm{b},j}$ in einem Suchfenster $S_h(\mathbf{x}_{j,t_{\mathrm{b},j}})$ aus $\mathbf{i}(t_{\mathrm{b},j})$ geschätzt. Die gesamte Vorgehensweise des Trackings wird im Folgenden erläutert:

Im ersten Schritt erfolgt das Tracking wie in [98, 99] in einem dichten Feld des optischen Flusses. Dort wird der Algorithmus [33] in der OpenCV-Implementierung [70] verwendet. Sei $\mathbf{x}_{t-1}$ die Position eines Aktionspunktes zum Zeitpunkt $t-1$ und $\mathbf{f}(\mathbf{x},t) = [f^x(\mathbf{x},t), f^y(\mathbf{x},t)]$ der optische Fluss zwischen den Bildern $\mathbf{i}(t-1)$ und $\mathbf{i}(t)$, dann wird die Trajektorienposition im Schritt t zunächst initialisiert gemäß

$$\mathbf{x}_t^* = \mathbf{x}_{t-1} + \mathbf{med}_{\mathcal{W}}\left(\mathbf{f}(\mathbf{x}_{t-1},t)\right) . \tag{4.51}$$

Dabei ist $\mathbf{med}_{\mathcal{W}}\left(\mathbf{f}(\mathbf{x}_t,t),\right)$ ein zweidimensionales Medianfilter, angewandt auf die beiden Komponenten des optischen Flusses in einer Umgebung $\mathcal{W}$ um die Position $\mathbf{x}_{t-1}$.

Die so ermittelte vorläufige Position wird im nächsten Schritt korrigiert mit Hilfe des zuvor initialisierten Objektmodells des aktuellen Merkmals-punktes. Zur Bestimmung der endgültigen Punktposition werden *Mean Shift*-Iterationen gemäß den Gleichungen (4.42) und (4.43) ausgeführt, um zum nächsten lokalen Maximum des Bhattacharyya-Koeffizienten fortzuschreiten.

Das Mean-Shift-Verfahren ist konvergent und findet in der Regel bereits nach wenigen Iterationen ein lokales Maximum. Allerdings gibt es keine Garantie dafür, dass dieses Maximum tatsächlich eine hohe Übereinstimmung zwischen Ziel- und Kandidatenmodell aufweist. Daher wird eine minimale Ähnlichkeit $\rho_{\min}$ gefordert. Liegt der Bhattacharyya-Koeffizient von Ziel- und Kandidatenmodell an der finalen Position unterhalb des geforderten Wertes, wird das Tracking als gescheitert betrachtet und der Punkt gilt als verloren.

4.3.4. Trajektorien-Deskriptoren

Nach dem Tracking stellt sich die Frage, welche Arten der *Repräsentation* für die gewonnenen Trajektorien geeignet sind. Die ermittelten Trajektorien werden durch unterschiedliche Deskriptoren dargestellt. Diese bilden die Grundlage der späteren Aktionsanalyse. Um auch bei komplexeren Szenarien eine gute Aktivitätserkennung zu erreichen, müssen neben den Positionsverläufen der Trajektorien weitere Informationen hinzugezogen werden. Hierzu werden Merkmale über Textur und Bewegung in lokalen Nachbarschaften um die Trajektorien gebildet.

Der Verlauf der Punktverschiebungen $\mathbf{v}_{j,t}$ einer Trajektorie liegt bereits vor. Daraus resultiert der *Dynamik-Deskriptor*

$$\mathbf{d}_{j,t}^{\mathrm{dyn}} = \mathbf{v}_{j,t}\,. \tag{4.52}$$

Zusätzlich dazu werden aus Volumen um die Trajektorien weitere Merkmale extrahiert. Diese beinhalten Informationen über Textur und Bewegung in einer lokalen Umgebung der Trajektorie. Im Folgenden werden unterschiedliche Deskriptoren erläutert, deren Eignung für die Aktionsanalyse in Kapitel 5 untersucht wird. In [98] wird eine Kombination von HOG-, HOF- und MBH-Deskriptoren zur Repräsentation

von Merkmals-trajektorien vorgeschlagen. Diese Methode hat sich als sehr erfolgreich erwiesen und wurde in einer Reihe weiterer Arbeiten in gleicher oder ähnlicher Form verwendet [46, 47, 99]. Zur Verwendung als Trajektorien-Deskriptoren werden nicht wie bei STIP-Deskriptoren Kuboide zur Merkmalsbildung herangezogen, sondern Volumen um die Trajektorien. Diese Volumen werden üblicherweise ebenso in örtliche und zeitliche Zellen eingeteilt, in [98] werden $n_\sigma = 3$ örtliche und $n_\tau = 2$ zeitliche Zellen gewählt. Die Verwendung zeitlicher Raster ist in den o. g. Fällen geeignet, da Trajektorien fester Länge vorliegen.

Eine Gruppe der hier betrachteten Merkmale bilden durch [98] inspirierte HOG-, HOF- und MBH-basierte Histogramm-Deskriptoren. In dieser Arbeit liegen Trajektorien unterschiedlicher Länge vor, daher wird im Gegensatz zu [98] auf ein zeitliches Raster verzichtet. Die HOG-Merkmale werden ohne Berücksichtigung des Vorzeichens der Gradientenrichtung bestimmt. Bei den Merkmalen des optischen Flusses werden Histogramme mit Betrachtung des Vorzeichens der Flussvektoren gebildet.

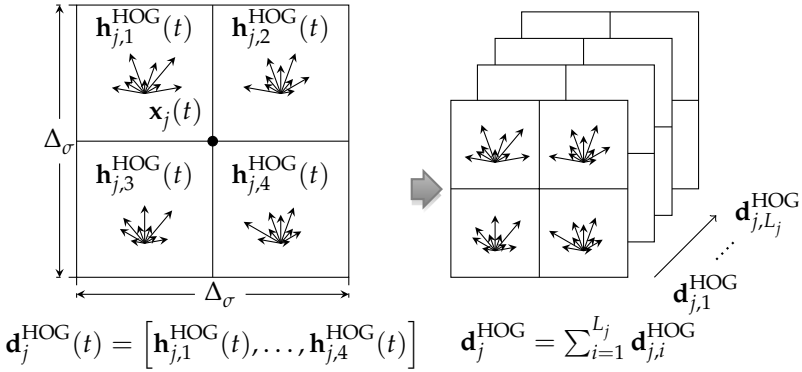


Abbildung 4.11. Veranschaulichung der Bestimmung des HOG-Deskriptors bei Verwendung eines 2×2- Rasters.

Abbildung 4.11 veranschaulicht die Bestimmung des HOG-Deskriptors. Für eine Trajektorie j wird in jedem Zeitschritt, in dem die Trajektorie aktiv ist, ein Bildbereich der Größe $\Delta_\sigma \times \Delta_\sigma$ um die aktuelle Trajektorienposition $\mathbf{x}_{j,t}$ zur Merkmalsbildung herangezogen. Dieser

Bildbereich wird in ein Raster mit $n_\sigma \times n_\sigma$ Zellen unterteilt. In einem Zeitschritt t wird in jeder Zelle c das Histogramm $\mathbf{h}_{j,c,t}^{\mathrm{HOG}}$ berechnet. Die Histogramme der einzelnen Zellen $c = 1, \ldots, C$ werden zusammengefügt, um das HOG-Merkmal des Zeitpunktes t zu bilden

$$\mathbf{d}_{j,t}^{\mathrm{HOG}} = \left[\mathbf{h}_{j,c,t}^{\mathrm{HOG}}, \ldots, \mathbf{h}_{j,C,t}^{\mathrm{HOG}} \right] . \tag{4.53}$$

Auf die entsprechende Weise werden die HOF-Merkmale $\mathbf{d}_{j,t}^{\mathrm{HOF}}$ aus dem bereits für das Tracking bestimmten optischen Fluss gebildet. Für die *Motion Boundary*-Histogramme ergeben sich nach diesem Vorgehen zwei Deskriptoren $\mathbf{d}_{j,t}^{\mathrm{MBH_x}}$ und $\mathbf{d}_{j,t}^{\mathrm{MBH_y}}$.

Für die Repräsentation der gesamten Trajektorie werden die Histogramm-Merkmale über die gesamte Dauer der Trajektorie gemittelt, um eine einheitliche, kompakte Form zu erhalten. Damit ergibt sich für den HOG- und HOF-Deskriptor:

$$\mathbf{d}_{j}^{\mathrm{HOG}} = \sum_{i=1}^{L_j} \mathbf{d}_{j,i}^{\mathrm{HOG}} , \tag{4.54}$$

$$\mathbf{d}_{j}^{\mathrm{HOF}} = \sum_{i=1}^{L_j} \mathbf{d}_{j,i}^{\mathrm{HOF}} . \tag{4.55}$$

Dabei ist L_j die Länge der j-ten Trajektorie.

Die einzelnen MBH-Deskriptoren können entweder als separate Merkmale betrachtet oder zu einem gemeinsamen Merkmalsvektor zusammengefügt werden:

$$\mathbf{d}_{j}^{\mathrm{MBH_x}} = \sum_{i=1}^{L_j} \mathbf{d}_{j,t}^{\mathrm{MBH_x}} , \tag{4.56}$$

$$\mathbf{d}_{j}^{\mathrm{MBH_y}} = \sum_{i=1}^{L_j} \mathbf{d}_{j,t}^{\mathrm{MBH_y}} , \tag{4.57}$$

$$\mathbf{d}_{j}^{\mathrm{MBH}} = \left[\mathbf{d}_{j}^{\mathrm{MBH_x}}, \mathbf{d}_{j}^{\mathrm{MBH_y}} \right] . \tag{4.58}$$

Inspiriert vom Erfolg des SURF-Deskriptors und seinen Vorzügen gegenüber den auf Histogrammen basierenden SIFT-Merkmalen (höhere Robustheit gegenüber kleinen Variationen, einfachere Berechnung und geringere Merkmalsdimension, da keine Histogramme verwendet werden) werden in dieser Arbeit Modifikationen der zuvor betrachteten Deskriptoren vorgeschlagen, die sich am Aufbau der SURF-Merkmale orientieren. Diese Deskriptor-Familie wird im Folgenden als Summen-Deskriptoren bezeichnet.

Zur Bildung eines Gradienten-Merkmals wird der betrachtete Bildbereich wie zuvor in $n_\sigma \times n_\sigma$ Zellen eingeteilt. Nun werden anstelle von Orientierungs-Histogrammen ähnlich wie beim SURF-Deskriptor die Gradienten sowie ihre Beträge jeweils in x- und y-Richtung aufsummiert. Für eine Zelle c ergibt sich somit

$$\mathbf{k}_{j,c,t}^{\mathrm{SG}} = \left[\sum_{\mathbf{x}\in\mathcal{X}_c} i_x(\mathbf{x},t), \sum_{\mathbf{x}\in\mathcal{X}_c} i_y(\mathbf{x},t), \sum_{\mathbf{x}\in\mathcal{X}_c} |i_x(\mathbf{x},t)|, \sum_{\mathbf{x}\in\mathcal{X}_c} |i_y(\mathbf{x},t)| \right] . \quad (4.59)$$

Die Menge $\mathcal{X}_c$ bezeichnet dabei alle Punktkoordinaten in der c-ten Zelle. Die Merkmale der einzelnen Zellen werden zu einem Vektor $\mathbf{d}_{j,t}^{\mathrm{SG}}$ zusammengefügt. Zur Repräsentation der gesamten Trajektorie wird wie früher das mittlere Merkmal betrachtet:

$$\mathbf{d}_j^{\mathrm{SG}} = \sum_{i=1}^{L_j} \mathbf{d}_{j,i}^{\mathrm{SG}} . \quad (4.60)$$

Das resultierende Merkmal wird als *Sum of Gradients* (SG)-Deskriptor bezeichnet.

Analog dazu werden die *Sum of Optical Flow* (SOF)- und die *Motion Boundary Sum* (MBS)-Deskriptoren ermittelt:

$$\mathbf{d}_j^{\mathrm{SOF}} = \sum_{i=1}^{L_j} \mathbf{d}_{j,i}^{\mathrm{SOF}}\,, \tag{4.61}$$

$$\mathbf{d}_j^{\mathrm{MBS_x}} = \sum_{i=1}^{L_j} \mathbf{d}_{j,i}^{\mathrm{MBS_x}}\,, \tag{4.62}$$

$$\mathbf{d}_j^{\mathrm{MBS_y}} = \sum_{i=1}^{L_j} \mathbf{d}_{j,i}^{\mathrm{MBS_y}}\,, \tag{4.63}$$

$$\mathbf{d}_j^{\mathrm{MBS}} = \left[\mathbf{d}_j^{\mathrm{MBS_x}}, \mathbf{d}_j^{\mathrm{MBS_y}}\right]\,. \tag{4.64}$$

Da während des Trackings bereits LBPs in der Umgebung der Trajektorien ermittelt werden, wird aus diesen ein weiterer Deskriptor gebildet. Hierzu wird der Bildbereich wie oben in Zellen eingeteilt und in jeder Zelle das Histogramm der Hauptmuster $\mathbf{h}_{j,c,t}^{\mathrm{LBP}}$ bestimmt. Die Histogramme der Zellen werden zu einem Merkmal zusammengefügt und die Merkmale der einzelnen Zeitschritte gemittelt:

$$\begin{aligned}\mathbf{d}_{j,t}^{\mathrm{LBP}} &= \left[\mathbf{h}_{j,c,t}^{\mathrm{LBP}}, \ldots, \mathbf{h}_{j,C,t}^{\mathrm{LBP}}\right]\,,\\ \mathbf{d}_j^{\mathrm{LBP}} &= \sum_{i=1}^{L_j} \mathbf{d}_{j,i}^{\mathrm{LBP}}\,.\end{aligned} \tag{4.65}$$

4.4. Versuche

In diesem Abschnitt wird auf Versuche des Merkmalstrackings eingegangen. Im Gegensatz zum Körper-Tracking ist es bei merkmalsbasierter Bewegungserfassung äußerst schwierig, eine Grundwahrheit zur Bewertung der gewonnenen Trajektorien zu formulieren. Ohne diese kann allein basierend auf den Trajektorien kein Qualitätsmaß definiert werden. Das Ziel bei der Extraktion der Merkmalstrajektorien besteht

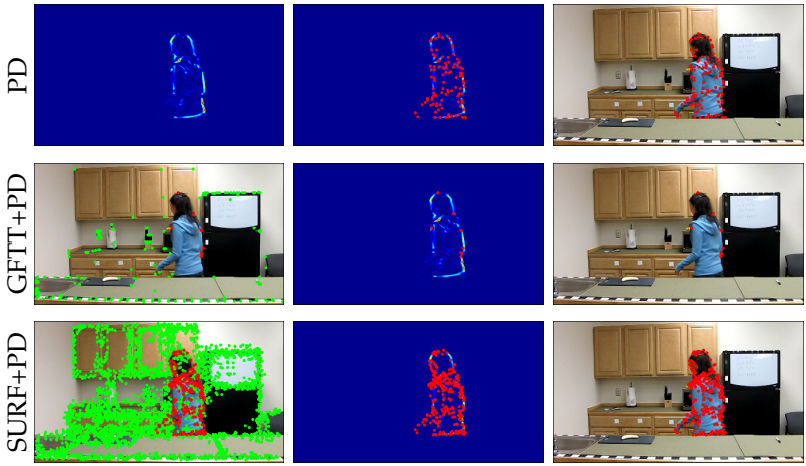


Abbildung 4.12. Beispiel für die Detektion von Aktionspunkten für den ADL-Datensatz mit dem periodischen Detektor (PD), der Kombination aus periodischem und *Good Features to Track*-Detektor (GFTT+PD) und der Kombination aus periodischem und SURF-Detektor (SURF+PD).

darin, basierend auf den ermittelten Deskriptoren eine möglichst gute Aktionserkennung zu erreichen. Diese stellt somit das wesentliche Bewertungskriterium über die Qualität der gewonnenen Trajektorien dar. Eine quantitative Bewertung der vorgestellten Methoden wird daher im Hinblick auf die Ergebnisse der Aktionserkennung in Kapitel 5 vorgenommen. Eine Evaluation und Vergleich der Methoden der Initialisierung und des Trackings der Aktionspunkte erfolgt im Rahmen eines Algorithmus zur Aktionserkennung in Abschnitt 5.6. In diesem Abschnitt wird zunächst eine qualitative Beurteilung durchgeführt. Es wird zunächst auf die Detektion und anschließend auf das Tracking der Aktionspunkte eingegangen.

4.4.1. Detektion von Aktionspunkten

Die Abbildungen 4.12, 4.13 und 4.14 zeigen jeweils Beispiele für die Aktionspunkt-Detektion für die drei in Abschnitt 4.3.2 erläuterten Detektionsmethoden. Die ersten beiden der verwendeten Sequenzen stammen

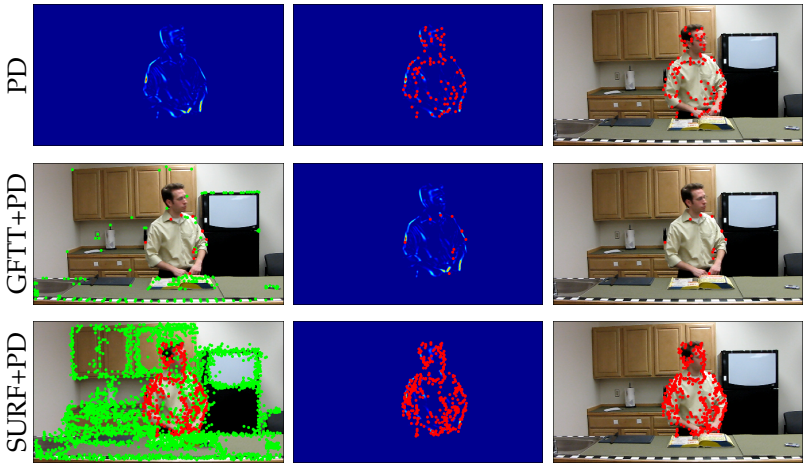


Abbildung 4.13. Beispiel für die Detektion von Aktionspunkten für den ADL-Datensatz mit dem periodischen Detektor (PD), der Kombination aus periodischem und *Good Features to Track*-Detektor (GFTT+PD) und der Kombination aus periodischem und SURF-Detektor (SURF+PD).

aus dem *Activities of Daily Living* (ADL)-Datensatz [62]. Dieser enthält Videos von Testpersonen, die alltägliche Aktivitäten in einem Küchenszenario ausführen (siehe Abschnitt 5.6). Da dieser Datensatz für die Entwicklung von Methoden des Merkmalstrackings erzeugt wurde, wurden die Videos in einer hohen Auflösung von 1280×720 Bildpunkten aufgenommen. Die Sequenz aus Abbildung 4.14 stammt aus dem *Hollywood Human Actions* (HOHA)-Datensatz [58]. Dieser besteht aus Ausschnitten von Hollywood-Filmen und enthält somit Aufnahmen unterschiedlicher Umgebungsbedingungen mit Kamerabewegungen, Beleuchtungsschwankungen, dynamischen Hintergründen etc.

Für jede der Detektionsmethoden werden maximal $N_{\max}^{\text{init}} = 50$ Aktionspunkte initialisiert. Die Detektionsschwelle des periodischen Detektors wird zu $r_{\min}^{\text{init}} = 200$ gewählt. Mit dem GFTT- und SURF-Detektor werden zunächst bis zu $N^{\text{ip}} = 200$ Bildmerkmale extrahiert. Die Detektion der SURF-Merkmale erfolgt mittels der Implementierung aus der *OpenCV*-Bibliothek [70].

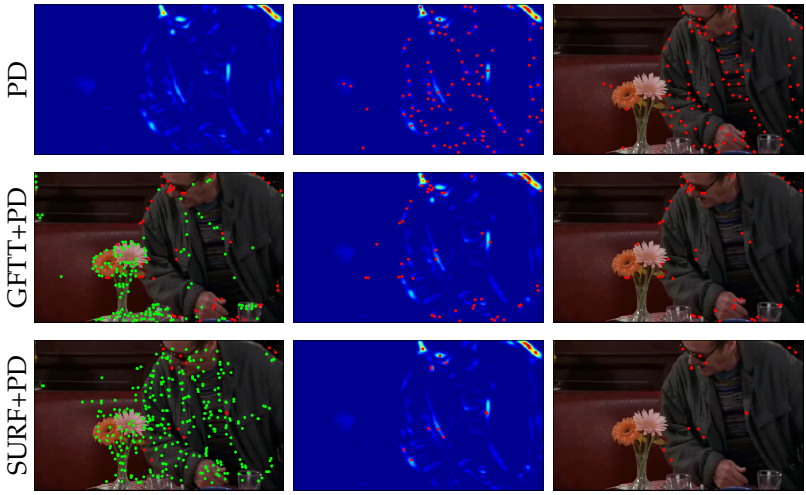


Abbildung 4.14. Beispiel für die Detektion von Aktionspunkten für den HOHA-Datensatz mit dem periodischen Detektor (PD), der Kombination aus periodischem und *Good Features to Track*-Detektor (GFTT+PD) und der Kombination aus periodischem und SURF-Detektor (SURF+PD).

In der ersten Zeile ist links zunächst die Antwort des periodischen Detektors abgebildet. In der Mitte sind die mit dem periodischen Detektor ermittelten Punkte auf dem Bewegungs- und rechts auf dem Eingangsbild zu sehen. Die zweite Zeile zeigt die Detektion für die Kombination aus GFTT und periodischem Detektor (GFTT+PD). Links sind die detektierten GFTT-Bildmerkmale in Grün und die mit dem kombinierten Akzeptanzkriterium selektierten Aktionspunkte in Rot auf dem Eingangsbild abgebildet. Die nächsten beiden Bilder zeigen, wie in der ersten Zeile, die ausgewählten Punkte auf dem Bewegungs- und Originalbild. In der dritten Zeile sind die entsprechenden Darstellungen für die Verwendung des SURF+PD-Detektors zu sehen.

Als Bewertungskriterien für die detektierten Punkte können die Menge und Qualität der Bildmerkmale herangezogen werden. In fast allen Fällen wird erreicht, dass nur Punkte in Bildbereichen initialisiert werden, in denen Bewegung auftritt. Im Hintergrund detektierte Bildmerkmale werden erfolgreich vom periodischen Detektor gefiltert. Für den

ADL-Datensatz liefert die SURF+PD-Detektion die größte Anzahl an Punkten. Der GFTT-Detektor mit der automatisch bestimmten Schwelle nach Gleichung (4.50) detektiert dagegen nur sehr wenige Bildmerkmale.

Dies wirkt sich natürlich auf die gesamte Anzahl an Trajektorien aus, die aus einer Videosequenz gewonnen werden. Dazu wird mit den drei Detektionsmethoden jeweils ein Merkmalstracking durchgeführt, wobei alle anderen Parameter wie in Tabelle 4.2 gewählt werden. Bei Verwendung des GFTT+PD-Detektors mit den oben angegebenen Parametern werden so im Mittel 367 Trajektorien aus einem Video extrahiert. Bei der Initialisierung mit dem periodischen Detektor sind es 719 und mit SURF+PD 750 Trajektorien pro Video (siehe Tabelle 4.1). Die mit SURF+PD initialisierten Punkte sind dichter als die mit dem periodischen Detektor alleine detektierten Merkmale, obwohl in beiden Fällen die gleiche Schwelle $r_{\text{min}}^{\text{init}}$ verwendet wurde. Der Unterschied der beiden Varianten ist darin begründet, dass bei der PD-Detektion lokale Maxima des periodischen Detektors gesucht werden, während bei SURF+PD für die detektierten SURF-Merkmale lediglich geprüft wird, ob der Wert des periodischen Detektors an den entsprechenden Koordinaten den Schwellwert $r_{\text{min}}^{\text{init}}$ übersteigt. Für den HOHA-Datensatz werden dagegen mehr Punkte mit GFTT+PD als mit SURF+PD ermittelt. Außerdem werden hier aufgrund von Kamerabewegungen mehr Punkte im Hintergrund detektiert.

Die Qualität der Initialisierung bezieht sich hier auf die Fähigkeit, die Punkte zu tracken, um eine Aktionserkennung zu ermöglichen. Die Bewertung der Qualität der verschiedenen Detektoren erfolgt daher in Kapitel 5 anhand von Ergebnissen der Aktionserkennung. In Abschnitt 5.6.1 werden dazu Ergebnisse mit den vorgestellten Trajektorien-Merkmalen für verschiedene Initialisierungen miteinander verglichen.

4.4.2. Merkmalstracking

Bild 4.15 zeigt einige Beispiele extrahierter Trajektorien. Dabei sind den dargestellten Bildern die Punktpositionen der Trajektorien vergangener Zeitschritte im Zeitraum von einer Sekunde überlagert. Die Trajektorien weisen einen glatten Verlauf auf. Es gelingt, die Punkte über einen längeren Zeitraum erfolgreich zu verfolgen. Im unteren mittleren Beispiel

des HOHA-Datensatzes treten verursacht durch Kamerabewegungen auch einige Trajektorien im Hintergrund auf.

Tabelle 4.1. Trajektorien-Kennwerte für verschiedene Initialisierungsmethoden für den ADL-Datensatz, gemittelt über alle Videos aller Klassen. Die restlichen Trackingparameter sind in Tabelle 4.2 aufgelistet. Die mittlere Trajektorienlänge $\overline{L}$, der Mittelwert der maximalen Länge $\overline{L}_{\max}$ sowie die Standardabweichung der Punktkoordinaten $\overline{\sigma}$ liegen für alle Methoden in einem ähnlichen Bereich. Die Gesamtzahl an resultierenden Trajektorien ist bei der Initialisierung mit GFTT+PD deutlich geringer als bei den anderen Methoden.

Initialisierung	$\overline{L}$	$\overline{L}_{\max}$	$\overline{\sigma}$	$\overline{N}$
PD	28	118	32,01	719
GFTT+PD	30	112	32,81	367
SURF+PD	27	116	31,94	750

Abbildung 4.15. Beispiele für extrahierte Trajektorien.

In Tabelle 4.3 sind Kennwerte der Trajektorien des ADL-Datensatzes für die einzelnen Klassen gegeben. Die verwendeten Parameter des Trackings sind in Tabelle 4.2 aufgelistet. Für alle Versuche wurde der SURF+PD-Detektor verwendet. Die mittlere Länge der Trajektorien $\overline{L}$ und die Standardabweichung der Punktpositionen liegen für alle Klassen in einem ähnlichen Bereich. Es kann somit ausgeschlossen werden, dass diese Werte die Klassifikation maßgeblich beeinflussen. Verwandte Arbeiten, die Merkmale nur während einer kurzen Zeitdauer verfolgen,

begrenzen die Trajektorienlänge häufig auf $L_{\max} = 15$. Hier ist die mittlere Länge fast doppelt so groß, und es werden auch deutlich längere Trajektorien über einen Zeitraum von bis zu ca. fünf Sekunden gewonnen. Auf die Bedeutung der Trajektoriendauer für die Aktionserkennung wird in Abschnitt 5.6.1 eingegangen.

Tabelle 4.4 vergleicht die Rechenzeit der in diesem Abschnitt verwendeten Methode mit Angaben verwandter Arbeiten aus der Literatur. Die Rechenzeit des hier durchgeführten Merkmalstrackings ist mit dem Stand der Technik vergleichbar. Es sei angemerkt, dass die Implementierung nicht für maximale Effizienz optimiert wurde und Einsparungen möglich sind. Bei den Angaben handelt es sich um durchschnittliche Werte. Sie beinhalten die Detektion und Verfolgung der Aktionspunkte sowie de Berechnung der Dynamik-, HOG-, HOF-, MBH- und LBP-Deskriptoren. Es wurden hier nicht die Berechnungszeiten aller in Abschnitt 4.3.4 vorgestellten Deskriptoren miteinbezogen, da die Verwendung der Histogramm- und Summen-Deskriptoren als Alternativen zueinander aufzufassen sind.

Tabelle 4.2. Parameter des Merkmalstrackings der hier dargestellten Versuche bei Detektion mit SURF+PD.

Parameter	Wert
Detektion der Aktionspunkte	SURF+PD
Skalenfaktoren des periodischen Detektors	$\sigma = 3, \tau = 1$
Anzahl SURF-Merkmale N^{ip}	200
Detektionsschwelle Bewegung $r_{\min}^{\text{init}}$	200
Fenster Initialisierung	7×7
Maximal gleichzeitig aktive Punkte $N_{\text{a,max}}$	100
Schwelle Verschiebung statischer Punkte $v_{\min}$	1,5
Zeitschritte bis Prüfung statischer Punkt τ_{static}	15
Zeitschritte bis Prüfung inaktiver Punkt τ_{d}	15
Maximale Trajektorienlänge	$L_{\max} = \infty$
Größe Suchfenster optischer Fluss $\mathcal{W}$	3×3
Größe Suchfenster S_h bei *Mean Shift*	9×9
Verwendete Farbkanäle	rot, blau
Schwelle Ähnlichkeitsmaß $\rho_{\min}$	0,70

Tabelle 4.3. Trajektorien-Kennwerte nach Klassen für eine Parametrierung des Aktionspunkt-Trackings für den ADL-Datensatz. Die Werte sind jeweils über alle Videos einer Klasse gemittelt. $\overline{L}$ ist die mittlere Länge aller Trajektorien. $\overline{L}_{\max}$ ist der Mittelwert der maximalen Trajektorienlänge. $\overline{\sigma}$ ist die Standardabweichung der Punktpositionen innerhalb einer Trajektorie und $\overline{N}$ ist die mittlere Anzahl an Trajektorien.

Klasse	$\overline{L}$	$\overline{L}_{\max}$	$\overline{\sigma}$	$\overline{N}$
Telefon abnehmen	28	86	35	256
Banane schneiden	28	110	38	506
Anruf tätigen	25	73	27	300
Wasser trinken	29	143	37	890
Banane essen	22	67	24	479
Snack essen	28	139	28	1488
In Telefonbuch nachschlagen	26	141	27	1471
Banane schälen	30	160	29	814
Mit Besteck essen	29	145	32	955
Auf Whiteboard schreiben	28	93	43	339

Tabelle 4.4. Vergleich des Rechenaufwandes der Trajektorienextraktion mit verwandten Arbeiten in der Literatur. Der Rechenaufwand wird in *Frames* pro Sekunde (fps) angegeben.

Methode	Verarbeitungszeit / fps
Saliente Trajektorien [110]	0,63 fps bei 360×240
Spärliche Bewegungstrajektorien [47]	35 fps bei 180×144
Langzeit-Trajektorien [91]	5 fps bei 320×240
Diese Arbeit	2,50 fps bei 240×400

5. Aktivitätsanalyse

5.1. Einleitung

Nachdem bestimmte Merkmale aus einer Videosequenz extrahiert wurden, erfolgt basierend darauf eine Modellierung verschiedener Aktivitäten, um eine Erkennung zu ermöglichen. Hierzu werden zwei Herangehensweisen betrachtet – der „*Bag of Words*“ (BoW)-Ansatz und ein sequenzielles Modell.

Zunächst wird eine Aktionserkennung mittels der in Kapitel 4 gewonnenen Merkmalstrajektorien durchgeführt. Dazu wird das sehr weit verbreitete BoW-Modell herangezogen, welches in den meisten verwandten Arbeiten eingesetzt wird. Die zugrundeliegende Idee dabei ist, lediglich zu modellieren, *welche* Merkmale in einer Sequenz auftreten. Die Frage, in welcher örtlichen oder zeitlichen Konfiguration die einzelnen Merkmale zueinander stehen, wird dabei nicht oder nur schwach berücksichtigt.

Zur Modellierung von Aktionen werden unterschiedliche Merkmale herangezogen, um möglichst komplementäre Informationsarten auszunutzen. Diese sind in Abb. 5.1 veranschaulicht. Es interessiert, *was* in einer Szene auftritt (Bildinformation) und *wie* sich dies verhält (Information über auftretende Bewegung). Dies wird durch die unterschiedlichen Trajektorien-Deskriptoren in das Modell eingebracht.

Obgleich der BoW-Ansatz sich in vielen Fällen als sehr mächtiges Werkzeug erwiesen hat, ist die Struktur des Auftretens der Merkmale v. a. für komplexe Aktivitäten durchaus wichtig. Es spielt eine Rolle, *wann* und in welchem Zusammenhang, örtlich sowie zeitlich, verschiedene Ereignisse auftreten. Besonders die Betrachtung der zeitlichen Abfolge der Merkmale wird meist nicht oder nur schwach berücksichtigt. Solche Zusammenhänge können in gewissem Maße dem Modell hinzugefügt werden. Die Frage, *wo* Merkmale auftreten, wird durch die

getrennte Betrachtung lokaler Bildbereiche modelliert. Auf die Berücksichtigung zeitlicher Zusammenhänge wird im Rahmen dieser Methode verzichtet, dies erfolgt bei der sequenziellen Modellierung.

Beim BoW-Modell wird davon ausgegangen, dass in einer gegebenen Sequenz nur eine Aktion auftritt, und es wird ein globaler Merkmalsvektor für das gesamte Video erstellt. Für die Verarbeitung der üblichen Datensätze ist dieses Vorgehen geeignet, da diese bereits segmentierte Aktionen enthalten. In praktischen Anwendungen liegen jedoch kontinuierliche Aufnahmen vor, so dass nicht nur erkannt werden muss, welche Aktionen vorliegen, sondern auch die Zeitintervalle ermittelt werden müssen, in denen sie auftreten.

Daher wird alternativ zum BoW-Ansatz eine sequenzielle Modellierung vorgenommen. Aufgrund des Erfolges diskriminativer Modelle [89] wird ein lineares kettenförmiges *Conditional Random Field* (CRF) verwendet. Dieses modelliert Zusammenhänge zwischen einer Abfolge von Aktionen und einer Beobachtungssequenz. Dabei wird zu jedem Zeitpunkt ein Merkmalsvektor gebildet und die resultierende Sequenz von Merkmalsvektoren zur Modellierung herangezogen. Die Verwendung des CRF erlaubt eine sehr flexible Repräsentation zeitlicher Zusammenhänge zwischen Merkmalen.

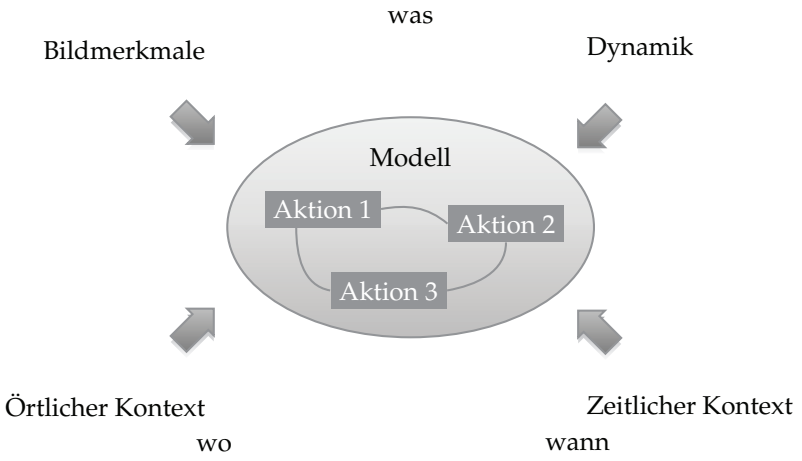


Abbildung 5.1. Betrachtete Informationsarten beim merkmalsbasierten Aktivitätsmodell.

Das Sequenzmodell wird angewandt, um auf Basis der Deskriptoren der Merkmalstrajektorien eine gemeinsame Segmentierung und Erkennung von Aktionen durchzuführen. Außerdem wird seine Eignung zur Modellierung von Posenverläufen untersucht, indem es auf *Motion Capture*-Daten sowie Ergebnisse des Körper-Trackings aus Kapitel 3 angewandt wird.

In Abschnitt 5.2 werden zunächst einige Grundlagen zum maschinellen Lernen gegeben. Abschnitt 5.3 stellt die Theorie probabilistischer Sequenzmodelle dar, welche für die sequenzielle Aktivitätsmodellierung verwendet werden. Anschließend werden in Abschnitt 5.4 die Aktionserkennung mit dem BoW-Ansatz und in Abschnitt 5.5 die sequenzielle Modellierung erläutert. In Abschnitt 5.6 werden schließlich Versuche und Ergebnisse besprochen.

5.2. Grundlagen

In diesem Abschnitt wird eine Einführung in die für diese Arbeit relevanten Grundlagen der Klassifikation und des maschinellen Lernens gegeben. Zunächst wird eine kurze Einführung in die Thematik gegeben, bevor in Abschnitt 5.2.2 auf die Klassifikation mit *Support Vector Machines* eingegangen wird. In Abschnitt 5.2.3 wird die bei der Vektorquantisierung eingesetzte *K-means*-Clusteringmethode erläutert.

5.2.1. Klassifikation und maschinelles Lernen

Die Aufgabe der Klassifikation besteht darin, Objekte in Kategorien – *Klassen* – einzuteilen, basierend auf Informationen bestimmter Merkmale, die über die Objekte vorliegen [31].

Ein Ansatz dazu ist das Lernen aus Beispielen. Dabei ist eine Menge an Objekten vorhanden, die typisch für die betrachteten Klassen sind. Wenn die Klassen und die Klassenzugehörigkeiten der Beispielobjekte bekannt sind, spricht man von *überwachtem* Lernen. Sollen zunächst geeignete Klassen ermittelt werden, wird der Lernprozess als *unüberwacht* bezeichnet [31]. Beim überwachten Lernen liegt der Fokus also auf einer möglichst genauen Prädiktion der Klasse neuer Objekte, während beim unüberwachten Lernen das Hauptanliegen ist, eine genaue und kompak-

te Darstellung der Daten zu finden [6]. Zu Methoden des unüberwachten Lernens zählen beispielsweise Verfahren der Clusteranalyse oder der Dimensionsreduktion [6, 31, 39].

Überwachtes Lernen

Im Folgenden wird eine Übersicht des überwachten Lernens gegeben, die sich an [6] orientiert. Dadurch soll die in dieser Arbeit verwendete Notation vorgestellt werden.

Die betrachteten Objekte werden durch Merkmalsvektoren $\mathbf{y}$ repräsentiert. Jedes Objekt besitzt eine zugehörige Ausgangsvariable $x \in \{1, \ldots, S\}$, welche die Zugehörigkeit zu den Klassen $s \in \{1, \ldots, S\}$ angibt. Es sei eine Menge annotierter Merkmale, d. h. Paare von Merkmalen und ihren zugehörigen Klassenvariablen, $\mathcal{D}^{\text{train}} = \{(\mathbf{y}_i, x_i),\ i = 1, \ldots, N^{\text{train}}\}$ vorhanden.

Die Aufgabe des überwachten Lernens besteht darin, bei gegebenen Trainingsdaten $\mathcal{D}^{\text{train}}$ den Zusammenhang zwischen der Eingangsgröße $\mathbf{y}$ und der Ausgangsgröße x so zu lernen, dass die prädizierte Klasse x^* für einen neues Merkmal $\mathbf{y}^*$ korrekt ist. Der Begriff *Training* bezeichnet den Vorgang, eine Entscheidungsfunktion zu lernen. Häufig entspricht dies einer Parameterschätzung für ein bestimmtes Modell. *Testen* bedeutet die Bestimmung des Wertes der Entscheidungsfunktion $x^*(y^*)$ für ein neues Merkmal, d. h. die Prädiktion des Ausgangswertes für einen bestimmten Eingangswert.

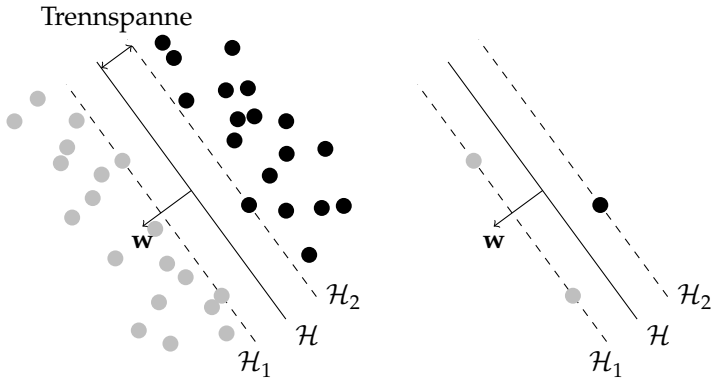
Die Ausgangsvariable x soll demnach beschrieben werden unter der Bedingung, dass $\mathbf{y}$ bekannt ist. Betrachtet man dies aus der Perspektive der probabilistischen Modellierung, ist die bedingte Verteilungsdichte $p(x|\mathbf{y}, \mathcal{D}^{\text{train}})$ gesucht.

In der hier gegebenen Erklärung wird davon ausgegangen, dass die Ausgangsvariable x eine endliche Anzahl an diskreten Werten, den sog. Klassen, annehmen kann. In diesem Fall spricht man von einem *Klassifikationsproblem*. Falls x ein kontinuierlicher Wert ist, spricht man von *Regression*.

5.2.2. Klassifikation mit Support Vector Machines

*Support Vector Machine*s (SVMs) sind Methoden des überwachten Lernens. Sie können zur Klassifikation oder Regression eingesetzt werden. In dieser Arbeit werden sie zur Klassifikation verwendet, auf ihre Anwendung zur Regression wird nicht eingegangen.

Das Lernen der Klassifikationsparameter einer SVM ist ein konvexes Optimierungsproblem, jede lokale Lösung entspricht also dem globalen Optimum [8]. SVMs gehören zu den *Maximum Margin*-Klassifikatoren, d. h. sie maximieren die Trennspanne zwischen den unterschiedlichen Klassen, um die bestmögliche Fähigkeit der Verallgemeinerung zu erlangen. Neben linearen Klassifikationsproblemen eignen sie sich auch für die nichtlineare Klassifikation, indem sie den *Kernel-Trick* anwenden.



(a) Lineares Zwei-Klassen-Problem **(b)** *Support*-Vektoren

Abbildung 5.2. Veranschaulichung der Klassifikation mit SVMs für ein linear separierbares Zwei-Klassen-Problem. Die Merkmale der Klasse s_1 sind als graue und die Merkmale der Klasse s_2 als schwarze Kreise dargestellt.

Zunächst wird von einem Zwei-Klassen-Problem ausgegangen, wie es beispielhaft in Abbildung 5.2(a) dargestellt ist. Es wird der einfachste Fall angenommen, dass die Daten *linear separierbar* sind. Gegeben seien Trainingsdaten $\mathcal{D}^{\text{train}} = \{(\mathbf{y}_i, x_i),\, i = 1, \ldots, N^{\text{train}}\}$, die aus N^{train} Paaren von Merkmalsvektoren $\mathbf{y}_i \in \mathbb{R}^D$ und zugehörigen Klassenvariablen $x_i \in \{-1, +1\}$ bestehen.

Gesucht ist eine lineare Trennfunktion, die die Trainingsmerkmale der beiden Klassen optimal trennt. Diese stellt eine *Hyperebene* im D-dimensionalen Raum dar:

$$\mathcal{H} = \{\mathbf{y} | \langle \mathbf{w}, \mathbf{y} \rangle + b = 0\}. \tag{5.1}$$

Die Trennebene $\mathcal{H}$ wird durch den Normalenvektor $\mathbf{w}$ und die Verschiebung b definiert.

Der Vorgang des Trainings ist nun, die Hyperebene so zu bestimmen, dass die Trainingsdaten getrennt werden. Die Klassifikation eines neuen Merkmals erfolgt durch Prüfen, auf welcher Seite der Ebene es liegt [8], was durch das Vorzeichen des Ausdrucks

$$z(\mathbf{y}) = \langle \mathbf{w}, \mathbf{y} \rangle + b \tag{5.2}$$

angegeben wird. $\mathbf{y}$ wird der Klasse s_1 zugeordnet, wenn $z(\mathbf{y}) \geq 0$ ist und der Klasse s_2, wenn $z(\mathbf{y}) < 0$ ist. Die Entscheidungsgrenze liegt also bei $z(\mathbf{y}) = 0$.

Die Lösung für dieses Problem ist nicht eindeutig. Es gibt unendlich viele Ebenen, die die Daten trennen können. Daher stellt sich die Frage nach der „besten“ Ebene. SVMs verfolgen die Strategie, diejenige Lösung zu wählen, die sich am besten verallgemeinern lässt.

Hierzu wird der Begriff der *Trennspanne* (engl. „*margin*“) eingeführt. Die Trennspanne ist der kleinste Abstand zwischen der Trennebene und den Trainingsmerkmalen. SVMs gehören zu den *Maximum Margin*-Klassifikatoren. Die Entscheidungsgrenze wird so gewählt, dass die Trennspanne maximal wird. Die Aufgabe des Trainings kann somit folgendermaßen formuliert werden: Finde die Hyperebene, die die beiden Klassen trennt und die Trennspanne maximiert. Um *Overfitting* zu vermeiden, wird in der Praxis die Forderung nach der genauen Trennung etwas gelockert, so dass einige Fehler erlaubt werden.

In Abbildung 5.2 wird die Trennspanne mittels zweier Hilfsebenen angezeigt, die parallel zu $\mathcal{H}$ sind und durch die Punkte mit minimalem Abstand zu $\mathcal{H}$ gehen. Es ist ersichtlich, dass sich die größte Trennspanne ergibt, wenn die Hyperebene gerade in der Mitte zwischen $\mathcal{H}_1$ und $\mathcal{H}_2$ liegt. Um die Trennspanne zu ermitteln, wird nun je ein Punkt der beiden Klassen $(\mathbf{y}_1, x_1 = 1)$ und $(\mathbf{y}_2, x_2 = -1)$ betrachtet, der jeweils gerade

den minimalen Abstand annimmt, d. h. $\mathbf{y}_1$ liegt auf $\mathcal{H}_1$ und $\mathbf{y}_2$ liegt auf $\mathcal{H}_2$. Die Trennebene $\mathcal{H}$ sei so skaliert, dass

$$\begin{aligned} \langle \mathbf{w}, \mathbf{y}_1 \rangle + b &= 1\,, \\ \langle \mathbf{w}, \mathbf{y}_2 \rangle + b &= -1 \end{aligned} \tag{5.3}$$

gilt. Die Abstände dieser beiden Punkte zur Hyperebene lauten:

$$\begin{aligned} d(\mathcal{H}, \mathbf{y}_1) &= x_1 \left(\left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \mathbf{y}_1 \right\rangle + \frac{b}{\|\mathbf{w}\|} \right) = \frac{1}{\|\mathbf{w}\|}\,, \\ d(\mathcal{H}, \mathbf{y}_2) &= x_2 \left(\left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \mathbf{y}_2 \right\rangle + \frac{b}{\|\mathbf{w}\|} \right) = \frac{1}{\|\mathbf{w}\|}\,. \end{aligned} \tag{5.4}$$

Die Trennspanne ist also gerade $\frac{1}{\|\mathbf{w}\|}$. Um die Trennspanne zu maximieren, muss $\|\mathbf{w}\|$ minimiert werden. Dies entspricht der Minimierung von $\|\mathbf{w}\|^2$. Es ergibt sich das Optimierungsproblem mit Nebenbedingungen

$$\begin{aligned} &\|\mathbf{w}\|^2 \;\rightarrow\; \min \\ &\text{mit } x_i \cdot (\langle \mathbf{w}, \mathbf{y}_i \rangle + b) \geq 1\,, \quad i = 1, \ldots, N^{\text{train}}\,. \end{aligned} \tag{5.5}$$

Zur Lösung wird ein Lagrange-Ansatz gewählt [8, 15]. Dafür gibt es zwei Gründe [15]: Erstens werden die Randbedingungen in Gleichung (5.5) durch Randbedingungen der Lagrange-Multiplikatoren selbst ersetzt, welche sich leichter handhaben lassen. Zweitens werden die Trainingsmerkmale nur noch als Skalarprodukte zwischen Vektoren auftauchen. Diese Eigenschaft wird später bei der Erweiterung auf nichtlineare SVMs nützlich sein. Die Lösung des Optimierungsproblems kann in Anhang A.2 nachvollzogen werden.

Es ergibt sich folgende Entscheidungsfunktion, um die Klasse x^* eines neuen Merkmals $\mathbf{y}^*$ zu bestimmen:

$$x^* = \operatorname{sign}\left(\sum_{i=1}^{N^{\text{train}}} \alpha_i x_i \langle \mathbf{y}_i, \mathbf{y}^* \rangle + b \right)\,. \tag{5.6}$$

Dabei gibt es für jedes der N^{train} Trainingsmerkmale einen Lagrange-Multiplikator α_i, $i = 1, \ldots, N^{\text{train}}$.

Die Trainingspunkte, für die $\alpha_i \neq 0$ gilt, sind die *Support-Vektoren*. Dies sind gerade die Punkte, die am nächsten an der trennenden Hyperebene liegen. Abbildung 5.2(b) zeigt die *Support*-Vektoren für das verwendete Beispiel. Für die Entscheidung werden nur noch diese Punkte benötigt. Würde man alle anderen Trainingspunkte entfernen und das Training wiederholen, würde man die gleiche Hyperebene erhalten [15].

Im Folgenden wird darauf eingegangen, wie mit SVMs *nichtlineare* Klassifikatoren entworfen werden können. Man stelle sich eine Problemstellung vor, bei der die Daten nicht linear trennbar in $\mathbb{R}^D$ sind. Möglicherweise gelingt aber eine lineare Trennung der Daten in einem höherdimensionalen Raum. Dann können die Daten mittels einer Merkmalstransformation $\boldsymbol{\phi}(\mathbf{y})$ in eine neue Darstellung überführt werden und dort durch eine lineare Klassifikation getrennt werden. Diese Darstellung wird als *Merkmalsraum* bezeichnet. Der Merkmalsraum kann eine beliebig hohe Dimensionalität besitzen, auch unendlich, um eine lineare Separierbarkeit zu erreichen [8]. Allerdings müssen Skalarprodukte der transformierten Daten berechnet werden, was zu Problemen führt, wenn der Merkmalsraum zu groß wird.

Hier kommt der sog. *Kernel-Trick* ins Spiel [8, 15]. Bei der Entscheidungsfunktion in Gleichung (5.6) tauchen die Merkmalsvektoren nur noch in Form von Skalarprodukten auf. Das Gleiche gilt dank dem Lagrange-Ansatz für das Training (siehe Anhang A.2). Bei Verwendung einer Merkmalstransformation fließen die Daten somit in Form von Skalarprodukten der transformierten Merkmale $\langle \boldsymbol{\phi}(\mathbf{y}), \boldsymbol{\phi}(\tilde{\mathbf{y}}) \rangle$ ein.

Wird nun eine Kernfunktion als Innenprodukt im Merkmalsraum formuliert [8],

$$K(\mathbf{y}, \tilde{\mathbf{y}}) = \langle \boldsymbol{\phi}(\mathbf{y}), \boldsymbol{\phi}(\tilde{\mathbf{y}}) \rangle \ , \tag{5.7}$$

können die Skalarprodukte durch diesen Kern ersetzt werden.

Es erfolgt hiermit immer noch eine lineare Trennung, jedoch in einem anderen Raum. Dieser Raum und die Merkmalstransformation $\boldsymbol{\phi}(\mathbf{y})$ müssen nicht explizit bekannt sein, es wird nur noch der Kern als Maß der Ähnlichkeit zwischen zwei Merkmalen benötigt.

Beispiele für Kerne und Eigenschaften, die eine Funktion erfüllen muss, um ein gültiger Kern zu sein, können in [8, 15] nachgelesen werden. Auf die in dieser Arbeit verwendeten Kernfunktionen wird in Abschnitt 5.4.4 eingegangen.

Bisher wurde von einem Zwei-Klassen-Problem ausgegangen. Häufig möchte man jedoch mehr als zwei Klassen voneinander unterscheiden. Eine übliche Methode, dies zu bewerkstelligen ist es, mehrere SVMs zu kombinieren, um eine Mehrklassen-SVM zu erhalten [8]. Dazu kann z. B. ein „*Einer gegen Alle (one versus all)*"-Ansatz gewählt werden. Hierbei wird für jede Klasse ein eigenes Modell gelernt, indem die Trainingsmerkmale der aktuellen Klasse als positive und die Trainingsmerkmale aller anderen Klassen als negative Beispiele verwendet werden. Bei S Klassen ergeben sich also S Klassifikatoren. Eine andere Möglichkeit ist der „*Einer gegen Einen (one versus one)*"-Ansatz. Bei diesem werden $(S(S-1)/2)$ Klassifikatoren für alle Paare von Klassen gelernt. Zur Klassifikation wird die Klasse gewählt, die die meisten Stimmen erhalten hat [8].

5.2.3. K-means Clustering

Der *K-means*-Algorithmus ist ein weit verbreitetes Verfahren zur Clusteranalyse. Die Aufgabe des Verfahrens besteht darin, aus gegebenen Daten in einem mehrdimensionalen Raum verschiedene Gruppen, auch *Cluster*, zu identifizieren [8].

Es seien Merkmalsvektoren $\{\mathbf{y}_n\}$, $n = 1,\ldots,N$, in einem D-dimensiona-len euklidischen Vektorraum gegeben. Das Ziel ist es, diese N Punkte in K Cluster einzuteilen. Es wird angenommen, dass die Anzahl K der Cluster bekannt ist. Die Gruppeneinteilung soll so erfolgen, dass die Abstände zwischen Punkten innerhalb eines Clusters klein sind im Vergleich zu Abständen zu Punkten außerhalb. Dazu werden Cluster-Prototypen $\boldsymbol{\mu}_k$, $k = 1,\ldots,K$, definiert, welche die einzelnen Cluster repräsentieren sollen. Gesucht sind nun diese Prototypen sowie die Zuordnung der gegebenen Datenpunkte zu den Clustern.

Die Cluster-Zugehörigkeit der $\mathbf{y}_n$ wird durch binäre Indikatorvariablen $r_{nk} \in \{0,1\}$ angegeben [8]. Es gilt $r_{nk} = 1$ und $r_{nj} = 0$ für $j \neq k$, wenn $\mathbf{y}_n$ dem k-ten Cluster zugeteilt wird. Damit kann die Summe der quadratischen Distanzen zwischen den Elementen der Cluster und ihren Prototypen als Zielfunktion formuliert werden [8]:

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\mathbf{y}_n - \boldsymbol{\mu}_k\|^2 \;\rightarrow\; \min. \tag{5.8}$$

Die Minimierung von J erfolgt iterativ in zwei Schritten. Als Startwerte für die Cluster-Prototypen werden zufällig Punkte aus den gegebenen Daten $\{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ ausgewählt. Im ersten Schritt wird J bezüglich $\{r_{nk}\}$ bei festen $\boldsymbol{\mu}_k$ optimiert. Dabei wird jeder Punkt dem nächsten Prototyp zugeordnet:

$$r_{nk} = \begin{cases} 1 & \text{wenn } k = \arg\min_j \|\mathbf{y}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{sonst} \end{cases} . \tag{5.9}$$

Als nächstes werden die optimalen Werte für $\{\boldsymbol{\mu}_k\}$ berechnet. Durch Differentiation und zu Null setzen von (5.8) ergibt sich

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} r_{nk}\mathbf{y}_n}{\sum_{n=1}^{N} r_{nk}}, \quad k = 1, \ldots, K. \tag{5.10}$$

$\boldsymbol{\mu}_k$ entspricht also gerade dem Mittelwert der Elemente des Clusters k. Die Schritte (5.9) und (5.10) werden solange wiederholt, bis die Lösung konvergiert oder eine maximale Anzahl an Iterationen durchlaufen wurde.

Das *K-means*-Clustering ist konvergent; es ist jedoch nicht garantiert, dass das globale Optimum gefunden wird [8]. Das Ergebnis hängt stark von der Initialisierung ab. Daher wird der Algorithmus häufig mehrmals wiederholt und die Lösung verwendet, bei der J am Ende des Clusterings den kleinsten Wert annimmt.

5.3. Probabilistische Sequenzmodelle

Für die sequenzielle Aktivitätsmodellierung in Abschnitt 5.5 wird ein sequenzielles probabilistisches Modell verwendet. In diesem Abschnitt werden die dafür benötigten Grundlagen erläutert. Zunächst wird eine kurze Übersicht über probabilistische Modelle gegeben, danach wird auf graphische Modelle eingegangen.

5.3.1. Generative und diskriminative Modelle

Probabilistische Modelle modellieren Zusammenhänge zwischen Variablen. Hier wird auf die Verwendung solcher Modelle in der Entscheidungstheorie eingegangen. Die modellierten Größen lassen sich einteilen in Zufallsvariablen, die beobachtet werden und die latenten Variablen bzw. Zustände. Dies ist analog zur Diskussion von Klassifikationsmethoden in Abschnitt 5.2.1 zu betrachten. Dort sind die Beobachtungen die Merkmale, durch die die Objekte repräsentiert werden, und die Klassenzugehörigkeiten stellen die latenten Variablen dar.

Es soll ein Modell formuliert werden, welches die probabilistischen Zusammenhänge zwischen Beobachtungen $\mathcal{Y}$ und latenten Variablen bzw. Zustandsvariablen $\mathcal{X}$ darstellt. Es wird unterschieden zwischen *generativen* und *diskriminativen* Modellen [93]. Ein *generatives* Modell beschreibt die Verbunddichte $p(\mathcal{X}, \mathcal{Y})$ der Beobachtungs- und Zustandsvariablen. Ein einfaches Beispiel eines generativen Modells ist der *naive Bayes-Klassifikator* (NBC).

Es sei eine Menge von N Beobachtungsvariablen $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ gegeben und es soll eine einzelne latente Variable $x \in \{1, \ldots, S\}$ prädiziert werden, welche die Klassenzugehörigkeit angibt. Dem NBC liegt die Annahme zugrunde, dass die Beobachtungen bei bekannter Klasse unabhängig voneinander sind. Die Verbunddichte lautet dann

$$p(x, \mathcal{Y}) = p(x) \prod_{n=1}^{N} p(\mathbf{y}_n | x)\,. \tag{5.11}$$

Ein *diskriminatives* Modell hingegen modelliert nicht die Verbunddichte, sondern die bedingte Dichte der Klassenvariable bei gegebenen Beobachtungen. Der Gedanke dahinter ist, dass die Beobachtungen bei Klassifikationsproblemen ohnehin bekannt sind, weshalb Modellierungsaufwand eingespart werden kann, indem auf die Modellierung der Beobachtungen verzichtet wird. Die diskriminative Entsprechung zum NBC ist die *logistische Regression* [93]. Hierbei wird angenommen, dass die logarithmierte Dichte $\log p(x|\mathcal{Y})$ jeder Klasse eine lineare Funktion der Beobachtungen ist. Die bedingte Dichte wird hierbei dargestellt gemäß [93]

$$p(x|\mathcal{Y}, \boldsymbol{\lambda}) = \frac{1}{Z(\mathcal{Y})} \exp\left(\lambda_x + \sum_{n=1}^{N} \lambda_{x,n} \mathbf{y}_n\right). \tag{5.12}$$

Das Gewicht λ_x erfüllt hierbei die Funktion der A-priori-Dichte $p(x)$ beim NBC. $Z(\mathcal{Y})$ ist eine Normierungskonstante

$$Z(\mathcal{Y}) = \sum_{x \in \mathcal{X}} \exp\left(\lambda_x + \sum_{n=1}^{N} \lambda_{x,n} \mathbf{y}_n\right). \tag{5.13}$$

Um den Zusammenhang zu den später eingeführten *Conditional Random Field*s (CRFs) zu verdeutlichen, wird eine alternative Notation eingeführt. Es werden gemeinsame Gewichte für alle Klassen und separate Merkmale für jede Klasse eingeführt. Jedes dieser Merkmale nimmt nur für eine Klasse einen Wert ungleich null an [93]:

$$f_k(x, \mathbf{y}) := f_{s,n}(x, \mathbf{y}) = \mathbf{1}_{\{x=s\}} \mathbf{y}_n, \tag{5.14}$$

wobei $\mathbf{1}_{\{\}}$ den Identitätsoperator darstellt. Hierbei ergeben sich $K = S \cdot N$ Merkmale, deren Laufindex durch k angegeben wird. Das Modell wird damit zu

$$p(x|\mathcal{Y}, \boldsymbol{\lambda}) = \frac{1}{Z(\mathcal{Y})} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(x, \mathbf{y})\right). \tag{5.15}$$

Diskriminative Modelle haben bezüglich der Klassifikation einige Vorteile gegenüber generativen Modellen [93]. Sie sind besser dazu geeignet, komplexe, überlappende Merkmale zu verwenden. Der Grund hierfür ist, dass die Verteilung der Beobachtungen nicht modelliert wird und demnach auch Zusammenhänge zwischen Beobachtungsvariablen nicht dargestellt werden müssen. Daher können die Beobachtungen in beliebiger Weise durch die Merkmale miteinander verknüpft werden. Dies resultiert in einer größeren Freiheit in der Modellierung. Dabei besteht allerdings die Gefahr des *Overfittings*. Ein Vorteil generativer Modelle ist, dass unüberwachtes Lernen einfacher umgesetzt werden kann, d. h. nicht oder nur teilweise annotierte Trainingsdaten können besser gehandhabt werden [93].

5.3.2. Probabilistische graphische Modelle

Nun werden die für das Verständnis der später diskutierten Aktivitätsmodelle hilfreichen Grundlagen graphischer Modelle erläutert. Außerdem wird auf ihre Bedeutung hinsichtlich den beiden im vorigen Abschnitt diskutierten Modellformen – generativ und diskriminativ – eingegangen.

Graphische Darstellungen sind ein sehr nützliches Hilfsmittel zur Formulierung und Analyse probabilistischer Modelle. Hierbei wird die Wahrscheinlichkeitsdichte mehrerer voneinander abhängiger Zufallsvariablen mit Hilfe von Graphen repräsentiert. Zufallsvariablen, oder Mengen von Zufallsvariablen, werden als Knoten dargestellt. Kanten drücken probabilistische Zusammenhänge zwischen Zufallsvariablen aus. Graphische Modelle geben an, wie die Verbunddichte aller Zufallsvariablen als Produkt lokaler *Faktoren*, welche nur von Teilmengen der Zufallsvariablen abhängen, angegeben werden kann. Dadurch wird eine intuitive Veranschaulichung und Modellierung der Struktur probabilistischer Modelle ermöglicht, indem beispielsweise Abhängigkeiten verschiedener Zufallsvariablen anhand des Graphen untersucht werden können. Gerade das Fehlen von Kanten – (bedingte) Unabhängigkeiten – ist häufig die wichtige Information. Auch Methoden der Inferenz und Parameterschätzung können durch graphische Manipulationen ausgedrückt werden [8].

Graphische Modelle können in gerichtet und ungerichtet unterteilt werden. Gerichtete graphische Modelle (GGM) werden auch als Bayes-Netze bezeichnet und werden häufig zur Darstellung generativer Modelle verwendet. Ungerichtete graphische Modelle (UGM) werden auch *Markov Random Fields* oder Markov-Netze genannt und eignen sich für diskriminative Modelle. Eine ausführliche Diskussion findet sich z. B. in [8].

Gerichtete graphische Modelle – Bayes-Netze

Bei Bayes-Netzen wird die Verbunddichte einer Menge von Zufallsvariablen als Produkt bedingter Dichten repräsentiert [8]. Sie werden als gerichtete Graphen dargestellt, d. h. die Verbindungen zwischen Variablen besitzen eine Richtung. In Abbildung 5.3(a) ist ein graphisches

Modell dreier Zufallsvariablen abgebildet. Die Verbunddichte einer Verteilung, die durch dieses Modell beschrieben wird, lässt sich ausrücken als

$$p(\mathcal{X}) = p(x_2|x_1)p(x_3|x_1)p(x_1)\,. \tag{5.16}$$

Mit dem Bayes'schen Gesetz kann die Verteilung (5.16) umgeformt werden zu

$$p(\mathcal{X}) = p(x_1|x_2)p(x_3|x_1)p(x_2)\,, \tag{5.17}$$

was dem Graphen 5.3(b) entspricht. Die durch diese beiden Graphen beschriebenen Verteilungen weisen also die gleichen Markov-Eigenschaften auf.

Die Verbunddichte einer Menge von Zufallsvariablen $\mathcal{X} = \{x_1, \ldots, x_N\}$ lässt sich allgemein schreiben als

$$p(\mathcal{X}) = \prod_{n=1}^{N} p(x_n|\mathcal{X}_n^{\mathrm{p}})\,. \tag{5.18}$$

$\mathcal{X}_n^{\mathrm{p}}$ ist hierbei die Menge der Eltern von x_n, d. h. alle Knoten, von denen aus Kanten zum Knoten x_n führen. Der Ausdruck (5.18) ist immer korrekt normiert, wenn die einzelnen Dichten normiert sind. Hieran werden die Faktorisierungseigenschaften von Bayes-Netzen ersichtlich: Die Verbunddichte wird als Produkt (lokaler Faktoren) der bedingten Dichten dargestellt. Eine wichtige Einschränkung bei Bayes-Netzen ist, dass keine gerichteten Zyklen erlaubt sind, es handelt sich also um gerichtete azyklische Graphen.

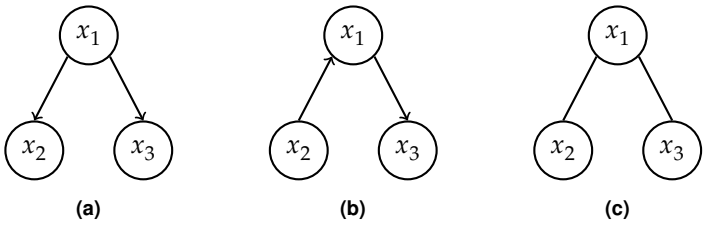


Abbildung 5.3. Beispiel graphischer Modelle.

Wichtige Eigenschaften zur Analyse der Modelle sind hierbei Markov-Eigenschaften, d. h. bedingte Unabhängigkeiten zwischen Variablen. Wenn für die Dichte dreier Zufallsvariablen a,b und c $p(a|b,c) = p(a|c)$ gilt, dann ist a bedingt unabhängig von b, bei bekanntem c [8]. Für die Verbunddichte folgt daraus

$$p(a,b|c) = p(a|b,c)p(b|c) = p(a|c)p(b|c)\,. \tag{5.19}$$

Bayes-Netze werden häufig zur Darstellung generativer Modelle verwendet. Gerichtete Graphen können kausale Zusammenhänge zwischen Zufallsvariablen ausdrücken. Als Beispiel hierfür ist in Abbildung 5.4(a) ein Modell einer Zustandsvariable x und Messgröße y mit der Verbunddichte

$$p(x,y) = p(y|x)p(x) \tag{5.20}$$

zu sehen. Dieser Graph drückt den Entstehungsprozess der Beobachtungsgröße aus, daher stammt die Bezeichnung als *generatives* Modell.

Nun soll eine Inferenz in diesem Modell durchgeführt werden. Ausgangspunkt sind die A-priori-Dichte $p(x)$ und die *Likelihood*-Dichte $p(y|x)$. Die Messgröße y wird beobachtet, was durch eine graue Färbung des Knotens in Abb. 5.4(b) angezeigt wird. Gesucht ist die A-posteriori-Dichte $p(x|y)$. Diese kann ausgedrückt werden gemäß

$$p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(y|x)p(x)}{p(y)}\,, \tag{5.21}$$

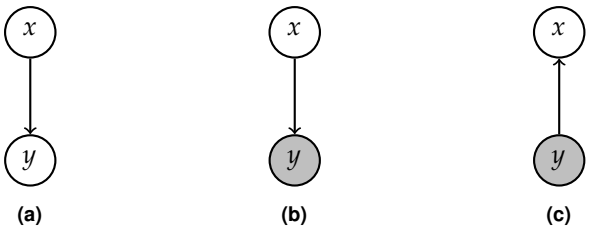


Abbildung 5.4. Inferenz in gerichteten Modellen.

wobei die Randdichte der Messung berechnet wird durch

$$p(y) = \sum_x p(y|x)p(x)\,. \tag{5.22}$$

Bei der Inferenz wird die Verbunddichte also durch $p(x,y) = p(x|y)p(y)$ repräsentiert, was dem Graphen in Abbildung 5.4(c) entspricht Dabei wird somit die Richtung der Kanten „umgekehrt".

Ungerichtete graphische Modelle – Markov Random Fields

Markov-Netze werden häufig durch ungerichtete graphische Modelle (UGMs) dargestellt [8]. Hier entfallen die Pfeile der Kanten, d. h. es werden keine kausalen Zusammenhänge mehr dargestellt, sondern allgemeine Abhängigkeiten zwischen Variablen. Auch hier wird die Verbunddichte als Produkt lokaler Faktoren ausgedrückt, welche jeweils von einer Untermenge der Zufallsvariablen abhängen. Im Gegensatz zu den Bayes-Netzen müssen diese Faktoren aber keine Wahrscheinlichkeitsdichten darstellen, sondern werden als *Potentiale* bezeichnet. Die Potentiale werden über sog. *Cliquen* definiert. Eine Clique ist eine Untermenge von Knoten, wobei Kanten zwischen allen Knoten der Clique existieren. Eine *maximale Clique* liegt vor, wenn kein weiterer Knoten hinzugenommen werden kann.

Die Verbunddichte wird bei Markov-Netzen als Produkt von Cliquenpotentialen dargestellt:

$$p(\mathcal{X}) = \frac{1}{Z}\prod_{c=1}^{C}\psi_c(\mathcal{X}_c)\,. \tag{5.23}$$

Die Potentiale $\psi_c(\mathcal{X}_c)$ seien nicht-negative Funktionen der maximalen Cliquen $\mathcal{X}_c$. Da die Potentialfunktionen keine Verteilungsdichten sein müssen, ist eine Normierungskonstante Z nötig, welche auch als *Partitionierungsfunktion* bekannt ist:

$$Z = \sum_{\mathbf{x}}\prod_{c=1}^{C}\psi_c(\mathbf{x}_c)\,. \tag{5.24}$$

Zur Bestimmung von Z muss über alle Kombinationen der Zustände summiert werden. Die Partitionierungsfunktion wird für die Parameterschätzung benötigt. Ihre Berechnung kann sehr aufwändig werden und ist nicht immer möglich. Beispielsweise müssen bei N Variablen, die jeweils S Werte annehmen können, N^S Summen ausgewertet werden. In der Praxis sind dazu Approximationen nötig. Dies ist die größte Einschränkung für UGMs. Bei der Bestimmung lokaler bedingter Dichten und Randdichten entfällt die Bestimmung von Z.

Die Potentialfunktionen geben keine kausalen Zusammenhänge zwischen den Variablen, d. h. keine physikalischen Informationen über den zugrunde liegenden Prozess, an. Stattdessen beschreiben sie, welche Konfigurationen lokaler Variablen bevorzugt auftreten. Eine hohe Wahrscheinlichkeit einer globalen Konfiguration wird erreicht, wenn diese eine gute Balance zwischen lokalen Cliquenpotentialen darstellt. UGMs werden daher häufig zur Darstellung diskriminativer Modelle eingesetzt.

Eine wichtige Eigenschaft von UGMs ist die lokale Markov-Eigenschaft [6]. Sind die Nachbarn $\mathcal{X}_n^{\text{nb}}$ eines Knoten x_n gegeben, ist der Knoten bedingt unabhängig von den restlichen Knoten des Graphen:

$$p(x_n|\mathcal{X} \setminus x_n) = p(x_n|\mathcal{X}_n^{\text{nb}})\,. \tag{5.25}$$

Um einen Zusammenhang zwischen gerichteten und ungerichteten Graphen herzustellen, wird noch einmal das Beispiel aus Bild 5.3 aufgegriffen. Mit

$$\psi_1(x_1, x_2) := p(x_2|x_1)p(x_1)\,, \tag{5.26}$$

$$\psi_2(x_1, x_3) := p(x_3|x_1) \tag{5.27}$$

kann die Verteilung (5.16) geschrieben werden als

$$p(\mathcal{X}) = \psi_1(x_1, x_2)\psi_2(x_1, x_3)\,. \tag{5.28}$$

Der entsprechende ungerichtete Graph ist in Bild 5.3(c) zu sehen.

Gerichtete Modelle können auch als UGMs dargestellt werden, wobei bedingte Dichten in geeigneter Weise zu Potentialen zusammengefasst werden. Für die Partitionierungsfunktion gilt in diesem Fall $Z = 1$. Es können dabei allerdings Unabhängigkeitsinformationen verloren gehen,

da es nötig sein kann, zusätzliche Kanten hinzuzufügen. Umgekehrt kann jede Dichte als Bayes-Netz dargestellt werden, welches im schlechtesten Fall Verbindungen zwischen allen Paaren von Knoten besitzt. Es können aber nicht alle Abhängigkeiten, die von einem UGM repräsentiert werden können, von einem gerichteten Graphen dargestellt werden.

5.3.3. Hidden Markov-Modelle

Im Folgenden wird eine kurze Vorstellung von *Hidden Markov*-Modellen (HMM) gegeben, um anschließend auf CRFs überzugehen. Hier wird aus der Perspektive der graphischen Modellierung auf *Hidden Markov*-Modells (HMMs) eingegangen, was später hilfreich sein wird, um den Zusammenhang zu CRFs herzustellen. Ein HMM modelliert die Verteilung einer Sequenz latenter Variablen $\mathbf{x} = [x_1, \ldots, x_T]$ oder Zustände und einer Beobachtungssequenz $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_T]$. Hier wird von diskreten Zuständen und Beobachtungen ausgegangen, mit S Zuständen s und O Beobachtungssymbolen $\mathbf{o}$. Die Verbunddichte lässt sich schreiben als

$$p(\mathbf{x}, \mathbf{y}) = \prod_{t=1}^{T} p(x_t|x_{t-1}) p(\mathbf{y}_t|x_t)\,, \tag{5.29}$$

wobei die A-priori-Dichte des ersten Zustandes $p(x_1)$ als $p(x_1|x_0)$ dargestellt wird. Es sei angemerkt, dass hier eine vektorielle Notation der

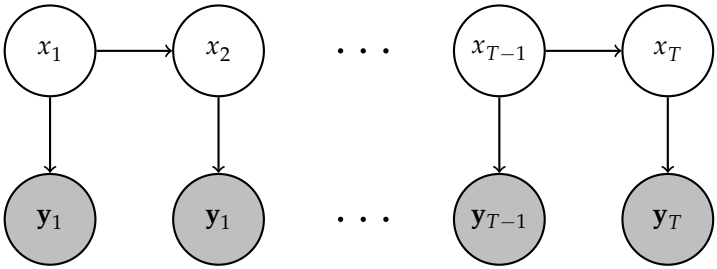


Abbildung 5.5. *Hidden Markov*-Modell (HMM).

Zustands- und Beobachtungsvariablen statt der Mengen-Notation verwendet wird, um den sequenziellen Charakter explizit zu verdeutlichen. Diesem Modell liegen die folgenden Markov-Annahmen zugrunde:

1. Der aktuelle Zustand hängt nur vom vorigen Zustand ab:

$$p(x_t|x_{1:t-1}) = p(x_t|x_{t-1})\,. \tag{5.30}$$

2. Die aktuelle Beobachtung hängt nur vom aktuellen Zustand ab:

$$p(\mathbf{y}_t|x_{1:t}) = p(\mathbf{y}_t|x_t)\,. \tag{5.31}$$

Die graphische Darstellung eines HMM ist in Abbildung 5.5 zu sehen.

Zusammenhang zwischen HMM und CRF

Das Modell (5.29) kann umgeformt werden zu [93]

$$p(\mathbf{x},\mathbf{y}) = \frac{1}{Z}\prod_{t=1}^{T}\exp\Big(\sum_{r\in\mathcal{S}}\sum_{s\in\mathcal{S}}\vartheta_{r,s}\mathbf{1}_{\{x_t=r\}}\mathbf{1}_{\{x_{t-1}=s\}} + \sum_{s\in\mathcal{S}}\sum_{\mathbf{o}\in\mathcal{O}}\mu_{\mathbf{o},s}\mathbf{1}_{\{x_t=s\}}\mathbf{1}_{\{\mathbf{y}_t=\mathbf{o}\}}\Big) \tag{5.32}$$

mit dem Parametern

$$\begin{aligned}\vartheta_{r,s} &= \log p(x_t = r|x_{t-1} = s)\\ \mu_{\mathbf{o},s} &= \log p(\mathbf{y}_t = \mathbf{o}|x_t = s)\,.\end{aligned} \tag{5.33}$$

Dabei ist $\mathbf{1}_{\{a=b\}}$ die Indikatorfunktion

$$\mathbf{1}_{\{a=b\}} = \begin{cases}1 & \text{falls } a = b\\ 0 & \text{sonst}\end{cases}\,. \tag{5.34}$$

Z ist eine Normierungskonstante, die sicherstellt, dass eine gültige Wahrscheinlichkeitsdichte vorliegt. Nun können für die Kombinationen von Zuständen und Beobachtungen Merkmalsfunktionen eingeführt werden:

$$\begin{aligned}f_{r,s}(x,\tilde{x}) &= \mathbf{1}_{\{x=r\}}\mathbf{1}_{\{\tilde{x}=s\}}\,,\\ f_{\mathbf{o},s}(x,\mathbf{y}) &= \mathbf{1}_{\{x=s\}}\mathbf{1}_{\{\mathbf{y}=\mathbf{o}\}}\,.\end{aligned} \tag{5.35}$$

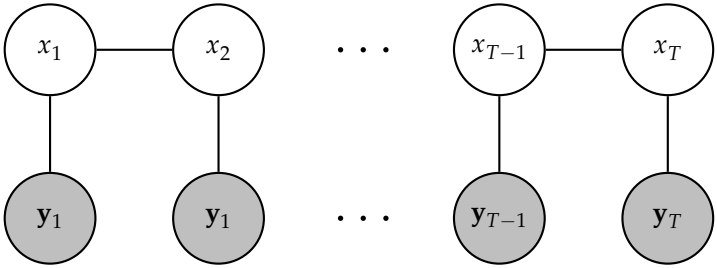


Abbildung 5.6. Lineares kettenförmiges *Conditional Random Field* (CRF).

Als Nächstes werden alle Merkmalsfunktionen zusammengefasst und als $f_k(x, \tilde{x}, \mathbf{y}), k = 1, \ldots, K$, geschrieben und die Parameter zusammengefasst zu $\boldsymbol{\lambda} = \{\vartheta_{r,s}, \mu_{\mathbf{o},s}\}$. Damit ergibt sich

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(x_t, x_{t-1}, \mathbf{y}_t) \right). \tag{5.36}$$

Die bedingte Dichte der Zustandssequenz bei gegebenen Beobachtungen lautet nun

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \frac{p(\mathbf{x}, \mathbf{y})}{\sum_{\mathbf{x}'} p(\mathbf{x}', \mathbf{y})} \\ &= \frac{\prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(x_t, x_{t-1}, \mathbf{y}_t) \right)}{\sum_{\tilde{\mathbf{x}}} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(\tilde{x}_t, \tilde{x}_{t-1}, \mathbf{y}_t) \right)}. \end{aligned} \tag{5.37}$$

Dies entspricht einem linearen kettenförmigen CRF mit den Merkmalsfunktionen (5.35). Die graphische Darstellung mit einem UGM ist in Abbildung 5.6 zu sehen.

Analog zum Verhältnis zwischen dem naiven Bayes-Klassifikator und der logistischen Regression ist das lineare kettenförmige CRF die diskriminative Entsprechung des HMM.

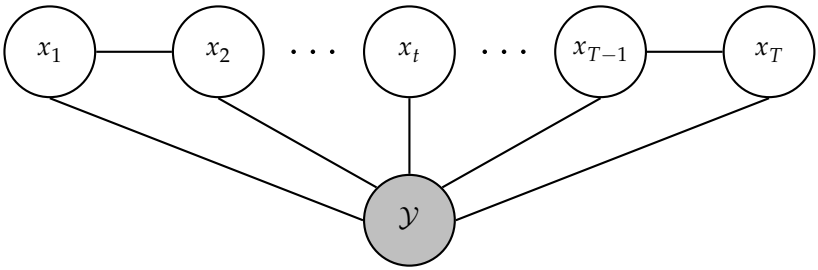


Abbildung 5.7. Allgemeine Form des kettenförmigen *Conditional Random Field* (CRF).

5.3.4. Conditional Random Fields

Die allgemeine Form eines linearen kettenförmigen CRF lautet

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{Z(\mathbf{y})} \prod_{t=1}^{T} \psi_t(x_t, x_{t-1}, \mathbf{y}) \tag{5.38}$$

und ist in Abbildung 5.7 mit einem UGM veranschaulicht. Die Partitionierungsfunktion ist

$$Z(\mathbf{y}) = \sum_{\mathbf{x}} \prod_{t=1}^{T} \psi_t(x_t, x_{t-1}, \mathbf{y}) \,. \tag{5.39}$$

Dabei sind $\psi_t(x_t, x_{t-1}, \mathbf{y})$ die Potentiale der maximalen Cliquen $\{x_t, x_{t-1}, \mathbf{y}\}$.

Häufig wird die Strategie des *Parameter-Tying* angewandt. Dabei werden sog. *Cliquen-Templates* mit jeweils festen Parametern angenommen. Jedes *Template* besteht aus einem Satz von Faktoren und Parametern. Bei linearen Ketten-CRFs bedeutet dies, dass zu jeden Zeitpunkt dieselben Parameter verwendet werden, d. h. es wird ein zeitinvarianter Prozess angenommen. Die Potentiale $\psi_t = \psi$ lassen sich damit schreiben als

$$\psi(x_t, x_{t-1}, \mathbf{y}) = \exp\left(\sum_{k=1}^{K} \lambda_k f_k(x_t, x_{t-1}, \mathbf{y})\right) . \tag{5.40}$$

Damit resultiert schließlich die analoge Formulierung zu Gleichung (5.37):

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{Z(\mathbf{y})} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(x_t, x_{t-1}, \mathbf{y})\right) \tag{5.41}$$

mit

$$Z(\mathbf{y}) = \sum_{\mathbf{x}} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(x_t, x_{t-1}, \mathbf{y})\right) . \tag{5.42}$$

Die Partitionierungsfunktion hängt von der gesamten Beobachtungssequenz ab, d. h. das CRF ist global normiert. Die Berechnung der Partitionierungsfunktion wird für die Parameterschätzung benötigt. Hierbei muss über alle möglichen Zustandssequenzen summiert werden. Eine effiziente Berechnung bei linearen Ketten-CRFs wird durch den Vorwärts-Rückwärts-Algorithmus ermöglicht. Dieser ist aus der Theorie der HMMs bekannt und kann ohne Weiteres auf lineare CRFs erweitert werden. Ebenso kann der Viterbi-Algorithmus zur Decodierung bzw. Inferenz verwendet werden, um die wahrscheinlichste Zustandssequenz zu berechnen. Für eine detaillierte Erläuterung der Anwendung dieser Algorithmen für CRFs sei auf [93] verwiesen.

Die Parameterschätzung erfolgt nach dem *Maximum Likelihood*-Prinzip mit Hilfe numerischer Optimierungsverfahren. Dabei wird in der Praxis häufig eine Regularisierung bezüglich der Gewichte vorgenommen. Diese kann mittels der L2-Norm erfolgen, um *Overfitting* zu vermeiden oder mit der L1-Norm, um möglichst spärliche Gewichte zu erhalten. In Anhang A.3 wird weiter auf die Parameterschätzung eingegangen.

Die Anzahl an Merkmalen kann bei CRFs sehr groß werden, auch mehrere Millionen [93]. Methoden der Merkmalsselektion zielen darauf ab, automatisch die besten Merkmale zu wählen. Dies kann beispielsweise durch Regularisierung mit der L1-Norm erfolgen. Hierzu gibt es außerdem Methoden der Merkmalsinduktion, welche zunächst von einfachen Standardmerkmalen ausgehen und diese auf bestimmte Weise kombinieren, um neue Merkmale zu erhalten [93].

5.4. Aktivitätserkennung mit dem „Bag of Words“-Modell

Der „*Bag of Words*“ (BoW)-Ansatz stammt aus der Textverarbeitung. Er basiert auf der vereinfachenden Annahme, dass das Wissen darüber, *welche* Worte in einem Text auftreten, bereits sehr aussagekräftige Informationen über den Inhalt des Textes enthält. Die genaue Anordnung der Worte besitzt dabei eine untergeordnete Bedeutung. Übertragen auf die Verarbeitung von Videosignalen bedeutet dies, eine lose Ansammlung von Merkmalen zu betrachten und nur zu modellieren, *welche* Merkmale auftreten.

Als Ausgangspunkt für die Aktionserkennung dienen die Ergebnisse des Merkmals-Trackings aus Kapitel 4. Durch die verschiedenen Deskriptoren, die nach Abschnitt 4.3.4 zur Repräsentation der Merkmalstrajektorien ermittelt werden, stehen sowohl Informationen über den Bildinhalt und den optischen Fluss in der Nachbarschaft der Trajektorien sowie der dynamische Verlauf der Merkmalspunkte zur Verfügung. Informationen über die örtliche Konfiguration der Merkmale wird durch die separate Betrachtung verschiedener Bildbereiche berücksichtigt, indem das Bild in bestimmte Raster unterteilt und Merkmale in den Rastern lokal analysiert werden.

Im Folgenden wird die Vorgehensweise bei der BoW-Modellierung und die genaue Umsetzung in dieser Arbeit dargestellt. Zunächst wird eine Übersicht über den Gesamtablauf gegeben, bevor die einzelnen Schritte im Detail erläutert werden.

5.4.1. Übersicht

Abbildung 5.8 zeigt den schematischen Ablauf der Modellerstellung. Es seien annotierte Trainingssequenzen

$$\mathcal{D}^{\text{train}} = \{(\mathbf{i}_i(t), x_i)\,,\ i = 1, \ldots, N^{\text{train}}\} \tag{5.43}$$

vorhanden. Aus den Trainingssequenzen werden zunächst Merkmalstrajektorien nach dem Vorgehen aus Abschnitt 4.3 extrahiert. Für jede Sequenz ergibt sich eine Menge an Trajektorien $\mathcal{Z}_i = \{Z_1, \ldots, Z_{N_i}\}$. Jede Trajektorie setzt sich aus mehreren Trajektorien-Deskriptoren gemäß

Abschnitt 4.3.4 zusammen. Zunächst muss die Menge an Merkmalstrajektorien $\mathcal{Z}$ einer Beobachtungssequenz in eine geeignete, einheitliche Form überführt werden, welche eine Klassifikation ermöglicht. Diese Darstellung wird als *Videodeskriptor* bzw. *Aktionsdeskriptor* bezeichnet und hat die Form eines Merkmalsvektors $\mathbf{y}$.

Das im Folgenden erläuterte Vorgehen wird für jeden Deskriptortyp separat durchgeführt. Das heißt, es gibt zunächst je einen Aktionsdeskriptor für die Dynamikmerkmale, die HOG-Merkmale etc. Die Fusion der unterschiedlichen Merkmale wird in Abschnitt 5.4.4 erläutert. Auf die Berücksichtigung des örtlichen Kontextes wird zunächst noch nicht explizit eingegangen.

Die Überführung der Fülle an Deskriptoren für alle Trajektorien einer Sequenz in den Aktionsdeskriptor erfolgt durch *Vektorquantisierung*.

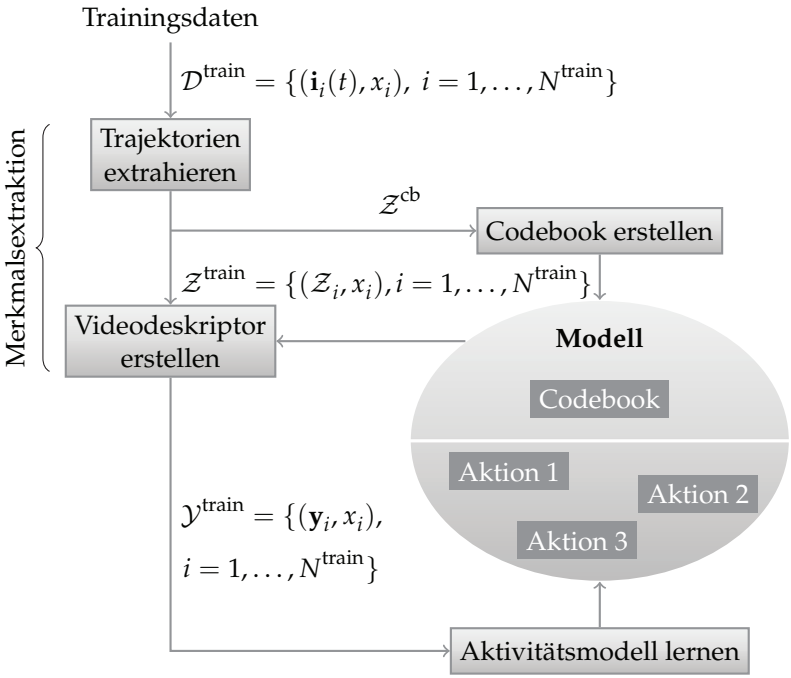


Abbildung 5.8. Ablauf der Modellerstellung am Beispiel eines einzelnen Merkmalstyps.

Dazu wird eine Menge an Prototypen-Deskriptoren benötigt. Die Prototypen sind in einem sog. *Codebook* gespeichert

$$\mathcal{CB} = \{\tilde{\mathbf{d}}_w\}\,, \quad w = 1, \ldots, W\,, \tag{5.44}$$

welches aus W Deskriptor-Prototypen besteht. Dieses wird aus den Trajektorien aller Trainingsvideos bzw. einer repräsentativen Untermenge davon $\mathcal{Z}^{\text{cb}} = \{Z_1, \ldots, Z_{\sum_i N_i}\}$ gelernt. Die Vektorquantisierung wird in Abschnitt 5.4.2 im Detail erklärt.

Nachdem das *Codebook* erstellt wurde, kann die Aktionsmodellierung vorgenommen werden. Für jede Eingangssequenz wird mit Hilfe des *Codebooks* ein *Videodeskriptor* $\mathbf{y}_i$ gebildet. Durch Projektion auf das *Codebook* wird jeder Deskriptor $\mathbf{d}_j$ dem ihm am ähnlichsten Prototyp w_j zugeordnet. w_j stellt hierbei den Index des gewählten Prototyps dar und wird auch als „Wort“ bezeichnet. Der Videodeskriptor ergibt sich aus der Statistik der in einer Sequenz auftretenden Worte. Auf den Aufbau dieser Deskriptoren wird in Abschnitt 5.4.3 eingegangen.

Die resultierenden Aktionsdeskriptoren können nun verwendet werden, um einen Klassifikator zu trainieren. Hierzu kann ein einzelner Merkmalstyp herangezogen oder eine Merkmalsfusion durchgeführt werden. Die Klassifikation wird in Abschnitt 5.4.4 erläutert.

Abbildung 5.9 zeigt die Vorgehensweise bei der Aktivitätsanalyse. Der Ausgangspunkt sind hier ein vorhandenes Modell und eine Eingangssequenz $\mathbf{i}(t)$ mit unbekannter Klasse. Aus $\mathbf{i}(t)$ werden, wie vorher, Merkmalstrajektorien $\mathcal{Z}$ extrahiert. Durch Vektorquantisierung mittels der im Trainingsschritt gelernten *Codebooks* wird für jeden Deskriptortyp ein Videodeskriptor $\mathbf{y}$ ermittelt. Mit den Videodeskriptoren eines oder mehrerer Merkmale und dem trainierten Klassifikator wird nun die Aktionserkennung durchgeführt.

5.4.2. Vektorquantisierung

Die Vektorquantisierung besitzt die Aufgabe, eine Menge an Merkmalen in eine einheitliche, klar vorgegebene Struktur zu bringen. Dies erfolgt durch Vergleich mit den im *Codebook* des jeweiligen Deskriptortyps gespeicherten Prototypen. Der schematische Ablauf ist in Abbildung 5.10 zu sehen. Das *Codebook* wird aus den vorhandenen Trainingsdaten mittels Clustering gelernt.

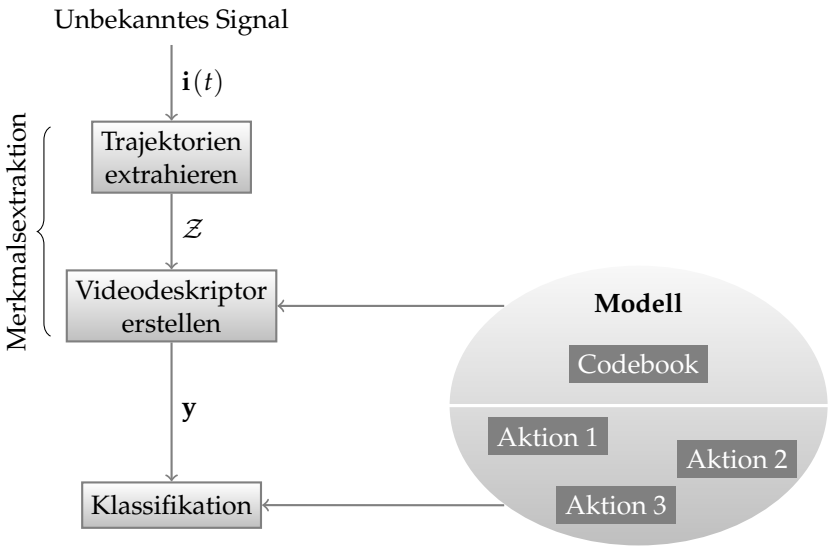


Abbildung 5.9. Allgemeiner Ablauf der Aktionsanalyse am Beispiel eines einzelnen Merkmalstyps.

Die Deskriptoren der Bild- und Flussinformation aus Abschnitt 4.3.4 beschreiben den Mittelwert der Merkmale über die gesamte Trajektoriendauer und besitzen eine einheitliche und kompakte Dimension. Sie können also direkt dem Cluster-Algorithmus zugeführt werden. Die Dynamik-Deskriptoren werden nicht gemittelt, sondern es liegt der gesamte Verlauf der Verschiebungsvektoren einer Trajektorie vor. Sie haben demnach unterschiedliche Dimensionalitäten, da in dieser Arbeit die Trajektorien eine beliebige Länge aufweisen können. Um einen zu großen Aufwand des Clusterings zu vermeiden, wird im Falle der Dynamik-Deskriptoren zuvor eine Dimensionsreduktion durchgeführt. Im Folgenden wird zunächst auf die Vektorquantisierung ohne und danach mit Dimensionsreduktion eingegangen.

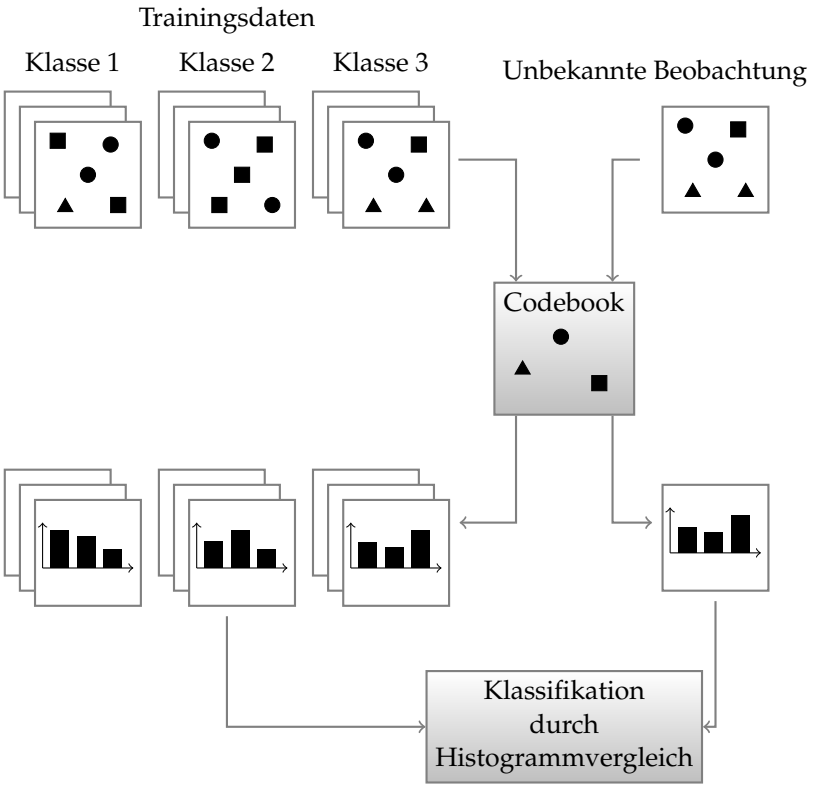


Abbildung 5.10. Überführung von Merkmalen in Aktionsdeskriptoren beim *„Bag of Words“*-Modell mittels Vektorquantisierung.

K-means-Clustering

Zur Vektorquantisierung wird die gesamte Menge an Trajektorien aller Trainingssequenzen herangezogen. Hieraus kann eine sehr große Menge an Merkmalen resultieren. Damit der Aufwand des Clusterings nicht zu hoch wird, wird eine zufällige Untermenge an Trajektorien ausgewählt.

Die Deskriptoren werden mit dem *K-means*-Verfahren aus Abschnitt 5.2.3 geclustert. Die gewünschte Cluster-Anzahl W muss hierbei vorgegeben werden. Da eine zufällige Initialisierung durchgeführt wird,

ist es sinnvoll, das Clustering mehrfach zu wiederholen und das beste Ergebnis zu verwenden, wie in Abschnitt 5.2.3 erläutert wurde. Als Distanzmaß wird die euklidische Distanz verwendet. Cluster, die eine minimale Größe unterschreiten, werden verworfen. Die Mittelpunkte der gefundenen Cluster bilden die Deskriptorprototypen $\tilde{\mathbf{d}}$ des *Codebooks*. Das *Codebook* ergibt sich somit zu

$$\mathcal{CB} = (\{\tilde{\mathbf{d}}_1, \ldots, \tilde{\mathbf{d}}_W\}) . \tag{5.45}$$

Zur Vektorquantisierung werden für einen Deskriptor $\mathbf{d}_j$ die Distanzen zu allen *Codebook*-Elementen berechnet und das Element mit der minimalen Distanz ausgewählt. Dem Deskriptor $\mathbf{d}_j$ wird somit ein „*Wort*“ w_j zugeordnet, welches dem Index des ähnlichsten Prototyps entspricht:

$$w_j = \arg\min_w d^{\text{euclid}}(\mathbf{d}_j, \tilde{\mathbf{d}}_w) . \tag{5.46}$$

Karhunen-Loève-Transformation und Clustering

Die Trajektorien besitzen unterschiedliche Längen, welche hier nicht begrenzt sind. Ein Distanzmaß, welches für Signale unterschiedlicher Länge geeignet ist, ist die *Dynamic Time Warping* (DTW)-Distanz [77], welche für die Vektorquantisierung für Merkmalstrajektorien bereits Einsatz fand [74]. In dieser Arbeit wird die euklidische Distanz in Verbindung mit einer Dimensionsreduktion angewandt, um die Dynamik-Deskriptoren in eine einheitliche und kompakte Form zu überführen. Die Komplexität des *K-means*-Verfahrens hängt linear von der Berechnungsdauer der Distanz zwischen zwei Merkmalen ab. Bei der euklidischen Distanz steigt der Aufwand somit linear mit der Merkmalsdimension, während er bei der DTW-Distanz i. Allg. quadratisch mit der Signallänge steigt.

Die Transformation der Dynamik-Deskriptoren in eine Darstellung geringerer Dimensionalität erfolgt mittels diskreter Karhunen-Loève-Transformation bzw. Hauptkomponentenanalyse [51, 71]. Dazu werden die Deskriptoren zunächst als Spaltenvektoren dargestellt und durch Anhängen von Nullen auf eine einheitliche Länge gebracht, welche der maximal vorkommenden Länge der Trainings-Deskriptoren entspricht. Die so erhaltenen Vektoren werden mit $\mathbf{u}_j$ bezeichnet. Mit Hilfe einer

Projektion auf eine orthonormale Basis $\mathbf{\Phi}$ sollen die Deskriptoren in eine niederdimensionale Darstellung überführt werden.

Zunächst wird der Mittelwert der N' Deskriptoren, die in die *Codebook*-Erstellung einfließen, ermittelt:

$$\boldsymbol{\mu}_u = \frac{1}{N'} \sum_{j=1}^{N'} \mathbf{u}_j \,. \tag{5.47}$$

Anschließend werden die Deskriptoren vom Mittelwert befreit und in einer Datenmatrix zusammengefasst:

$$\mathbf{Z} = \left[\mathbf{u}_1 - \boldsymbol{\mu}_u, \ldots, \mathbf{u}_{N'} - \boldsymbol{\mu}_u \right] \,. \tag{5.48}$$

Die Eigenvektoren $\boldsymbol{\varphi}_k$ der Kovarianzmatrix $\mathbf{C}_{uu} = \mathrm{E}\{\mathbf{Z}\mathbf{Z}^{\mathrm{H}}\}$ sind die gesuchten Basisvektoren. $\mathbf{Z}^{\mathrm{H}}$ steht dabei für die transponierte und konjugiert komplexe Matrix $\mathbf{Z}$: $\mathbf{Z}^{\mathrm{H}} = \mathbf{Z}^{\mathrm{T}*}$. $\mathbf{C}_{uu}$ enthält N' reelle, nicht-negative Eigenwerte. Durch die Beschränkung auf die $K < N'$ Basisvektoren mit den größten Eigenwerten erhält man eine kompakte Signalapproximation. Die reduzierte Basis ist somit

$$\mathbf{\Phi}_K = \left[\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_K \right] \,. \tag{5.49}$$

Durch Projektion

$$\mathbf{b}_j = \mathbf{\Phi}_K^{\mathrm{H}} \left(\mathbf{u}_j - \boldsymbol{\mu}_u \right) \tag{5.50}$$

erhält man die Koeffizienten der Signaldarstellung.

Die Prototypen $\tilde{\mathbf{b}}_w$, $w = 1, \ldots, W$, werden durch Clustering der Koeffizienten $\mathbf{b}_j$ mit dem *K-means*-Verfahren wie zuvor ermittelt. Das *Codebook* ergibt sich damit zu

$$\mathcal{CB} = (\mathbf{\Phi}_K, \boldsymbol{\mu}_u, \{\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_W\}) \,. \tag{5.51}$$

Zur Vektorquantisierung eines unbekannten Signals wird dieses zunächst gemäß Gleichung (5.50) mit der reduzierten Basis (5.49) projiziert. Die Zuordnung der erhaltenen Koeffizienten $\mathbf{b}$ zu einem Wort w erfolgt nun analog zu Gleichung (5.46) mit dem *Codebook* (5.51).

5.4.3. Aktionsdeskriptoren

Im Schritt der Vektorquantisierung werden die Deskriptoren einer Sequenz $\{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$ in Worte $\{w_1, \ldots, w_N\}$, d. h. Indices der *Codebook*-Elemente, überführt. Der Aktions- oder Videodeskriptor $\mathbf{y}$ wird als Histogramm der Wortvorkommen gebildet

$$\begin{aligned} &\mathbf{y} = [y_1, \ldots, y_W]^{\mathrm{T}} \text{ mit} \\ &y_w = \sum_{j=1}^{N} \delta_{w_j, w}\,, \quad w = 1, \ldots, W\,. \end{aligned} \tag{5.52}$$

Obwohl der BoW-Ansatz seine Fähigkeiten in vielen Aufgabenstellungen bereits unter Beweis gestellt hat, weist er Schwächen auf. So ist die örtliche und zeitliche Anordnung der Merkmale bei der Aktivitätserkennung durchaus eine relevante Information, welche hier komplett ignoriert wird. Besonders in komplexen Szenarien hat sich diese Vorgehensweise als unzureichend erwiesen.

In jüngerer Zeit werden daher verstärkt Zusammenhänge zwischen Merkmalen berücksichtigt und in den BoW-Ansatz integriert. Ein beliebtes Beispiel dafür stellt die zellenweise Verarbeitung dar. Dabei werden die Wort-Histogramme nicht global für die gesamte Videosequenz, sondern separat in mehreren lokalen Bereichen ermittelt. Dazu wird das Bild in ein bestimmtes Raster (engl. *grid*) an Zellen eingeteilt. Die Zelleneinteilung kann sowohl in örtlicher als auch zeitlicher Richtung erfolgen. Häufig werden redundante Raster verwendet. In [58] kommen als örtliche Raster $G_{C_x C_y}$ gleichmäßige Einteilungen in C_x vertikale und C_y horizontale Zellen sowie um den Bildmittelpunkt zentrierte, sich überlappende 2×2-Raster zum Einsatz. Des Weiteren erfolgt eine zeitliche Zelleneinteilung in drei Zellen. Die besten Ergebnisse erzielen die Autoren mit den Rastern G_{11}, G_{13}, G_{31} und G_{22} Bei der zeitlichen Einteilung erweist sich eine einzige Zelle – das Verzichten auf eine zeitliche Einteilung – als am besten geeignet. Sun et al. [92] verwenden ebenso die in [58] vorgeschlagenen Raster. Wang et al. [99] wählen die Einteilungen G_{11}, G_{22} und G_{31} jeweils für ein bis zwei zeitliche Zellen.

In dieser Arbeit werden lediglich örtliche Zelleneinteilungen berücksichtigt. Das Eingangsbild wird zunächst in die von einem bestimmten

Raster vorgegebene Menge an Zellen unterteilt. Der Auftrittsort einer Trajektorie Z_j wird durch ihren örtlichen Mittelpunkt repräsentiert:

$$\mathbf{m}_j = \frac{1}{L_j} \sum_{t=1}^{L_j} \mathbf{x}_j(t) \,. \tag{5.53}$$

Damit wird jeder Trajektorie eine Zellenzugehörigkeit c_j zugeordnet, indem geprüft wird, in welcher Bildzelle sich der Trajektorienmittelpunkt befindet.

Nun wird für jede Zelle c ein Aktionsdeskriptor $\mathbf{y}^c$ wie in Abschnitt 5.4.3 gebildet. Der gesamte Aktionsdeskriptor ergibt such durch Zusammenfügen der Deskriptoren der einzelnen Zellen

$$\mathbf{y} = \left[\mathbf{y}^1, \ldots, \mathbf{y}^C\right] \tag{5.54}$$

und besitzt $W \cdot C$ Elemente

$$y_w^c = \sum_{j=1}^{N} \delta_{w_j,w} \delta_{c_j,c} \,, \quad w = 1, \ldots, W \,,\; c = 1, \ldots, C \,, \tag{5.55}$$

wobei C die Anzahl der Zellen ist. Werden redundante Raster verwendet, wird das obige Vorgehen für jedes von ihnen wiederholt. Für jedes Raster resultiert somit ein Aktionsdeskriptor. Diese redundanten Deskriptoren werden im Modellierungs- bzw. Klassifikationsschritt fusioniert.

5.4.4. Klassifikation

Zur Aktionserkennung muss auf Basis der Aktionsdeskriptoren der Trainingsdaten ein Klassifikator trainiert werden, welcher anschließend für unbekannte Beobachtungen die Aktionsklasse prädizieren kann. Die Videodeskriptoren werden für jeden Merkmalstyp separat bestimmt. Somit ergeben sich für jede Sequenz mehrere Deskriptoren. Zur Klassifikation kann ein einziger Deskriptor herangezogen oder eine Merkmalsfusion durchgeführt werden.

In [111] werden unterschiedliche Merkmale und Klassifikationsmethoden zur Objekt- und Texturerkennung untersucht. Die Autoren ziehen

das Fazit, dass selbst das beste Merkmal alleine nicht so gut ist wie eine Kombination komplementärer Merkmale. Bei der Kombination von Deskriptoren sind laut [111] Kernel-Klassifikatoren besser geeignet als beispielsweise der *Nearest Neighbour*-Klassifikator, bei dem die Merkmalsfusion in manchen Fällen schlechter als die Verwendung einzelner Merkmale abschneidet, wenn ein Merkmal schlechter als die anderen ist.

Die Aktionserkennung erfolgt hier sowohl für einzelne als auch mehrere Merkmale mit einem *Support Vector Machine*-Klassifikator (SVM). In den folgenden Abschnitten wird zunächst auf die Klassifikation mit nur einem Deskriptortyp und anschließend auf die Fusion mehrerer Merkmale eingegangen.

Klassifikation mit einzelnen Merkmalen

Für die SVM-Klassifikation muss eine geeignete Kernfunktion gewählt werden, die je zwei Merkmalsvektoren miteinander vergleicht. Dafür können verschiedene Distanzmaße $d(\mathbf{y}_i, \mathbf{y}_j)$ eingesetzt werden.

In [111] werden verschiedene Kernfunktionen untersucht. Dazu gehören der lineare, quadratische, RBF- und χ^2-Kern. Diese stellen alle sog. erweiterte Gauß-Kerne dar und werden aus den entsprechenden Distanzen gemäß

$$K(\mathbf{y}_i, \mathbf{y}_j) = \exp\left(-\frac{1}{A} d(\mathbf{y}_i, \mathbf{y}_j)\right) \tag{5.56}$$

gebildet. Dabei werden in [111] mit dem χ^2-Kern bessere Ergebnisse als mit den anderen Kernen berichtet. A ist ein Skalierungsparameter, eine gute Wahl dafür ist die mittlere Distanz zwischen den Trainingsinstanzen. Der χ^2-Kern wird für M-dimensionale Merkmale bestimmt gemäß

$$\begin{aligned} d^{\chi^2}(\mathbf{y}_i, \mathbf{y}_j) &= \frac{1}{2} \sum_{m=1}^{M} \frac{(y_{i,m} - y_{j,m})^2}{y_{i,m} + y_{j,m}}, \\ K^{\chi^2}(\mathbf{y}_i, \mathbf{y}_j) &= \exp\left(-\frac{1}{A}\frac{1}{2} \sum_{m=1}^{M} \frac{(y_{i,m} - y_{j,m})^2}{y_{i,m} + y_{j,m}}\right). \end{aligned} \tag{5.57}$$

Weiterhin wird der *Histogram Intersection*-Kern (HIK) betrachtet. Dieser ergibt sich aus

$$K^{\mathrm{HIK}}(\mathbf{y}_i, \mathbf{y}_j) = \sum_{m=1}^{M} \min(y_{i,m}, y_{j,m}) \,. \tag{5.58}$$

Dieser Kern zählt die Anzahl der gemeinsamen Werte in Histogramm-Abschnitten.

Vor der Klassifikation mit SVMs sollten die Deskriptoren zunächst skaliert werden [17]. Dies hat zum einen den Vorteil, dass bestimmte Merkmale nicht aufgrund ihres Wertebereiches das Modell dominieren und zum anderen werden numerische Probleme vermieden [17]. Dabei wird jede Merkmalsdimension separat auf den Wertebereich $[0, 1]$ normiert.

Merkmalsfusion

Bei Mehrkanal-SVMs erfolgt die Fusion mehrerer Kanäle durch Addition der Distanzen der einzelnen Kanäle. Gegeben seien Deskriptoren für N_{ch} Kanäle $\mathbf{y}^{ch}$, $ch = 1, \ldots, N_{\mathrm{ch}}$. Es werden der χ^2- und der *Histogram Intersection*-Kern verwendet. Für den χ^2-Kern hat sich in dieser Arbeit folgende Vorgehensweise als vorteilhaft erwiesen: Vor der Summation erfolgt zunächst eine Skalierung der einzelnen Kanäle

$$\begin{aligned} d^{\chi^2,\mathrm{mc}}(\mathbf{y}_i, \mathbf{y}_j) &= \sum_{ch=1}^{N_{\mathrm{ch}}} \frac{1}{A_{ch}} d^{\chi^2}(\mathbf{y}_i^{ch}, \mathbf{y}_j^{ch}) \quad \text{mit} \\ A_{ch} &= \frac{1}{(N^{\mathrm{train}})^2} \sum_{i=1}^{N^{\mathrm{train}}} \sum_{j=1}^{N^{\mathrm{train}}} d^{\chi^2}(\mathbf{y}_i^{\mathrm{train},ch}, \mathbf{y}_j^{\mathrm{train},ch}) \,. \end{aligned} \tag{5.59}$$

Zur Bestimmung des χ^2-Kerns der Mehrkanal-SVM werden die fusionierten Distanzen ein weiteres Mal skaliert. Es ergibt sich schließlich

$$K^{\chi^2,\mathrm{mc}}(\mathbf{y}_i, \mathbf{y}_j) = \exp\left(-\frac{1}{A} d^{\chi^2,\mathrm{mc}}(\mathbf{y}_i, \mathbf{y}_j)\right) \quad \text{mit}$$

$$A = \frac{1}{(N^{\mathrm{train}})^2} \sum_{i=1}^{N^{\mathrm{train}}} \sum_{j=1}^{N^{\mathrm{train}}} d^{\chi^2,\mathrm{mc}}(\mathbf{y}_i^{\mathrm{train}}, \mathbf{y}_j^{\mathrm{train}}) \,. \tag{5.60}$$

Durch die Skalierung der einzelnen Kanäle und der Gesamtdistanz wird eine deutlich bessere Klassifikation erreicht als wenn nur die fusionierte Distanz skaliert wird.

Bei der Klassifikation mit dem *Histogram Intersection*-Kern ist keine zusätzliche Skalierung notwendig, da die Deskriptoren bereits auf den Wertebereich $[0, 1]$ normiert sind. Somit ergibt sich

$$K^{\mathrm{HIK,mc}}(\mathbf{y}_i, \mathbf{y}_j) = \sum_{ch=1}^{N_{\mathrm{ch}}} \sum_{m=1}^{M} \min(y_{i,m}^{ch}, y_{j,m}^{ch}) \,. \tag{5.61}$$

5.5. Sequenzielle Aktionsmodellierung

Die *„Bag of Words"*-Modellierung ist eine verbreitete und erprobte Methode zur Aktionserkennung mit lokalen Merkmalen. Da der Videodeskriptor dabei Informationen über die gesamte Aktion bzw. Videosequenz enthält, ist er bereits sehr aussagekräftig. Durch die lose Sammlung der Merkmale ist diese Methode außerdem robust gegenüber Variationen innerhalb der einzelnen Klassen, wie der Ausführungsgeschwindigkeit oder der Reihenfolge einzelner Teil-Ereignisse. Auch Unterschiede der örtlichen Anordnung von Szenen werden teilweise kompensiert – abgesehen von den Zellen-Deskriptoren und von Effekten auf die lokalen Merkmale selbst. Jedoch geht durch das Ignorieren der Zusammenhänge zwischen Merkmalen eine Menge Information verloren, die insbesondere bei komplexen und ähnlichen Handlungen wichtig ist. Durch die Verwendung von Rastern in Abschnitt 5.4.3 wird zusätzlich Wissen darüber berücksichtigt, *wo* in der Szene bestimmte Ereignisse stattfinden. In einigen Arbeiten, wie z. B. [58, 92, 99] wurde der Versuch unternommen,

die *zeitliche* Anordnung von Merkmalen ebenso durch die Einteilung von Videos in zeitliche Zellen zu repräsentieren. Dies ist jedoch nicht immer erfolgreich – in [58] erweisen die zeitlichen Zellen sich nicht als vorteilhaft.

Ein weiterer Nachteil an der BoW-Vorgehensweise ist, dass davon ausgegangen wird, in einer vorhandenen Sequenz lediglich eine Aktion vorzufinden. Die Verarbeitung unsegmentierter Beobachtungen ist hier nicht inhärent berücksichtigt. Was einerseits die Stärke dieser Methode ist – das Betrachten der gesamten Information über eine Sequenz – wird zur Schwierigkeit, wenn man sich mit der Aufgabe konfrontiert sieht, eine kontinuierliche Beobachtungssequenz mit mehreren aufeinanderfolgenden Aktionen vorzufinden, ohne die Start- und Endzeiten dieser Aktionen zu kennen.

Hier ist neben der Aktionserkennung auch eine Aktions*segmentierung* gefragt. Die Segmentierung komplett getrennt von der Erkennung durchzuführen, ist schwierig und häufig ungenau [136, 101]. Hier sollen diese beiden Schritte gemeinsam mittels eines Sequenzmodells erfolgen. HMMs und CRFs wurden bereits erfolgreich für die Modellierung menschlicher Bewegungen eingesetzt. Dies erfolgt allerdings meist im Zusammenhang mit Posenverläufen oder globalen Merkmalen, z. B. Sequenzen von Körpersilhouetten. Solche Modelle haben den Vorteil, dass die Modellierung zeitlicher Abläufe in ihrer Natur liegt. Neben der Modellierung zeitlicher Abfolgen von Merkmalen berücksichtigen sie die Wahrscheinlichkeiten von Zustandsübergängen auf natürliche Weise, d. h. die Informationen darüber, welche Aktionen häufig nacheinander auftreten.

In diesem Abschnitt werden die bisher betrachteten Merkmalstrajektorien im Rahmen eines Sequenzmodells eingesetzt. Um dies zu bewerkstelligen, stellen sich zunächst zwei Fragen: Welche Modellform eignet sich und wie können die lokalen Merkmale in zeitabhängige Beobachtungsfolgen überführt werden? Um die Eignung des verwendeten Modells zu untersuchen, wird dieses außerdem für die Modellierung von Posenverläufen eingesetzt.

5.5.1. Modell

Die sequenzielle Aktionsmodellierung erfolgt mit einem linearen Ketten-CRF nach Abschnitt 5.3.4. Dadurch können Beobachtungen und Wissen über Zustandsübergänge holistisch modelliert werden. Das Wegfallen der Unabhängigkeitsannahmen bezüglich der Beobachtungen im Vergleich zu HMMs ermöglicht die Modellierung von zeitlichem *Kontext*, d. h. der Zusammenhang des Zustandes zu einem bestimmten Zeitpunkt mit Beobachtungen *anderer* Zeitpunkte kann in das Modell integriert werden. Zur Erkennung der Aktion zu einem bestimmten Zeitpunkt kann damit eine längere Zeitspanne betrachtet werden, woraus ein reicheres Modell resultiert.

CRFs wurden bereits in einigen Arbeiten zur Modellierung menschlicher Bewegungen verwendet, sowohl für atomare Aktionen als auch für komplexe Aktivitäten. In [89] werden lineare Ketten-CRFs mit HMMs und MEMMs (*Maximum Entropy Markov Models*) verglichen. Dort werden zwei unterschiedliche Arten von Merkmalen betrachtet, nämlich zum einen Sequenzen von (synthetisch erzeugten) Bildsilhouetten und zum anderen Gelenkwinkel-Verläufe, die beide aus *Motion Capture*-Daten gewonnen werden. Es werden zehn Aktivitäten betrachtet. Außerdem wird die Wirkung von Kontext-Merkmalen untersucht, wobei Beobachtungen von bis zu drei Schritten in die Zukunft und Vergangenheit betrachtet werden. Es wird gezeigt, dass CRFs den beiden anderen Modellformen überlegen sind und dass die Verwendung von Kontext-Information einen Mehrwert für die Erkennung bringt. Ein ähnliches Vorgehen wie in [89] wird auch hier angewandt.

Das verwendete lineare Ketten-CRF besitzt die Form

$$
\begin{aligned}
p(\mathbf{x}|\mathbf{y}) &= \frac{1}{Z(\mathbf{y})} \prod_{t=1}^{T} \psi(x_t, x_{t-1}, \mathbf{y}, t)\,, \\
Z(\mathbf{y}) &= \sum_{\mathbf{x}} \prod_{t=1}^{T} \psi(x_t, x_{t-1}, \mathbf{y}, t)\,.
\end{aligned}
\tag{5.62}
$$

Dabei beschreibt $\mathbf{x} = [x_1, \ldots, x_T]$ die Zustandssequenz mit der Beobachtungsdauer T. Der Zustand x_t stellt die Aktion dar, die zum diskreten Zeitpunkt t ausgeführt wird. Die Beobachtungen sind eine Sequenz von Merkmalsdeskriptoren $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_T]$.

Die Potentiale werden formuliert zu

$$\psi(x_t, x_{t-1}, \mathbf{y}) = \psi_{\text{tr}}(x_t, x_{t-1}) + \psi_{\text{so}}(x_t, \mathbf{y}, t)\,. \tag{5.63}$$

Für die Transitions-Potentiale $\psi_{\text{tr}}(x_t, x_{t-1})$ werden Merkmalsfunktionen verwendet, die nur von Zustandsübergängen zweier aufeinanderfolgender Zeitpunkte abhängen. $\psi_{\text{so}}(x_t, \mathbf{y})$ modelliert die Zusammenhänge zwischen Aktionen und Beobachtungen. Die Potentiale in Gleichung (5.63) setzen sich jeweils aus bestimmten Merkmalsfunktionen zusammen, welche aus den Zuständen und Beobachtungen gebildet werden. Es werden drei Merkmalstypen verwendet: die aktuellen Deskriptoren selbst, Informationen über Zustandsübergänge und Kontextmerkmale.

5.5.2. CRF-Merkmale

In Gleichung (5.63) wird eine Formulierung der Potentiale gewählt, die eine Unterscheidung zwischen der Modellierung der Zustandsübergänge und den Zusammenhängen zwischen Zuständen und Beobachtungen trifft. Daraus ergeben sich zwei unterschiedliche Arten von Merkmalen. Die Potentiale der Zustandsübergänge werden aus den Merkmalsfunktionen $f_{\text{tr}}(x_t, x_{t-1})$ gebildet:

$$\psi_{\text{tr}}(x_t, x_{t-1}) = \exp\left(\sum_{k=1}^{K_{\text{tr}}} \lambda_{\text{tr},k} f_{\text{tr},k}(x_t, x_{t-1})\right)\,. \tag{5.64}$$

Die Clique zweier benachbarter Zustände x_t und x_{t-1} wird nun als $\mathcal{X}_t = \{x_{t-1}, x_t\}$ bezeichnet. Die Transitions-Merkmale besitzen die Form

$$f_{\text{tr},k}(x_t, x_{t-1}) = \mathbf{1}_{\{\mathcal{X}_t = \tilde{\mathcal{X}}_{t,k}\}}\,, \quad k = 1, \ldots, K_{\text{tr}}. \tag{5.65}$$

Die Mengen $\tilde{\mathcal{X}}_{t,k}$ beinhalten dabei alle möglichen Paare an Werten, die die beiden aufeinanderfolgenden Zustände x_t und x_{t-1} annehmen können. $\mathbf{1}_{\{\mathcal{X}_t = \tilde{\mathcal{X}}_{t,k}\}}$ ist die Indikatorfunktion

$$\mathbf{1}_{\{\mathcal{X}_t = \tilde{\mathcal{X}}_{t,k}\}} = \begin{cases} 1 & \text{falls } x_{t-1} = \tilde{x}_{t-1,k} \text{ und } x_t = \tilde{x}_{t,k} \\ 0 & \text{sonst} \end{cases}\,. \tag{5.66}$$

Jede Merkmalsfunktion nimmt nur für eine Konfiguration von $\tilde{\mathcal{X}}_{t,k}$ einen Wert ungleich null an. Dies bedeutet, dass für jeden möglichen Zustandsübergang ein Gewicht $\lambda_{\text{tr},k}$ verwendet wird. Können S mögliche Zustände, d. h. Aktionsklassen, auftreten, ergeben sich somit $K_{\text{tr}} = S^2$ Merkmalsfunktionen für die Zustandsübergänge.

Die Zustands-Beobachtungs-Potentiale lauten

$$\psi_{\text{so}}(x_t, \mathbf{y}, t) = \exp\left(\sum_{k=1}^{K_{\text{so}}} \lambda_{\text{so},k} f_{\text{so},k}(x_t, \mathbf{y}, t)\right) \tag{5.67}$$

mit den Merkmalen $f_{\text{so},k}(x_t, \mathbf{y}, t)$, welche den Zustand des Zeitpunktes t mit den Beobachtungen in Verbindung setzen:

$$f_{\text{so},k}(x_t, \mathbf{y}, t) = \mathbf{1}_{\{x_t = \tilde{x}_k\}} f_{\text{obs},k}(x_t, \mathbf{y}, t)\,, \quad k = 1, \ldots, K_{\text{so}}\,. \tag{5.68}$$

Dabei nimmt das k-te Merkmal nur für einen Zustand $\tilde{x}_k \in \{1, \ldots, S\}$ Werte ungleich null an. Die Funktionen $f_{\text{obs},k}(\mathbf{y}, t)$ hängen nur noch von den Beobachtungen ab. Dies kann so verstanden werden, dass die Merkmalsfunktionen nur von den Beobachtungen abhängen, jedoch für jeden Zustand andere Gewichte λ verwendet werden [93]. Um Verwechslungen mit den Merkmalen $f_{\text{so},k}(x_t, \mathbf{y}, t)$ zu vermeiden, werden die $f_{\text{obs},k}(x_t, \mathbf{y})$ analog zu [93] nicht als Merkmale, sondern als *Beobachtungsfunktionen* bezeichnet. Wenn für jeden Zustand die gleichen K_{obs} Beobachtungsfunktionen verwendet werden, ergeben sich $K_{\text{so}} = K_{\text{obs}} \cdot S$ Merkmale.

Als Beobachtungsfunktionen werden Deskriptoren mit Berücksichtigung eines gewissen zeitlichen Kontextes eingesetzt. Der Kontext wird definiert durch eine Menge an Zeitverschiebungen, für die der Deskriptor ausgewertet wird. Es sei die Menge an Verschiebungen

$$\tau = \{\tau_l\}\,, \quad l = 1, \ldots, N_\tau\,. \tag{5.69}$$

Dann lauten die Beobachtungsfunktionen

$$f_{\text{obs},k}(\mathbf{y}, t) = y_{m_k}(t - \tau_k)\,, \quad m_k \in \{1, \ldots, M\}\,,\ \tau_k \in \tau. \tag{5.70}$$

Dabei ist τ_k der Kontext des k-ten Merkmals und M die Deskriptordimension. Es ergeben sich $M \cdot N_\tau$ Beobachtungsfunktionen und $K_{\text{so}} = M \cdot N_\tau \cdot S$ Zustands-Beobachtungsmerkmale.

5.5.3. Sequenzielle Deskriptoren für Merkmalstrajektorien

Die Bestimmung der sequenziellen Deskriptoren stellt eine Erweiterung des Vorgehens bei der BoW-Modellierung dar. Im Gegensatz zu einem Merkmalsvektor $\mathbf{y}$ für eine gesamte Sequenz ist nun ein zeitabhängiger Merkmalsvektor $\mathbf{y}(t)$ gesucht. Im Folgenden wird das Vorgehen zum Berechnen der Merkmalssequenz beispielhaft für einen Deskriptortyp erläutert.

Der Ablauf der merkmalsbasierten Aktivitätsanalyse mit sequenzieller Modellierung ist in Abbildung 5.11 veranschaulicht. Analog zu Abschnitt 5.4.2 wird zunächst eine Vektorquantisierung vorgenommen. Dazu wird nach dem gleichen Ablauf wie zuvor das *Codebook* ermittelt und jedem Trajektorien-Deskriptor $\mathbf{d}_j$ durch Projektion auf das *Codebook* ein Wort w_j zugeordnet.

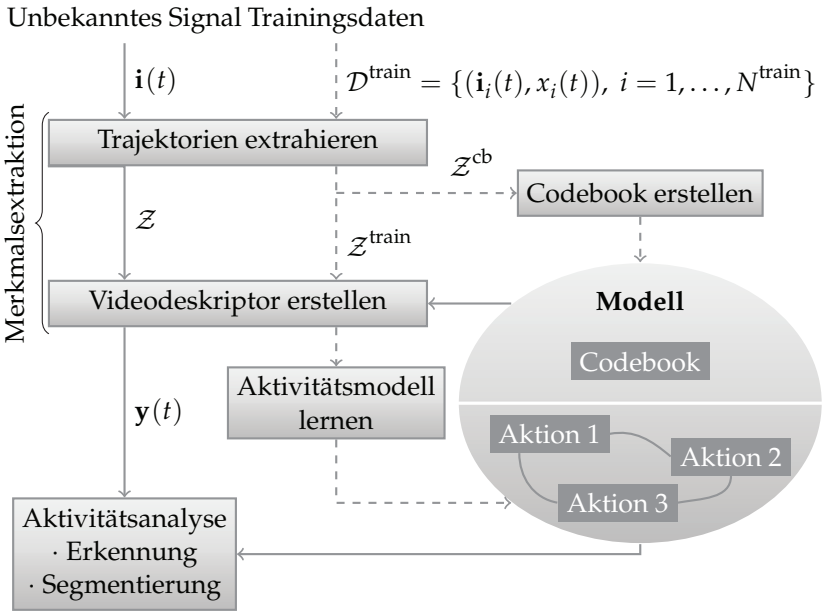


Abbildung 5.11. Übersicht über die merkmalsbasierte Aktivitätsanalyse mit sequenzieller Modellierung.

Der sequenzielle Deskriptor hat die Form

$$\mathbf{y}(t) = [y_1(t), \ldots, y_W(t)]^{\mathrm{T}} \,. \tag{5.71}$$

Die Komponente $y_w(t)$ gibt dabei den Verlauf des Auftretens des w-ten Wortes über der Zeit an. Im Gegensatz zu früher wird hierbei der Zeitbereich τ_j berücksichtigt, in dem die Trajektorie Z_j aktiv ist. Dies ist das Zeitintervall zwischen dem Zeitpunkt $t_{\mathrm{b},j}$, an dem Z_j initialisiert und dem Zeitpunkt $t_{\mathrm{d},j}$, an dem Z_j deaktiviert wurde. Damit ergibt sich für die w-te Komponente des Deskriptors

$$y_w(t) = \sum_{j=1}^{N} \delta_{w_j,w} \cdot \Big(\sigma(t - t_{\mathrm{b},j}) - \sigma(t - t_{\mathrm{d},j}))\Big) \,, \quad w = 1, \ldots, W \,. \tag{5.72}$$

Dabei ist N die Gesamtzahl der Trajektorien der aktuellen Sequenz und $\sigma(t)$ die Sprungfunktion

$$\sigma(t) = \begin{cases} 0 \,, & t < 0 \\ 1 \,, & t \geq 0 \end{cases} \,. \tag{5.73}$$

Der resultierende sequenzielle Deskriptor $\mathbf{y}(t)$ wird auf den Wertebereich $[0, 1]$ normiert.

Werden mehrere Merkmalstypen $\{\mathbf{y}^{ch}\}$, $ch = 1, \ldots, N_{\mathrm{ch}}$, verwendet, ergibt sich der gesamte Deskriptor durch Zusammenfügen der einzelnen Deskriptoren

$$\mathbf{y}(t) = \left[\mathbf{y}^{1,\mathrm{T}}(t), \ldots, \mathbf{y}^{N_{\mathrm{ch}},\mathrm{T}}(t)\right]^{\mathrm{T}} \,. \tag{5.74}$$

Dabei ist zu beachten, dass die individuellen Deskriptoren vor dem Zusammenfügen separat auf den Wertebereich $[0, 1]$ normiert werden. Die einzelnen Deskriptoren können mit oder ohne Rastereinteilung gemäß Abschnitt 5.4.3 bestimmt werden.

5.5.4. Sequenzielle Deskriptoren für Posenverläufe

Posenverläufe, die durch *Motion Capturing* oder markerloses Körper-Tracking ermittelt werden, stellen bereits sequenzielle Deskriptoren dar.

Sie können daher ohne Weiteres in das Sequenzmodell integriert werden. Es werden dazu lediglich einige Vorverarbeitungsschritte durchgeführt. Im Falle von *Motion Capture*-Daten liegen Verläufe von Markerpositionen vor. Zur Bildung der Deskriptoren werden 19 Marker ausgewählt. Diese repräsentieren die Pose des Torsos und die Gelenkpositionen von Schultern, Ellbogen, Händen, Hüften, Knien und Füßen. Die Markerverläufe hängen von der Position der Person im dreidimensionalen Raum ab. Daher werden alternativ zu den absoluten Werten relative Markerpositionen betrachtet. Dazu wird die Position der Person durch einen Bezugspunkt repräsentiert, welcher aus den Markern des Torsos berechnet wird, und für die restlichen Gelenkpositionen wird der Abstand zum Bezugspunkt ermittelt. Die Zustandsverläufe des markerlosen Körper-Trackings liegen in Form von Gelenkwinkeln vor. Diese können entweder selbst als Deskriptoren verwendet werden oder in die virtuellen Markerpositionen transformiert werden, wie sie in Abschnitt 3.4.2 zur Fehlerberechnung eingesetzt werden. Ebenso wie im vorigen Abschnitt erfolgt eine Normierung der sequenziellen Deskriptoren auf den Wertebereich $[0, 1]$.

Alternativ zu den Posenverläufen selbst werden sog. *Relational Features* [66] betrachtet. Diese sind geometrische Merkmale, die aus den Posenverläufen berechnet werden und qualitativ die räumlichen Beziehungen zwischen Körperteilen wiedergeben. Sie wurden entwickelt, um die Semantik von Posen zu repräsentieren. Der Gedanke dahinter ist, dass die gleiche Aktion starke Unterschiede im Posenverlauf aufweisen kann. Diese Merkmale haben somit zum Ziel, die wesentlichen Charakteristika von Aktionen zu modellieren und irrelevante Bewegungsinhalte zu ignorieren. Für weiterführende Informationen und Details zur Umsetzung sei auf [66, 136] verwiesen.

5.6. Ergebnisse

Im Folgenden werden die Ergebnisse der Aktivitätsanalyse besprochen. In Abschnitt 5.6.1 wird die Aktionserkennung mit dem BoW-Modell betrachtet. Dazu wird zunächst die vorgestellte Methode des Merkmalstrackings untersucht. Danach werden die verschiedenen Trajektorien-Deskriptoren miteinander verglichen und schließlich auf die Fusion

verschiedener Merkmale eingegangen. In Abschnitt 5.6.2 werden die Ergebnisse der sequenziellen Modellierung diskutiert. Dies erfolgt zunächst anhand der Segmentierung von Aktionen basierend auf den Merkmalstrajektorien. Anschließend wird das Sequenzmodell für modellbasierte Posenverläufe betrachtet.

5.6.1. Bag of Words-Modell

Datensätze

Viele etablierte Datensätze für die Aktivitätserkennung bestehen aus einfachen Aktionen und Aufnahmen geringer Auflösung. In [62] wurde ein Datensatz für die Aktivitätserkennung mittels Merkmalstrajektorien erstellt, welcher ein Szenario umsetzt, für das Merkmalstrajektorien besonders geeignet sind. Dieser Datensatz wird als *Activities of Daily Living* (ADL)-Datensatz bezeichnet und enthält hoch aufgelöste Aufnahmen komplizierter Aktivitäten. Er besteht aus zehn verschiedenen Aktivitäten, die in jeweils drei Wiederholungen von fünf Personen ausgeführt werden. Es wird ein Alltags-Szenario in einer Küchen-Umgebung umgesetzt, bei der die Personen mit einem Abstand von ca. zwei Metern mit 30 Hz und einer Bildauflösung von 1289×720 gefilmt werden. Die Testpersonen wurden so ausgewählt, dass sie sich in ihren Erscheinungsformen relativ stark unterscheiden. Es werden komplexe Aktivitäten verwendet, die sich auf Basis einzelner Merkmalstypen schwierig voneinander unterscheiden lassen [62]. Einige Aktivitäten beinhalten ähnliche Bewegungen, wie das Essen verschiedener Mahlzeiten, jedoch verschiedene beteiligte Objekte. In anderen Fällen sind die gleichen Objekte an unterschiedlichen Aktionen beteiligt, wie z. B. das Schälen, Schneiden und Essen von Bananen oder das Entgegennehmen und Betätigen eines Telefonanrufes. Seit seiner Veröffentlichung wurde der ADL-Datensatz in vielen weiteren Arbeiten angewandt; in Tabelle 5.1 sind Ergebnisse aus der Literatur aufgelistet. Da das hier vorgeschlagene Merkmalstracking auf ein solches Anwendungsszenario abzielt, wird auch hier dieser Datensatz angewandt.

Tabelle 5.1. Ergebnisse in der Literatur für den ADL-Datensatz.

Methode	*acc* /%
Temporal Templates [11] (getestet von [62])	33
Kuboide [29] (getestet von [62])	36
STIPs [58] (getestet von [62])	59
Quantisierte Geschwindigkeiten [62]	63
Latente quantisierte Geschwindigkeiten [62]	67
Angereicherte Geschwindigkeitsverläufe [62]	89
Tracklets [74]	82,67
Saliente Trajektorien [110]	98
Spärlicher optischer Fluss [54]	82

Die Auswertung erfolgt nach dem *Leave One Out* (LOO)-Prinzip. Dabei werden jeweils alle Aufnahmen einer Person als Test- und die der restlichen Personen als Trainingsdaten verwendet. Dieses Vorgehen wird wiederholt, so dass jede Person einmal die Testperson darstellt. Das Ergebnis ergibt sich als die mittlere Genauigkeit *acc* der Klassifikation über alle Durchläufe. Die gesamte Auswertung wird dreimal wiederholt und die mittlere Genauigkeit betrachtet.

Messing et al. [62] vergleichen ihren Trajektorien-basierten Ansatz mit zwei Methoden, welche lokale STIP-Merkmale verwenden (Dollárs Kuboide [29] und Laptevs STIP-Merkmale [58]), sowie einem Verfahren, welches ein globales Merkmal für eine gesamte Sequenz ermittelt (*Temporal Templates* [11]). Die in [62] erzielten Ergebnisse sind in den ersten drei Zeilen von Tabelle 5.1 abgebildet, woraus ersichtlich wird, dass in diesem Szenario Trajektorien-Ansätze deutlich überlegen sind. Alleine mit Verschiebungsvektoren von Merkmalstrajektorien erzielen Messing et al. [62] eine Genauigkeit von 63 %. Durch Verwenden von Zusatzinformationen erreichen sie 89 %. Die salienten Trajektorien von Yi et al. erzielen eine Genauigkeit von 98 %.

Am Ende dieses Abschnitts wird die hier vorgestellte Methode anhand zweier weiterer Datensätze getestet. Der Weizmann-Datensatz [9, 41] setzt ein sehr einfaches Szenario um. Er besteht aus Videos geringer Auflösung (180×144) von zehn einfachen Aktionen wie Gehen, Laufen, Springen, Winken etc., die jeweils von neun Personen ausgeführt

werden. Für ein solches Szenario eignen sich globale Methoden besonders, und es wurden bereits Erkennungsergebnisse von 100 % (siehe z. B. [102]) erzielt. Auch wenn die hier vorgeschlagene Methode nicht für ein solches Szenario entwickelt wurde, soll getestet werden, wie sich das Verfahren bei diesem etablierten Datensatz verhält. Schließlich wird der *Hollywood Human Actions* (HOHA)-Datensatz betrachtet. Dies ist ein sehr anspruchsvoller Datensatz, da er Herausforderungen der Erkennung realistischer Aktionen in unkontrollierten Umgebungen bietet. Er enthält Ausschnitte aus Hollywood-Filmen, welche in Trainings-, Test- und Validierungsdaten unterteilt sind. Die verwendeten Klassen sind relativ einfache, kurze Aktionen, darunter Händeschütteln, Telefon abnehmen und Küssen. Die große Schwierigkeit liegt hier in den Umgebungsbedingungen. Es gibt große Unterschiede der Ausführung der Aktionen und der Aufnahmebedingungen, z. B. in der Aufnahmeperspektive oder der Beleuchtung. Außerdem treten häufig Kamerabewegungen, Bewegungen im Hintergrund, abrupte Szenenwechsel sowie Interaktionen zwischen Personen auf. Betrachtet man den Stand der Technik bezüglich dieses Datensatzes (siehe Tabelle 5.2), wird deutlich, dass die Interpretation realistischer Videosequenzen in unkontrollierten Szenarien ein ungelöstes Problem bleibt. Sun et al. [92] können hierbei durch Kombination verschiedener Kontextmerkmale eine Genauigkeit von 47,10 % erreichen. Die Methode in [105] mit einem Ergebnis von 47,60 % befasst sich explizit mit der Präsenz von Kamerabewegungen, indem Trajektorien zerlegt werden in einen Teil, der von Kamerabewegung herrührt, und einen aktionsspezifischen Teil.

Tabelle 5.2. Ergebnisse in der Literatur für den HOHA-Datensatz.

Methode	*acc* /%
STIPs [58]	38,40
Tracklets [74]	34,30
Trajectons [61]	31,10
SIFT-Trajektorien [92]	47,10
Lagrangian Particle Trajectories [105]	47,60

Vergleich der Merkmalstracker

Im Folgenden werden verschiedene Varianten des Merkmalstrackings und der Trajektorien-Initialisierung aus Kapitel 4 anhand des ADL-Datensatzes untersucht. Dazu werden zunächst verschiedene Tracker mit der gleichen Initialisierungsmethode betrachtet und anschließend wird bei gleichem Tracking die Initialisierung variiert. Für die Auswertung wird zunächst ein einzelner Merkmalstyp, der Dynamik-Deskriptor (Gleichung (4.52)), betrachtet. Es wird eine Aktionserkennung mit dem BoW-Modell gemäß Abschnitt 5.4 durchgeführt. Zur Erstellung der *Codebooks* werden aus allen Trainingsmerkmalen 20 000 zufällig ausgewählt. Für die Dimensionsreduktion der Trajektorien-Deskriptoren werden 20 Hauptkomponenten verwendet. Die Auswertung erfolgt für *Codebook*-Größen von 1000 und 2000 Worten, wobei die Ergebnisse für beide Werte gemittelt werden. Die Klassifikation erfolgt mittels SVM mit χ^2-Kern. Es wird die Implementierung der *LIBSVM*-Bibliothek [17] verwendet.

In Tabelle 5.3 sind Ergebnisse verschiedener Varianten der Merkmalsextraktion zu sehen. Zusätzlich zu der Genauigkeit der Aktionserkennung *acc* sind einige Kennwerte der ermittelten Trajektorien angegeben, jeweils gemittelt über alle Aktionsklassen. $\overline{L_{\text{mean}}}$ ist die mittlere Länge aller Trajektorien in Zeitschritten. $\overline{L_{\text{max}}}$ ist der Mittelwert der maximalen Trajektorienlänge und $\overline{N}$ ist die mittlere Anzahl an extrahierten Trajektorien.

Tabelle 5.3. Vergleich verschiedener Tracking- und Initialisierungsmethoden der Merkmalspunkte.

Tracker	$\overline{L_{\text{mean}}}$	$\overline{L_{\text{max}}}$	$\overline{N}$	*acc* / %
① OFT_{15}	15	15	1013	75,78
② OFT_{∞}	28	120	704	77,44
③ $\text{OFT-LBP}_{0,7}$	25	110	596	77,78
④ $\text{OFT-LBP}_{0,8}$	19	72	354	68,56
⑤ KLT	29	118	642	81,22
⑥ SURF+PD	27	116	750	81,44
⑦ GFTT+PD	30	112	367	71,87
⑧ PD	28	118	719	79

Die Parameter der unterschiedlichen Versuche sind in Tabelle B.1 aufgelistet. Für die Szenarien ① bis ⑤ werden verschiedene Tracker-Varianten mit der gleichen Initialisierungsmethode verwendet. Die Initialisierung erfolgt mit der Kombination aus periodischem und SURF-Detektor (siehe Abschnitt 4.3.2).

In verwandten Arbeiten wird häufig die maximal erlaubte Trajektorienlänge begrenzt, so dass nur Trajektorien von sehr kurzer Dauer extrahiert werden. Der Grund dafür ist, dass die Tracker schnell abdriften und somit lange Trajektorien häufig fehlerhaft sind. Es stellt sich allerdings die Frage, ob lange Trajektorien von guter Qualität mehr Informationen enthalten, wie beispielsweise in [91] konstatiert wird. In Tabelle 5.3 wird der sehr erfolgreiche Tracker aus [98] betrachtet (hier als „optischer-Fluss-Tracker"–OFT bezeichnet), mit Begrenzung der Trajektorien auf $L_{\max} = 15$ Schritte wie in [98] (① OFT_{15}). Als Nächstes wird dieser Tracker ohne Begrenzung der Trajektorienlänge verwendet (② OFT_{∞}). Die Szenarien ③ und ④ verwenden die in dieser Arbeit vorgestellte Methode, wobei für ④ ein strengeres *Matching*-Kriterium gewählt wurde, was in kürzeren und weniger Trajektorien resultiert. Zum Vergleich wird der KLT-Tracker [60] betrachtet ⑤ . Es wird die Implementierung aus der *OpenCV*-Bibliothek [12] verwendet. Der KLT-Tracker wird in den selben Rahmenalgorithmus wie die anderen Methoden eingebettet, so dass die Überprüfung der Trajektorien etc. auf die gleiche Weise durchgeführt wird. Außerdem werden Trajektorien verworfen, die große Sprünge in der Punktposition aufweisen, wie sie beim KLT-Tracker häufig auftreten.

Die Szenarien ⑥ bis ⑧ verwenden das gleiche Tracking-Verfahren wie Methode ③ bei unterschiedlicher Initialisierung. Bei ⑥ wird wie bei ③ der SURF- mit dem Bewegungsdetektor verknüpft, jedoch wird eine niedrigere Schwelle für den Bewegungsdetektor gewählt, so dass mehr Punkte, auch bei schwächerer Bewegung, initialisiert werden. Als Nächstes wird anstelle von SURF-Merkmalen der *Good Features to Track* (GFTT)-Detektor verwendet ⑦ bei gleichem Schwellwert für Bewegung wie in ⑥ . Schließlich wird bei ⑧ ausschließlich der Bewegungsdetektor zur Initialisierung eingesetzt.

Die verschiedenen Methoden und Parametrierungen resultieren in teilweise sehr unterschiedlichen Trajektorien-Kennwerten (Länge und Anzahl der Trajektorien). Dadurch wird ein fairer Vergleich der Verfahren erschwert. Es stellt sich beispielsweise die Frage, welche Bedeutung

die bloße Anzahl an Trajektorien für die Erkennungsrate hat und ob man durch Variation der Parameter bei einem vermeintlich schlechteren Tracker vergleichbare Ergebnisse wie bei einem vermeintlich besseren Tracker erhalten würde. Ergebnisse in der Literatur besagen, dass Trajektorien nicht zu spärlich sein dürfen, weshalb beispielsweise in [98, 99] auf sehr dichte Merkmale gesetzt wird, die aus einem gleichmäßigen Raster ohne weitere Merkmalsdetektion ausgewählt werden. In [110] wird allerdings gezeigt, dass durch eine selektivere Auswahl an Merkmalen bessere Ergebnisse bei weniger Trajektorien erzielt werden können.

Um einem fairen Vergleich der Methoden näher zu kommen, werden die Klassifikationsergebnisse der Menge und Länge der Trajektorien gegenübergestellt. Diese beiden Kennwerte sollten allerdings nicht getrennt voneinander betrachtet werden. Dies wird beispielsweise anhand Szenario ① (OFT_{15}) in Tabelle 5.3 ersichtlich, bei dem sich sehr viele, jedoch kurze Trajektorien ergeben. In Abbildung 5.12 sind die Klassifikationsergebnisse über dem Produkt aus $\overline{L_{\mathrm{mean}}}$ und $\overline{N}$ aufgetragen, welches als grobes Maß für die Größe des durch die Trajektorien abgedeckten Bereiches der Videos angesehen werden kann. Es ist zu erkennen, dass Szenarien, die insgesamt dichtere Merkmale ergeben, deutlich bessere Ergebnisse als spärliche Darstellungen liefern.

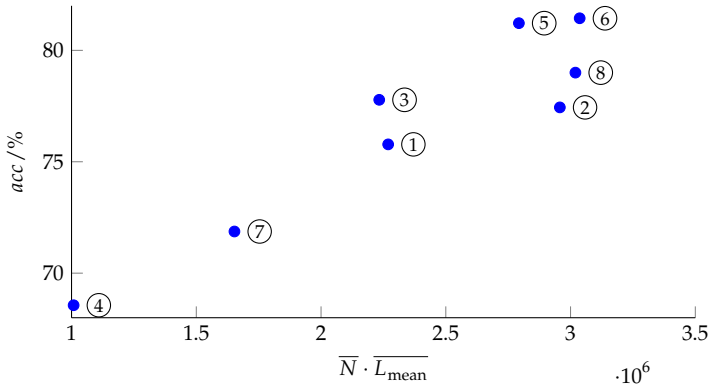


Abbildung 5.12. Zusammenhang der Kennzahlen der Trajektorien und dem Klassifikationsergebnis.

Szenario ③ befindet sich in Abbildung 5.12 oberhalb von ① und weist demnach bessere Ergebnisse bei vergleichbarer Trajektorienabdeckung vor. Bei ② , ⑤ , ⑧ und ⑥ liegt $\overline{L_{\text{mean}}} \cdot \overline{N}$ in einem ähnlichen Bereich. Hierbei schneiden die Trajektorien von OFT_{∞} am schlechtesten ab. Die Initialisierungen bei ⑥ (SURF+Bewegung) und ⑧ (nur Bewegung) ergeben zwar vergleichbare Trajektorien-Kennwerte, jedoch sind die Ergebnisse ohne SURF-Merkmale schlechter. Die beiden Methoden ④ und ⑦ liefern schlicht zu wenige Trajektorien, um mit den anderen Szenarien mithalten zu können. Dies liegt bei ersterer daran, dass die Trajektorien aufgrund der strengeren *Matching*-Schwelle schnell verworfen werden. Im zweiten Fall wurde der GFTT-Merkmalsdetektor verwendet, der mit der Parametrierung wie in [99] deutlich weniger Merkmale als der SURF-Detektor liefert.

Mit dem KLT-Tracker werden hier auch sehr gute Ergebnisse erzielt. Dies steht im Gegensatz zu Angaben der Literatur [99], wonach er deutlich schlechter als der auf optischem Fluss basierende Tracker abschneidet. Dort wird dies jedoch u. a. durch häufig auftretende „Sprünge“ des Trackers und eine zu geringe Anzahl an Merkmalen begründet. In dieser Arbeit werden jedoch Trajektorien, die solche Sprünge aufweisen, verworfen. Außerdem wird eine andere Initialisierungsmethode verwendet, wodurch sich dichtere Trajektorien ergeben. Die Länge der KLT-Trajektorien ist mit denen des LBP-Trackers vergleichbar. Da später ohnehin das dichte OF-Feld benötigt wird, wird auf den KLT-Tracker verzichtet. Dieser stellt jedoch eine interessante Alternative dar, wenn auf die Flussmerkmale verzichtet werden soll.

Vergleich der Trajektorien-Deskriptoren

Nachdem im vorigen Abschnitt das Augenmerk auf dem Tracking lag, werden nun die verschiedenen Deskriptortypen und die Klassifikation mit dem BoW-Modell untersucht. In diesem Abschnitt werden die einzelnen Merkmalsarten separat voneinander betrachtet, bevor auf die Fusion von Merkmalen eingegangen wird. Die Auswertung erfolgt wie im vorigen Abschnitt nach dem LOO-Prinzip. Es werden wieder drei Iterationen durchgeführt und die Ergebnisse gemittelt. Die *Codebook*-Größe wird hier jeweils von 1000 bis 4000 Wörtern variiert. Auf der Ebene

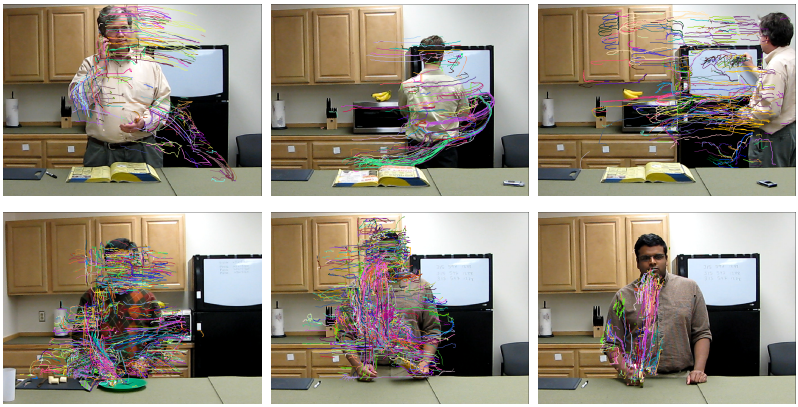


Abbildung 5.13. Beispiele geclusterter Trajektorien für den Dynamik-Deskriptor. Trajektorien, die einem bestimmten Wort zugeordnet werden, sind in derselben Farbe abgebildet.

der Aktionsdeskriptoren werden die vier Raster G_{11}, G_{13}, G_{31} und G_{22} verwendet.

Es werden alle in Abschnitt 4.3.4 erläuterten Deskriptoren ermittelt. Bei den Histogramm-Deskriptoren (HOG, HOF, MBH) werden wie in [98] $n_\sigma = 2$ örtliche Zellen verwendet. Bei den Summen-Deskriptoren (SG, SOF, MBS) werden außerdem Einteilungen in $n_\sigma = 4$ Zellen betrachtet, da diese mit geringerem Aufwand als die Histogramm-Deskriptoren berechnet werden können. In Anhang B, Tabelle B.2 wird gezeigt, dass die 4×4- den 2×2-Summen-Deskriptoren überlegen sind. Daher wird für diese im Folgenden $n_\sigma = 4$ verwendet. Alle Deskriptoren werden in einem quadratischen Bildbereich mit $\Delta_\sigma = 64$ bestimmt.

In den Abbildungen 5.13, 5.14 und 5.15 sind Beispiele für die ermittelten Trajektorienwörter für den Dynamik-, SG und den MBH-Deskriptor zu sehen. Dabei werden alle Trajektorien, die demselben Prototypen zugeordnet wurden, in der gleichen Farbe dargestellt.

Tabelle 5.4 zeigt die Klassifikationsergebnisse der verschiedenen Merkmalstypen für die einzelnen Raster sowie die Fusion aller Raster für Szenario ⑥ . Zur Klassifikation wird der χ^2-Kern verwendet. Die Fusion der Raster erfolgt mittels Mehrkanal-SVM, wie in Abschnitt 5.4.4 erläutert. Dabei werden die Einzeldeskriptoren vor der Fusion zunächst

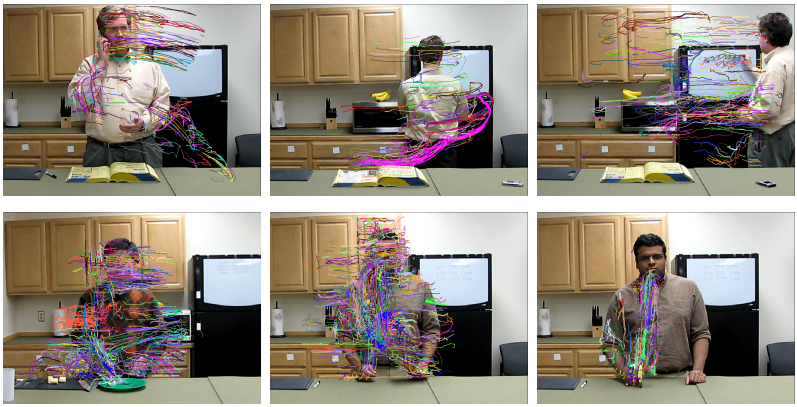


Abbildung 5.14. Beispiele geclusterter Trajektorien für den SG-Deskriptor. Trajektorien, die einem bestimmten Wort zugeordnet werden, sind in derselben Farbe abgebildet.

separat auf den Wertebereich $[0, 1]$ skaliert, wie es im Theorieteil dargelegt wurde. Zusätzlich erfolgt nach der Fusion eine weitere Skalierung der fusionierten Distanzen. Die letzte Zeile zeigt jeweils den Mittelwert über alle Merkmalstypen der jeweiligen Raster bzw. der Fusion.

Betrachtet man die Raster einzeln, so schneiden G_{31} und G_{13} am besten ab. G_{22} liefert die schlechtesten Ergebnisse. Durch Fusion der Raster lassen sich die Ergebnisse im Mittel weiter verbessern.

Vergleicht man die beiden Deskriptoren HOG und SG, welche beide Informationen über Bildgradienten im Bereich der Trajektorien beinhalten, ist zu erkennen, dass der SG-Deskriptor dem HOG-Deskriptor klar überlegen ist. Bei den Deskriptoren für den optischen Fluss schneiden dagegen die Histogramm-Deskriptoren besser ab. Bei den *Motion Boundary*-Deskriptoren liefert der MBH- deutlich bessere Ergebnisse als der MBS-Deskriptor. Die HOF- und SOF-Merkmale unterscheiden sich dagegen nur gering. Das beste Merkmal insgesamt ist der SG-Deskriptor mit einer Genauigkeit von 91,78 %, gefolgt vom MBH-Deskriptor mit 88,78 %. Der LBP- und der Dynamik-Deskriptor liefern die schlechtesten Ergebnisse.

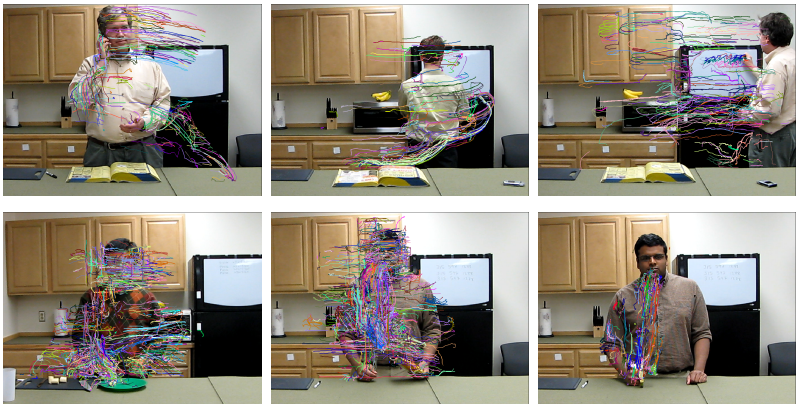


Abbildung 5.15. Beispiele geclusterter Trajektorien für den MBH-Deskriptor. Trajektorien, die einem bestimmten Wort zugeordnet werden, sind in derselben Farbe abgebildet.

Fusion verschiedener Merkmale

Nun werden nicht nur verschiedene Raster, sondern auch Deskriptortypen miteinander fusioniert.

Vorab werden unterschiedliche Zusammensetzungen der *Motion Boundary*-Deskriptoren betrachtet. Diese bestehen je aus zwei Komponenten MBH_x und MBH_y bzw. MBS_x und MBS_y. In [98] werden die beiden Komponenten der *Motion Boundary*-Histogramme zu einem Merkmalsvektor MBH zusammengefügt. In einer Nachfolgearbeit [99] wird dieser verglichen mit der separaten Verwendung der x- und y-Kanäle. Es werden leicht bessere Ergebnisse erzielt, wenn die Kanäle getrennt als eigenständige Merkmale in den Klassifikator einfließen. Reddy und Shah [75] adressieren ebenfalls die Frage, ob es besser ist, Merkmale früh oder spät zu fusionieren. Mit früher Fusion bezeichnen die Autoren eine Fusion vor der Klassifikation, beispielsweise durch Zusammenfügen der Merkmalsvektoren. Bei der späten Fusion werden Ergebnisse der Klassifikation einzelner Merkmale fusioniert. Die späte Fusion schneidet im Mittel besser als die frühe Fusion ab.

Tabelle 5.4. Deskriptoren einzelner Raster und fusioniert für verschiedene Merkmalstypen. Die Klassifikation erfolgt mit dem χ^2-Kern.

Deskriptor	$acc_{G_{11}}/\%$	$acc_{G_{31}}/\%$	$acc_{G_{13}}/\%$	$acc_{G_{22}}/\%$	$acc_{\text{fus}}/\%$
HOG	80,56	84,89	83	74,67	86,89
SG	86,22	87,33	85,67	79,78	**91,78**
LBP	72,56	83,11	75,22	73,67	83,33
HOF	81,44	86,56	85,11	78,89	86
SOF	80,11	85,67	82,44	77,56	86,11
MBH	84,44	86,22	85,78	76,44	88,78
MBS	77	84,44	79,22	74	85,67
Dyn	82,11	83,67	83,67	77,11	84,78
Mittelwert	80,56	85,24	82,51	76,52	86,67

Für den MBH und MBS-Deskriptor werden nun unterschiedliche Fusionsmechanismen betrachtet, welche in Tabelle 5.5 miteinander verglichen werden. Es werden die gleiche Auswertungsmethode und Parameter wie im vorigen Abschnitt verwendet. Es sind die Ergebnisse bei Fusion aller vier Raster abgebildet. Als Erstes wird eine *frühe Fusion* betrachtet, welche sich ergibt, wenn die x- und y-Kanäle zu einem Deskriptor zusammengefügt werden. Dies entspricht dem MBH bzw. MBS-Deskriptor, der in den vorigen Abschnitten bereits betrachtet wurde. In Tabelle 5.5 werden die Ergebnisse der frühen Fusion zusammenfassend mit F_{xy} bezeichnet und sind in der ersten Spalte zu sehen. Als Nächstes wird eine Variante getestet, bei der die x-/y-Kanäle als separate Deskriptoren in die Mehrkanal-SVM eingehen. Diese Methode wird in [99] der frühen Fusion gegenüber bevorzugt. Diese Methode wird als *späte Fusion* bezeichnet und in Tabelle 5.5 durch F_xF_y repräsentiert. Es sei angemerkt, dass der Begriff auf andere Weise als in [75] verwendet wird, denn hier erfolgt die späte Fusion nicht nach der Klassifikation, sondern indem die Merkmale als eigenständige Kanäle für den Klassifikator verwendet werden. Schließlich sind in der dritten Spalte Ergebnisse für eine Fusion mit *redundanten Deskriptoren* dargestellt, welche sich aus Kombination der frühen und späten Fusion ergeben. Hierbei fließen drei Merkmalskanäle in die Mehrkanal-SVM ein, die separaten x- und y-Deskriptoren

Tabelle 5.5. Vergleich von Fusionsmethoden für den MBH- und MBS-Deskriptor anhand dreier Szenarien. Die erste Spalte zeigt die Ergebnisse der frühen Fusion (F_{xy}), die zweite der späten Fusion (F_xF_y) und die dritte die Ergebnisse mit redundanten Merkmalen ($F_{xy}F_xF_y$).

Deskriptor	$acc_{F_{xy}}$ / %	$acc_{F_xF_y}$ / %	$acc_{F_{xy}F_xF_y}$ / %
MBH ③	83,67	82,67	84,44
MBS ③	84,67	85,44	86,67
MBH ⑧	89,89	88,11	90
MBS ⑧	85,11	87,11	87,56
MBH ⑥	88,78	88,67	90,33
MBS ⑥	85,67	87,67	86,67
Mittelwert	86,30	86,61	87,61

F_x und F_y sowie die kombinierten Merkmale der frühen Fusion F_{xy}. Die Auswertung wird für drei Versuchsszenarien durchgeführt. Im Mittel schneidet die späte Fusion besser als die frühe ab. Dies bestätigt die Aussagen in [99] und passt auch zu den Erkenntnissen in [75], obgleich dort die späte Fusion auf andere Weise durchgeführt wird. Interessant ist ein Blick in die dritte Spalte. Es lassen sich im Mittel weitere Verbesserungen erzielen, wenn die beiden vorigen Varianten kombiniert und redundante Deskriptoren verwendet werden.

Nun werden alle zur Verfügung stehenden Deskriptoren betrachtet. Tabelle 5.6 zeigt die Klassifikationsergebnisse bei der Fusion verschiedener Deskriptortypen. Es werden jeweils alle vier Raster herangezogen. Die Auswertung wurde mit und ohne redundante *Motion Boundary*-Deskriptoren durchgeführt und jeweils die besten Ergebnisse verwendet. Zur Klassifikation werden der HIK und der χ^2-Kern eingesetzt.

Zunächst werden lediglich die Histogramm- bzw. Summen-Deskriptoren getrennt voneinander betrachtet. Die Verwendung der Histogramm-Merkmale HOG+HOF+MBH resultiert in besseren Ergebnissen als die der Summen-Deskriptoren SG+SOF+MBS. Im vorigen Abschnitt (siehe Tabelle 5.4) wurde jedoch ersichtlich, dass, einzeln betrachtet, der SG-Deskriptor die besten Ergebnisse erzielt und deutlich besser als der HOG-Deskriptor ist. HOF und SOF liegen in einem

Tabelle 5.6. Fusion von Merkmalen und Rastern. Die erste Spalte zeigt die Ergebnisse mit dem HIK und die zweite die mit dem χ^2-Kern.

Deskriptoren	acc_{HIK} / %	acc_{χ^2} / %
HOG+HOF+MBH	91,78	91,67
SG+SOF+MBS	90,67	90,22
alle	92,33	91,44
SG+MBH	92,67	92,11
SG+HOF+MBH	92,44	92,78
SG+MBH+LBP	93,56	93,89
SG+HOF+MBH+LBP	93,56	93,33
SG+MBH+Dyn	93,56	93,67
SG+HOF+MBH+Dyn	91,67	92
SG+MBH+LBP+Dyn	**94,11**	93,78
SG+HOF+MBH+LBP+Dyn	93	92,78
SG+MBH+LBP+Dyn redundante Modelle	**95,11**	94,44

ähnlichen Bereich, während bei den *Motion Boundary*-Deskriptoren die Histogramm-Merkmale deutlich besser sind.

Es liegt daher nahe, jeweils die für den vorliegenden Datensatz am besten geeigneten Deskriptorkanäle zu selektieren. Daher werden nun Histogramm- und Summen-Deskriptoren miteinander kombiniert. Tabelle 5.6 zeigt Ergebnisse unterschiedlicher Kombinationen von Merkmalen, wobei bei den Gradienten- und Flussmerkmalen jeweils diejenigen verwendet werden, die einzeln die besseren Ergebnisse erzielen. Außerdem werden die Dynamik- und LBP-Merkmale hinzugezogen. Im Hinblick auf eine Reduzierung des Aufwandes bei der Merkmalsextraktion wird außerdem geprüft, ob auf den HOF-Deskriptor verzichtet werden kann, da dieser alleine deutlich schlechter als SG und MBH ist und der MBH-Deskriptor bereits Informationen über den optischen Fluss enthält.

Es erweist sich als sinnvoll, eine Kombination der jeweils besten Deskriptorarten vorzunehmen. Verwendet man nur SG+MBH oder SG+HOF+MBH, erhält man bessere Ergebnisse als in den darüber liegenden Zeilen in Tabelle 5.6, obwohl weniger Merkmale verwendet werden. In den meisten Fällen lassen sich die Ergebnisse verbessern, indem auf

die HOF-Merkmale verzichtet wird. Das Hinzunehmen der LBP- und Dynamik-Deskriptoren bringt dagegen weitere Verbesserungen. Dies ist der Fall, obwohl diese beiden Merkmale einzeln die schlechtesten waren. Die beste Konstellation ist die Kombination der SG-, MBH-, LBP- und Dynamik-Deskriptoren. Abbildung 5.16 zeigt dazu die Konfusionsmatrizen der fusionierten Merkmale verglichen mit den Einzelergebnissen für den SG-, MBH- und Dynamik-Deskriptor.

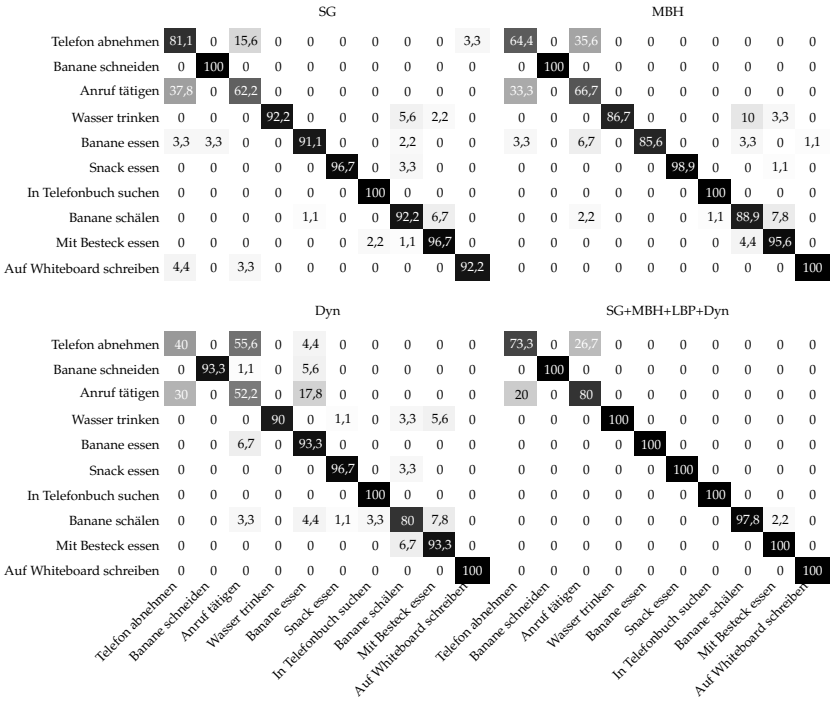


Abbildung 5.16. Konfusionsmatrizen zu SG+MBH+LBP+Dyn aus Tabelle 5.6: Einzelergebnisse für die SG-, MBH- und Dyn-Kanäle sowie Fusionsergebnis.

Aus den diskutierten Ergebnissen wird ersichtlich, dass die Verwendung komplementärer Informationen wichtig für die Aktionserkennung ist. Zum Vergleich werden alle zur Verfügung stehenden Deskriptoren

miteinander fusioniert. Die Ergebnisse sind in der dritten Zeile von Tabelle 5.6 abgebildet. Durch eine Selektion der besten Kanäle unter Berücksichtigung der Verwendung möglichst vielfältiger Informationsquellen ergeben sich bessere Ergebnisse als bei Fusion aller Merkmale.

Für die in Tabelle 5.6 dargestellten Versuche wurden Modelle mit unterschiedlichen *Codebook*-Größen verwendet und die mittleren Ergebnisse betrachtet. Aus den bisherigen Erkenntnissen lässt sich schließen, dass eine gewisse Redundanz der verwendeten Merkmalskanäle durchaus vorteilhaft sein kann. Beispielsweise ist die Fusion von Deskriptoren verschiedener Rastereinteilungen besser als das beste Raster alleine. Das Gleiche gilt für die redundanten *Motion Boundary*-Deskriptoren in Tabelle 5.5. Daher wird im Folgenden die Verwendung redundanter Modelle in Betracht gezogen. Dazu werden die Deskriptoren, die man durch Projektion auf *Codebooks* unterschiedlicher Größen, 1000 und 2000 Wörter, erhält, als einzelne Kanäle für einen gemeinsamen Klassifikator verwendet. Die Ergebnisse sind in Tabelle 5.6 in der untersten Zeile abgebildet. Hierdurch lassen sich die Erkennungsergebnisse weiter steigern auf 95,11 % im Falle des HIK.

Zum Abschluss soll die hier vorgestellte Methode auf die Weizmann- und HOHA-Datensätze angewandt werden. Aufgrund der geringen Auflösung und der Aufnahmeperspektive des Weizmann-Datensatzes driften die Trajektorien hier besonders schnell ab. Daher ist es hier vorteilhaft, keine Langzeit-Trajektorien zu betrachten. Die Trajektoriendauer wird daher auf $L_{\max} = 15$ begrenzt. Außerdem werden keine Bildraster verwendet. Für alle Trajektorien-Deskriptoren wird ein Bereich der Größe $\Delta_\sigma = 32$ und eine Zelleneinteilung mit $n_\sigma = 2$ angewandt. Wie bereits erwähnt, sind für ein solches Szenario globale Merkmale besser als Merkmalstrajektorien geeignet. Dennoch wird mit der hier vorgeschlagenen Methode eine Genauigkeit von 96,30 % erzielt. Die beste Deskriptorkombination ist hier die Verwendung von Dynamik- und Summen-Deskriptoren. Im Gegensatz zu Weizmann stellt der HOHA-Datensatz eine sehr große Herausforderung dar und wird noch nicht gut beherrscht. Zur Aktionspunkt-Detektion ist hier die Verwendung von GFTT+PD besser als SURF+PD geeignet, da sich damit dichtere Merkmale ergeben. Die Trajektoriendauer wird hier nicht begrenzt. Zur Deskriptorbestimmung wird wie zuvor $\Delta_\sigma = 32$ und $n_\sigma = 2$ gewählt. Bei

Verwendung von HOG- und MBH-Deskriptoren wird eine Genauigkeit von 37,17 % erzielt. Dieses Ergebnis übertrifft die beiden auf Merkmalstrajektorien basierten Ansätze [61, 74] und ist vergleichbar mit den STIP-Deskriptoren in [58] (siehe Tabelle 5.2). Bessere Ergebnisse werden dagegen von Methoden erzielt, die sich stärker auf Kontext-Merkmale fokussieren [92] oder das Auftreten von Kamerabewegungen explizit berücksichtigen [105].

Diskussion

Der Vergleich der verschiedenen Methoden zur Gewinnung der Merkmalstrajektorien zeigt, dass die bloße Menge an Trajektorien die Aktivitätserkennung entscheidend beeinflusst. Mit dichten Trajektorien lassen sich deutlich bessere Ergebnisse erzielen als mit spärlichen. Jedoch spielt auch die Qualität der Trajektorien eine wichtige Rolle. Zur Detektion der Aktionspunkte hat sich die Kombination der Salienzmaße für Bild- und Bewegungsinformation als am Besten geeignet erwiesen. Für komplexe Aktivitäten liefert die Gewinnung von Langzeit-Trajektorien bessere Ergebnisse als das Tracking über einen kurzen Zeitraum. Die hier vorgestellte zweistufige Tracking-Methode ist dafür dem reinen Tracking mit optischem Fluss überlegen, da ein Abdriften der Punkte verhindert wird.

In Bezug auf die verschiedenen Deskriptoren erlaubt die Fusion von Merkmalen eine deutlich bessere Aktionserkennung als die Verwendung einzelner Merkmale. Die Fusion von Rastern bringt eine leichte Verbesserung, während die Verwendung unterschiedlicher Merkmalstypen z. T. starke Verbesserungen mit sich bringt. Besonders lohnt es sich, komplementäre Kanäle zu kombinieren. Damit kann eine bessere Klassifikation auch durch Hinzunehmen schlechterer Einzel-Kanäle erreicht werden. Dies stimmt mit den Schlussfolgerungen in [111] überein. Es kann keine allgemeine Aussage darüber getroffen werden, ob die etablierten Histogramm-Deskriptoren oder die in dieser Arbeit vorgestellten Summen-Deskriptoren zu bevorzugen sind. Dazu ist eine Analyse des vorliegenden Szenarios und eine darauf basierende Merkmalsselektion lohnenswert.

5.6.2. Ergebnisse des Sequenzmodells

Datensätze

Das in Abschnitt 5.5 besprochene Sequenzmodell wird sowohl auf Merkmalstrajektorien als auch Posenverläufe angewandt. Zur Evaluation der sequenziellen Deskriptoren der Merkmalstrajektorien wird die gekoppelte Erkennung und Segmentierung von Aktionen betrachtet. Dazu werden die komplexen Aktivitäten des ADL-Datensatzes in atomare Aktionen zerlegt. Beispielsweise wird die Aktivität „Wasser trinken" unterteilt in die Aktionen „Kühlschrank öffnen", „Wasser holen", „Glas füllen" und „Trinken". Die Annotationen dazu sind in Tabelle B.3 gegeben. Es werden jeweils die Daten aller Personen und Wiederholungen verwendet.

Für die Bewegungsanalyse basierend auf Posenverläufen werden zwei Datensätze herangezogen. Die *CMU Graphics Lab Motion Capture Database* [20] enthält umfangreiche *Motion Capturing*-Aufnahmen sehr vieler verschiedener Bewegungsarten. Die Daten werden von zwölf Infrarot-Kameras mit 120 Hz aufgenommen. Die Testpersonen tragen spezielle *Motion Capturing*-Anzüge mit 41 Markern. Der Datensatz enthält Aufnahmen von insgesamt 144 Personen mit unterschiedlichen Aktionen. Die Aktionen zählen u. a. zu den Kategorien Fortbewegung, Sport und Interaktionen, wobei für viele Aktionen nur wenige Wiederholungen zur Verfügung stehen. Weiterhin wird der HumanEva-Datensatz (siehe Abschnitt 3.4) verwendet. Dieser wurde für die Evaluation von Methoden des Körper-Trackings entworfen und enthält nur wenige verschiedene Aktionen. Dennoch sollen die Ergebnisse des Körper-Trackings aus Kapitel 3 zur Bewegungsanalyse eingesetzt werden. Außerdem werden die *Motion Capture*-Daten des HumanEva-I-Datensatzes verwendet.

Sequenzielle Modellierung der Merkmalstrajektorien

Zunächst wird die Eignung der in Abschnitt 5.5 vorgestellten sequenziellen Deskriptoren und des CRF-Modells zur gemeinsamen Segmentierung und Erkennung von Aktionen geprüft. Die Auswertung erfolgt wie zuvor nach dem LOO-Prinzip.

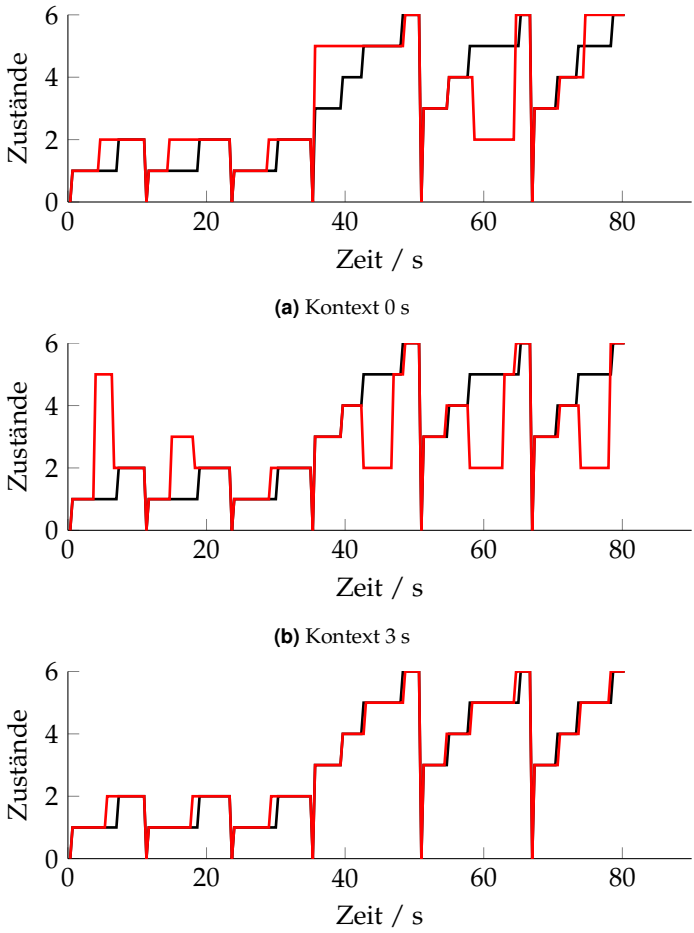
Die Merkmalstrajektorien selbst modellieren bereits lokale zeitliche Verläufe der Aktionspunkte. Im Rahmen der sequenziellen Modellierung

werden durch die Kontext-Merkmale (5.70) längerfristige Zusammenhänge hinzugenommen. Diese repräsentieren außerdem keine Zusammenhänge innerhalb einer Trajektorie, sondern zwischen den in einer Sequenz auftretenden Merkmalen, da sie die Verläufe der *Wortvorkommen* der Merkmale modellieren.

Zunächst wird die Auswirkung der Kontextlänge auf die Genauigkeit der Prädiktion der Zustandsverläufe untersucht. Dazu werden die Trajektorien aus Versuch ⑥ aus dem vorigen Abschnitt verwendet. Zur Verringerung des Aufwandes wird nur jeder 10-te Zeitschritt ausgewertet. Die Parameterschätzung und Inferenz des CRF erfolgt mit Hilfe der Implementierung der UGM-Toolbox [79]. Es werden hier keine Bildraster berücksichtigt, um die Merkmalsdimension einzuschränken. Diese können jedoch ohne Weiteres in das Modell integriert werden. Die sequenzielle Aktionserkennung wird für Kontextlängen ausgewertet, die von 0 bis 10 Sekunden variiert wird. Das Kontextfenster ist dabei um den aktuellen Zeitpunkt zentriert. Abbildung 5.17 zeigt beispielhaft die Segmentierungsergebnisse für eine Sequenz für Kontextfenster von 0, 3 und 10 Sekunden. Hierbei wird zunächst nur der Dynamik-Deskriptor verwendet. Es ist deutlich zu erkennen, dass die Miteinbeziehung längerfristiger Zusammenhänge zwischen Merkmalen eine deutliche Verbesserung des Modells ergibt.

In Abbildung 5.18 sind die Ergebnisse der Zustandsprädiktion über der Kontextlänge für den Dynamik–Deskriptor aufgetragen. Als Maß für die Genauigkeit wird hierbei der Prozentsatz korrekt prädizierter Zustandsvariablen, d. h. Aktionsklassen, acc_{lab} betrachtet. Die Ergebnisse sind über alle Personen gemittelt. Mit Verlängerung des Kontextes steigt die Genauigkeit deutlich an bis zu einer Kontextlänge von ca. zehn Sekunden, danach tritt eine Sättigung ein.

Im Folgenden erfolgt eine Fusion der verschiedenen Deskriptoren. Im Rahmen des Sequenzmodells bedeutet Fusion, dass die CRF-Merkmale für jeden Deskriptor gebildet und in das Modell einfließen. Dabei werden die einzelnen Deskriptoren zunächst auf den Wertebereich $[0, 1]$ normiert. Es wird ein Kontextfenster von 10 Sekunden verwendet. Zur Auswahl von Merkmalen wird auf die Erkenntnisse des vorigen Abschnittes zurückgegriffen und diejenigen Deskriptoren verwendet, deren Kombination bei der BoW-Modellierung die besten Ergebnisse lieferte.



(a) Kontext 0 s

(b) Kontext 3 s

(c) Kontext 10 s

Abbildung 5.17. Beispiel der Zustandsverläufe für drei verschiedene Kontextlängen. Schwarz zeigt die wahren, rot die prädizierten Zustandsverläufe.

Tabelle 5.7 zeigt die Einzelergebnisse des SG- MBH-, Dyn- und LBP-Deskriptors sowie die Ergebnisse aus Kombinationen dieser Merkmale.

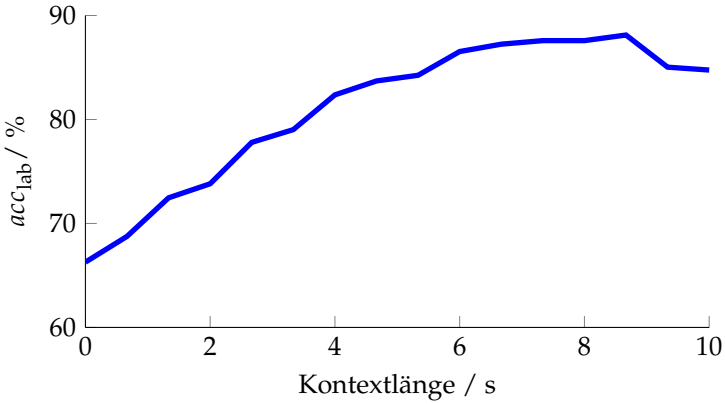


Abbildung 5.18. Genauigkeit der prädizierten Zustände über der Kontextlänge für den Dynamik-Deskriptor.

Zusätzlich zur Genauigkeit der prädizierten Zustände acc_{lab} wird hierbei ein weiteres Auswertekriterium betrachtet. Dieses prüft, ob die einzelnen Aktionen während der Zeiträume ihres Auftretens überwiegend korrekt erkannt werden. Dabei wird für jeden Zeitabschnitt, in dem in den annotierten Daten eine Aktion auftritt, derjenige geschätzte Zustand ermittelt, der am häufigsten während dieses Zeitintervalls auftritt. Dieser wird dann als prädizierte Klasse für diesen Abschnitt betrachtet und damit die Klassengenauigkeit acc_{cl} berechnet. Der Grund für die zusätzliche Betrachtung dieses Wertes ist, dass Übergänge zwischen Aktionen nur sehr ungenau definiert und annotiert werden können. Somit gehen Ungenauigkeiten in der Bestimmung der Start- und Endzeitpunkte der Aktionsintervalle weniger stark in die Auswertung ein.

Wie zuvor liefern auch hier der SG- und MBH-Deskriptor einzeln die besten Ergebnisse. Die Fusion des SG- und MBH-Deskriptors bringt eine Verbesserung im Vergleich zu den beiden Einzelergebnissen. Durch Hinzunahme weiterer, komplementärer Informationen lässt sich das Ergebnis i. Allg. weiter steigern. Die beste Kombination ist auch hier die Fusion von SG-, MBH-, LBP- und Dynamik-Deskriptoren, welche eine mittlere Zustandsgenauigkeit von 89,97 % und Klassengenauigkeit von 94,93 % ergibt.

Tabelle 5.7. Genauigkeit der sequenziellen Erkennung für verschiedene Deskriptoren und deren Kombinationen. Es sind jeweils die mittlere Zustands- und Klassengenauigkeit acc_{lab} / acc_{cl} in Prozent dargestellt.

Deskriptoren	acc_{lab} / %	acc_{cl} / %
SG	83,35	87,96
MBH	83,43	90,23
Dyn	81,22	89,05
LBP	75,44	79
SG+MBH	87,12	93,76
SG+MBH+Dyn	89,26	92,57
SG+MBH+LBP+Dyn	89,97	94,93

Sequenzielle Modellierung von Posenverläufen

Nun wird das Sequenzmodell auf modellbasierte Posenverläufe angewandt. Dazu werden zunächst *Motion Capture*-Daten der HumanEva- und CMU-Datensätze betrachtet. Anschließend werden die Ergebnisse des Körper-Trackings aus Kapitel 3 verwendet.

Für die *Motion Capture*-Daten werden sechs verschiedene Szenarien betrachtet, welche in Tabelle B.4 beschrieben werden. Zwei Szenarien bestehen aus *Motion Capture*-Daten des HumanEva-Datensatzes mit vier bzw. fünf Aktionsklassen. Für die CMU-Daten werden vier Szenarien verwendet mit fünf bis acht Aktionen. Die CMU-Szenarien bestehen aus Fortbewegungsarten und sportlichen Aktivitäten und enthalten teilweise Aktionen, die ähnlich zueinander sind. Dazu zählt die Unterscheidung zwischen normalem Gehen und dem Gehen auf unebenem Grund.

Für die *Motion Capturing*-Daten werden in Tabelle 5.8 verschiedene Methoden zur Bildung der sequenziellen Deskriptoren anhand zweier Szenarien miteinander verglichen. Die sequenziellen Deskriptoren werden aus 19 Markern gebildet, welche die Verläufe verschiedener Gelenkpositionen wiedergeben. Dabei werden zunächst unnormierte Marker betrachtet, d. h. die Verläufe der 19 Marker werden ohne weitere Verarbeitung zusammengefügt, um den Sequenz-Deskriptor zu erhalten.

Tabelle 5.8. Vergleich verschiedener Methoden zur Bildung der sequenziellen Deskriptoren für *Motion Capture*-Daten anhand der Szenarien HE-1 und CMU-1. Es sind jeweils die Zustands- und Klassengenauigkeit acc_{lab} / acc_{cl} in Prozent dargestellt.

Deskriptor	HE-1	CMU-1
Marker unnormiert	81,89 / 91,67	49,36 / 50
Marker normiert	92,79 / 91,67	83,57 / 93,75
Relative Marker, normiert	83,59 / 87,50	91,15 / 75
RelF	89,94 / 87,50	89,42 / 87,50

Bei den normierten Markern werden die einzelnen Verläufe zunächst wie in Abschnitt 5.5.4 erläutert normiert. Relative Markerpositionen werden ermittelt, indem die Gelenkpositionen auf das Körperzentrum bezogen werden. Schließlich werden die *Relational Features* betrachtet, in deren Berechnung dieselben 19 Marker einfließen. Insgesamt schneiden hier die normierten Marker am besten ab. Die unnormierten Marker sind besonders für den CMU-Datensatz deutlich schlechter als die anderen Merkmale. Die Verwendung relativer Marker stellt sich im Falle des HE-1 Szenarios nicht als nützlich heraus, sie schneiden schlechter as die absoluten Positionen ab. Das Gleiche gilt für die Klassengenauigkeit des CMU-1 Szenarios. Betrachtet man hierfür jedoch die Zustandsgenauigkeit, ergeben die relativen Marker die besten Ergebnisse. Die *Relational Features* liefern gute Ergebnisse, liegen jedoch hinter den normierten Markern zurück.

Tabelle 5.9 zeigt Ergebnisse der sequenziellen Erkennung basierend auf den durch *Motion Capturing* gewonnenen Posenverläufen für verschiedene Szenarien. Es werden die absoluten, normierten Markerpositionen betrachtet. Die CMU-Szenarien bestehen aus sich teilweise ähnelnden Aktionen verschiedener Geh-Arten und Sport. Bei Szenarien mit mehr Klassen verschlechtern sich die Ergebnisse im Allgemeinen. Dies ist allerdings teilweise dadurch begründet, dass bei einigen der betrachteten Aktionen nur wenige Trainings-Instanzen vorhanden sind.

Tabelle 5.9. Ergebnisse sequenzieller Erkennung basierend auf *Motion Capture*-Daten für verschiedene Szenarien.

Szenario	Anzahl Klassen	acc_{lab} / %	acc_{cl} / %
HE-1	4	92,79	91,67
HE-2	5	91,22	86,30
CMU-1	5	83,57	93,75
CMU-2	6	97,62	95,24
CMU-3	7	92,94	83,33
CMU-4	8	61,84	76,92

Zuletzt soll die vorgestellte Methode auf die Tracking-Ergebnisse aus Abschnitt 3.4 angewandt werden. Hierzu werden die beiden Sequenzen aus dem HumanEva-II Datensatz verwendet, jeweils eine zum Testen und die andere zum Lernen des Modells. Es werden die Ergebnisse des evolutionären Posentrackings verwendet. Als Sequenz-Deskriptoren zur Bildung der CRF-Merkmale werden drei Varianten betrachtet. Zunächst werden die Verläufe der Gelenkwinkel, die aus dem Körper-Tracking resultieren, direkt verwendet. Als Zweites werden aus den Winkel-Verläufen die Positionen der virtuellen Marker berechnet, die auch zur Berechnung der Fehler der geschätzten Posen in Abschnitt 3.4.2 verwendet werden. Hierbei werden die absoluten Positionen der virtuellen Marker betrachtet. Schließlich werden basierend auf den virtuellen Markern die *Relational Features* bestimmt. Die Ergebnisse sind in Tabelle 5.10 aufgetragen. Dabei wird nur die Genauigkeit der Zustände betrachtet, da die Klassen in allen Fällen korrekt zugeordnet werden. Hier lassen sich mittels der *Relational Features* die besten Ergebnisse erzielen. Dies steht im Gegensatz zu den Versuchen mit den *Motion Capture*-Daten. Der Grund dafür ist darin zu vermuten, dass die *Relational Features* robuster gegenüber Ungenauigkeiten der geschätzten Posen sind, da sie keine absoluten Werte, sondern qualitative Relationen zwischen Körperteilen codieren.

Tabelle 5.10. Sequenzielle Erkennung der Ergebnisse des markerlosen Körper-Trackings. Es ist jeweils die Genauigkeit der prädizierten Zustände acc_{lab} in Prozent für die Winkelverläufe, die Verläufe der virtuellen Marker und die *Relational Features* (RelF) dargestellt.

Deskriptor	Winkel / %	Marker / %	RelF / %
Sequenz 1	93,60	82,13	94,20
Sequenz 2	85,20	73,73	99,87

Diskussion

Die in dieser Arbeit vorgestellte Methode zur Bestimmung sequenzieller Deskriptoren der Merkmalstrajektorien ist für die Aktionserkennung und -segmentierung im Rahmen des verwendeten Sequenzmodells geeignet. Die durch das verwendete CRF-Modell ermöglichte Modellierung von zeitlichem Kontext resultiert in starken Verbesserungen der Ausdrucksfähigkeit des Modells. Hierbei hat sich eine Kontextlänge von ca. zehn Sekunden als ausreichend herausgestellt. Wie bei der BoW-Modellierung ist auch hier die Fusion von Merkmalen besser als die Betrachtung einzelner Deskriptoren. Auf die Verwendung von Raster-Deskriptoren wurde hier verzichtet, da der Fokus darauf gelegt wurde, die Eignung der Sequenz-Deskriptoren und der gewählten Modellform insbesondere im Hinblick auf die Modellierung langfristiger Zusammenhänge zwischen Merkmalen zu prüfen. Raster-Deskriptoren können jedoch ohne Weiteres in das Modell integriert werden, wodurch weitere Verbesserungen zu erwarten sind. Die Modellierung von Posenverläufen gelingt mit dem verwendeten Sequenzmodell auf einfache Weise. Da hierbei bereits sequenzielle Deskriptoren vorliegen, sind nur wenige Vorverarbeitungsschritte zur Bildung der Deskriptoren vonnöten. Zur Modellierung sich ähnelnder Aktivitäten auf Basis von *Motion Capture*-Daten wurden bessere Ergebnisse mit direkter Verwendung der Markerverläufe als durch die *Relational Features* erzielt. Im Falle der Ergebnisse des markerlosen Körper-Trackings sind die *Relational Features* dagegen vielversprechend. Allerdings muss betont werden, dass hier lediglich vorläufige Versuche zur Beurteilung der grundsätzlichen Eignung der Tracking-Ergebnisse unternommen wurden. Für weiterführende Betrachtungen sind ausführlichere Untersuchungen mit größeren Szenarien und mehr Aktionen notwendig.

6. Zusammenfassung und Ausblick

Im Folgenden werden die behandelten Themen zusammengefasst und es wird dabei auf die Beiträge dieser Arbeit eingegangen. Außerdem erfolgt ein Ausblick auf weiterführende Betrachtungen.

In Kapitel 3 wurde eine Methode zum markerlosen, dreidimensionalen Körper-Tracking mit einem evolutionären Algorithmus vorgestellt. Das entwickelte Verfahren weist eine ähnliche Vorgehensweise wie das *Interacting Simulated Annealing* (ISA)-Partikelfilter auf und verwendet ebenso *Simulated Annealing* zur globalen Optimierung. Die wesentliche Stärke des evolutionären Posen-Trackings ist das Zusammenspiel von Rekombination und Mutation zur Erzeugung von Variation in der Population. Hierdurch wird eine intelligente Durchsuchung des Zustandsraumes erreicht und somit ein erfolgreiches Tracking mit halb so vielen Individuen ermöglicht, als es das ISA benötigt. Um Mehrdeutigkeiten insbesondere im Falle weniger Kameraperspektiven zu reduzieren, wurde anhand von Geh-Bewegungen ein erweitertes Dynamikmodell untersucht, welches im Falle von Überlappungen von Körperteilen deren vergangenen Verlauf zur Prädiktion heranzieht. Mit dieser Vorgehensweise gelingt es, Zuordnungsprobleme im Falle sich überlappender Körperteile zu reduzieren.

Weiterführende Themen im Bereich des markerlosen Körper-Trackings könnten sich mit der Entwicklung von Dynamikmodellen befassen, die sich auf weitere Bewegungsarten anwenden lassen. Hierzu kann das in dieser Arbeit für Geh-Bewegungen angewandte, erweiterte Dynamikmodell auf weitere Bewegungsarten verallgemeinert werden. Eine interessante Thematik ist auch die Kopplung des Posen-Trackings mit der Aktivitätserkennung wie in [108]. Dabei werden niederdimensionale Dynamikmodelle für verschiedene Aktivitäten gelernt und eine Akti-

vitätserkennung durchgeführt um geeignete Dynamikmodelle für das Tracking zu wählen. Weiterhin kann das Posen-Tracking durch aussagekräftigere Gewichtungsfunktionen erweitert werden, beispielsweise durch Detektion bestimmter Körperteile wie in [88].

Alternativ zum Körper-Tracking wurde in Kapitel 4 ein merkmalsbasierter Ansatz der Bewegungserfassung verfolgt. Hierzu wurde eine Methode zur Gewinnung von Merkmalstrajektorien vorgestellt, welche den dynamischen Verlauf von für die auftretenden Bewegungen charakteristischen Aktionspunkten repräsentieren. Diese Punkte werden an Orten detektiert, welche markante und sich bewegende Bildmerkmale enthalten. Dazu wurde ein Salienzmaß entwickelt, welches Bild- und Bewegungsinformation kombiniert. Damit werden bereits bei der Initialisierung besonders interessante Merkmale selektiert und nur diese verfolgt. Die Versuche in Abschnitt 5.6.1 zeigten, dass diese Initialisierungsmethode sich positiv auf die Qualität der Trajektorien und damit auf deren Eignung zur Aktivitätserkennung auswirkt. Da die Selektion der Aktionspunkte hier bereits bei der Initialisierung erfolgt und nicht durch nachträgliche Filterung der Trajektorien, wird der Aufwand des Trackings reduziert. Das Tracking erfolgt durch die Verwendung von optischem Fluss und *Mean Shift*-Tracking, bei dem die Merkmale durch *Local Binary Patterns* modelliert werden. Die im Rahmen dieser Arbeit entwickelte Methode verhindert das schnelle Abdriften der Aktionspunkte und ermöglicht so das Verfolgen der Punkte über einen längeren Zeitraum. Aus den erzielten Ergebnissen wurde die Schlussfolgerung gezogen, dass einerseits die Menge extrahierter Trajektorien ein sehr wichtiges Kriterium für die Aktionserkennung ist. Andererseits ist auch die Qualität der Trajektorien von großer Bedeutung, weshalb eine gute Initialisierung der Aktionspunkte und ein robustes Tracking wichtig sind. Die hier gewonnenen Langzeit-Trajektorien sind besser in der Lage, komplexe Ereignisse in Videos zu modellieren, als Trajektorien kurzer Dauer.

Die Trajektorien werden durch verschiedene Deskriptoren dargestellt, welche neben dem Positionsverlauf Informationen über Textur und Bewegung in der Umgebung der Trajektorien enthalten. Neben den etablierten Histogramm-Deskriptoren HOG, HOF und MBH wurden außerdem Summen-Deskriptoren vorgeschlagen. Diese sind vom Auf-

bau des SURF-Deskriptors inspiriert und können mit weniger Aufwand berechnet werden. Analog zu den Histogramm-Deskriptoren wurden *Sum of Gradients* (SG), *Sum of Optical Flow* (SOF) und *Motion Boundary Sum* (MBS)-Deskriptoren formuliert, welche Informationen über Bildgradienten bzw. optischen Fluss repräsentieren. Außerdem werden die zum Tracking ohnehin benötigten *Local Binary Patterns* zur Bildung eines weiteren Textur-Deskriptors herangezogen.

Die betrachteten Deskriptoren wurden in Kapitel 5 zur Aktivitätserkennung eingesetzt. Dazu wurden zwei Modellierungsarten betrachtet. Zunächst wurde das auch in verwandten Arbeiten am meisten verwendete „*Bag of Words*“ (BoW)-Modell eingesetzt. Dabei erfolgt eine Fusion der verschiedenen Deskriptoren mit einer Mehrkanal-*Support Vector Machine*. Die Ergebnisse in Abschnitt 5.6 verdeutlichen, dass zur erfolgreichen Aktivitätserkennung die Kombination verschiedener Merkmale wichtig ist. Wesentlich hierbei ist die Verwendung komplementärer Informationsarten. Eine gewisse Redundanz beispielsweise durch ähnliche Merkmale kann durchaus sinnvoll sein, den größten Mehrwert bringen jedoch unterschiedliche Informationskanäle. Beim Vergleich der Summen- und Histogramm-Deskriptoren hat sich keine der beiden Deskriptorarten als durchgehend überlegen erwiesen. Im Falle des *Activities of Daily Living* (ADL)-Datensatzes lieferte der SG-Deskriptor einzeln betrachtet die besten Ergebnisse, während zur Darstellung des optischen Flusses die Histogramm-Deskriptoren besser abschnitten. Daher ist es sinnvoll – auch um den Aufwand der Merkmalsextraktion zu reduzieren –, für einen speziellen Einsatzfall die besten Deskriptoren zu ermitteln und auszuwählen.

Für den ADL-Datensatz, welcher aus komplexen Aktivitäten besteht, werden durch die hier vorgestellte Methode bessere Ergebnisse erzielt als von den meisten verwandten Arbeiten, welche kurze, spärliche Trajektorien extrahieren oder schwache Kriterien zur Merkmalsdetektion einsetzen (vgl. Tabelle 5.1 und 5.6). Lediglich die salienten Trajektorien [110] erzielen eine höhere Klassifikationsgenauigkeit, sind jedoch deutlich aufwändiger zu extrahieren (vgl. Tabelle 4.4). Im Falle des *Hollywood Human Actions* (HOHA)-Datensatzes, welcher kurze, einfache Aktionen in unkontrollierten, realistischen Umgebungen enthält, sind die hier erzielten Ergebnisse besser als bei anderen, Trajektorien-basierten Ansät-

zen und vergleichbar mit den STIP-Deskriptoren in [58] (vgl. Tabelle 5.2). Eine höhere Genauigkeit wird von Verfahren erreicht, die mehr Zusatzinformationen heranziehen, beispielsweise über den Szenenkontext oder Kamerabewegungen.

Neben dem BoW-Modell wurde ein sequenzielles Aktivitätsmodell mit einem *Conditional Random Field* (CRF) betrachtet. Damit wird neben der Erkennung auch eine Segmentierung von Aktivitäten ermöglicht. Diese ist für reale Anwendungen eine wichtige Thematik, wird allerdings häufig ignoriert. Da es sich bei CRFs um diskriminative Modelle handelt, können auch komplexe Zusammenhänge der Beobachtungen verschiedener Zeitpunkte modelliert werden. Die Untersuchung der Modellierung von zeitlichem Kontext wurde in der Literatur bereits für *Motion Capture*-Daten oder Sequenzen von Bildsilhouetten eingesetzt. Im Zusammenhang mit Trajektorien lokaler Merkmale wurden diese Aspekte nach Wissen des Autors in verwandten Arbeiten noch nicht behandelt. Zur Darstellung der hier vorliegenden Trajektorien kann nicht das übliche Vorgehen wie bei der BoW-Modellierung angewandt werden, bei der ein globaler Deskriptor für die gesamte Beobachtungssequenz ermittelt wird. Stattdessen wird eine zeitliche Abfolge von Deskriptoren benötigt. Für deren Bestimmung wurde die bei der BoW-Modellierung angewandte Vorgehensweise erweitert und die zeitlichen Verläufe des Auftretens der Merkmalswörter ermittelt. Die in Abschnitt 5.6.2 durchgeführten Experimente haben gezeigt, dass die Betrachtung eines längeren zeitlichen Kontextes wichtig für die Modellierung komplexer Aktivitäten ist.

Ein mögliches Thema für künftige Arbeiten im Bereich der merkmalsbasierten Aktivitätserkennung ist zum einen die Umsetzung einer automatisierten Deskriptor-Selektion. Dies ist sowohl für die BoW- als auch die sequenzielle Modellierung relevant. Eine solche Methode kann sowohl für die Selektion von Merkmalsarten, Rastern als auch Kontextmerkmalen verwendet werden. In Bezug auf das Sequenzmodell können außerdem Methoden der Merkmals*induktion* untersucht werden, um automatisch Merkmalsfunktionen zu erzeugen, welche in das CRF-Modell einfließen. Für die gemeinsame Aktivitätserkennung und -segmentierung wurden in dieser Arbeit bereits Wahrscheinlichkeiten der Zustandsübergänge aufeinanderfolgender Aktionen betrachtet. Für komplexe Aktivitäten oder Szenen, die sich aus mehreren Teil-Aktionen

zusammensetzen, ist die Modellierung von längerfristigen, logischen Zusammenhängen zwischen den Aktionen eine Thematik, die künftig betrachtet werden sollte. Ein möglicher Ansatz hierzu ist die Verwendung hierarchischer graphischer Modelle. In einem solchen Modell kann auch weiteres Zusatzwissen, beispielsweise über den Kontext der betrachteten Szene oder Interaktionen mit Objekten, integriert werden.

Für das Sequenzmodell wurden außerdem Versuche zur Modellierung von Posenverläufen durchgeführt, und zwar für *Motion Capture*-Daten und Ergebnisse des markerlosen Körper-Tracking-Algorithmus aus Kapitel 3. Im Falle der *Motion Capture*-Daten hat sich die direkte Verwendung der Markerpositionen zur Bildung der Sequenz-Deskriptoren als geeignet erweisen, um ähnliche Bewegungsarten voneinander zu unterscheiden. Für die Posenverläufe des markerlosen Körper-Trackings sind relationale Merkmale vielversprechend, welche logische Zusammenhänge zwischen Körperteilen codieren und robuster gegenüber Ungenauigkeiten der Posenschätzung sind. Für weiterführende Betrachtungen sind hierfür ausführlichere Experimente mit größeren Datensätzen und mehr Aktivitäten notwendig, was über den Rahmen dieser Arbeit hinausgeht. Ein interessanter Aspekt hierbei ist es, Aktivitätsmodelle aus *Motion Capture*- oder synthetischen Daten zu lernen und diese für die Analyse markerloser Verfahren einzusetzen.

A. Herleitungen

A.1. Iterationsvorschrift beim Mean Shift-Tracking

Die Gradientenschätzung des Bhattacharyya-Koeffizienten aus Gleichung (4.40) lautet:

$$\begin{aligned}\widehat{\nabla}\rho(\mathbf{x}) &= \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}} \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{x})\hat{q}_u} \\ &= \frac{1}{2} \sum_{u=1}^{m} \frac{1}{\sqrt{\hat{p}_u(\mathbf{x})\hat{q}_u}} \cdot \hat{q}_u \cdot \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}} \hat{p}_u(\mathbf{x}). \end{aligned} \tag{A.1}$$

Zu dessen Berechnung muss das Kandidatenmodell nach $\mathbf{x}$ abgeleitet werden

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}} \hat{p}_u(\mathbf{x}) &= \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}} \left[C_h \sum_{i=1}^{n_h} k\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \delta(b(\mathbf{x}_i) - u) \right] \\ &= \frac{2C_h}{h^2} \sum_{i=1}^{n_h} (\mathbf{x} - \mathbf{x}_i) k'\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \delta(b(\mathbf{x}_i) - u). \end{aligned} \tag{A.2}$$

Mit

$$g\left(\|\mathbf{x}\|^2\right) = -k'\left(\|\mathbf{x}\|^2\right) \tag{A.3}$$

wird der Gradient des Bhattacharyya-Koeffizienten zu

$$\begin{aligned}
\widehat{\nabla}\rho(\mathbf{x}) &= \frac{C_h}{h^2}\sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{x})}}\cdot\sum_{i=1}^{n_h}(\mathbf{x}_i-\mathbf{x})g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)\delta(b(\mathbf{x}_i)-u) \\
&= \frac{C_h}{h^2}\sum_{i=1}^{n_h}(\mathbf{x}_i-\mathbf{x})g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)\underbrace{\sum_{u=1}^{m}\delta(b(\mathbf{x}_i)-u)\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{x})}}}_{=:w_i} \\
&= \frac{C_h}{h^2}\left(\sum_{i=1}^{n_h}\mathbf{x}_i w_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)-\mathbf{x}\sum_{i=1}^{n_h}\mathbf{w}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)\right) \\
&= \frac{C_h}{h^2}\sum_{i=1}^{n_h}\mathbf{w}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)\underbrace{\left[\frac{\sum_{i=1}^{n_h}\mathbf{x}_i w_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h}\mathbf{w}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}-\mathbf{x}\right]}_{=\mathbf{m}_G(\mathbf{x})}.
\end{aligned} \tag{A.4}$$

Aus Gleichung (A.4) wird ersichtlich, dass der *Mean Shift*-Vektor $\mathbf{m}_G(\mathbf{x})$ eine Gradientenschätzung des Bhattacharyya-Koeffizienten darstellt. Es wird iterativ bis zum Maximum von $\widehat{\nabla}\rho(\mathbf{x})$ fortgeschritten entsprechend

$$\hat{\mathbf{x}}_{j+1} = \frac{\sum_{i=1}^{n_h}\mathbf{x}_i w_{ij} g\left(\left\|\frac{\hat{\mathbf{x}}_j-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h}\mathbf{w}_{ij} g\left(\left\|\frac{\hat{\mathbf{x}}_j-\mathbf{x}_i}{h}\right\|^2\right)}. \tag{A.5}$$

Gleichung (A.5) entspricht der *Mean Shift*-Iteration nach Gleichung (4.33) mit den zusätzlichen Gewichten w_{ij}, die den Vergleich der beiden Modelle beinhalten:

$$w_{ij} = w_i(\hat{\mathbf{x}}_j) = \sum_{u=1}^{m}\delta(b(\mathbf{x}-i)-u)\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{x}_j)}}. \tag{A.6}$$

A.2. Bestimmen der optimalen Hyperebene bei linearen SVMs

Zur Bestimmung der Hyperebene mit maximaler Trennspanne liegt das folgende Optimierungsproblem mit Nebenbedingungen vor:

$$\begin{aligned} &\|\mathbf{w}\|^2 \rightarrow \min \\ &\text{mit } x_i \cdot (\langle \mathbf{w}, \mathbf{y}_i \rangle + b) \geq 1\,, \quad i = 1, \ldots, N^{\text{train}}\,. \end{aligned} \tag{A.7}$$

$\mathbf{w}$ ist der Normalenvektor und b die Verschiebung der Trennebene. Es seien N^{train} Trainingsmerkale $\mathcal{D}^{\text{train}} = \{(\mathbf{y}_i, x_i),\, i = 1, \ldots, N^{\text{train}}\}$ gegeben. Die Lagrange-Funktion lautet

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N^{\text{train}}} \alpha_i \left(x_i(\langle \mathbf{w}, \mathbf{y}_i \rangle + b) - 1\right)\,,\ \alpha_i \geq 0\,. \tag{A.8}$$

Dabei gibt es für jeden Trainingspunkt einen Lagrange-Multiplikator $\alpha_i \geq 0$, $i = 1, \ldots, N^{\text{train}}$. Nun muss bezüglich $\mathbf{w}$ und b minimiert und bzgl. der α_i maximiert werden [15]. Differenzieren und Nullsetzen von L nach $\mathbf{w}$ und b ergibt

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \mathbf{w} - \sum_{i=1}^{N^{\text{train}}} \alpha_i x_i \mathbf{y}_i = 0 \;\Rightarrow\; \mathbf{w} = \sum_{i=1}^{N^{\text{train}}} \alpha_i x_i \mathbf{y}_i\,, \tag{A.9}$$

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{N^{\text{train}}} \alpha_i x_i = 0\,. \tag{A.10}$$

Setzt man die Lösung für $\mathbf{w}$ aus Gleichung (A.9) wieder in L ein, ergibt sich

$$\begin{aligned} L_{\text{d}}(b, \boldsymbol{\alpha}) = &\frac{1}{2}\| \sum_{i=1}^{N^{\text{train}}} \alpha_i x_i \mathbf{y}_i\|^2 \\ &- \sum_{i=1}^{N^{\text{train}}} \alpha_i \left(x_i \left(\sum_{j=1}^{N^{\text{train}}} \alpha_j x_j \langle \mathbf{y}_j, \mathbf{y}_i \rangle + b \right) - 1 \right)\,. \end{aligned} \tag{A.11}$$

Durch Umformen und Verwenden von (A.10) erhält man

$$
\begin{aligned}
&L_{\mathrm{d}}(\alpha) = \sum_{i=1}^{N^{\text{train}}} \alpha_i - \frac{1}{2} \sum_{i=1}^{N^{\text{train}}} \sum_{j=1}^{N^{\text{train}}} \alpha_i \alpha_j x_i x_j \langle \mathbf{y}_i, \mathbf{y}_j \rangle \text{ mit} \\
&\alpha_i \geq 0\,, \quad i = 1, \ldots, N^{\text{train}}\,, \\
&\sum_{i=1}^{N^{\text{train}}} \alpha_i x_i = 0\,.
\end{aligned} \tag{A.12}
$$

Jetzt liegt ein Optimierungsproblem vor, das nur noch von den Lagrange-Multiplikatoren abhängt. Es wird auch als *duales Problem* bezeichnet [15]. L_{d} muss bezüglich der α_i maximiert werden unter Berücksichtigung der Nebenbedingungen aus Gleichung (A.12). Das Lösen des Optimierungsproblems erfolgt durch numerische Methoden, weiterführende Informationen sind z. B. in [15] zu finden.

Sind die α_i gefunden, kann $\mathbf{w}$ durch Einsetzen in Gleichung (A.9) bestimmt werden. Die Verschiebung b taucht im Optimierungsproblem (A.12) nur noch implizit auf und wird durch dessen Lösung ebenfalls ermittelt.

Setzt man die Lösung für $\mathbf{w}$ in die Gleichung der Entscheidungsfunktion $x^* = \operatorname{sign}(\langle \mathbf{w}, \mathbf{y}^* \rangle + b)$ ein, erhält man

$$
x^* = \operatorname{sign}\left(\sum_{i=1}^{N^{\text{train}}} \alpha_i x_i \langle \mathbf{y}_i, \mathbf{y}^* \rangle + b \right)\,. \tag{A.13}
$$

Sowohl im Optimierungsproblem (A.12) als auch bei der Klassifikation nach Gleichung (A.13) tauchen die Merkmalsvektoren $\mathbf{y}$ nur noch in Form von Skalarprodukten auf. Diese Tatsache stellt sich als sehr nützlich heraus, um SVMs auf den nichtlinearen Fall zu erweitern (siehe Abschnitt 5.2.2).

A.3. Parameterschätzung bei linearen Ketten-CRFs

Für die Parameterschätzung bei CRFs wird die *Maximum Likelihood*-Methode (ML) angewandt. Hier wird von linearen Ketten-CRFs mit *Parameter Tying* ausgegangen. Gegeben sind Trainingsdaten $\mathcal{D}^{\text{train}} = \{\mathbf{x}_m, \mathbf{y}_m\}$, $m = 1, \ldots, N^{\text{train}}$, welche aus N^{train} Zustands- und Beobachtungssequenzen

$$\begin{aligned} \mathbf{x}_m &= \{x_{m,t}, \ldots, x_{m,T_m}\}\,, \\ \mathbf{y}_m &= \{\mathbf{y}_{m,t}, \ldots, \mathbf{y}_{m,T_m}\} \end{aligned} \tag{A.14}$$

bestehen. Dabei ist T_m die Länge der m-ten Sequenz. Aufgrund des *Parameter Tying* können die Knotenanzahlen und damit die Sequenzlängen unterschiedlich sein. Um die Notation zu vereinfachen wird im Folgenden, ohne Verlust der Allgemeingültigkeit, von einer konstanten Knotenanzahl T ausgegangen.

Die ML-Parameter maximieren die bedingte logarithmische Likelihood-Funktion

$$l(\boldsymbol{\lambda}) = \sum_{m=1}^{N^{\text{train}}} \log p(\mathbf{x}_m|\mathbf{y}_m)\,. \tag{A.15}$$

In der Praxis wird statt dem Term (A.15) häufig die *penalisierte* Likelihood-Funktion

$$l_{\text{L}_2}(\boldsymbol{\lambda}) = l(\boldsymbol{\lambda}) - \sum_{k=1}^{K} \frac{{\lambda_k}^2}{2\sigma^2} \tag{A.16}$$

maximiert. Dabei wird eine Regularisierung mit der L2-Norm vorgenommen, um *Overfitting* bezüglich der Trainingsdaten zu vermeiden [93]. Der zusätzliche Term bestraft die euklidische Norm der Gewichte, wodurch vereinzelte sehr große Gewichte verhindert werden. Eine andere Interpretation ist, dass eine Maximum A-posteriori-Schätzung mit gaußverteilten Parametern mit dem Mittelwert $\mathbf{0}$ und der Varianz $\sigma^2\mathbf{I}$ durchgeführt wird. Je größer σ^2 gewählt wird, desto größere Gewichte

werden erlaubt. Die Wahl der Varianz ist abhängig von der Größe des Trainingsdatensatzes; ein typischer Wert ist 10.

Eine andere mögliche Regularisierungsvariante verwendet die L1-Norm

$$l_{\mathrm{L}_1}(\boldsymbol{\lambda}) = l(\boldsymbol{\lambda}) - \alpha \sum_{k=1}^{K} |\lambda_k| \,. \tag{A.17}$$

Im Gegensatz zu (A.16) wird hier Spärlichkeit ermutigt und die meisten Gewichte werden null. Diese Variante kann zur Merkmalsselektion eingesetzt werden. Setzt man das Modell (5.41) in Gleichung (A.16) ein, so erhält man

$$\begin{aligned} l_{\mathrm{L}_2}(\boldsymbol{\lambda}) &= \sum_{m=1}^{N^{\text{train}}} \log \left[\frac{1}{Z(\mathbf{y}_m)} \prod_{t=1}^{T} \exp \left(\sum_{k=1}^{K} \lambda_k f_k(x_{m,t}, x_{m,t-1}, \mathbf{y}_m) \right) \right] \\ &\quad - \sum_{k=1}^{K} \frac{{f_k}^2}{2\sigma^2} \\ &= \sum_{m=1}^{N^{\text{train}}} \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(x_{m,t}, x_{m,t-1}, \mathbf{y}_m) - \sum_{m=1}^{N^{\text{train}}} \log Z(\mathbf{y}_m) \\ &\quad - \sum_{k=1}^{K} \frac{{f_k}^2}{2\sigma^2} \,. \end{aligned} \tag{A.18}$$

Die partiellen Ableitungen nach λ_k lauten

$$\begin{aligned} \frac{\partial l_{\mathrm{L}_2}}{\partial \lambda_k} &= \sum_{m=1}^{N^{\text{train}}} \sum_{t=1}^{T} f_k(x_{m,t}, x_{m,t-1}, \mathbf{y}_m) \\ &\quad - \frac{\partial}{\partial \lambda_k} \sum_{m=1}^{N^{\text{train}}} \log Z(\mathbf{y}_m) - \frac{\lambda_k}{\sigma^2} \,. \end{aligned} \tag{A.19}$$

Die Ableitung der Partitionierungsfunktion einer Sequenz ist

$$\begin{aligned}\frac{\partial}{\partial\lambda_k}\log Z(\mathbf{y}_m) &= \frac{\partial}{\partial\lambda_k}\sum_{\mathbf{x}}\sum_{t=1}^{T}\sum_{k=1}^{K}\lambda_k f_k(x_t, x_{t-1}, \mathbf{y}_m)\\ &= \sum_{\mathbf{x}}\sum_{t=1}^{T} f_k(x_t, x_{t-1}, \mathbf{y}_m)\\ &= \sum_{t=1}^{T}\sum_{x,x'} f_k(x, x', \mathbf{y}_m)p(x, x'|\mathbf{y}_m)\,. \end{aligned} \tag{A.20}$$

Um diesen Term zu berechnen, muss über alle möglichen Zustandssequenzen $\mathbf{x}$ summiert werden. Da dies sehr aufwändig bzw. unmöglich sein kann, wird stattdessen über alle in den Trainingsdaten vorkommenden Transitionen summiert. Damit ergibt sich

$$\begin{aligned}\frac{\partial l_{L_2}}{\partial\lambda_k} = &\sum_{m=1}^{N^{\text{train}}}\sum_{t=1}^{T} f_k(x_{m,t}, x_{m,t-1}, \mathbf{y}_m)\\ &- \sum_{m=1}^{N^{\text{train}}}\sum_{t=1}^{T}\sum_{x,x'} f_k(x, x', \mathbf{y}_m)p(x, x'|\mathbf{y}_m) - \frac{\lambda_k}{\sigma^2}\,. \end{aligned} \tag{A.21}$$

Die Optimierung kann mittels Gradientenabstiegs erfolgen, hierbei sind allerdings viele Iterationen nötig. Methoden, die sich als erfolgreich erwiesen haben, sind Quasi-Newton-Verfahren und konjugierte Gradientenverfahren [93]. Der Aufwand liegt in $\mathcal{O}(TS^2N^{\text{train}}G)$, wobei G die Anzahl der Trainingsiterationen darstellt. Der Aufwand ist also stark durch die Anzahl an Zuständen beeinflusst.

B. Details zu Versuchen der Aktionsanalyse

B.1. Parametrierungen des Merkmals-Trackings

Tabelle B.1. Parameter des Merkmals-Trackings der unterschiedlichen Versuche. Nicht genannte Werte entsprechen denen in Tabelle 4.2

Tracker	Parameter Detektion	Parameter Tracking
①	SURF+PD, $r_{\text{min}}^{\text{init}} = 1000$	OFT, $L_{\text{max}} = 15$
②	SURF+PD, $r_{\text{min}}^{\text{init}} = 1000$	OFT, $L_{\text{max}} = \infty$
③	SURF+PD, $r_{\text{min}}^{\text{init}} = 1000$	OFT-LBP
④	SURF+PD, $r_{\text{min}}^{\text{init}} = 1000$	OFT-LBP, $\rho_{\text{min}} = 0{,}80$
⑤	SURF+PD, $r_{\text{min}}^{\text{init}} = 1000$	KLT
⑥	SURF+PD, $r_{\text{min}}^{\text{init}} = 200$,	OFT-LBP
⑦	GFTT+PD, $r_{\text{min}}^{\text{init}} = 200$,	OFT-LBP
⑧	PD, $r_{\text{min}}^{\text{init}} = 200$,	OFT-LBP

B.2. Berechnung der Summen-Deskriptoren

In Tabelle B.2 werden die Summen-Deskriptoren SG, SOF und MBS, jeweils berechnet für $n_\sigma = 2$ und $n_\sigma = 4$ Zellen (siehe Abschnitt 4.3.4). Zum Vergleich sind die Histogramm-Deskriptoren HOG, HOF und MBH mit $n_\sigma = 2$ abgebildet. Die Auswertung erfolgt hier beispielhaft für Szenario ③ nach dem in Abschnitt 5.6.1 beschriebenen Vorgehen. Die *Codebook*-Größe wird hier von 1000 bis 4000 Wörtern variiert und

die Ergebnisse gemittelt. Es ist zu erkennen, dass die Verwendung von 4 Zellen z. T. deutlich bessere Ergebnisse liefert.

Tabelle B.2. Vergleich der Summen-Deskriptoren für die Berechnung in Bildbereichen, die in $n_\sigma = 2$ bzw. $n_\sigma = 4$ horizontale und vertikale Zellen unterteilt werden.

Deskriptor	$acc\ n_\sigma = 2$ /%	$acc\ n_\sigma = 4$ /%
HOG	78	–
SG	72,17	79,06
LBP	74	–
HOF	79,06	–
SOF	77,61	79,06
MBH	74,33	–
MBS	73,27	73,67

B.3. Annotationen

Tabelle B.3. Annotationen für die Aktionssegmentierung anhand des ADL-Datensatzes. In der ersten Spalte sind die betrachteten Teilaktionen aufgelistet und in Klammern die jeweiligen Aktivitäten derer sie Bestandteil sind. In den weiteren Spalten sind die Zeitintervalle der fünf Personen (P1–P5) aufgelistet mit einer Zeile für jede Wiederholung.

Aktion	P1	P2	P3	P4	P5
Messer holen	1–150	1–105	1-181	1–194	1–160
(Banane schneiden)	–	1–135	1-150	1–215	1–132
	–	1–130	1-157	1–190	1–99
Banane schneiden	151–410	106–272	182–375	195-313	161–294
(Banane schneiden)	1–260	136–279	151–346	216–358	133–271
	1–249	131-245	158–356	191–361	100-222
Kühlschrank öffnen	1–98	1–103	1–127	1–118	1–87
(Wasser trinken)	1–90	1–66	1–106	1–105	1–89
	1–82	1–77	1–116	1–93	1–117
Wasser holen	99-219	104–212	128–227	119–208	88–197
(Wasser trinken)	91–200	67–163	107–208	106–192	90–194
	83–205	78–180	117–226	94–188	118–212
Glas füllen	220–353	213–413	228–351	209–379	198–313
(Wasser trinken)	201–331	164–300	209–313	193–411	195–300
	206–320	181–328	227–326	189–333	213–297
Trinken	354–475	414–504	352–431	380–473	314–362
(Wasser trinken)	332–487	301–471	314–420	412–524	307–357
	321–470	329–407	327–382	334–419	298–364

Tabelle B.4. Szenarien zur Evaluation der sequenziellen Modellierung von Posenverläufen mit *Motion Capture*-Daten. Die ersten beiden Datensätze werden aus dem HumanEva-I-Datensatz [85] gebildet, die restlichen aus der *CMU Graphics Lab Motion Capture Database* [20].

Szenario	Beschreibung
HE-1	Klassen: Gehen, Joggen, Boxen, Gestikulieren. Jeweils zwei Wiederholungen der Personen 1–3. Test mit *Leave One Out* (LOO)-Prinzip.
HE-2	Klassen: Gehen, Joggen, Boxen, Gestikulieren, Werfen. Jeweils zwei Wiederholungen der Personen 1–3. Test mit *Leave One Out* (LOO)-Prinzip.
CMU-1	Klassen: Gehen, Joggen, Kicken, Gehen auf unebenem Grund, Springen. 31 Trainings-, 16 Testinstanzen; Personen 8–11, 13, 16, 35, 36.
CMU-2	Klassen: Gehen, Joggen, Kicken, Gehen auf unebenem Grund, Springen, Tanzen. 45 Trainings-, 21 Testinstanzen; Personen 8–11, 13, 16, 35, 36, 60, 61.
CMU-3	Klassen: Gehen, Joggen, Kicken, Gehen auf unebenem Grund, Springen, Tanzen, Basketball spielen. 50 Trainings-, 24 Testinstanzen; Personen 6, 8–11, 13, 16, 35, 36, 60, 61.
CMU-4	Klassen: Gehen, Joggen, Kicken, Gehen auf unebenem Grund, Springen, Tanzen, Basketball spielen, Boxen. 54 Trainings-, 26 Testinstanzen; Personen 6, 8–11, 13–17, 35, 36, 60, 61.

Literaturverzeichnis

[1] **Agarwal**, **Ankur** und **Triggs**, **Bill**. *Tracking articulated motion using a mixture of autoregressive models*. In: *Computer Vision-ECCV 2004*. Springer, 2004, S. 54–65.

[2] **Ahad**, **Md Atiqur Rahman**, **Tan**, **Joo Kooi**, **Kim**, **Hyoungseop** und **Ishikawa**, **Seiji**. *Motion history image: its variants and applications*. In: *Machine Vision and Applications* 23.2 (2012), S. 255–281.

[3] **Amin**, **Sikandar**, **Andriluka**, **Mykhaylo**, **Rohrbach**, **Marcus** und **Schiele**, **Bernt**. *Multi-view Pictorial Structures for 3D Human Pose Estimation*. In: *British Machine Vision Conference*. Bd. 2. 2013.

[4] **Andriluka**, **Mykhaylo**, **Roth**, **Stefan** und **Schiele**, **Bernt**. *Discriminative appearance models for pictorial structures*. In: *International journal of computer vision* 99.3 (2012), S. 259–280.

[5] **Arulampalam**, **M Sanjeev**, **Maskell**, **Simon**, **Gordon**, **Neil** und **Clapp**, **Tim**. *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*. In: *Signal Processing, IEEE Transactions on* 50.2 (2002), S. 174–188.

[6] **Barber**, **David**. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[7] **Bay**, **Herbert**, **Tuytelaars**, **Tinne** und **Van Gool**, **Luc**. *Surf: Speeded up robust features*. In: *Computer Vision–ECCV 2006*. Springer, 2006, S. 404–417.

[8] **Bishop**, **Christopher M** et al. *Pattern recognition and machine learning*. Bd. 1. springer New York, 2006.

[9] **Blank**, **Moshe**, **Gorelick**, **Lena**, **Shechtman**, **Eli**, **Irani**, **Michal** und **Basri**, **Ronen**. *Actions as space-time shapes*. In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Bd. 2. IEEE. 2005, S. 1395–1402.

[10] **Bo**, **Liefeng** und **Sminchisescu**, **Cristian**. *Structured output-associative regression*. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, S. 2403–2410.

[11] **Bobick**, **Aaron F.** und **Davis**, **James W.** *The recognition of human movement using temporal templates*. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.3 (2001), S. 257–267.

[12] **Bradski**, **G.** In: *Dr. Dobb's Journal of Software Tools* (2000).

[13] **Brubaker**, **Marcus A** und **Fleet**, **David J**. *The kneed walker for human pose tracking*. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, S. 1–8.

[14] **Brubaker**, **Marcus A**, **Fleet**, **David J** und **Hertzmann**, **Aaron**. *Physics-based person tracking using the anthropomorphic walker*. In: *International Journal of Computer Vision* 87.1-2 (2010), S. 140–155.

[15] **Burges**, **Christopher JC**. *A tutorial on support vector machines for pattern recognition*. In: *Data mining and knowledge discovery* 2.2 (1998), S. 121–167.

[16] **Canny**, **John**. *A computational approach to edge detection*. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 6 (1986), S. 679–698.

[17] **Chang**, **Chih-Chung** und **Lin**, **Chih-Jen**. *LIBSVM: A library for support vector machines*. In: *ACM Transactions on Intelligent Systems and Technology* 2.6 (2011), S. 1–27.

[18] **Cheng**, **Y.** *Mean shift, Mode seeking, and Clustering*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (1995), S. 790–799.

[19] **Cheung**, **Warren** und **Hamarneh**, **Ghassan**. *N-sift: N-dimensional scale invariant feature transform for matching medical images*. In: *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*. IEEE. 2007, S. 720–723.

[20] *CMU Graphics Lab Motion Capture Database*. http://mocap.cs.cmu.edu/.

[21] **Comaniciu**, **D.** und **Meer**, **P.** *Mean Shift Analysis and Applications*. In: *The Proceedings of the Seventh IEEE International Conference on Computer Vision* 2 (1999), S. 1197–1203.

[22] **Comaniciu, D.** und **Meer, P.** *Mean shift: A Robust Approach Toward Feature Space Analysis*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), S. 603–619.

[23] **Comaniciu, D., Ramesh, V.** und **Meer, P.** *Real-Time Tracking of Non-Rigid Objects Using Mean Shift*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 2 (2000), S. 142–149.

[24] **Comaniciu, D., Ramesh, V.** und **Meer, P.** *Kernel-Based Object Tracking*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2003), S. 564–575.

[25] **Dalal, Navneet** und **Triggs, Bill**. *Histograms of oriented gradients for human detection*. In: 1 (2005), S. 886–893.

[26] **Dalal, Navneet, Triggs, Bill** und **Schmid, Cordelia**. *Human detection using oriented histograms of flow and appearance*. In: (2006), S. 428–441.

[27] **Deutscher, J.** und **Reid, I.** *Articulated body motion capture by stochastic search*. In: *International Journal of Computer Vision* 61.2 (2005), S. 185–205.

[28] **Deutscher, Jonathan, Blake, Andrew** und **Reid, Ian**. *Articulated body motion capture by annealed particle filtering*. In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. Bd. 2. IEEE. 2000, S. 126–133.

[29] **Dollár, P., Rabaud, V., Cottrell, G.** und **Belongie, S.** *Behavior Recognition via Sparse Spatio-Temporal Features*. In: (2006), S. 65–72.

[30] **Doucet, Arnaud, Godsill, Simon** und **Andrieu, Christophe**. *On sequential Monte Carlo sampling methods for Bayesian filtering*. In: *Statistics and computing* 10.3 (2000), S. 197–208.

[31] **Dougherty, Geoff**. *Pattern Recognition and Classification: An Introduction*. Springer-Verlag New York, 2012.

[32] **Efros, Alexei A, Berg, Alexander C, Mori, Greg** und **Malik, Jitendra**. *Recognizing action at a distance*. In: *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE. 2003, S. 726–733.

[33] **Farnebäck**, **Gunnar**. *Two-frame motion estimation based on polynomial expansion*. In: *Image Analysis*. Springer, 2003, S. 363–370.

[34] **Filipovych**, **Roman** und **Ribeiro**, **Eraldo**. *Learning human motion models from unsegmented videos*. In: (2008), S. 1–7.

[35] **Fleet**, **David** und **Weiss**, **Yair**. *Optical flow estimation*. In: *Handbook of Mathematical Models in Computer Vision*. Springer, 2006, S. 237–257.

[36] **Fukunaga**, **Keinosuke** und **Hostetler**, **Larry**. *The estimation of the gradient of a density function, with applications in pattern recognition*. In: *Information Theory, IEEE Transactions on* 21.1 (1975), S. 32–40.

[37] **Gall**, **Juergen**, **Rosenhahn**, **Bodo**, **Brox**, **Thomas** und **Seidel**, **Hans-Peter**. *Optimization and filtering for human motion capture*. In: *International journal of computer vision* 87.1-2 (2010), S. 75–92.

[38] **Gall**, **J.**, **Potthoff**, **J.**, **Schnörr**, **C.**, **Rosenhahn**, **B.** und **Seidel**, **H.P.** *Interacting and annealing particle filters: Mathematics and a recipe for applications*. In: *Journal of Mathematical Imaging and Vision* 28.1 (2007), S. 1–18.

[39] **Ghahramani**, **Zoubin**. *Unsupervised learning*. In: *Advanced Lectures on Machine Learning*. Springer, 2004, S. 72–112.

[40] **Goldberg**, **D.E.** *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional, 1989.

[41] **Gorelick**, **Lena**, **Blank**, **Moshe**, **Shechtman**, **Eli**, **Irani**, **Michal** und **Basri**, **Ronen**. *Actions as Space-Time Shapes*. In: *Transactions on Pattern Analysis and Machine Intelligence* 29.12 (Dez. 2007), S. 2247–2253.

[42] **Harris**, **Chris** und **Stephens**, **Mike**. *A combined corner and edge detector.* In: *Alvey vision conference*. Bd. 15. Manchester, UK. 1988, S. 50.

[43] **Heikkilä**, **M.**, **Pietikäinen**, **M.** und **Schmid**, **C.** *Description of interest regions with local binary patterns*. In: *Pattern recognition* 42.3 (2009), S. 425–436.

[44] *Internetseite des HumanEva-I und HumanEva-II Datensatzes*. http://humaneva.is.tue.mpg.de/. 2016.

[45] **Jähne**, **Bernd**. *Digitale Bildverarbeitung*. Springer-Verlag, 2013.

[46] **Jain**, **Mihir**, **Jégou**, **Hervé**, **Bouthemy**, **Patrick** et al. *Better exploiting motion for better action recognition*. In: (2013).

[47] **Jargalsaikhan**, **Iveel**, **Little**, **Suzanne**, **Direkoglu**, **Cem** und **O'Connor**, **Noel E**. *Action recognition based on sparse motion trajectories*. In: (2013).

[48] **Jhuang**, **Hueihan**, **Gall**, **Juergen**, **Zuffi**, **Silvia**, **Schmid**, **Cordelia** und **Black**, **Michael J**. *Towards understanding action recognition*. In: *International Journal of Computer Vision* 101.3 (2013), S. 437–458.

[49] **John**, **Vijay**, **Trucco**, **Emanuele** und **Ivekovic**, **Spela**. *Markerless human articulated tracking using hierarchical particle swarm optimisation*. In: *Image and Vision Computing* 28.11 (2010), S. 1530–1547.

[50] **Johnson**, **Sam** und **Everingham**, **Mark**. *Learning effective human pose estimation from inaccurate annotation*. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, S. 1465–1472.

[51] **Kiencke**, **Uwe**, **Schwarz**, **Michael** und **Weickert**, **Thomas**. *Signalverarbeitung: Zeit-frequenz-analyse und Schätzverfahren*. Oldenbourg Verlag, 2008.

[52] **Kirkpatrick**, **Scott**, **Vecchi**, **MP** et al. *Optimization by simmulated annealing*. In: *science* 220.4598 (1983), S. 671–680.

[53] **Kläser**, **Alexander** und **Marszalek**, **Marcinw**. *A spatio-temporal descriptor based on 3d-gradients*. In: (2008).

[54] **Kuehne**, **Hildegard**, **Gehrig**, **Dirk**, **Schultz**, **Tanja** und **Stiefelhagen**, **Rainer**. *On-line Action Recognition from Sparse Feature Flow.* In: *VISAPP (1)*. 2012, S. 634–639.

[55] **Laptev**, **Ivan**. *On space-time interest points*. In: *International Journal of Computer Vision* 64.2-3 (2005), S. 107–123.

[56] **Laptev**, **Ivan** und **Lindeberg**, **Tony**. *Space-time Interest Points*. In: *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)* (2003).

[57] **Laptev**, **Ivan** und **Lindeberg**, **Tony**. *Local descriptors for spatio-temporal recognition*. In: (2006), S. 91–103.

[58] **Laptev**, **Ivan**, **Marszalek**, **Marcin**, **Schmid**, **Cordelia** und **Rozenfeld**, **Benjamin**. *Learning realistic human actions from movies*. In: (2008), S. 1–8.

[59] **Lowe**, **David G**. *Object recognition from local scale-invariant features*. In: *The proceedings of the seventh IEEE international conference on Computer vision*. Bd. 2. Ieee. 1999, S. 1150–1157.

[60] **Lucas**, **Bruce D**, **Kanade**, **Takeo** et al. *An iterative image registration technique with an application to stereo vision.* In: *IJCAI*. Bd. 81. 1981, S. 674–679.

[61] **Matikainen**, **P.**, **Hebert**, **M.** und **Sukthankar**, **R.** *Trajectons: Action recognition through the motion analysis of tracked features*. In: *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)* (2009), S. 514–521.

[62] **Messing**, **R.**, **Pal**, **C.** und **Kautz**, **H.** *Activity recognition using the velocity histories of tracked keypoints*. In: *IEEE 12th International Conference on Computer Vision* (2009), S. 104–111.

[63] **Moeslund**, **Thomas B**. *Visual analysis of humans: looking at people*. Springer, 2011.

[64] **Moeslund**, **Thomas B** und **Granum**, **Erik**. *A survey of computer vision-based human motion capture*. In: *Computer Vision and Image Understanding* 81.3 (2001), S. 231–268.

[65] **Moeslund**, **Thomas B**, **Hilton**, **Adrian** und **Krüger**, **Volker**. *A survey of advances in vision-based human motion capture and analysis*. In: *Computer vision and image understanding* 104.2 (2006), S. 90–126.

[66] **Müller**, **Meinard** und **Röder**, **Tido**. *Motion templates for automatic classification and retrieval of motion capture data*. In: *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association. 2006, S. 137–146.

[67] **Ning**, **Jifeng**, **Zhang**, **Lei**, **Zhang**, **David** und **Wu**, **Chengke**. *Robust object tracking using joint color-texture histogram*. In: *International Journal of Pattern Recognition and Artificial Intelligence* 23.07 (2009), S. 1245–1263.

[68] **Noguchi**, **Akitsugu** und **Yanai**, **Keiji**. *A SURF-based spatio-temporal feature for feature-fusion-based action recognition*. In: *Trends and Topics in Computer Vision*. Springer, 2012, S. 153–167.

[69] **Ojala**, **Timo**, **Pietikainen**, **Matti** und **Maenpaa**, **Topi**. *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.7 (2002), S. 971–987.

[70] *Open Source Computer Vision Library*. https://github.com/itseez/opencv. 2015.

[71] **Pearson**, **Karl**. *LIII. On lines and planes of closest fit to systems of points in space*. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), S. 559–572.

[72] **Poppe**, **R.** *A Survey on Vision-Based Human Action Recognition*. In: *Image and Vision Computing* 28 (2010), S. 976–990.

[73] **Poppe**, **Ronalds**. *Vision-based human motion analysis: An overview*. In: *Computer vision and image understanding* 108.1 (2007), S. 4–18.

[74] **Raptis**, **M.** und **Soatto**, **S.** *Tracklet descriptors for action modeling and video analysis*. In: *Computer Vision–ECCV 2010* 577-590 (2010).

[75] **Reddy**, **Kishore K** und **Shah**, **Mubarak**. *Recognizing 50 human action categories of web videos*. In: *Machine Vision and Applications* 24 (2013), S. 971–981.

[76] **Rutenbar**, **Rob A**. *Simulated annealing algorithms: An overview*. In: *Circuits and Devices Magazine, IEEE* 5.1 (1989), S. 19–26.

[77] **Sakoe**, **Hiroaki** und **Chiba**, **Seibi**. *Dynamic programming algorithm optimization for spoken word recognition*. In: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 26.1 (1978), S. 43–49.

[78] **Sand**, **Peter** und **Teller**, **Seth**. *Particle video: Long-range motion estimation using point trajectories*. In: *International Journal of Computer Vision* 80.1 (2008), S. 72–91.

[79] **Schmidt**, **Mark**. *UGM: A Matlab toolbox for probabilistic undirected graphical models*. http://www.cs.ubc.ca/ schmidtm/Software/UGM.html. 2007.

[80] **Scovanner**, **Paul**, **Ali**, **Saad** und **Shah**, **Mubarak**. *A 3-dimensional sift descriptor and its application to action recognition*. In: *Proceedings of the 15th international conference on Multimedia*. ACM. 2007, S. 357–360.

[81] **Shao**, **Ling** und **Mattivi**, **Riccardo**. *Feature detector and descriptor evaluation in human action recognition*. In: (2010), S. 477–484.

[82] **Shi**, **Jianbo** und **Tomasi**, **Carlo**. *Good features to track*. In: *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE. 1994, S. 593–600.

[83] **Sidenbladh**, **H.**, **Black**, **M.** und **Sigal**, **L.** *Implicit probabilistic models of human motion for synthesis and tracking*. In: *Proceedings of the European Conference on Computer Vision* (2002), S. 784–800.

[84] **Sidenbladh**, **Hedvig**, **Black**, **Michael J** und **Fleet**, **David J**. *Stochastic tracking of 3D human figures using 2D image motion*. In: *Computer Vision—ECCV 2000*. Springer, 2000, S. 702–718.

[85] **Sigal**, **L.**, **Balan**, **A.O.** und **Black**, **M.J.** *Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion*. In: *International Journal of Computer Vision* 87.1 (2010), S. 4–27.

[86] **Sigal**, **Leonid**. *Visual Analysis of Humans*. In: Hrsg. von **Moeslund**, **T.B. et al.** Springer, 2011. Kap. Articulated Pose Estimation and Tracking: Introduction, S. 131–137.

[87] **Sigal**, **Leonid** und **Black**, **Michael J**. *Guest editorial: state of the art in image-and video-based human pose and motion estimation*. In: *International Journal of Computer Vision* 87.1 (2010), S. 1–3.

[88] **Sigal**, **L.**, **Isard**, **M.**, **Haussecker**, **H.** und **Black**, **M.J.** *Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation*. In: *International journal of computer vision* (2012), S. 1–34.

[89] **Sminchisescu**, **Cristian**, **Kanaujia**, **Atul** und **Metaxas**, **Dimitris**. *Conditional models for contextual human motion recognition*. In: *Computer Vision and Image Understanding* 104.2 (2006), S. 210–220.

[90] **Sminchisescu**, **Cristian**, **Telea**, **Alexandru** et al. *Human pose estimation from silhouettes. a consistent approach using distance level sets*. In: 10 (2002).

[91] **Sun**, **Ju**, **Mu**, **Yadong**, **Yan**, **Shuicheng** und **Cheong**, **Loong-Fah**. *Activity recognition using dense long-duration trajectories*. In: *Multimedia and Expo (ICME), 2010 IEEE International Conference on*. IEEE. 2010, S. 322–327.

[92] **Sun**, **J.**, **Wu**, **X.**, **Yan**, **S.**, **Cheong**, **L.F.**, **Chua**, **T.S.** und **Li**, **J.** *Hierarchical spatio-temporal context modeling for action recognition*. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2009), S. 2004–2011.

[93] **Sutton**, **Charles** und **McCallum**, **Andrew**. *An introduction to conditional random fields*. In: *Machine Learning* 4.4 (2011), S. 267–373.

[94] **Takala**, **V.** und **Pietikainen**, **M.** *Multi-object tracking using color, texture and motion*. In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. Ieee. 2007, S. 1–7.

[95] **Tomasi**, **Carlo** und **Kanade**, **Takeo**. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[96] **Turaga**, **P.**, **Chellappa**, **R.**, **Subrahmanian**, **V.S.** und **Udrea**, **O.** *Machine Recognition of Human Activities: A Survey*. In: *IEEE Transactions on Circuits and Systems for Video Technology* 18 (2008), S. 1473–1488.

[97] **Wang**, **Heng**, **Ullah**, **Muhammad Muneeb**, **Klaser**, **Alexander**, **Laptev**, **Ivan**, **Schmid**, **Cordelia** et al. *Evaluation of local spatio-temporal features for action recognition*. In: *BMVC 2009-British Machine Vision Conference*. 2009.

[98] **Wang**, **Heng**, **Klaser**, **Alexander**, **Schmid**, **Cordelia** und **Liu**, **Cheng-Lin**. *Action recognition by dense trajectories*. In: (2011), S. 3169–3176.

[99] **Wang**, **Heng**, **Kläser**, **Alexander**, **Schmid**, **Cordelia** und **Liu**, **Cheng-Lin**. *Dense trajectories and motion boundary descriptors for action recognition*. In: *International Journal of Computer Vision* (2013), S. 1–20.

[100] **Weicker**, **K.** *Evolutionäre Algorithmen*. Vieweg+Teubner Verlag, 2007.

[101] **Weinland**, **D.**, **Ronfard**, **R.** und **Boyer**, **E.** *A Survey of Vision-Based Methods for Action Representation, Segmentation and Recognition*. In: *Computer Vision and Image Understanding* 115 (2010), S. 224–241.

[102] **Weinland**, **Daniel** und **Boyer**, **Edmond**. *Action recognition using exemplar-based embedding*. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, S. 1–7.

[103] **Willems**, **Geert**, **Tuytelaars**, **Tinne** und **Van Gool**, **Luc**. *An efficient dense and scale-invariant spatio-temporal interest point detector*. In: (2008), S. 650–663.

[104] **Wilson**, **Andrew David** und **Bobick**, **Aaron F**. *Learning visual behavior for gesture analysis*. In: *Computer Vision, 1995. Proceedings., International Symposium on*. IEEE. 1995, S. 229–234.

[105] **Wu**, **Shandong**, **Oreifej**, **Omar** und **Shah**, **Mubarak**. *Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories*. In: (2011), S. 1419–1426.

[106] **Yamato**, **Junji**, **Ohya**, **Jun** und **Ishii**, **Kenichiro**. *Recognizing human action in time-sequential images using hidden markov model*. In: *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*. IEEE. 1992, S. 379–385.

[107] **Yang**, **Yi** und **Ramanan**, **Deva**. *Articulated pose estimation with flexible mixtures-of-parts*. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, S. 1385–1392.

[108] **Yao**, **Angela**, **Gall**, **Juergen** und **Van Gool**, **Luc**. *Coupled action recognition and pose estimation from multiple views*. In: *International journal of computer vision* 100.1 (2012), S. 16–37.

[109] **Yeffet**, **Lahav** und **Wolf**, **Lior**. *Local trinary patterns for human action recognition*. In: (2009), S. 492–497.

[110] **Yi**, **Yang** und **Lin**, **Yikun**. *Human action recognition with salient trajectories*. In: *Signal Processing* 93 (2013), S. 2932–2941.

[111] **Zhang**, **Jianguo**, **Marszałek**, **Marcin**, **Lazebnik**, **Svetlana** und **Schmid**, **Cordelia**. *Local features and kernels for classification of texture and object categories: A comprehensive study*. In: *International journal of computer vision* 73.2 (2007), S. 213–238.

[112] **Zhao**, **Guoying** und **Pietikainen**, **Matti**. *Dynamic texture recognition using local binary patterns with an application to facial expressions*. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29.6 (2007), S. 915–928.

Eigene Veröffentlichungen

[113] **Back**, **Kristine**, **Heiman**, **Ron** und **Puente León**, **Fernando**. *Modellbasierte Analyse menschlicher Bewegungen*. In: *Forum Bildverarbeitung* (2010), S. 277–288.

[114] **Back**, **Kristine**, **Hernández Mesa**, **Pilar** und **Puente León**, **Fernando**. *Modellbasiertes dreidimensionales Posentracking mittels evolutionärem Algorithmus*. In: *Forum Bildverarbeitung* (2012), S. 267–278.

[115] **Back**, **Kristine** und **Puente León**, **Fernando**. *Erfassung menschlicher Bewegungen durch merkmalsbasiertes Tracking*. In: *XXV. Messtechnisches Symposium des Arbeitskreises der Hochschullehrer für Messtechnik e.V. (AHMT)* (2011).

[116] **Back**, **Kristine** und **Puente León**, **Fernando**. *Erfassung und Erkennung menschlicher Aktionen durch Tracking von Bewegungsmerkmalen*. In: *Technisches Messen* 79 (2012).

[117] **Back**, **Kristine** und **Puente León**, **Fernando**. *Erkennung menschlicher Aktivitäten durch Körper-Tracking mit evolutionärem Algorithmus*. In: 2014.

[118] **Back**, **Kristine**, **Hernández Mesa**, **Pilar**, **Diebold**, **Maximilian** und **Puente León**, **Fernando**. *Dreidimensionales Körper-Tracking mit Hilfe eines evolutionären Algorithmus*. In: *Technisches Messen* 80 (2013), S. 335–342.

[119] **Christ**, **Konrad**, **Michelsburg**, **Matthias**, **Back**, **Kristine**, **Eidam**, **Alexander** und **Kiencke**, **Uwe**. *Möglichkeiten zur Injektorkalibrierung mit Hilfe von Klopfsensoren bei der Benzin-Direkteinspritzung*. In: *Sensoren und Messsysteme 2010, 15. ITG/GMA-Fachtagung* (2010), S. 374–379.

[120] **Christ**, **Konrad**, **Back**, **Kristine**, **Jiqqir**, **Mehdi**, **Kiencke**, **Uwe** und **Puente León**, **Fernando**. *Calibration of solenoid injectors for gasoline direct injection using the knock sensor*. In: *MTZ worldwide* 72(4) (2011), S. 64–70.

[121] **Christ**, **Konrad**, **Back**, **Kristine**, **Jiqqir**, **Mehdi**, **Kiencke**, **Uwe** und **Puente León**, **Fernando**. *Kalibrierung von Magnet-Injektoren für Benzin-Direkteinspritzung mittels Klopfsensor*. In: *MTZ - Motortechnische Zeitschrift* 72(4) (2011), S. 322–328.

[122] **Christ**, **Konrad**, **Back**, **Kristine**, **Kieweler**, **Thomas**, **Kiencke**, **Uwe** und **Puente León**, **Fernando**. *On applying the knock sensor for injector calibration*. In: *In International Journal of Engine Research* 15 (2014).

Betreute studentische Arbeiten

[123] **Dahlheimer**, **Niklas**. *Modellierung und Erkennung menschlicher Bewegungen durch Multiskalenanalyse*. Studienarbeit. Karlsruher Institut für Technologie (KIT), 2012.

[124] **Diebold**, **Maximilian**. *Bewegungstracking des menschlichen Körpers mittels Interacting Simulated Annealing Partikel-Filter*. Studienarbeit. Karlsruher Institut für Technologie (KIT), 2011.

[125] **Diebold**, **Maximilian**. *Entwicklung einer markerbasierten Bewegungsanalyse und Parameteroptimierung eines markerlosen Körpertrackings*. Diplomarbeit. Karlsruher Institut für Technologie (KIT), 2012.

[126] **Finkenbein**, **Felix**. *Tracking von SURF-Merkmalen für die Aktivitätserkennung*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2012.

[127] **Heiman**, **Ron**. *Implementierung und Vergleich von Raum-Zeit-Merkmalen zur Erkennung von menschlichen Bewegungen*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2011.

[128] **Jany**, **Josefin**. *Modellierung menschlicher Bewegungen durch Multiskalenanalyse*. Studienarbeit. Karlsruher Institut für Technologie (KIT), 2011.

[129] **Jian**, **Ren**. *Learning of Hidden Markov Models for Human Activities Recognition*. Masterarbeit. Karlsruher Institut für Technologie (KIT), 2012.

[130] **Laubach**, **Felix**. *Farbbasiertes Tracking von Menschen in Videos*. Studienarbeit. Karlsruher Institut für Technologie (KIT), 2012.

[131] **Martini**, **Johannes**. *Entwicklung und Analyse von Likelihoodfunktionen für markerloses Körpertracking*. Studienarbeit. Karlsruher Institut für Technologie (KIT), 2012.

[132] **Matthäus**, **Ralf**. *Detektion und Tracking von Gesichtern basierend auf Farbe*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2011.

[133] **Mesa**, **Pilar Hernández**. *Erfassung menschlicher Bewegungen durch markerloses dreidimensionales Körpertracking*. Diplomarbeit. Karlsruher Institut für Technologie (KIT), 2012.

[134] **Milella**, **Vito-Oronzo**. *Physikalische Modelle für die Dynamik menschlicher Bewegungen*. Diplomarbeit. Karlsruher Institut für Technologie (KIT), 2011.

[135] **Roith**, **Sebastian**. *Erkennung menschlicher Bewegungen mittels parametrischer Modelle*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2011.

[136] **Semmler**, **Michaela Alexandra**. *Segmentierung menschlicher Bewegungen durch Detektion von Aktionsgrenzen*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2011.

[137] **Varona**, **Maria Cruz**. *Lernen statistischer Modelle menschlicher Bewegungen aus Motion-Capture-Daten*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2011.

[138] **Weber**, **Tobias**. *Erkennung menschlicher Bewegungen in Videos mittels Wavelettransformation und Local Binary Patterns*. Studienarbeit. Karlsruher Institut für Technologie (KIT), 2014.

[139] **Winterbauer**, **Eric**. *Tracking von Bewegungsmerkmalen zur Erfassung menschlicher Aktionen*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2011.

[140] **Xu**, **Liqing**. *Bildbasiertes Tracking von Händen mittels Farb- und Bewegungsmerkmalen*. Masterarbeit. Karlsruher Institut für Technologie (KIT), 2012.

Forschungsberichte aus der Industriellen Informationstechnik (ISSN 2190-6629)

Institut für Industrielle Informationstechnik
Karlsruher Institut für Technologie (KIT)

Hrsg.: Prof. Dr.-Ing. Fernando Puente León

Band 1 Pérez Grassi, Ana
Variable illumination and invariant features for detecting and classifying varnish defects. (2010)
ISBN 978-3-86644-537-6

Band 2 Christ, Konrad
Kalibrierung von Magnet-Injektoren für Benzin-Direkteinspritzsysteme mittels Körperschall. (2011)
ISBN 978-3-86644-718-9

Band 3 Sandmair, Andreas
Konzepte zur Trennung von Sprachsignalen in unterbestimmten Szenarien. (2011)
ISBN 978-3-86644-744-8

Band 4 Bauer, Michael
Vergleich von Mehrträger-Übertragungsverfahren und Entwurfskriterien für neuartige Powerline-Kommunikationssysteme zur Realisierung von Smart Grids. (2012)
ISBN 978-3-86644-779-0

Band 5 Kruse, Marco
Mehrobjekt-Zustandsschätzung mit verteilten Sensorträgern am Beispiel der Umfeldwahrnehmung im Straßenverkehr. (2013)
ISBN 978-3-86644-982-4

Band 6 Dudeck, Sven
Kamerabasierte In-situ-Überwachung gepulster Laserschweißprozesse. (2013)
ISBN 978-3-7315-0019-3

Band 7 Liu, Wenqing
Emulation of Narrowband Powerline Data Transmission Channels and Evaluation of PLC Systems. (2013)
ISBN 978-3-7315-0071-1

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar oder als Druckausgabe bestellbar.

Band 8 Otto, Carola
Fusion of Data from Heterogeneous Sensors with Distributed Fields of View and Situation Evaluation for Advanced Driver Assistance Systems. (2013)
ISBN 978-3-7315-0073-5

Band 9 Wang, Limeng
Image Analysis and Evaluation of Cylinder Bore Surfaces in Micrographs. (2014)
ISBN 978-3-7315-0239-5

Band 10 Michelsburg, Matthias
Materialklassifikation in optischen Inspektionssystemen mithilfe hyperspektraler Daten. (2014)
ISBN 978-3-7315-0273-9

Band 11 Pallauf, Johannes
Objektsensitive Verfolgung und Klassifikation von Fußgängern mit verteilten Multi-Sensor-Trägern. (2016)
ISBN 978-3-7315-0529-7

Band 12 Sigle, Martin
Robuste Schmalband-Powerline-Kommunikation für Niederspannungsverteilernetze. (2016)
ISBN 978-3-7315-0539-6

Band 13 Opalko, Oliver
Powerline-Kommunikation für Batteriemanagement-Systeme in Elektro- und Hybridfahrzeugen. (2017)
ISBN 978-3-7315-0647-8

Band 14 Han, Bin
Characterization and Emulation of Low-Voltage Power Line Channels for Narrowband and Broadband Communication. (2017)
ISBN 978-3-7315-0654-6

Band 15 Alonso, Damián Ezequiel
Wireless Data Transmission for the Battery Management System of Electric and Hybrid Vehicles. (2017)
ISBN 978-3-7315-0670-6

Band 16 Hernández Mesa, Pilar
Design and analysis of a content-based image retrieval system. (2017)
ISBN 978-3-7315-0692-8

Band 17 Suchaneck, André
Energiemanagement-Strategien für batterieelektrische Fahrzeuge. (2018)
ISBN 978-3-7315-0773-4

Band 18 Bauer, Sebastian
Hyperspectral Image Unmixing Incorporating Adjacency Information. (2018)
ISBN 978-3-7315-0788-8

Band 19 Vater, Sebastian
Monokulare Blickrichtungsschätzung zur berührungslosen Mensch-Maschine-Interaktion. (2019)
ISBN 978-3-7315-0789-5

Band 20 Back, Kristine
Erkennung menschlicher Aktivitäten durch Erfassung und Analyse von Bewegungstrajektorien. (2019)
ISBN 978-3-7315-0909-7

ISSN 2190-6629
ISBN 978-3-7315-0909-7

Gedruckt auf FSC-zertifiziertem Papier