Fakultät für Informatik

Bildverstehen und wissensbasierte Systeme

Erkennung und Verfolgung von relevanten Objekten zur semantischen Annotierung von dynamischen, monokularen Szenen am Beispiel von Fußballübertragungen

Michael Barthel

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende(r):	Prof. Dr. Alois Chr. Knoll
Prüfer der Dissertation:	
	1. Prof. Dr. Bernd Radig
	2. Prof. Gudrun J. Klinker, Ph.D.

Die Dissertation wurde am 20.06.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 11.11.2017 angenommen.

Zusammenfassung

Die vorliegende Dissertation beschäftigt sich mit der schwierigen Aufgabe, in einer monokularen Videosequenz alle Objekte, die für eine semantische Annotation relevant sind, automatisch zu erkennen und zu verfolgen.

Was dabei für einen menschlichen Betrachter meist unterbewusst und ohne Schwierigkeit abläuft, stellt ein computerbasiertes System vor enorme Probleme, die bisher vom Stand der Technik noch unzureichend gelöst sind. Hierzu gehören unter anderem die im Vorhinein unbekannte Art und Anzahl der relevanten Objekte, die unterschiedliche optische Erscheinung von Objekten, Überdeckungen, fehlerbehaftete Sensordaten sowie rasche Positionsveränderungen in aufeinanderfolgenden Bildern.

Der vorgestellte Ansatz ist robust gegenüber veränderten Randbedingungen. Die Grundlage dafür ist eine Kombination aus allgemeingültigen Merkmalen, wie die Textur von Personen, aus objekt-spezifischen Merkmalen, wie die Farbe der individuellen Bekleidung einer Person sowie aus regionen- und formbasierten Merkmalen. Die Implementierung als Online-Verfahren ist potentiell echtzeitfähig und in der Lage mit teilweisen und vollständigen Überdeckungen umzugehen. Durch die automatische Initialisierung entfällt die Notwendigkeit von manuellen Eingriffen. Der Ansatz berücksichtigt dynamische Bewegungen der Kameraführung und bietet somit die Möglichkeit, Aufnahmen von schwenkenden und zoomenden Kameras erfolgreich zu prozessieren. Liegen Transformationen in ein Weltkoordinatensystem vor, bereinigt das vorgestellte System die generierten Bewegungstrajektorien der Objekte in einer effizienten Nachbearbeitung anhand von physikalischen Plausibilitätskriterien.

Durch eine empirische Evaluierung mit mehreren Stunden anspruchsvollem Videomaterial werden die Leistungsfähigkeit des Ansatzes und seine Vorzüge gegenüber dem Stand der Forschung demonstriert.

Die bewusst gewählte Fokussierung auf Aufnahmen von Fußballbegegnungen ermöglicht, durch das vorhandene Domänenwissen, die Konzentration auf einen Teilbereich der wissenschaftlichen Fragestellungen. Gleichzeitig bringt dieser Anwendungsbereich besondere Herausforderungen mit sich, wie eine hohe Dynamik der sich bewegenden Objekte, vermehrte Objektüberdeckungen oder einen verstärkten Einfluss durch äußere Bedingungen. Die Fülle an interessanten Anwendungen und erhältlichem Video- und Evaluierungsmaterial legt die in dieser Arbeit vorgenommene Spezialisierung zusätzlich nahe.

Abstract

This thesis tackles the challenging task of automatically detecting and tracking all objects in a monocular video sequence that are relevant for a sematic annotation.
Although easy to solve for a human observer using subconscious mechanisms, state-of-the-art computer-based systems still provide insufficient results for such kind of problems. This includes challenges such as unknown type and number of relevant objects, different optical appearances of objects, occlusions, noisy sensor output, as well as rapid movements between subsequent images.
The presented approach is robust with respect to changing external settings and is based on a combination of general features, such as the texture of persons, of object-specific features, such as the color of the individual clothes, as well as of region- and shape-based features. The implementation in form of an online approach is in principle capable to operate in real-time und to handle partial and full occlusions. By means of an automated self-initialization, the system is not dependent on manual input. The approach is able to handle dynamic movements of the camera and hence to process sequences created by pan-tilt-zoom cameras. If transformations from the image to the world coordinate system are available, the system performs a revision of the generated trajectories in a post-processing step using criteria of physical plausibility.
An extensive empiric evaluation using several hours of challenging video data demonstrates the performance of the system and its advantage over the state-of-the-art.
The intentional focus on records of soccer matches enables the concentration on subdomains of the general problem, using existing domain knowledge. Simultaneously, this application brings special challenges, such as highly dynamic objects, extensive object occlusion, or an increased influence of external conditions. Additionally, this kind of specialization is motivated by the plenty of available video and evaluation material and by possible applications.

Danksagung

Mein Dank geht an Prof. Dr. Bernd Radig, der durch seine Unterstützung und kompetente Betreuung diese Arbeit überhaupt ermöglicht hat.
Zudem möchte ich meinen Kollegen in der Forschungsgruppe Dr. Christoph Mayer und Dr. Martin Hörnig für die konstruktive Zusammenarbeit und die vielen fachlichen Anregungen, Quirin Lohr für den IT-Support sowie Andreas Bigontina, Martin Hopper und Trung Hieu Dao für ihre tollen studentischen Arbeiten danken.
Ein besonderer Dank geht an meine Frau Nicole, die mich immer unterstützt hat, an meinen Sohn Jonas, dafür dass er da ist sowie an meine Eltern, die mir meinen Bildungsweg ermöglicht haben.
Zuletzt möchte ich meiner Frau, Martin und meinen Eltern für die intensive Korrekturhilfe danken, durch die ich den einen oder anderen Flüchtigkeitsfehler noch beheben konnte.

Inhaltsverzeichnis

Danksagung v

Inhaltsverzeichnis vi

Abbildungsverzeichnis xiii

Tabellenverzeichnis xv

Liste der Algorithmen xvii

I Einleitung und Grundlagen 1

1 Einleitung 3
- 1.1 Einführung . . . 3
- 1.2 Motivation der Domäne Fußball . . . 4
- 1.3 Problemstellung . . . 10
- 1.4 Lösungsansätze und wissenschaftlicher Beitrag . . . 11
- 1.5 Stand der Forschung . . . 14
- 1.6 Inhaltsübersicht . . . 17

2 Grundlagen und Definitionen 19
- 2.1 Bild und Bildregionen . . . 19
  - 2.1.1 Das Bild . . . 19
  - 2.1.2 Die Bounding-Box . . . 20
- 2.2 Identität, Label und Qualitätsmaß . . . 20
- 2.3 Template . . . 21
- 2.4 Objekterkennung . . . 21
- 2.5 Aufnahmemodalitäten . . . 22
- 2.6 Das Spielfeld . . . 22

II Erkennung und Verfolgung von Spielern 25

3 Merkmale für die Spielerverfolgung 27
- 3.1 Einleitung . . . 27
- 3.2 Stand der Forschung . . . 30
- 3.3 Segmentierung des Vordergrundes . . . 35

3.3.1 Bestimmung der Feldhülle . . . 35
3.3.2 Bestimmung der Grasmaske . . . 36
3.4 Schätzung der Größe von Objekten im Bild . . . 37
3.5 Extraktion der Spielersilhouetten . . . 41
3.6 Farbtemplates . . . 45
3.6.1 Ausschluss von überdeckten Objekten . . . 48
3.6.2 Bestimmung der dominanten Farben . . . 49
3.6.3 Die Wahl des Farbraums . . . 52
3.6.4 Erzeugung von Farbtemplates für individuelle Objekte . . . 52
3.6.5 Gruppierung der individuellen Objekttemplates . . . 53
3.7 Die Konfidenzkarte (*Confidence Map*) . . . 56
3.7.1 Silhouettenbasierte Konfidenz . . . 58
3.7.2 Überlappungsbasierte Konfidenz . . . 59
3.7.3 Farbbasierte Konfidenz . . . 59
3.7.4 Texturbasierte Konfidenz . . . 60
3.7.5 Vorhersagebasierten Konfidenz . . . 61
3.7.6 Ensemble Averaging . . . 61
3.8 Auffinden lokaler Konfidenzmaxima . . . 62
3.9 Diskussion und Ausblick . . . 64

4 Spielerverfolgung **67**
4.1 Einleitung . . . 67
4.1.1 Online 2D-Spielerverfolgung . . . 67
4.1.2 Verfeinerung der 3D-Trajektorien . . . 68
4.2 Stand der Forschung . . . 69
4.3 Stochastische Filterung und Glättung . . . 73
4.3.1 Zustands- und Messmodell . . . 73
4.3.2 Kalman-Filter . . . 74
4.3.3 Stochastische Glättung . . . 75
4.3.4 Zustandsmodelle für die Spielerverfolgung . . . 76
4.4 2D-Spielerverfolgung . . . 76
4.4.1 Messvorgang für die Spielerpositionen . . . 76
4.4.2 Initiale Spielererkennung . . . 79
4.4.3 Zustandsmodell für die 2D-Spielerverfolgung . . . 84
4.4.4 Online-Spielerverfolgung in 2D . . . 85
4.4.5 Spielerverfolgung bei anderen Sportarten . . . 88
4.5 Verfeinerung der 3D-Trajektorien . . . 88
4.5.1 Einleitung . . . 88
4.5.2 Transformation der Bildpositionen zu Positionen im Spielfeld . . . 89
4.5.3 Auftrennung der Trajektorien in unterbrechungsfreie Stücke . . . 91
4.5.4 Glättung der Trajektorien . . . 91
4.5.4.1 Zustandsmodell für 3D-Trajektorien . . . 91
4.5.5 Entfernung potentiell falsch-positiver Erkennungen . . . 92
4.5.6 Vereinigung der Trajektorien . . . 93
4.5.7 Rückprojektion ins Bild . . . 94
4.6 Diskussion und Ausblick . . . 95

5 Automatisierte Bestimmung von Parametern 97
5.1 Einleitung . . . 97
5.2 Parameter und Parameterbestimmung . . . 99
5.2.1 Parameter für die 2D-Spielerverfolgung . . . 99
5.2.1.1 Kalman-Filter . . . 99
5.2.1.2 Statusbehandlung . . . 100
5.2.2 Automatische Parameterbestimmung . . . 100
5.2.2.1 Trainingsdatensatz . . . 101
5.2.2.2 Optimierung der Parameter . . . 102
5.2.2.3 Zufallsbasierte Suche mit adaptiven Wertebereichen . . . 103
5.2.2.4 Bewertungsmaße . . . 103
5.3 Diskussion und Ausblick . . . 107

6 Evaluierung 109
6.1 Einleitung . . . 109
6.1.1 Stand der Forschung . . . 111
6.2 Annotation . . . 113
6.3 Metriken zur Evaluierung von Verfahren zur Verfolgung mehrerer Ziele . . . 114
6.3.1 Eingabedaten . . . 114
6.3.2 Zuordnung . . . 116
6.3.2.1 Zuordnung ohne zeitliche Komponente . . . 116
6.3.2.2 Zuordnung mit zeitlicher Komponente . . . 117
6.3.2.3 Varianten . . . 119
6.3.3 Richtig-positiv, falsch-positiv und falsch-negativ . . . 120
6.3.4 Genauigkeit, Trefferquote und F-Maße . . . 120
6.3.5 Verwechslungen der Identität . . . 121
6.3.6 Verwechslungen der Mannschaftszuordnung . . . 123
6.3.7 Multiple Object Tracking Accuracy (MOTA) . . . 124
6.3.8 Multiple Object Tracking Precision (MOTP) . . . 124
6.3.9 Trajektorienbasierte Maße . . . 125
6.4 Implementierungsdetails . . . 126
6.4.1 Implementierung . . . 126
6.4.2 Auswahl der Bilder für die Outfitbestimmung . . . 126
6.4.3 Erkennung von dauerhafte Einblendungen . . . 127
6.5 Evaluierung der 2D-Spielerverfolgung . . . 128
6.5.1 Videosequenzen . . . 128
6.5.2 Vergleichsverfahren . . . 129
6.5.3 Modalitäten der 2D-Evaluierung . . . 131
6.5.4 Ergebnisse der 2D-Evaluierung . . . 132
6.6 Evaluierung der 3D-Spielerverfolgung . . . 136
6.6.1 Videosequenzen . . . 136
6.6.2 Modalitäten der 3D-Evaluierung . . . 136
6.6.3 Ergebnisse der 3D-Evaluierung . . . 138
6.7 Diskussion und Ausblick . . . 139

III Erkennung und Verfolgung des Balls 143

7 Verfolgung des Balls 145
7.1 Einleitung . . . 145
7.2 Stand der Forschung . . . 147
7.3 Erkennung des Balls . . . 149
7.3.1 Kaskadenklassifikation mit Boosting . . . 149
7.3.2 Local Binary Patterns . . . 151
7.3.3 Vor- und Nachverarbeitung . . . 151
7.3.4 Trainings- und Testdaten . . . 152
7.3.5 Implementierung und Auswertung . . . 153
7.4 Verfolgung des Balls . . . 153
7.4.1 Initialisierung . . . 154
7.4.2 Stochastische Schätzung der Balltrajektorie . . . 155
7.4.3 Messung der Ballposition . . . 155
7.4.3.1 Generierung der Suchregion . . . 155
7.4.3.2 Templatematching . . . 156
7.4.3.3 Geometrische Suche . . . 156
7.4.3.4 Grasmaske . . . 157
7.4.3.5 Balldetektor . . . 157
7.4.3.6 Kombination der Merkmale . . . 157
7.4.3.7 Auswahl des besten Kandidaten . . . 158
7.4.4 Aktualisierung des Templates . . . 158
7.4.5 Bestimmung des ballführenden Spielers . . . 159
7.4.6 Behandlung von Überdeckungen . . . 160
7.4.6.1 Verfolgung des ballführenden Spielers . . . 160
7.4.6.2 Überdeckung ohne ballführenden Spieler . . . 160
7.4.6.3 Wiedererkennung des Balls . . . 161
7.4.7 Hinweise durch Balldetektion . . . 161
7.4.8 Testdaten . . . 161
7.4.9 Implementierung und Auswertung . . . 162
7.5 Diskussion und Ausblick . . . 164

IV Anwendungen 167

8 Anwendungen 169
8.1 Einleitung . . . 169
8.2 Spielerverfolgung mit einer Webanwendung . . . 170
8.3 Bestimmung des Ballbesitzes . . . 172
8.4 Bestimmung der Posen . . . 173
8.4.1 Stand der Forschung . . . 173
8.4.2 Klassifikation von Körperteilen und Orientierung . . . 175
8.4.2.1 Random Forests . . . 175
8.4.2.2 Training . . . 176
8.4.2.3 Klassifikation . . . 177
8.4.2.4 Merkmale . . . 177

8.4.2.5 Klassifikation der Orientierung . . . 181
8.4.2.6 Klassifikation der Körperteile . . . 182
8.4.3 Anpassung der Pose . . . 183
8.4.3.1 Die Mittelpunkte der Körperteile . . . 184
8.4.3.2 Skelettanpassung . . . 185
8.4.4 Evaluierung . . . 187
8.4.4.1 Datensatz . . . 187
8.4.4.2 Klassifikation der Orientierung . . . 187
8.4.4.3 Klassifikation der Körperteile . . . 188
8.4.4.4 Anpassung der Pose . . . 189
8.4.5 Fazit und Ausblick . . . 191

9 Diskussion und Ausblick **193**

A Relevante Publikationen **199**
A.1 Relevante Publikationen des Autors . . . 199
A.2 Vom Autor betreute Arbeiten . . . 200

B Wahl der Parameterwerte **201**

C Unterschiede bei Implementierungen von CLEAR-MOT **205**

Literatur **209**

Abbildungsverzeichnis

1.1 Beispiele für Aufnahmen von Fußballbegegnungen 5
1.2 Ein Beispiel für die Umwandlung des Originalbilds 10
1.3 Eine Übersicht des Trackingverfahrens 13

2.1 Mögliche Größen des Spielfelds 22

3.1 Übersicht über das Vorgehen bei der Merkmalsberechnung 30
3.2 Bild einer simulierten Kamera 39
3.3 Zwischenergebnisse der Extraktion der Spielersilhouetten 44
3.4 Ein Beispiel für die Bestimmung der dominanten Farben 50
3.5 Ein Beispiel für das zusammengesetzte Farbhistogramm eines Spielers . . 54
3.6 Ein Beispiel für Teamhistogramme 55
3.7 Bildausschnitte (oben) und die zugehörigen Konfidenzkarten (unten). . . . 56
3.8 Beispiel überlappungsbasierte Konfidenz 59

4.1 Beispiel für die Ergebnisse der initialen Detektion 81
4.2 Beispiel für die Ergebnisse der NMS (*Non-Maximum Suppression*) 81
4.3 2D-Spielerverfolgung für Feldhockey 88

5.1 Eine kleine Auswahl an Bildern zum Einlernen der Parameter 101

6.1 Normierung des Seitenverhältnis 114
6.2 Ausschnitte 2D-Spielerverfolgung 130
6.3 Zu klein erkannte Spieler in **ML** (links) und **VS** (rechts). 134
6.4 Ergebnisse pro Videosequenz 2D 135
6.5 Ergebnisse im zeitlichen Verlauf pro Videosequenz 3D 137
6.6 Ergebnisse pro Videosequenz 3D 138

7.1 Verschiedene Erscheinungsformen des Balls 146
7.2 Das Prinzip einer Kaskade von Klassifikatoren 150

8.1 Eine Webanwendung zur Analyse von Fußballvideos. 171
8.2 Darstellung einiger untersuchter Merkmale 177
8.3 Beispiele für Spieler und ihre Orientierung 180
8.4 Klassifikation der Körperteile 183
8.5 Wahrscheinlichkeiten der Körperteile 183
8.6 3D-Skelettmodell 185
8.7 Vergleich mit dem Ansatz von Ansatz von Y. Yang und Ramanan 190

Tabellenverzeichnis

6.1 Übersicht Videosequenzen . . . 127
6.2 Übersicht Begegnungen . . . 127
6.3 Ergebnisse (MOTA und andere) für Schwellwert $\tau_d := 0{,}5$ . . . 133
6.4 Ergebnisse (MME und andere) für Schwellwert $\tau_d := 0{,}5$ . . . 133
6.5 Ergebnisse (MOTA und andere) für Schwellwert $\tau_d := 0{,}7$ . . . 133
6.6 Ergebnisse (MME und andere) für Schwellwert $\tau_d := 0{,}7$ . . . 133
6.7 Ergebnisse (MOTA und andere) für Schwellwert $\tau_d := 0{,}9$ . . . 133
6.8 Ergebnisse (MME und andere) für Schwellwert $\tau_d := 0{,}9$ . . . 133
6.9 Vergleich der Laufzeiten und Frameraten der Verfahren . . . 134
6.10 Übersicht Begegnungen für die 3D-Evaluierung . . . 136
6.11 Übersicht Videosequenzen für die 3D-Evaluierung . . . 136
6.12 Ergebnisse der 3D-Evaluierung für verschiedene Schwellwerte . . . 137

7.1 Ergebnis der Balldetektion . . . 153
7.2 Übersicht Videosequenzen . . . 162
7.3 Übersicht Begegnungen . . . 162
7.4 Ergebnisse ohne Spielerpositionen (50 Bilder / Sequenz) . . . 163
7.5 Ergebnisse mit Spielerpositionen (50 Bilder / Sequenz) . . . 163
7.6 Ergebnisse mit und ohne Hinweise (100 Bilder / Sequenz) . . . 164

8.1 Genauigkeit der Klassifikation der Orientierung (korrekt klassifizierte Bilder) . . . 187
8.2 Genauigkeit der Klassifikation der Körperteile (korrekt klassifizierte Pixel) 188
8.3 PCP . . . 189
8.4 PCK . . . 189
8.5 PCP (bei Billigung von Links-/Rechts-Vertauschungen) . . . 190
8.6 PCK (bei Billigung von Links-/Rechts-Vertauschungen) . . . 190

B.1 Parameter für die Größenbestimmung . . . 201
B.2 Parameter für die Extraktion der Spielersilhouetten . . . 202
B.3 Parameter für die Bestimmung der dominanten Farben . . . 202
B.4 Parameter für die Bestimmung der verschiedenen Outfit-Templates . . . 202
B.5 Parameter für die Berechnung der Konfidenzkarte . . . 202
B.6 Parameter für die Berechnung der Konfidenzkarte . . . 203
B.7 Parameter für die Berechnung der Konfidenzkarte . . . 203

C.1 Beispiel 1a . . . 206
C.2 Beispiel 1b . . . 206
C.3 Beispiel 2a . . . 206

C.4 Beispiel 2b 206
C.5 Beispiel 3a 206
C.6 Beispiel 3b 206
C.7 Beispiel 4a 207
C.8 Beispiel 4b 207
C.9 Beispiel 5a 207
C.10 Beispiel 5b 207
C.11 Beispiel 6a 207
C.12 Beispiel 6b 207

Liste der Algorithmen

3.1 Iterativ-neugewichtete kleinste Quadrate mit Randbedingungen 42

3.2 Extraktion der Spielersilhouetten 43

3.3 Bestimmung der dominanten Farben 51

3.4 Bestimmung der verschiedenen Outfit-Templates 55

3.5 Auffinden lokaler Konfidenzmaxima 63

4.1 Messvorgang 77

4.2 Initiale Spielererkennung 82

4.3 Vereinigung der Trajektorien 94

6.1 Gierige (*Greedy*) Zuordnung von Referenz- und Ergebnisobjekten 117

Teil I

Einleitung und Grundlagen

Kapitel 1

Einleitung

1.1 Einführung

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung eines robusten, echtzeitfähigen Systems zur Erkennung und Verfolgung von Objekten in monokularen Videosequenzen, die für deren semantische Interpretation relevant sind. Dabei versteht man unter Objektverfolgung „die Aufgabe, die Position von Objekten von Interesse in einer Bildsequenz über die Zeit hinweg zu schätzen“ (Maggio und Cavallaro 2011, S. xvii).

Um die Unmenge an weltweit verfügbarem Videomaterial den jeweiligen Interessengruppen angemessen zur Verfügung zu stellen, ist eine semantische Aufbereitung unumgänglich. Manuelle Prozesse sind extrem kostspielig und beschränken sich häufig auf sehr einfache Konstrukte, wodurch semi- und vollautomatische Systeme in den Vordergrund rücken. Trotz beachtlicher Fortschritte, wie man beispielsweise an der Methode von He u. a. (He u. a. 2015) erkennen kann, die in zahlreichen Kategorien des *ImageNet*-Wettbewerbs 2015 (Russakovsky u. a. 2015) gewinnen konnte, ist der Stand der Technik bei der semantischen Interpretation von visuellen Daten noch weit entfernt von der kognitiven Leistung eines Menschen.

Für die semantische Einordnung einer Videosequenz ist es essentiell, die Ereignisse und Handlungen (insbesondere von Menschen) innerhalb der Sequenz zu erkennen und zu analysieren. Um dies zu ermöglichen, müssen in einem vorherigen Schritt die relevanten Objekte lokalisiert, identifiziert und über die Zeit verfolgt werden. Dieser Vorgang der Objekterkennung und -verfolgung läuft bei einem menschlichen Betrachter unterbewusst ab und stellt in der Regel keine Schwierigkeit dar. Ein computerbasiertes kognitives System hingegen trifft hier auf eine Unzahl von Herausforderungen, die durch den gegenwärtigen Stand der Technik noch nicht zufriedenstellend gelöst sind. Dies zeigt sich

etwa an den Ergebnissen des *Multiple Object Tracking Benchmark* (Milan u. a. 2016), bei dem die Leistung von Verfahren zur Verfolgung von mehreren Objekten verglichen wird. Dort liegen die besten Einreichungen deutlich von einem fehlerfreien Leistungsbereich entfernt. Ein Grund dafür sind die zahlreichen Schwierigkeiten, mit denen ein solches automatisches System umgehen muss:

- Ohne kontextuelles Vorwissen ist nicht bekannt, welche Art von Objekte in der Sequenz erscheinen. Ein automatisches Erkennungssystem muss daher mit einer (beliebig) weiten Bandbreite an verschiedenen Klassen von zu erkennenden Objekten umgehen können.
- Dem System ist nicht bekannt, welche und wie viele der (ggf. zahlreichen) Objekte in einer Sequenz für die semantische Einordnung relevant sind.
- Auch wenn die Art der relevanten, zu erkennenden Objekte bekannt ist, kann die optische Erscheinung solcher Objekte stark unterschiedlich sein. Das kann an den Objekten selbst liegen (beispielsweise durch unterschiedliche Form und Farbe) oder an den Aufnahmemodalitäten (beispielsweise durch eine unterschiedliche Aufnahmeperspektive oder Beleuchtung der Szene).
- Objekte können durch schnelle Bewegungen und andere Transformationen ihr Aussehen in kurzer Zeit stark verändern.
- Objekte können im Laufe der Sequenz teilweise oder ganz verdeckt sein, sie können die Szene verlassen und müssen nach Wiedererscheinen richtig erkannt und identifiziert werden.
- Die Sensordaten können unsicher und fehlerbehaftet sein.

Diese Problemstellungen werden im Rahmen dieser Arbeit untersucht und in eine Domäne eingebettet, die zusätzliche Herausforderungen und spannende Anwendungsmöglichkeiten bietet: In die Analyse von monokularen Aufzeichnungen von feldbasierten Sportspielen (im Speziellen Fußball).

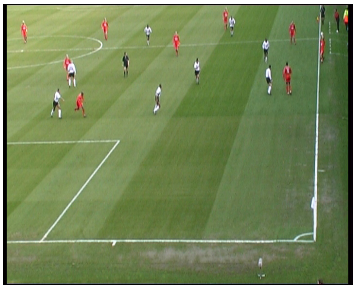
1.2 Motivation der Domäne Fußball

Vollständiges Domänenwissen bei Sportspielen

Wie bereits erwähnt, sind die Herausforderungen für ein automatisches kognitives System zur Erkennung und Verfolgung mehrerer Objekte ohne weiteres Kontextwissen immens. Die gezielte Beschränkung auf eine Domäne ermöglicht es, sich auf einen Teilbereich der wissenschaftlichen Fragestellungen zu konzentrieren. Das Domänenwissen bei

(a) Gute Aufnahme- und Witterungsverhältnisse (DFL 2014c)

(b) Schlechte Witterungsverhältnisse (Eurosport 2013a)



(c) Amateuraufnahme mit ungünstigem Blickfeld (University of Reading 2002)

Abbildung 1.1: Beispiele für Aufnahmen von Fußballbegegnungen

feldbasierten Sportspielen ist in der Theorie vollständig: Es gibt Spielregeln, die diverse Rahmenbedingungen eindeutig vorgeben. Unter anderem ist die Anzahl der relevanten Personen auf dem Spielfeld in der Regel bekannt, Mannschaften lassen sich durch farblich unterscheidbare Trikots auseinanderhalten und die Art und Form von Spielfeld und Spielgeräten ist definiert. Auch die Tatsache, dass sich alles (näherungsweise) auf einer Ebene abspielt, erleichtert die 3D-Rekonstruktion der Szene. Dieses Kontextwissen ermöglicht nicht nur die Fokussierung auf relevante Teilgebiete. Es erleichtert zudem die Implementierung von robusten, produktnahen Anwendungen, die auch andere wissenschaftliche Disziplinen, wie beispielsweise die Sportwissenschaft, unterstützen können.

Domänenspezifische Herausforderungen bei Sportspielen

Auch wenn das Wissen über die eingeschränkte Domäne ausgenutzt werden kann, steht die monokulare Analyse von feldbasierten Sportspielen vor diversen anspruchsvollen Herausforderungen, die in anderen Anwendungsbereichen (wie beispielsweise der Videoüberwachung) keine oder nur eine unbedeutende Rolle spielen:

- Im Sport kommt es häufig zu dynamischen Szenen, die sehr schnelle und nichtlineare Bewegungen enthalten, vor allem, wenn ein Ball das zentrale Element des Spiels ist. Eine solche Dynamik muss nicht nur im Bewegungsmodell eines automatischen Systems abgedeckt sein, sondern sie kann auch zu Objektverformungen

und unscharfen Darstellungen führen, die die Erkennung und Verfolgung weiter erschweren.

- Eine intrinsische Eigenschaft von monokularen Aufnahmen ist es, dass die dreidimensionale Rekonstruktion der Szene ein unterbestimmtes Problem ist. Hinzu kommt, dass es bei großen Spielfeldern zu Nachteilen führen kann, eine ganze Szene (also das gesamte Spielfeld) mit einer Kamera zu erfassen. Es gibt hier prinzipiell zwei gegensätzliche Herangehensweisen:
  - Es wird versucht möglichst viel von der Szene zu erfassen und die Kameraführung weist dabei wenig oder keine Bewegung auf. Allerdings kann dies zu Problemen bei Weitwinkelaufnahmen führen, wie die kleine Größe von Objekten im Bild (geringe Anzahl an Pixeln) oder die starke Verzeichnung durch die Optik.
  - Eine zweite Möglichkeit ist es, nur einen relevanten Teil der Szene (beispielsweise das Geschehen um den Ball) zu erfassen. Dabei stellt die geringe Größe von Objekten und die optische Verzeichnung eine weniger gewichtige Rolle dar. Allerdings wird das System mit anderen kritischen Punkten, wie einem eingeschränktem Blickfeld, schneller Kamerabewegungen und Zoomänderungen konfrontiert.
- Die Möglichkeiten monokulare Aufnahmen zu erstellen sind vielfältig. Die Spanne reicht von Hochglanzaufnahmen der Sportclubs (sogenannte *Scoutingfeeds*), über hochauflösende und weniger hochauflösende Fernsehaufnahmen, bis hin zu Amateuraufnahmen mit Camcorder oder Smartphone. Ein automatisches System muss mit dieser Vielfalt an Aufnahmemodalitäten zurechtkommen, die zu unterschiedlichen Objektgrößen im Bild und verschiedenen Bildqualitäten, beispielsweise bedingt durch Aufnahme- und Kompressionsartefakte, führen können. Eine Auswahl von Aufnahmen mit unterschiedlichen Bedingungen ist in Abbildung 1.1 dargestellt.
- Auch wenn die Rahmenbedingungen eines Sportspiels durch Regeln vorgegeben sind, kann es zu starken Variationen kommen. So gibt es häufig Toleranzen in den Regeln, wie beispielsweise bezüglich der vorgeschriebenen Feldgröße. Ebenso können sich die äußeren Bedingungen ändern und für Schwierigkeiten sorgen, zum Beispiel durch schlechte Wetterbedingungen, ausgeprägte Licht-Schatten-Bereiche in Stadien oder unterschiedliche Banden- und Bodenwerbung.
- Bei feldbasierten Sportspielen treffen meist zwei Mannschaften aufeinander, die versuchen sich gegenseitig an erfolgreichen Aktionen zu hindern. Dies führt dazu, dass sich sehr häufig zwei oder mehrere Spieler in unmittelbarer Nähe zueinander

befinden, beispielsweise beim Zweikampf oder bei der Manndeckung. Diese Spieler überdecken sich aus Sicht der Kamera teilweise oder komplett. Ebenso ist das Sportgerät selten im freien Feldbereich, sondern wird von Spielern geführt oder umkämpft. Des Weiteren gibt es Situationen, wie Frei- oder Eckstöße, in denen sich viele Spieler auf einem kleinen Bereich des Spielfelds befinden. Das stellt ein automatisches System vor Schwierigkeiten bei der Verfolgung und Identifizierung von Spielern. In anderen Domänen treten solche Situationen in abgeschwächter Form auf. Zum Beispiel versuchen sich Fußgänger normalerweise gegenseitig aus dem Weg zu gehen.

- Trikots erleichtern die optische Unterscheidung von Spielern unterschiedlicher Mannschaften. Die Identifizierung von Spielern derselben Mannschaft wird jedoch erschwert, da sich die optischen Erscheinungen nur geringfügig unterscheiden. Rückenbeschriftung, Haarfarbe oder Schuhfarbe sind Merkmale außerhalb des Mannschaftsoutfits, die bei der Identifizierung individueller Spieler behilflich sein können. Allerdings sind sie oft nicht sichtbar (wie im Fall von Rückennummern) oder bei schlechter Bildqualität und kleinen Objektgrößen im Bild schwierig zu extrahieren. Dies führt zu einer deutlich erschwerten Identifizierung bei Sportspielen im Vergleich zu Szenen mit heterogenen Erscheinungen der relevanten Objekte, wie beispielsweise in einer Szene mit Fußgängern.

Umfangreiche Datenverfügbarkeit im Bereich der Sportspiele

Eine wesentliche Problematik bei der Entwicklung von kognitiven Systemen zur Analyse von Videosequenzen ist, insbesondere bei der Erkennung und Verfolgung von Objekten, die Möglichkeit einer repräsentativen Evaluierung. Die Erzeugung einer sogenannten *Ground Truth* ist meist nur mit manuellen oder semi-automatischen Methoden möglich und generiert, neben anderen Schwierigkeiten, einen enormen zeitlichen Aufwand (siehe dazu auch Kapitel 6). So steht für den oben genannten *Multiple Object Tracking Benchmark* (Milan u. a. 2016) insgesamt nur ungefähr 30 Minuten (Stand Mai 2016) annotiertes Videomaterial zur Verfügung. Im professionellen Sportbereich werden seit einigen Jahren Bewegungsdaten von Spielern und Ball in den höherklassigen Ligen (wie beispielsweise der deutschen Fußball-Bundesliga oder der nordamerikanischen National Basketball Association (NBA)) mit aufwendigen Systemen automatisch erfasst (ChyronHego Corporation 2016; Prozone Sports 2016). Somit stehen diese Daten prinzipiell im großen Maßstab zur Verfügung, um die Auswertung von einfachen monokularen Aufnahmen (wie Fernsehaufnahmen) zu evaluieren und die Eignung solcher Systeme für den Einsatz außerhalb des gegenwärtigen professionellen Spielbetriebs (beispielsweise im Amateurbereich oder zur Auswertung von Begegnungen aus vergangenen Jahren) zu bewerten.

Analyse von lang andauernden Prozessen

Klassische Anwendungsmotivationen für die Erkennung und Verfolgung von mehreren Objekten kommen beispielsweise aus der Überwachungstechnik (Benfold und I. Reid 2011) oder aus der Fahrzeugtechnik (W. Choi u. a. 2013). Dabei handelt es sich in der Regel um die Aufgabe, innerhalb eines kurzen zeitlichen Rahmens, verdächtige Handlungen von Personen zu erkennen oder den Fahrweg von Fahrzeugen auf potentielle Kollisionen zu überwachen. Bei der semantischen Interpretation von Sportspielen stehen dagegen lange andauernde Prozesse mit einer Vielzahl von Ereignissen im Vordergrund. In vielen Anwendungsbereichen sind Analysen über ganze Spielphasen, Begegnungen oder sogar mehrere Spielzeiten von Interesse.

Anwendungsvielfalt im kommerziellen und wissenschaftlichen Bereich

Weltweit zeigen Menschen ein sehr großes Interesse daran, Sportereignisse mit Hilfe von medialen Produkten zu verfolgen. Die FIFA (*Fédération Internationale de Football Association*) schätzt, dass rund 3,2 Milliarden Zuschauer mindestens eine Spielminute der Fußball-Weltmeisterschaft 2014 in Brasilien verfolgt haben (Fédération Internationale de Football Association (FIFA) 2015a). Das globale Interesse ist jedoch nicht auf Fußball beschränkt und die Liste lässt sich beliebig fortführen mit Veranstaltungen und Sportarten, wie den olympischen Sommerspielen, American Football, Cricket, Tennis oder Radrennfahren.

Diese beeindruckenden Zahlen deuten auf einen gewinnbringenden Markt hin, auf dem Organisatoren, Sponsoren, Medien und Sportvereine agieren. Somit bietet sich eine Vielfalt an potentiellen kommerziellen und wissenschaftlichen Anwendungen. Jede dieser Anwendungen hat ihre speziellen Anforderungen und generiert dadurch bestimmte Herausforderungen für die Technik:

- **Datengenerierung:** Statistische Daten zu Sportereignissen werden in der Regel von Hand erstellt, wie beispielsweise bei der PERFORM Media Deutschland GmbH (PERFORM Media Deutschland GmbH 2016). Dabei ist das Verhältnis von Qualität zu Aufwand sehr gering. Automatische Systeme können hier zur Unterstützung der manuellen Aufzeichnungen sowie zu deren Qualitätskontrolle eingesetzt werden. Hier steht vor allem die reproduzierbare Generierung von präzisen Ergebnissen im Vordergrund. Dafür ist es wichtig, die Verfahren mit geeigneten Referenzdaten zu validieren.
- **Mediendienstleistungen:** Zuschauer sind es gewohnt, eine umfangreiche Sammlung an statistischen Daten, wie Ballbesitz oder Laufleistung, von medialen Anbietern geliefert zu bekommen. Folglich besteht ein Wettbewerb von Fernsehsendern

und anderen medialen Plattformen um immer attraktivere Berichterstattungen und Präsentationsmöglichkeiten, wie beispielsweise die freie Auswahl der Blickperspektive (ZDF 2016) oder die Darstellung mit virtueller Computergraphik (Virtually Live US, Inc. 2016). Hierbei werden die Möglichkeiten durch automatische Prozesse enorm erweitert. Eine wichtige Voraussetzung dafür ist die Lieferung der entsprechenden Ergebnisse in Echtzeit.

- **Leistungsanalyse:** Die gezielte Analyse und Bewertung der eigenen und gegnerischen Leistung ist in vielen Sportarten zum Standard geworden und hilft Athleten, Trainern und Vereinen, sich auf sportlicher Ebene zu verbessern und sich taktisch auf den Gegner einzustellen. Ebenso hilft eine solche Bewertung beim gezielten Abschluss von Spielertransfers. Automatische Systeme, die in kurzer Zeit große Datenmengen objektiv analysieren können, sind in der Lage den Aufwand zu reduzieren und sogar neue Möglichkeiten zu erschließen. Solche Systeme müssen robust gegenüber Änderungen der Umgebungsbedingungen sein, da Daten aus den verschiedensten Quellen herangezogen werden.

- **Sportwissenschaft:** Aus sportwissenschaftlicher Sicht steht unter anderem die Ermittlung von Leistungsindikatoren im Vordergrund. Dabei geht es um die Fragestellungen, in welchem Umfang individuelle Eigenschaften, statistische Kennzahlen und taktische Vorgehensweisen auf den Erfolg von Athleten und Mannschaften hinweisen und wie aus diesen Erkenntnissen neue Trainingsschwerpunkte und -methoden entwickelt werden können (Mackenzie und Cushion 2013). Automatische Systeme können hier nicht nur helfen neue Indikatoren zu ermitteln, sie ermöglichen zudem Studien in bisher unbekanntem Umfang, die statistisch belastbare Ergebnisse liefern. Daher ist es wichtig, dass solche Systeme vollautomatisch funktionieren und manuelle Eingriffe nicht notwendig sind.

- **Amateurbereich:** Im Amateurbereich lässt die finanzielle Lage meist den Einsatz von aufwändigen Kamerasystemen nicht zu. Dennoch kann die Analyse von Aufzeichnungen für die Trainings- und Wettkampfverbesserung oder für das Vereinsmarketing nützlich sein. Automatische Auswerteverfahren die dabei hilfreich sein können, müssen mit stark wechselnden Umgebungsbedingungen und einer schlechten Bildqualität von Camcordern oder Smartphones umgehen können.

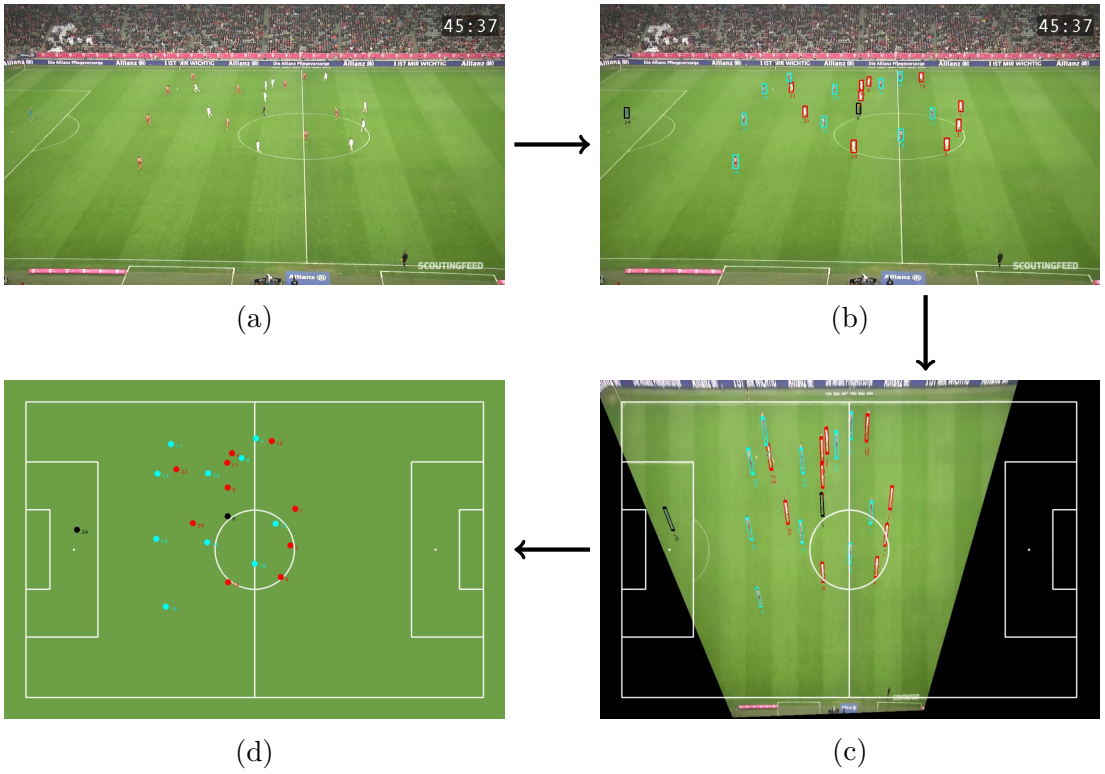


Abbildung 1.2: Ein Beispiel für die Umwandlung des Originalbilds (a) in berechnete Spielerpositionen im Bild (b), die Transformation des Bildes mit überlagerten Spielfeldmarkierungen (c) und die transformierten Spielerpositionen im Spielfeldkoordinatensystem (d). Originalbild aus (DFL 2014c)

1.3 Problemstellung

Gegenstand dieser Arbeit ist es, robuste, echtzeitfähige Methoden und Verfahren zu entwickeln und zu evaluieren, die in monokularen Aufnahmen von sogenannten Schwenk-Neige-Zoom-Kameras (*Pan-Tilt-Zoom-Cameras*) automatisch die relevanten Objekte erkennen und ihre Position in 2D (Bildkoordinaten) und 3D (Spielfeldkoordinaten) bestimmen. Dabei liegt der Fokus auf Videosequenzen von Sportspielen, insbesondere von Fußballbegegnungen. In Abbildung 1.2 ist exemplarisch die Umwandlung von den Originalbilddaten in 3D-Spielerpositionen dargestellt. In diesem Zusammenhang sei erwähnt, dass im Rahmen dieser Arbei die Position und Größe der Spieler (in Pixel) im Bildkoordinatensystem als 2D-Spielerpositionen bezeichnet werden. In Abgrenzung dazu, werden die Positionen der Spieler (in Meter) im Feldkoordinatensystem als 3D-Spielerpositionen bezeichnet, auch wenn diese genau genommen 2D-Koordinaten sind, da sie keine Höheninformation (Z-Achse) enthalten.

Als relevante Objekte bei Sportspielen werden die Objekte betrachtet, die für die spätere semantische Aufbereitung besonders interessant sind. Dies sind in erster Linie Spieler, Ball und Schiedsrichter. Allerdings können auch andere Objekte interessant sein, wie

beispielsweise die Körperteile und Posen der Spieler zur Unterscheidung von Kopfball und Schuss oder zur virtuellen 3D-Rekonstruktion.

Die Fokussierung auf monokulare Schwenk-Neige-Zoom-Kameras motiviert sich durch zwei Aspekte. Zum einen ermöglichen die einfache Erzeugung und die breite Verfügbarkeit (beispielsweise im Fernsehen oder Internet) solcher Aufnahmen das Erstellen von umfangreichen Test- und Trainingsdatenbanken und zum anderen ist diese Problemstellung im Allgemeinen und auch im Kontext von Sportspielen sowohl in kommerzieller als auch in wissenschaftlicher Hinsicht noch nicht befriedigend gelöst (siehe Abschnitt 1.5).

Bei der Robustheit der vorgestellten Verfahren geht es in erster Linie darum, mit oben genannten Toleranzen und Problematiken, wie schwierigen Wetterbedingungen oder schlechten Aufnahmequalitäten sowie Ereignissen, die von der erwarteten Norm abweichen, umgehen zu können. Dazu gehört auch eine geeignete Wahl der Parameter des Systems, um eine Sensitivität gegenüber Schwankungen der Eingabe zu vermeiden.

In manchen Anwendungen sind semantische Aufbereitungen in Echtzeit notwendig, beispielsweise für aktuelle Statistiken zur Spielzeit oder Taktik- und Leistungsanalysen in der Halbzeitpause. Daher ist es wichtig, Verfahren zu entwickeln, die eine sequentielle Dateneingabe unterstützen und potentiell echtzeitfähig sind. Das heißt, dass es mit den Methoden prinzipiell möglich sein muss, mit geeigneter Hardware, die Resultate in Echtzeit zu berechnen, beispielsweise durch eine hohe Parallelisierbarkeit.

Automatische Verfahren liefern zufriedenstellende Ergebnisse auch ohne manuelle Interaktion und sind in der Lage sich selbst zu initialisieren. Ein hoher Grad an Automatisierung ist ebenfalls eine wesentliche Voraussetzung für die Echtzeitfähigkeit.

Der Fokus auf die Sportart Fußball liegt nahe, da durch deren Popularität die Menge an frei erhältlichem Videomaterial und das Interesse an sportwissenschaftlichen und kommerziellen Anwendungen am größten sind. Zudem bietet die weite Verbreitung von professionellen Verfolgungssystemen mit aufwendigen Kamerainstallationen in den großen europäischen Ligen eine sehr gute Verfügbarkeit von Daten zur Evaluierung.

1.4 Lösungsansätze und wissenschaftlicher Beitrag

Um die obengenannten Herausforderungen zu bewältigen und die gesteckten Ziele zu erreichen, stellt diese Arbeit geeignete Verfahren vor und leistet damit folgende wissenschaftliche Beiträge:

- **Kombination aussagekräftiger Merkmale:** Eine sogenannte Konfidenzkarte (*Confidence Map*) ist der wesentliche Grundstein für die leistungsfähige Erkennung und Verfolgung von Objekten, die robust gegenüber optischen Artefakten, morphologischen Veränderungen und Überdeckungen sind. Diese Konfidenzkarte berechnet sich aus einer neuartigen Kombination geeigneter Merkmale, die auf Farbe, Textur und räumlichen Zusammenhängen basiert.

- **Unüberwachte, farbbasierte Bestimmung der Teamoutfits:** Die Zuordnung von Spielern zu einer Mannschaft ist für nachfolgende Auswertungsschritte von entscheidender Bedeutung. Beispielsweise bei der Reidentifizierung von Spielern nach einer Überdeckung oder bei der Bestimmung des Ballbesitzes einer Mannschaft. Innerhalb einer Videosequenz ist a priori nicht bekannt, wie viele verschiedene Teamoutfits darin vorkommen. Es kann beispielsweise sein, dass Torhüter nicht zu sehen sind oder dass Trainer oder Kameramänner auf dem Spielfeld erscheinen. In Abschnitt 3.6 wird ein Verfahren vorgestellt, das mit einer kontextsensitiven Quantisierung und einer unüberwachten Clusteranalyse die Anzahl verschiedener Outfits bestimmt und aussagekräftige Farbtemplates generiert. Diese dienen nicht nur der Zuordnung der Spieler zu der richtigen Mannschaft, sondern auch zu einer robusteren Erkennung und Verfolgung.

- **Effiziente Optimierung in der Konfidenzkarte:** Um Objekte in aufeinanderfolgenden Bildern wiederzufinden, sucht ein effizientes Verfahren optimale Positionen in der Konfidenzkarte. Die Berechnung der Merkmale ist der Flaschenhals in diesem Verfahren und eine erschöpfende Suche in der Konfidenzkarte ist nicht praktikabel. In dieser Arbeit wird eine intelligente Kombination von Bergsteigeralgorithmus (*Hill Climbing*) (Russell und Norvig 2003) und Gradientenabstieg (Nocedal und Wright 2006a) vorgestellt, welche die Anzahl der berechneten Positionen in der Konfidenzkarte gering hält und dennoch befriedigende Ergebnisse liefert.

- **Automatische initiale Spielererkennung:** Basierend auf den oben genannten Merkmalen und Farbtemplates, werden die Spieler in einem iterativen Verfahren erkannt und den Mannschaften zugeordnet. Hervorzuheben ist hier die iterative Schätzung der Spielergrößen im Bild abhängig von der Position im Bild, die zu einer robusten Erkennung in den späteren Iterationen beiträgt.

- **Effiziente 2D-Online-Verfolgung für mehrere Objekte:** Alle Objekte in der Szene werden mit einem effizienten, parallelisierbaren Online-Verfahren verfolgt. Das Prinzip des Kalman-Filters sorgt hierbei für die Inkorporation des zeitlichen Kontexts und für glatt verlaufende Bewegungstrajektorien. Ein regelbasierter Ansatz sorgt dabei für plausible Entscheidungen und Identifizierungen, wenn Objekte die Szene verlassen und betreten oder durch eine Überdeckung für einige Zeit nicht

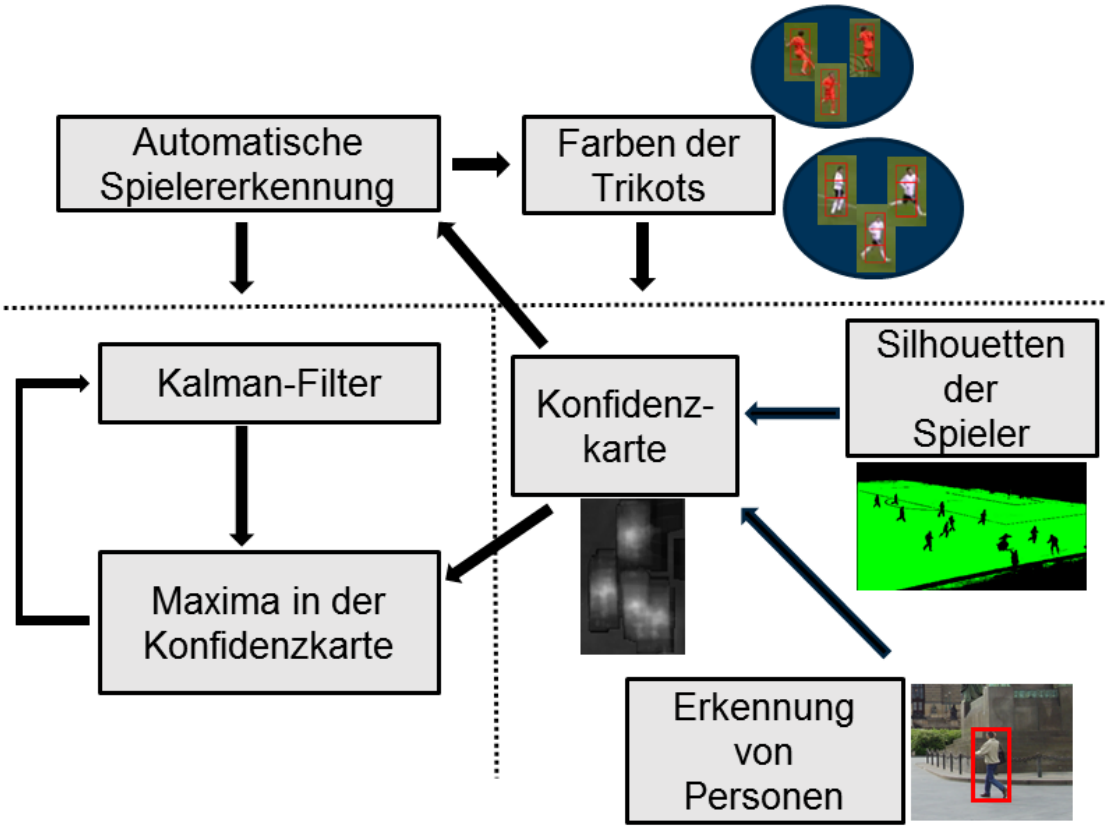


Abbildung 1.3: Eine Übersicht des Trackingverfahrens

sichtbar sind. Eine schematische Übersicht über das ganze Verfahren ist in Abbildung 1.3 dargestellt.

- **Effiziente Bereinigung der 3D-Trajektorien:** Nach Projektion der Spielerpositionen in das Koordinatensystem des Spielfelds werden die 3D-Trajektorien durch ein effizientes Verfahren bereinigt. Dabei werden Plausibilitätsannahmen ausgenutzt, wie beispielsweise physikalische Randbedingungen, die in der Bilddomäne nicht ohne Weiteres getroffen werden können.

- **Automatische Bestimmung von optimalen Parametern:** Die oben genannten Verfahren kommen nicht ohne Parameter aus, seien es Schrittweite, Maskengröße bei Filteroperationen oder der Grad der Quantisierung des Farbraums. Um ein robustes System zu entwickeln, sind Parameterkombinationen notwendig, welche die eingesetzten Methoden insensitiv gegenüber Änderungen in der Eingabe machen. Gleichzeitig verhält sich der Parameterraum exponentiell zu der Anzahl an verschiedenen Parameter. Die manuelle Ermittlung optimaler Parameter ist deswegen sehr zeitaufwändig und nur mit hohem Dokumentationsaufwand zu gewährleisten. Daher wird in dieser Arbeit ein zufallsbasiertes Verfahren vorgestellt,

das durch eine iterative Einschränkung des Parameterraums geeignete Parameterkombinationen ermittelt. Dabei wird auf einen repräsentativen Trainingsdatensatz zurückgegriffen, der im Rahmen dieser Arbeit erstellt wurde.

- **Umfangreiche Evaluierung der Verfahren mit einem neuen, repräsentativen Datensatz:** Für die Evaluierung von Systemen zur Verfolgung mehrerer Objekte gab es lange Zeit keine zentral gepflegte Methodik zum Vergleich der Leistung zweier Systeme. Häufig wurden in wissenschaftlichen Veröffentlichungen unpräzise definierte Kriterien genutzt. In dieser Arbeit werden gängige Metriken exakt definiert und neue Metriken eingeführt, die gezielt auf Anforderungen im Bereich der Sportspiele abzielen. Die vorgestellten Verfahren werden mittels repräsentativer Datensätzen anhand von sechs Sequenzen mit einer Gesamtdauer von mehr als 5 Minuten (2D) beziehungsweise vier Sequenzen mit einer Gesamtdauer von mehr als 180 Minuten (3D) ausgewertet und mit dem Stand der Forschung verglichen. Der Datensatz für die Evaluierung des 2D-Verfahrens wurde im Rahmen der vorliegenden Arbeit erstellt und kann für weitere wissenschaftliche Arbeiten genutzt werden.
- **Effizientes Verfahren zur Erkennung und Verfolgung des Balls:** Der Ball ist definitiv eines der entscheidenden Objekte in Sportspielen und die Kenntnis seiner Position für viele Anwendungen entscheidend. Die Bestimmung der Ballposition in monokularen Aufnahmen ist ein schwieriges Unterfangen, da der Ball ein verhältnismäßig kleines Objekt ist, hohe Geschwindigkeiten und Beschleunigungen aufweisen kann und häufig durch Spieler verdeckt ist. Dieser Herausforderung stellt sich diese Arbeit und präsentiert ein robustes Online-Verfahren, das dieses Problem zuverlässig löst. Hierbei kommen ähnliche Methoden wie bei der Spielerverfolgung zum Einsatz, die auf die speziellen Anforderungen bei der Erkennung und Verfolgung des Balls zugeschnitten sind. Die Leistungsfähigkeit des Verfahrens wird mit einer umfangreichen Evaluierung aufgezeigt.
- **Erkennung der Körperposen der Spieler:** Als eine Anwendung, die die generierten Spielerpositionen nutzt, wird die Erkennung der Spielerposen auf Basis von *Random Forest* Klassifikatoren und inverser Kinematik vorgestellt. Auch hier findet eine umfangreiche vergleichende Evaluierung statt.

1.5 Stand der Forschung

Obwohl im Bereich Erkennung und Verfolgung mehrerer Objekte, insbesondere im Bereich Fußgängerverfolgung, die Anzahl der Forschungsarbeiten und die dahinter stehende Forschungsgemeinde sehr groß sind (siehe unter anderem (Dollár, Wojek u. a.

2012; Fleuret u. a. 2008; Breitenstein u. a. 2011; Benfold und I. Reid 2011; Berclaz u. a. 2011; Izadinia u. a. 2012; Milan, Schindler u. a. 2015)), sind die Arbeiten im Bereich von Sportspielen und im Speziellen im Bereich von Fußball weniger zahlreich und weniger gebündelt. Wie es der Übersichtsartikel von D'Orazio und Leo (D'Orazio und Leo 2010) zeigt, gibt es zwar einige Veröffentlichungen, die sich mit dieser Thematik beschäftigen. Dennoch hebt sich das System, wie es in der vorliegenden Arbeit vorgestellt wird, in entscheidenden Punkten vom Stand der Forschung ab. Dies liegt schon alleine daran, dass sich nur wenige Arbeiten mit der vollständigen Kette von der Videoeingabe einer einzigen schwenk- und zoombaren Kamera bis hin zu den 3D-Trajektorien von Spieler und Ball auseinandersetzen.

Viele Ansätze basieren auf statischen Kameras, wie beispielsweise die Arbeiten von Kristan u. a. (Kristan u. a. 2005), Mazzeo u. a. (Mazzeo u. a. 2008), Itoh u. a. (Itoh u. a. 2012), Joo und Chellappa (Joo und Chellappa 2007) oder Schlipsing u. a. (Schlipsing u. a. 2013; Schlipsing 2014). Dies erleichtert die Segmentierung und Erkennung der Spieler enorm, da eine naive Hintergrundschätzung schon für befriedigende Ergebnisse sorgen kann. Zudem können durch eine initiale Kalibrierung der Kamera sehr einfach die Positionen der Spieler in Spielfeldkoordinaten bestimmt werden. Wenn zusätzlich die Annahme einer vorteilhaften Kameraposition getroffen wird, die beispielsweise zu einer Aufnahme aus der Vogelperspektive führt (Kristan u. a. 2005), können Schwierigkeiten wie die Überdeckung von Spielern deutlich abgemildert werden.

Beim Einsatz von mehreren Kameras kann zum einen das komplette Spielfeld mit einer ausreichenden Auflösung und ohne Weitwinkeleffekte abgedeckt werden, wie es beispielsweise Schlipsing u. a. (Schlipsing u. a. 2013) und Figueroa u. a. (Figueroa u. a. 2006b) vorschlagen. Zum anderen können, wie in der Arbeit von Nillius u. a. (Sullivan und Carlsson 2006; Nillius u. a. 2006), überlappende Blickfelder zur Auflösung von Überdeckungen oder gar zur Stereo-Rekonstruktion der Szene genutzt werden. Diese Tatsache nutzen auch die großen kommerziellen Anbieter wie Prozone Sports (Prozone Sports 2016) oder ChyronHego Corporation (ChyronHego Corporation 2016), die hochwertige Positionsdaten von Fußballspielen liefern. Dafür greifen sie jedoch auf aufwendige Kamerainstallationen zurück, die aufgrund der hohen Kosten nur für professionelle Vereine der oberen Ligen ökonomisch sinnvoll sind. Zudem sind die genannten Verfahren nicht ohne Weiteres geeignet, bei Aufnahmen mit einer schwenkenden Kamera sinnvolle Ergebnisse zu liefern.

Wie zu Beginn erwähnt, ist es für eine Bereitstellung von Auswertungsdaten zur Spielzeit erforderlich, dass das System in Echtzeit arbeitet. Dennoch werden in vielen Veröffentlichungen Verfahren vorgestellt, die nicht Online-fähig sind, wie zum Beispiel in den Arbeiten von J. Liu u. a. (J. Liu u. a. 2009), Miura und Kubo (Miura und Kubo 2008),

Nillius u. a. (Nillius u. a. 2006) oder Pallavi u. a. (Pallavi u. a. 2008). Meist werden zum Erstellen von Trajektorien graphentheoretische Ansätze präsentiert, die als Eingabe die Spielererkennungen in jedem Einzelbild einer Videosequenz benötigen. Das bedeutet, dass die Videosequenz zunächst komplett vorliegen muss. Bei ganzen Halbzeiten führt diese sogenannte *Batch*-Prozessierung zu sehr mächtigen Graphen. Diese müssten auf intelligente Weise in kleine Subgraphen unterteilt werden, wie es beispielsweise in der Arbeit von Berclaz u. a. (Berclaz u. a. 2011) vorgeschlagen wird.

Viele sinnvolle semantische Auswertungen können nur mit den Positionsdaten von Spieler und Ball im Feldkoordinatensystem durchgeführt werden. Dazu gehören zum Beispiel die Analyse der taktischen Aufstellung oder die Ermittlung der Laufleistung von Spielern. Zudem können die Spielertrajektorien mit solchen Koordinaten durch physikalische Annahmen bereinigt und verbessert werden. Zahlreiche wissenschaftliche Artikel beschäftigen sich dennoch ausschließlich mit der Extraktion von 2D-Bildpositionen, wie beispielsweise die Arbeiten von Joo und Chellappa (Joo und Chellappa 2007), J. Liu u. a. (J. Liu u. a. 2009), Nillius u. a. (Nillius u. a. 2006), Pallavi u. a. (Pallavi u. a. 2008) oder Sabirin u. a. (Sabirin u. a. 2015). Solche Systeme haben zunächst nur eine beschränkte Eignung für sinnvolle Anwendungen.

In den meisten Arbeiten werden die vorgeschlagenen Methoden zwar evaluiert, jedoch weisen diese Evaluierungen in der Regel nicht den wünschenswerten Umfang auf. So sind Dauer, Repräsentativität und Anzahl der genutzten Videosequenzen nicht aussagekräftig oder es wird gar auf synthetische Daten ausgewichen (Kim u. a. 2003). In den Arbeiten von Nillius u. a. (Nillius u. a. 2006) und Mazzeo u. a. (Mazzeo u. a. 2008) werden zwar Bilddaten mit einer Länge von circa 10 Minuten beziehungsweise 90 Minuten ausgewertet, diese Daten stammen aber aus nur einer Sequenz und weisen somit keine unterschiedlichen Rahmenbedingungen auf. Diesem Umstand begegnen die Arbeiten von Pallavi u. a. (Pallavi u. a. 2008) und Sabirin u. a. (Sabirin u. a. 2015), indem sie eine größere Anzahl (sechs beziehungsweise sieben) verschiedenartiger Sequenzen zur Evaluierung nutzen. Da die Sequenzen im Schnitt aber nur weniger als sieben Sekunden lang sind, ist die Aussagekraft der Auswertung ebenfalls beschränkt. Die eingesetzten Bewertungsmetriken beschränken sich häufig auf positiven Vorhersagewert und Sensitivität (J. Liu u. a. 2009; Pallavi u. a. 2008; Sabirin u. a. 2015) und ziehen die spezifische Aufgabenstellung der Verfolgung mehrerer Objekte und der Mannschaftszuordnung nicht in Betracht. Zudem werden die vorgestellten Verfahren selten mit ähnlichen Methoden des Stands der Forschung verglichen.

Einige Arbeiten beschäftigen sich mit Aufnahmen anderer Sportspiele wie Handball (Kristan u. a. 2005), Basketball (Lu u. a. 2011; Lu u. a. 2013), Eishockey (Okuma u. a. 2004; Lu u. a. 2009) oder American Football (T. Zhang u. a. 2012). In einzelnen Teilbereichen

können hier Parallelen gezogen werden. Wenn man jedoch das Gesamtsystem betrachtet, so sind die genannten Methoden, wenn überhaupt, nur durch umfangreiche Anpassungen an die Domäne Fußball nutzbar.

Am ähnlichsten zu den in dieser Arbeit vorgestellten Verfahren ist das System zur Spielerverfolgung von ASPOGAMO, welches von Beetz u. a. (Beetz u. a. 2007; Beetz u. a. 2009; Hoyningen-Huene 2011) beschrieben wurde. Das System kann sowohl mit mehreren statischen als auch mit einzelnen dynamischen Kameras umgehen. Die eingesetzten Merkmale sind dennoch deutlich weniger technisch ausgefeilt und stoßen vermutlich bei schwierigeren Verhältnissen schnell an ihre Grenzen. Weiterhin sind die meisten Verfahren auf manuelle Initialisierungen angewiesen und lassen eine robuste automatische Erkennung vermissen. Aufgrund der limitierten Evaluierung lässt sich darüber allerdings nur eine eingeschränkte Aussage treffen. In der Arbeit von Hoyningen-Huene (Hoyningen-Huene 2011) wurden die Verfahren mit Hilfe von Sequenzen von bewegten Kameras mit einer Länge von 45 Minuten und circa zwei Minuten aus zwei verschiedenen professionellen Begegnungen ausgewertet. Trotzdem ist die Repräsentativität des Datensatzes ausbaufähig und die angewandten Metriken sind in der Forschungsgemeinde nicht etabliert. Die Vergleichbarkeit der Ergebnisse mit anderen Verfahren ist somit kaum gegeben.

Neben den wissenschaftlichen Beiträgen zu den einzelnen Unterthematiken, auf die in den jeweiligen Kapiteln noch näher eingegangen wird, kann man zusammenfassend festhalten, dass ein System, wie es in dieser Arbeit vorgestellt und evaluiert wird, noch nicht in der einschlägigen Literatur behandelt wurde.

1.6 Inhaltsübersicht

Nach einer kurzen Einführung von Begrifflichkeiten und Notationen in Kapitel 2 werden in Kapitel 3 die genutzten Merkmale und angrenzenden Themen, wie die automatische Bestimmung der Teamfarben, die Erstellung der Konfidenzkarte und das Auffinden von Optima in dieser Karte, vorgestellt. In Kapitel 4 wird auf die Erkennung und Verfolgung der Spieler in der 2D-Bildebene sowie die Transformation und Verfeinerung der Trajektorien in Spielfeldkoordinaten eingegangen. Kapitel 5 erläutert die automatische Bestimmung der Parameter des Verfolgungsverfahrens. Die Ergebnisse einer umfangreiche Auswertung des Verfahrens für die Verfolgung der Spieler in 2D und 3D sind in Kapitel 6 dargestellt.

Ein Verfahren zur Erkennung und Verfolgung des Balls in der 2D-Bildebene wird in Kapitel 7 beschrieben.

Kapitel 8 beinhaltet eine kurze Übersicht über potentielle Anwendungen der vorgestellten Verfahren, wie beispielsweise die Quantifizierung von individuellem Ballbesitz oder die Rekonstruktion von Körperposen der Spieler.

Zuletzt wird in Kapitel 9 über Möglichkeiten, Einschränkungen und potentielle Weiterentwicklungen diskutiert.

Kapitel 2

Grundlagen und Definitionen

In dieser Arbeit werden essentielle Definitionen, Verfahren und mathematische Konzepte aus dem Bereich der digitalen Bild- und Signalverarbeitung als bekannt vorausgesetzt. Für detaillierte Informationen wird daher auf die gängigen Lehrbücher, wie beispielsweise von Steger u. a. (Steger u. a. 2008) oder Sonka u. a. (Sonka u. a. 2008), verwiesen. Dennoch werden für das bessere Verständnis einige Definitionen und Notationen eingeführt, die in der gängigen Literatur nicht üblich oder nicht einheitlich sind.

2.1 Bild und Bildregionen

2.1.1 Das Bild

Ein *Bild* ist eine Abbildung

$$I : R_I \to \mathbb{R}^{c_I}. \tag{2.1}$$

Dabei ist $R_I := \{0,\dots,W_I-1\}\times\{0,\dots,H_I-1\}$ eine rechteckige Teilmenge von $\mathbb{Z}^2$ sowie $W_I \in \mathbb{N}^*$ die Bildbreite, $H_I \in \mathbb{N}^*$ die Bildhöhe und $c_I \in \mathbb{N}^*$ die Anzahl der Farbkanäle von I. Im Bildkoordinatensystem läuft die x-Achse von links nach rechts und die y-Achse von oben nach unten. Somit kann der Bildwert eines Pixels mit den Koordinaten (x,y) mit $x \in \{0,\dots,W_I-1\}$ und $y \in \{0,\dots H_I-1\}$ als der Eintrag einer Matrix $I_{y,x} := I(x,y)$ interpretiert werden. In einer Videosequenz wird das Bild zum Zeitpunkt $t \in \mathbb{N}$ als I^t bezeichnet.

Eine *Bildregion* ist eine beliebige Teilmenge von $\mathbb{Z}^2$, die in der Regel vollständig in R_I enthalten ist. Im folgenden Absatz wird eine Form der Bildregion vorgestellt, die im Bereich der Objekterkennung und -verfolgung eine wichtige Rolle spielt: die Bounding-Box.

2.1.2 Die Bounding-Box

Das Ziel von Objekterkennung und -verfolgung ist es, Objekte im Bild zu erkennen und über die Zeit hinweg zu verfolgen. Im Kontext der vorliegenden Arbeit wird häufig der allgemeine Begriff Objekt benutzt, wobei es sich dabei in der Regel um Fußballspieler oder den Fußball handeln wird. Die Position eines Objektes i in einem Bild I wird repräsentiert durch eine Bildregion als achsenparalleles minimal umschließendes Rechteck

$$B(x_i, y_i, w_i, h_i) := \{(x,y) \in \mathbb{Z}^2 \mid (x_i \leq x < x_i + w_i) \wedge (y_i \leq y < y_i + h_i)\}, \tag{2.2}$$

wobei $(x_i, y_i) \in \mathbb{Z}^2$ die linke obere Ecke, $w_i \in \mathbb{N}^*$ die Breite und $h_i \in \mathbb{N}^*$ die Höhe des Rechtecks darstellen. Das Rechteck umfasst die Ausmaße des beschriebenen Objekts im Bild. Im Sinne der Übersicht, wird B_i anstelle von $B(x_i, y_i, w_i, h_i)$ verwendet. In den folgenden Kapiteln wird ein achsenparalleles minimal umschließendes Rechteck als *Bounding-Box* bezeichnet.

Oft ist es wichtig zu überprüfen, inwieweit die Position zweier Objekte übereinstimmt (beispielsweise bei der Evaluierung von Ergebnissen, siehe Kapitel 6). Daher wird das Überdeckungsmaß für zwei Objekte, repräsentiert durch die Bounding-Boxen B_1 und B_2, über den Jaccard-Koeffizient (Jaccard 1912; Everingham u. a. 2010) wie folgt definiert:

$$o_J(B_1, B_2) := \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|} \tag{2.3}$$

mit $o_J(B_1, B_2) \in [0; 1]$.

2.2 Identität, Label und Qualitätsmaß

Jedem Objekt i wird eine *Identitätsnummer* id_i zugeordnet. Diese ermöglicht es, eine *Trajektorie* für das Objekt zu ermitteln, das heißt eine zeitliche (meist zusammenhängende) Abfolge von Positionen.

Die verfolgten Objekte werden in Klassen unterteilt. In dieser Arbeit werden, neben der Klasse Ball, sogenannte *Outfitklassen* behandelt. Unter einem *Outfit* wird dabei die farbliche Erscheinung einer Objektklasse verstanden, also in der Regel das Aussehen der Trikots einer Mannschaft. Bei einem Fußballspiel gibt es normalerweise fünf Outfitklassen auf dem Spielfeld: Feldspieler von Mannschaft 1, Feldspieler von Mannschaft 2, Torwart von Mannschaft 1, Torwart von Mannschaft 2 und die Schiedsrichter. Zusätzlich kann es gelegentlich vorkommen, dass noch andere Outfits in einer Videosequenz auf dem Spielfeld zu sehen sind, beispielsweise ein Trainer, der direkt an der Außenlinie

steht. Aus diesem Grund wird jedem Spieler ein Klassenlabel lbl_i zugeordnet. In der Regel definiert das Label die Zugehörigkeit zu einer Outfitklasse (also Mannschaft).

Die Positionen der Objekte werden meist durch automatisierte Messungen bestimmt. Durch Rauschen, Bildartefakte und anderen Störungen sind solche Messungen nicht immer verlässlich. Aus diesem Grund wird der Position beziehungsweise Bounding-Box ein Qualitätsmaß zugeordnet, dass abhängig vom Bild I^t ist: $q_i^{I^t} := q(B_i, I^t) \in [0; 1]$. Dieses Maß trifft eine Aussage über den Grad der Sicherheit, mit der das Objekt an dieser Position geschätzt wird.

2.3 Template

Unter einem *Template* wird eine Vorlage verstanden, die das Aussehen eines Objekts oder einer Klasse von Objekten beschreibt. Diese Vorlage wird genutzt, um gesuchte Objekte in einem Bild zu finden. So wird zum Beispiel beim klassischen *Template Matching* das Template als separates (meist kleines) Bild gespeichert. Das Vorkommen dieses Vorlagenbilds wird dann an verschiedenen Positionen und Skalierungen im eigentlichen Bild über geeignete Vergleichsmaße untersucht (siehe beispielsweise (Steger u. a. 2008, S. 211)). Ein anderer Ansatz, der in der vorliegenden Arbeit verfolgt wird, ist es, Templates über die Farbverteilung innerhalb der gesuchten Objekte zu modellieren (siehe Abschnitt 3.6.4).

2.4 Objekterkennung

Ziel der Objekterkennung ist es, Position und Ausdehnung aller Objekte einer Klasse (wie Personen oder Fußbälle) in einem Bild zu bestimmen. Bei der Erkennung mit Hilfe eines Klassifikators wird dabei in der Regel ein Suchfenster fester Größe über das Bild geschoben (*Sliding Window*). Innerhalb dieses Suchfensters wird ein Merkmalsvektor berechnet, beispielsweise von Haar-ähnlichen Merkmalen (*Haar-like Features* (Viola und Jones 2001)) oder HOG-Merkmalen (*Histogram of Oriented Gradients* (Dalal und Triggs 2005)). Anhand dieses Vektors entscheidet ein Klassifikator, ob sich an der Position des Suchfensters ein gesuchtes Objekt mit der Ausdehnung des Suchfensters befindet. Um verschiedene Objektgrößen erkennen zu können, wird meist die Suche in einer sogenannten Bildpyramide auf unterschiedlichen Bildskalierungen durchgeführt (wobei die Größe des Suchfensters gleich bleibt). Die Antwort des Klassifikators ist in der Regel binär, also „Objekt vorhanden“ oder „Objekt nicht vorhanden“. Allerdings wird häufig auch der nicht-binäre, interne Entscheidungswert (*Decision Value*) des Klassifikators für eine differenzierte Analyse des Ergebnisses genutzt.

2.5 Aufnahmemodalitäten

In dieser Arbeit werden unterschiedlichste Videosequenzen von Fußballspielen untersucht. Gemeinsam haben diese Sequenzen, dass sie aus einer totalen Kameraeinstellung aufgezeichnet wurden, das heißt, das Spielfeld nimmt einen Großteil des Bildes ein und es sind zumeist mehrere Spieler (> 5) im Bild zu sehen. Nahaufnahmen, Aufnahmen von Hintertorkameras und Ähnliches werden nicht betrachtet. Zudem zeichnen sich die ausgewählten Sequenzen durch eine heterogene Verteilung von Quellen (Fernsehsender, Amateuraufnahmen, DFL-Scoutingfeeds), Wettbewerben (International, National, Jugend, Frauen) und Aufnahmequalität und -format aus.

2.6 Das Spielfeld

In dieser Arbeit wird angenommen, dass das Spielfeld eben ist und sich alle Spieler auf derselben Ebene befinden. Das entspricht nicht unbedingt der Realität, da zur besseren Entwässerung Fußballfelder häufig mit einem leichten Gefälle zum Rand hin konstruiert werden (siehe (Puhalla u. a. 2010)). Da ein zu starkes Gefälle die Rolleigenschaften des Balls beeinflussen würde, wird die Abweichung zu einer Ebene als vernachlässigbar angenommen. Dies ist auch im Einklang mit der *Ground Plane Assumption* von Beetz u. a. (Beetz u. a. 2007).

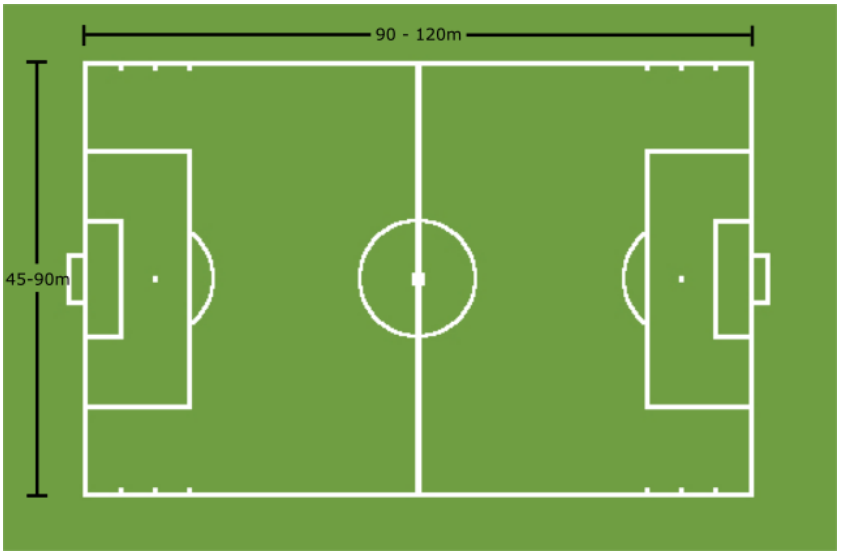


Abbildung 2.1: Mögliche Größen des Spielfelds

Die Breite und Länge eines Spielfelds haben nach den offiziellen Regeln einen große Toleranzbereich, wie in Abbildung 2.1 dargestellt (siehe (Fédération Internationale de Football Association (FIFA) 2015b)). In dieser Arbeit wird davon ausgegangen, dass das Spielfeld den gängigen Maßen der Profiklassen mit einer Länge von 105 Metern und

einer Breite von 68 Metern entspricht. Die Abmessungen von Strafraum, Torraum und Mittelkreis sind fest von der FIFA vorgegeben. Das verwendete Weltkoordinatensystem hat seinen Ursprung am Mittelpunkt des Spielfelds, wobei die X-Achse vom (im Bild) linken Tor zum anderen Tor und die Y-Achse von der (im Bild) unteren Seitenauslinie zur gegenüberliegenden Seitenauslinie verlaufen.

Teil II

Erkennung und Verfolgung von Spielern

Kapitel 3

Merkmale für die Spielerverfolgung

Die Inhalte dieses Kapitels basieren zu Teilen auf folgender Veröffentlichung:

Herrmann, M., Hoernig, M. und Radig, B. (2014). „Online Multi-player Tracking in Monocular Soccer Videos“. In: *AASRI Procedia* 8, S. 30–37.

3.1 Einleitung

Für ein Verfahren, das Objekte in Videosequenzen erkennen und verfolgen soll, ist es notwendig, geeignete Merkmale aus der Bildinformation zu extrahieren und einzusetzen. Dieses Kapitel stellt Merkmale vor, die aus Einzelbildern extrahiert werden können und besonders für die Erkennung und Verfolgung von Fußballspielern in Videosequenzen geeignet sind (siehe Kapitel 4):

- **Gras- bzw. Vordergrundsegmentierung:** Die Farbe des Grases ist in der Regel die dominante Farbe in Videoaufzeichnungen von Fußballbegegnungen und wird genutzt, um zum einen die Ausmaße des Spielfeldes im Bild zu erkennen und zum anderen eine Unterscheidung zwischen Gras und Vordergrund (Spieler, Linien, Ball, etc.) zu treffen. Diese Arbeit nutzt dazu das Verfahren von Hoernig u. a. (Hoernig u. a. 2015), welches eine effiziente Bestimmung der Spielfeldhülle ermöglicht sowie eine Trennung von Gras und Vordergrund durchführt, die robust gegenüber Helligkeits- und Farbtonschwankungen im Gras ist.
- **Schätzung der Größe der Spieler (im Bild):** Die Größe der Spieler ist aufgrund der perspektivischen Abbildung annähernd linear entlang der y-Achse. Das

ermöglicht eine Schätzung der Spielergrößen, ohne auf ein komplexes Kameramodell mit internen und externen Parameter zurückzugreifen. Die Schätzung der Größe (in Abhängigkeit von den Bildkoordinaten) ist ein entscheidender Faktor bei der robusten und effizienten Erkennung von Spielern und bei der Abgrenzung von anderen Vordergrundobjekten.

- **Extraktion der Spielersilhouetten:** Das Ergebnis der Vordergrundsegmentierung ist im Prinzip eine binäre Maske des Vordergrunds. Um daraus die Silhouetten der Spieler zu bestimmen, ist es notwendig die Regionen der Spieler von anderen Vordergrundregionen wie Feldlinien, Toren oder Bandenwerbung abzugrenzen. Dabei kommen die geschätzte Größe, morphologische Eigenschaften sowie lokale Schwellwertoperationen zum Einsatz.

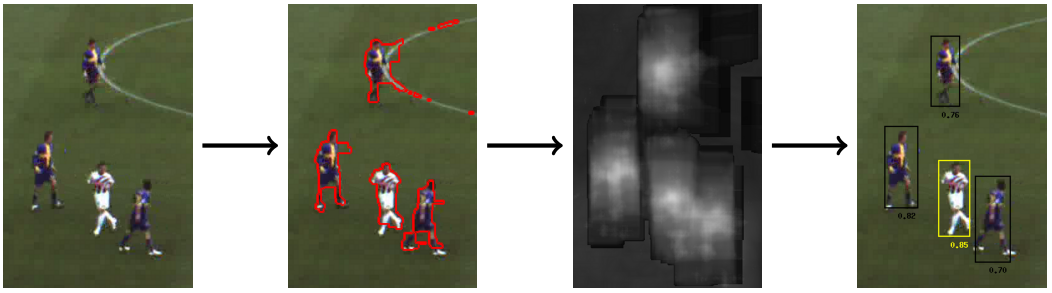
- **Farbtemplates:** Die Trikotfarbe ist regeltechnisch das entscheidende Merkmal, um Spieler zweier Mannschaften optisch auseinanderzuhalten (Fédération Internationale de Football Association (FIFA) 2015b, S. 22). Zwar gibt es unter Einbezug der Schiedsrichter prinzipiell fünf verschiedene Outfits auf dem Platz, allerdings ist für eine Videosequenz im Vorhinein die genaue Anzahl nicht bekannt. Es können weniger als fünf sein, wenn die Kamera zum Beispiel nicht in den Bereich der Torhüter schwenkt. Genauso können es aber auch mehr als fünf sein, wenn sich Trainer, Kameramänner oder Zuschauer am Spielfeldrand befinden. Diese Personen am Spielfeldrand sind semantisch nicht relevant, ihre äußere Erscheinung muss aber erfasst werden, um sie von den Spielern und Schiedsrichtern zu unterscheiden. Das vorgestellte Verfahren kann anhand weniger Einzelbilder den Farbraum durch eine Clusteranalyse auf Basis des *k-means*-Algorithmus (Lloyd 1982) geeignet quantisieren und die dominanten Farben der einzelnen Outfits extrahieren. Somit können Outfits durch aussagekräftige Histogramme mit wenigen Einträgen repräsentiert und effizient berechnet werden. Durch Aufteilung der Spieler in semantische Regionen (Kopf, Oberkörper, Unterkörper) werden dabei räumliche Abhängigkeiten beachtet. Da die Wahl der Trikotfarben für die menschliche Wahrnehmung optimiert ist, werden die Histogramme im CIE L*a*b-Farbraum (Plataniotis und Venetsanopoulos 2000, S. 35) berechnet. Mit einer weiteren Clusteranalyse auf Basis des *k-means*-Algorithmus (Lloyd 1982) werden die berechneten Histogramme, die jeweils einen Spieler repräsentieren, in die verschiedenen Outfitklassen eingeteilt. Dabei ist die Anzahl (also das k) nicht bekannt und wird mit einer automatischen iterativen Methode bestimmt.

- **Konfidenzkarte:** Mit Hilfe von Spielersilhouetten und Farbtemplates wird für jedes Einzelbild eine sogenannte Konfidenzkarte (*Confidence Map*) definiert. Diese

Karte ist ein Bild, das angibt, mit welcher Wahrscheinlichkeit sich an einer Position ein Spieler befindet. Die Konfidenzkarte setzt sich als Summe verschiedener aussagekräftiger Einzelmerkmale zusammen:

- **Silhouettenbasierte Konfidenz:** Die silhouettenbasierten Merkmale bewerten für jede beliebige Position im Bild das Verhältnis der (geschätzten) Größe eines Spielers und der Fläche seiner Silhouette. Damit werden Positionen im Bild höher bewertet, an denen eine Silhouette in Spielergröße und -form aufzufinden ist.
- **Überlappungsbasierte Konfidenz:** Dieses Merkmal bestraft starke Überlappungen zweier (oder mehrerer) Spieler. Damit wird der Tatsache Rechnung getragen, dass sich Spieler zwar sehr häufig überlappen, aber sehr selten komplett überdecken. Dieses Merkmal verhindert in erster Linie, dass die Verfolgung eines überdeckten Spielers auf einen dominanten Spieler im Vordergrund überspringt und dieser dann doppelt verfolgt wird (siehe Abbildung 3.8).
- **Farbbasierte Konfidenz:** Die Übereinstimmung mit einem vorgegebenen Farbtemplate (Histogramm) wird durch dieses Merkmal bewertet. Es ist zum einen wichtig für die Unterscheidung von Spielern unterschiedlicher Mannschaften sowie zur Abgrenzung von fälschlicherweise segmentierten Grasflächen, Feldlinien und ähnlichem.
- **Texturbasierte Konfidenz:** Dieses Merkmal basiert auf einem Personendetektor, der mit sogenannten HOG-Merkmalen (*Histogram Of Oriented Gradients* (Dalal und Triggs 2005)) arbeitet. Dieser Detektor führt eine binäre Klassifizierung auf Bildregionen durch. Er wurde mit Trainingsbildern von Fußgängern eingelernt und bringt somit eine allgemeingültige Komponente in die Konfidenzkarte, die unabhängig von der Domäne und äußeren Bedingungen ist.
- **Vorhersagenbasierte Konfidenz:** Die Abweichungen von der Vorhersage einer stochastischen Filtermethode (Kalman-Filter (Kálmán 1960)) werden durch dieses Merkmal bestraft. Ihm liegt implizit die Annahme zu Grunde, dass die Spieler im Bild eine glatte Bewegungstrajektorie aufweisen, die keine plötzlichen Sprünge enthält.

- **Suche lokaler Maxima in der Konfidenzkarte:** Ausgehend von einer Startposition (beispielsweise die letzte Position eines Spielers im vorherigen Einzelbild) wird in einem effizienten hybriden Ansatz das nächstgelegene lokale Maximum gesucht (beispielsweise als Messergebnis für die Spielerposition im aktuellen Einzelbild).



(a) (b) (c) (d)

ABBILDUNG 3.1: Übersicht über das Vorgehen bei der Merkmalsberechnung: (a) Originalbild, (b) Vordergrundsegmentierung, (c) Konfidenzkarte und (d) Spielererkennung / -verfolgung. Originalbild aus (D'Orazio u. a. 2009a).

Die in diesem Kapitel vorgestellten Merkmale sind die Grundlage für die eigentliche Erkennung und Verfolgung von Spielern, wie sie in Kapitel 4 vorgestellt werden. Ein Beispielszenario für das gesamte Vorgehens ist in Abbildung 3.1 dargestellt.

3.2 Stand der Forschung

Gras- bzw. Vordergrundsegmentierung

Die Entfernung des Hintergrunds (*Background Subtraction*), also die Trennung von beweglichen (Vordergrund) und unbeweglichen Objekten (Hintergrund) ist ein gängiges Mittel bei der Objektverfolgung (Stauffer und Grimson 1999). Meist sind solche Techniken für statische Kameras ausgelegt und lassen sich nicht ohne Weiteres auf schwenkende Kameras übertragen. Auf dem Fußballfeld ist die einheitliche Grasfarbe ein entscheidender Faktor bei der Unterscheidung von Vorder- und Hintergrundobjekten. Dennoch sind zahlreiche Schwierigkeiten zu beachten, wie etwa deutliche Helligkeitsunterschiede auf dem Feld. Dies führt zu Problemen bei Ansätzen, die sich auf eine Grasfarbe limitieren, sei sie manuell vorgegeben (Utsumi u. a. 2002; Tong u. a. 2004; Pallavi u. a. 2008) oder automatisch bestimmt (Gedikli 2009). Auch die Suche nach globalen Maxima im Histogramm (Seo u. a. 1997; Ekin u. a. 2003; Yu u. a. 2003) geht das Risiko ein zu scheitern, wenn in bestimmten Situationen (zum Beispiel bei einem Eckball) das Spielfeld nicht mehr den größten Teil des Bildes einnimmt. Das in dieser Arbeit genutzte Verfahren ist in der Lage mit beweglichen Kameras umzugehen und grenzt sich damit deutlich von Ansätzen ab, die auf einer Hintergrundschätzung von statischen Kameras basieren (Figueroa u. a. 2006a; Marco u. a. 2013; Schlipsing 2014). Für eine detaillierte Betrachtung sei auf Hoernig u. a. (Hoernig u. a. 2015) verwiesen.

Schätzung der Spielergrößen

Das Wissen über die Größe der gesuchten Objekte in einem Bild kann helfen, diese mit höherer Robustheit zu erkennen und vor allem den Suchraum deutlich einzuschränken. Im Allgemeinen ist es dafür jedoch erforderlich, die Kameraparameter zu kennen. Für den Fall, dass sich alle relevanten Objekte auf einer Ebene befinden, gibt es Ansätze, die Kameraparameter teilweise oder ganz anhand von schon bereits erkannten Objekten schätzen, wie von Rodriguez u. a. (Rodriguez u. a. 2011) oder J. Liu u. a. (J. Liu u. a. 2011) vorgeschlagen. Aufgrund der vielen Freiheitsgrade werden dabei allerdings einschränkende Annahmen getroffen (wie das Wissen über den ungefähren Kamerawinkel) oder eine ausreichend große Menge an Objekten im Bild vorausgesetzt. Da beides bei Fußballaufnahmen nicht zwingend zutrifft, wird in dieser Arbeit ein ähnliches Verfahren angewendet, wie es von Rujikietgumjorn und Collins (Rujikietgumjorn und Collins 2013) vorgeschlagen wird. Rujikietgumjorn und Collins (Rujikietgumjorn und Collins 2013) schätzen die Größe von Personen im Bild einer statischen Kamera anhand der linearen Abhängigkeit der Größe zur y-Koordinate. Diese Abhängigkeit wird durch eine einmalige lineare Regression anhand mehrerer Personenerkennungen initial ermittelt. Dabei wird keine spezielle Ausreißerbehandlung durchgeführt. Im Gegensatz dazu behandelt der Ansatz dieser Arbeit Aufnahmen einer nicht-stationären Kamera. Das bedeutet, die Größen werden in jedem Einzelbild neu bestimmt, wobei die Annahme zugrunde liegt, dass sich die Größen von einem Bild zum nächsten nur geringfügig ändern. Des Weiteren werden bei der Regression Verfahren der robusten Statistik angewendet, da ansonsten die Gefahr besteht, dass bei wenigen Objekten eine falsche Erkennung die Größenschätzung empfindlich stört. Im Ansatz von J. Zhang u. a. (J. Zhang u. a. 2012) wird ebenfalls die geschätzte Personengröße bei der Fußgängerverfolgung genutzt, um falsch-positive Erkennungen zu verwerfen. Allerdings wird dabei das Bild in 8×8 Zellen aufgeteilt und für jede Zelle ein Konfidenzintervall für die Personengröße ermittelt. Bei einer hohen Auflösung oder einem sehr spitzen Kamerawinkel kann hier das Intervall sehr breit ausfallen und dessen Nutzen in Frage stellen. Zudem wird ebenfalls keine dedizierte Ausreißerbehandlung durchgeführt.

Bestimmung der Spielersilhouetten

Für eine robuste Erkennung und Verfolgung der Spieler ist die reine Trennung von Gras und Vordergrund nicht ausreichend. So werden nicht nur Feldlinien, Tore, Werbungen und Senderlogos in der Regel als Vordergrund erkannt. Häufig hat das Gras auch lokale Inhomogenitäten, die bei einer globalen Farbsegmentierung dazu führen können, dass große Bereiche des Grases als Vordergrund interpretiert werden. Es sind Nachbearbeitungsschritte notwendig, um die Vordergrundregionen zu identifizieren, die mit hoher Wahrscheinlichkeit Spielersilhouetten darstellen. Der hier vorgestellte Ansatz wird dem

gerecht, indem morphologische Annahmen und geschätzte Spielergrößen sowie lokale Farbinformationen in Betracht gezogen werden. Pallavi u. a. (Pallavi u. a. 2008) nutzen eine Hough-Transformation, um die Mittellinie zu erkennen und zu entfernen. Für die Bestimmung von Spielerregionen greifen die Autoren auf einen festen Schwellwert bezüglich der Pixelfläche zurück. Im Gegensatz zu der dynamischen Größenschätzung in der vorliegenden Arbeit, ist dieser Ansatz nicht geeignet für Videos mit unterschiedlicher Auflösung und mit Spielergrößen im Bild, die durch die perspektivische Abbildung stark unterschiedlich sind. Lokale Farbunterschiede werden von Pallavi u. a. gar nicht beachtet. Sabirin u. a. (Sabirin u. a. 2015) nutzen ein semi-automatisches Verfahren, bei dem sich überlappende Vorder- und Hintergrundobjekte manuell markiert werden und mit einer Wasserscheidentransformation getrennt werden. Übrig gebliebene Linien werden mit einer Hough-Transformation entfernt. Das manuelle Markieren von Objekten in einer Videosequenz stellt dabei einen sehr aufwendigen Prozess dar und ist für sinnvolle Anwendungen unpraktikabel. In der Arbeit von Schlipsing (Schlipsing 2014) wird zur Unterdrückung von Rauschen eine morphologische Opening-Operation auf der binären Vordergrundmaske durchgeführt. Da nur mit einer statischen Kamera gearbeitet wird, sind die Ergebnisse der Hintergrundsubtraktion weniger fehleranfällig. Im Ansatz von Hoyningen-Huene (Hoyningen-Huene 2011) werden die Spielfeldlinien in das Bild projiziert und durch eine intelligente Linienentfernung von den Spielerregionen getrennt. Dazu wird allerdings eine bekannte Transformation vom Spielfeld in das Bild vorausgesetzt, die in der vorliegenden Arbeit zunächst als nicht vorliegend angenommen wird.

Farbtemplates

Neben der Tatsache, dass das Regelwerk im Fußball eine farbliche Unterscheidung der einzelnen Akteure vorschreibt (Fédération Internationale de Football Association (FIFA) 2015b, S. 22), können Erkennung und Verfolgung von Objekten durch den Einsatz von farblichen Merkmalen robuster gestaltet werden. So zeigen Weijer u. a. (Weijer u. a. 2006) und Burghouts und Geusebroek (Burghouts und Geusebroek 2009) in ihren Arbeiten, dass mit Hilfe von spektraler Information (im Vergleich zur reinen Helligkeitsinformation) die Leistung von Verfahren zum Auffinden markanter Punkte im Bild signifikant verbessert werden kann. Zu einer ähnlichen Schlussfolgerung kommen die Experimente von Sebastian u. a. (Sebastian u. a. 2008) bei der Verfolgung von Objekten und von Kviatkovsky u. a. (Kviatkovsky u. a. 2013) bei der Wiedererkennung von Personen. Der Einsatz von Farbhistogrammen zur Repräsentation der optischen Erscheinung ist ein gängiges Mittel bei der Verfolgung von Objekten (Comaniciu und Meer 2002; Comaniciu u. a. 2003). Die Berücksichtigung von räumlicher Information bei der Berechnung von Histogrammen hingegen wird seltener genutzt, obwohl Birchfield und Rangarajan (Birchfield und Rangarajan 2005) zeigen konnten, dass damit die Ergebnisse der

Objektverfolgung verbessert werden können. Sie augmentieren jede Histogrammklasse (*Bin*) mit räumlichen Informationen wie Schwerpunkt und Kovarianz. Das Kontextwissen, dass Spielertrikots aus Hemd, Hose und Stutzen bestehen, kann damit allerdings nicht ausgenutzt werden. Der hier vorgestellte Ansatz stellt daher eine Erweiterung der Idee von Beetz u. a. (Beetz u. a. 2006; Beetz u. a. 2007) dar, die Farbhistogramme aus verschiedenen Körperregionen eines Spielers zu kombinieren.

Die Quantisierung des Farbraums ist notwendig, um dünn besiedelte Histogramme mit wenig Aussagekraft zu vermeiden. Dabei wird der Farbraum von mehreren Millionen auf wenige dominante Farben reduziert und jede der ursprünglichen Farben diesen sogenannten Repräsentanten zugeordnet. Eine uniforme Quantisierung scheidet dafür aus, da sie nicht berücksichtigt, dass in den Trikots insgesamt nur wenige dominante Farben vorkommen. Sabirin u. a. (Sabirin u. a. 2015) nutzen einen solchen Ansatz für die Berechnung von Farbhistogrammen bei der Verfolgung von Fußballspielern. Dabei werden allerdings nur der Farbton (*Hue*) und die Farbsättigung (*Saturation*) des HSV-Farbraums berücksichtigt und die Helligkeit komplett ignoriert. Umgekehrt würde die Wahl der häufigsten Farben (*Popularity Algorithm* (Heckbert 1982)) dazu führen, dass leicht unterschiedliche Farbtöne der großen Flächen am Oberkörper dominieren. Andere gängige Ansätze zur Farbquantisierung wie die sogenannten *Octrees* (Gervautz und Purgathofer 1988) oder das *Median Cut*-Verfahren (Heckbert 1982) teilen den Raum in rechteckige Bereiche auf. Dies entspricht im Allgemeinen nicht der tatsächlichen Verteilung der dominanten Farben. Der in dieser Arbeit verwendete *k-means*-Algorithmus (Lloyd 1982) unterliegt nicht diesen Einschränkungen, da er den Raum in Voronoi-Regionen zerlegt und auch Cluster in weniger besetzten Teilen des Raums finden kann.

Sind die Histogramme für jede erkannte Person auf dem Spielfeld ermittelt, werden diese in verschiedene Outfitklassen eingeteilt. Wie schon erwähnt, ist dabei die Anzahl der Klassen unbekannt. Allerdings benötigen die meisten bekannten Clusterverfahren (Xu und Wunsch 2005) als Eingabe die Anzahl der zu bestimmenden Cluster (Klassen). Zwar gibt es Ansätze bei denen die Anzahl der Klassen vom Verfahren bestimmt wird, wie zum Beispiel die *Growing Neural Gas* Methode (Fritzke 1995; Andreakis u. a. 2009) oder diverse Erweiterungen des *k-means*-Algorithmus wie *X-means* (Pelleg und Moore 2000), *G-means* (Hamerly und Elkan 2004) und *PG-means* (Feng und Hamerly 2007). Diese Verfahren werden allerdings meist mit synthetischen Datensätzen evaluiert und tendieren in der Praxis zur Überschätzung der tatsächlichen Clusteranzahl. Zudem sind auch diese Verfahren nicht frei von Parametern und häufig ist die Semantik dieser Parameter schwierig zu interpretieren. Der in dieser Arbeit vorgestellte Ansatz ist ein *k-means*-Clustering mit einer iterativen Durchführung, beginnend mit einem kleinem k. In jeder Iteration wird dabei das k inkrementiert und das Abbruchkriterium basiert auf einem einzigen Parameter, der einfach zu interpretieren ist.

Die automatische Bestimmung der Mannschaftsfarben ist keine triviale Aufgabe, weswegen bei vielen vorgestellten Verfahren mir einer manuellen Initialisierung gearbeitet wird, wie bei den Arbeiten von Beetz u. a. (Beetz u. a. 2007) oder von Miura und Kubo (Miura und Kubo 2008). Der Ansatz von Figueroa u. a. (Figueroa u. a. 2006b) nutzt ausschließlich die Helligkeitsinformation zur Unterscheidung der beiden Mannschaften. Dies kann im Allgemeinen zu Schwierigkeiten führen, wenn beide Mannschaften ähnlich helle oder dunkle Trikots haben, die sich aber in Farbton und -sättigung sehr gut unterscheiden. In der Arbeit von Gerke u. a. (Gerke u. a. 2013) wird ebenfalls ein *k-means*-Clustering auf Farbmerkmalen durchgeführt, um die Spieler zweier Mannschaften zu unterscheiden. Sie wählen $k = 2$ und lassen somit Torhüter und Schiedsrichter außen vor. Sehr ähnlich zu der vorliegenden Arbeit ist der Ansatz von J. Liu u. a. (J. Liu u. a. 2009), der zur Bestimmung von repräsentativen Histogrammen für die Outfitklassen vorgestellt wurde. Dabei wird ebenfalls nur von genau drei sichtbaren Outfits (Feldspieler Mannschaft A, Feldspieler Mannschaft B und Schiedsrichter) ausgegangen. Dies kann zur Folge haben, dass die Torhüter nicht richtig erkannt und verfolgt werden.

Konfidenzkarte

Die Kombination von verschiedenartigen Merkmalen ist ein gängiger Ansatz bei der Erkennung und Verfolgung von Objekten. So kombinieren Breitenstein u. a. (Breitenstein u. a. 2011) die Konfidenzdichte (*Confidence Density*) eines texturbasierten Personendetektors mit der Konfidenz von online eingelernten Klassifikatoren auf Basis von Farbhistogrammen und einer vorhersagebasierten Konfidenz. Die Konfidenzdichte des Personendetektors wird schon im Vorhinein komplett für jedes Einzelbild berechnet und erzeugt so einen signifikanten Berechnungsaufwand, der im vorliegenden Ansatz vermieden wird. Viele Ansätze nutzen bei der Erkennung und Verfolgung von Personen eine sogenannte *Occupancy Map*, die angibt mit welcher Wahrscheinlichkeit sich an einer bestimmten Position im Raum eine Person befindet. Dazu benötigt man allerdings mehrere Kameras, wie beim Ansatz von Fleuret u. a. (Fleuret u. a. 2008; Berclaz u. a. 2011; Ben Shitrit u. a. 2011) oder zumindest eine kalibrierte Kamera, wie sie von Carr u. a. (Carr u. a. 2012) eingesetzt wird. Der hier vorgestellte Ansatz für die Berechnung der Konfidenzkarte basiert auf der Arbeit von Beetz u. a. (Beetz u. a. 2007). Dabei fließen neben Farbhistogrammen auch Kompaktheits- und Größenbedingungen der Spielerregionen ein, die vergleichbar mit den silhouettenbasierten Merkmalen der vorliegenden Arbeit sind und in ähnlicher Form bei Ansätzen mit statischer Kamera, wie zum Beispiel von Rujikietgumjorn und Collins (Rujikietgumjorn und Collins 2013), eingesetzt werden. Im Gegensatz zu den genannten Veröffentlichungen werden diese Merkmale in der vorliegenden Arbeit zusätzlich von den Erkennungen eines Personendetektors (Dalal und Triggs 2005) unterstützt. Damit werden Spieler robuster erkannt, insbesondere

auch in Situationen, in denen ihre Silhouette nicht vollständig mit Gras umgeben ist, sondern sich zum Beispiel mit einer Bandenwerbung überschneidet. Zudem wird mit der überlappungsbasierten Konfidenz der Tatsache Rechnung getragen, dass sich zwei Spieler nicht gleichzeitig an der gleichen Position im Spielfeld aufhalten können. Sie ist angelehnt an den Abstoßungsterm (*Repulsion*) von W. Choi u. a. (W. Choi u. a. 2013) ohne dabei auf 3D-Koordinaten angewiesen zu sein.

3.3 Segmentierung des Vordergrundes

Zur Detektion und Verfolgung von Objekten im Spielfeld, wie Spieler, Ball und Feldlinien, liegt es nahe, den Farbkontrast zum Spielfeld auszunutzen. Die Zielsetzung für die Verfolgung der Spieler ist es, durch farbbasierte Segmentierungsmethoden die Ausmaße des Spielfeldes zu bestimmen, Vordergrundobjekte innerhalb des Spielfeldes zu lokalisieren und aus diesen Objekten die einzelnen Spieler herauszufiltern und zu vereinzeln. Im Folgenden wird das Verfahren von Hoernig u. a. (Hoernig u. a. 2015) vorgestellt.

3.3.1 Bestimmung der Feldhülle

Die Ausmaße des Spielfeldes im Bild werden durch eine sogenannte Feldhülle beschrieben. Eine solche Hülle ist nicht nur nützlich zur Bestimmung der Vordergrundpixel. Sie kann auch als Hilfsmittel für nachfolgende Verfahrensschritte genutzt werden, beispielsweise um Objekte außerhalb des Spielfeldes (wie zum Beispiel das Publikum) von der Spielerverfolgung auszuschließen.

Ein primitives Vorgehen im Bereich Fußball ist es, die Menge der Pixel mit grüner Farbe zu extrahieren und deren konvexe Hülle zu berechnen. Solch ein Verfahren ist sehr anfällig gegenüber Herausforderungen, wie einer inhomogene Farbverteilung im Rasen oder grüner Bereiche außerhalb des Rasens (zum Beispiel Bandenwerbung). Zudem werden hilfreiche geometrische Randbedingungen außer Acht gelassen. Aus diesem Grund kommt in der vorliegenden Arbeit ein robusteres Verfahren, wie in (Hoernig u. a. 2015) beschrieben, zum Einsatz.

Es kann mit Hilfe des Algorithmus von Sutherland-Hodgman (Sutherland und Hodgman 1974) nachgewiesen werden, dass ein rechteckiges Spielfeld unter bestimmten Voraussetzungen, welche bei Aufnahmen von Fußballspielen aus der totalen Perspektive erfüllt sind, als ein konvexes Polygon mit mindestens drei und höchstens acht Ecken ins Bild projiziert wird (für Details siehe (Hoernig u. a. 2015) und (Hartley und Zisserman 2003)).

Die Bestimmung der Feldhülle erfolgt in zwei Schritten:

1. Die Ermittlung der Menge N von grünen Pixeln im Bild, im Wesentlichen durch eine Schwellwertoperation im Farbtonkanal (Hue) des HSV-Farbraums.
2. Die Bestimmung einer Menge C von Ecken mit $3 \leq |C| \leq 8$, welche die Fehlerfunktion $E(C, N)$ minimiert. Dabei ist $E(C, N) := |\texttt{conv}(C) \triangle N|$ und $\texttt{conv}(C)$ die konvexe Hülle der Menge C sowie $A \triangle B$ die symmetrische Differenz der Mengen A und B . Für die Optimierung von $E(C, N)$ kommt ein gieriges Optimierungsverfahren (*Greedy*) zum Einsatz.

Es wird also ein Polygon mit drei bis acht Ecken gesucht, das möglichst viele grüne Pixel und möglichst wenig andersfarbige Pixel einschließt. Das Resultat des Verfahrens ist eine Bildregion H, welche die konvexe Hülle der optimalen Menge von Eckpunkten repräsentiert.

3.3.2 Bestimmung der Grasmaske

Nach der Bestimmung der Hülle des Spielfelds im Bild werden nun die Pixel innerhalb der Hülle klassifiziert. Das geschieht durch die Zuweisung eines Wertes aus $[0; 1]$ für jedes Pixel. Dieser Wert sagt aus, zu welchem Grad ein Pixel als Gras klassifiziert werden kann (1 bedeutet „Pixel ist mit hoher Wahrscheinlichkeit Vordergrund“; 0 bedeutet „Pixel ist mit hoher Wahrscheinlichkeit Gras“).

Eine einfache Schwellwertsegmentierung wie in 3.3.1 ist unter anderem aus folgenden Gründen nicht geeignet:

- Da es starke Helligkeits- und Farbunterschiede des Grases innerhalb eines Bildes (beispielsweise durch Schatten eines Stadiondaches) und vor allem zwischen Aufnahmen verschiedener Spiele geben kann, müssen die Schwellwerte sehr tolerant gewählt werden.
- Tolerant gewählte Schwellwerte haben zur Folge, dass andere grüne Objekte segmentiert werden, die kein Gras sind (beispielsweise grüne Spielertrikots oder Bandenwerbung).
- Eine einfache Schwellwertsegmentierung ist binär und ermittelt keine graduellen Werte im Intervall $[0; 1]$.

Der verwendete Ansatz von Hoernig et al. (Hoernig u. a. 2015) berücksichtigt bei der Berechnung der Grasmaske die Homogenität der Rasenflächen. Zudem wird die Korrelation der RGB-Farbkanäle innerhalb des Rasens genutzt. Das Verfahren geht davon

aus, dass alle Rasenpixel im RGB-Würfel in der Nähe einer Geraden liegen, die parallel zur Schwarz-Weiß-Diagonalen des Würfels ist. Das Verfahren ermittelt Ellipsoide die auf dieser Gerade liegen und die Graspixel umschließen. Als Resultat erhält man ein Bild $I_{Grass} : R_I \rightarrow [0; 1]$.

3.4 Schätzung der Größe von Objekten im Bild

In vielen Fällen der Objekterkennung und -verfolgung ist es hilfreich die ungefähre Größe des gesuchten Objektes im Bild zu kennen. Zum einen können dadurch verschiedene Objekte voneinander abgegrenzt werden (z.B. Spieler und Feldlinien) und zum anderen kann die Treffergenauigkeit bei der Wiederfindung von Objekten (*Template Matching*) durch eine Anpassung der Vorlagengröße erhöht werden. Idealerweise sind die internen und externen Parameter der Kamera bekannt. In diesem Fall können die Ausmaße eines Objektes im Bild anhand der Ausmaße im Weltkoordinatensystem (z.B. ein Spieler mit einer Größe von 1,80 m) zuverlässig bestimmt werden. Sind die Kameraparameter nicht bekannt und sonst kein Vorwissen vorhanden, ist es zunächst nicht möglich die Größe von Objekten im Bild abzuschätzen.

Voraussetzungen

Wurden allerdings schon mehrere Objekte ähnlicher Größe (z.B. Spieler) im Bild oder in einer Videosequenz in dem vorhergehenden Bild erkannt, so kann daraus eine Größenabschätzung für alle Positionen im aktuellen Bild abgeleitet werden. Dazu werden zwei Voraussetzungen als erfüllt angenommen:

1. Die Objekte befinden sich alle auf der gleichen Ebene im Weltkoordinatensystem.
2. Die Objekte (im Weltkoordinatensystem) sind annäherungsweise gleich groß.

Da das Spielfeld als Ebene angenommen wird, ist es leicht einzusehen, dass Voraussetzung 1 im Rahmen von Fußballspielen in der Regel erfüllt ist. Nur in den seltenen Fällen, in denen Spieler weit vom Boden abspringen, trifft die Voraussetzung nicht zu. Man kann auch davon ausgehen, dass Voraussetzung 2 erfüllt ist. Auch wenn einzelne Spieler unterschiedliche Körpergrößen haben, sind die Abweichungen von Mittelwert insgesamt vernachlässigbar. Zu Problemen führt dieses Modell, wenn sich Spieler beispielsweise bücken oder am Boden liegen.

Das lineare Modell

Sind beide Voraussetzungen erfüllt, können die Größen der Objekte trotz perspektivischer Abbildung mit einem linearen Modell in Abhängigkeit von der y-Koordinate im Bild geschätzt werden, das heißt:

$$\hat{h}(y_o) = c_0 \cdot y_o + c_1 \tag{3.1}$$

Für Personen im Bild wird zusätzlich Folgendes angenommen:

$$\hat{w}(y_o) = 0{,}41 \cdot \hat{h}(y_o) \tag{3.2}$$

Dabei ist y_o die y-Koordinate der Unterkante des Objektes o im Bild, $\hat{w}(y_o)$ und $\hat{h}(y_o)$ sind die geschätzte Breite bzw. Höhe des Objektes o im Bild und c_0 und c_1 sind die Parameter des linearen Modells, die anhand von schon bekannten Objektpositionen und -größen im Bild zu bestimmen sind. Gleichung 3.2 leitet sich aus den Erkenntnissen von Dollar u.a. (Dollár, Wojek u. a. 2012) ab, die besagen, dass Personen in Bildern im Mittel ein Seitenverhältnis von Breite zu Höhe von 0,41 zu 1 haben.

Motivation für das lineare Modell

Eine Motivation für das lineare Modell in Gleichung 3.1 liefert folgende Fallstudie. Im Fußballszenario wird eine virtuelle Full-HD-Kamera (1920×1024) simuliert mit einer plausiblen Position und Konfiguration (z.B. in einem größeren Fußballstadion). Die Kamera ist auf der Höhe der Mittellinie in 20 Meter Abstand zur Seitenlinie und in 15 Meter Höhe positioniert. Dabei hat das Objektiv eine Brennweite von 4 mm und die Pixel auf dem Sensor eine Größe von 2,5 μm (das entspricht in etwa einem Sensorformat von $1/3''$) (siehe beispielsweise (Canon Deutschland GmbH 2016)). Das optische Zentrum ist gleichzeitig der Mittelpunkt des Sensors und ist auf den Feldpunkt mit den Koordinaten $(0, -20)$ ausgerichtet. Die gerenderte Aufnahme eines Fußballfeldes mit Standardmaßen (105 m $\times$ 68 m) mit dieser Kamerakonfiguration ist in Abbildung 3.2 zu sehen. Für die Details zum verwendeten Kameramodell sei auf (Steger u. a. 2008, S. 180 ff.) verwiesen.

Für die simulierte Kamera sind die internen und externen Kameraparameter bekannt (eine optische Verzeichnung wird nicht simuliert) und somit auch die Transformation vom Weltkoordinatensystem in das Bildkoordinatensystem. Das Spielfeld (mit Standardmaßen) wird nun in X-Richtung von -52 m bis 52 m und in Y-Richtung von -34 m bis 34 m mit einer Schrittweite von 1 m durchlaufen. An jedem Abtastpunkt i werden die Punkte in 0 m und 1,80 m Höhe in das Bild auf die Bildkoordinaten $(x_i^0, y_i^0)^T$ und $(x_i^{1,8}, y_i^{1,8})^T$

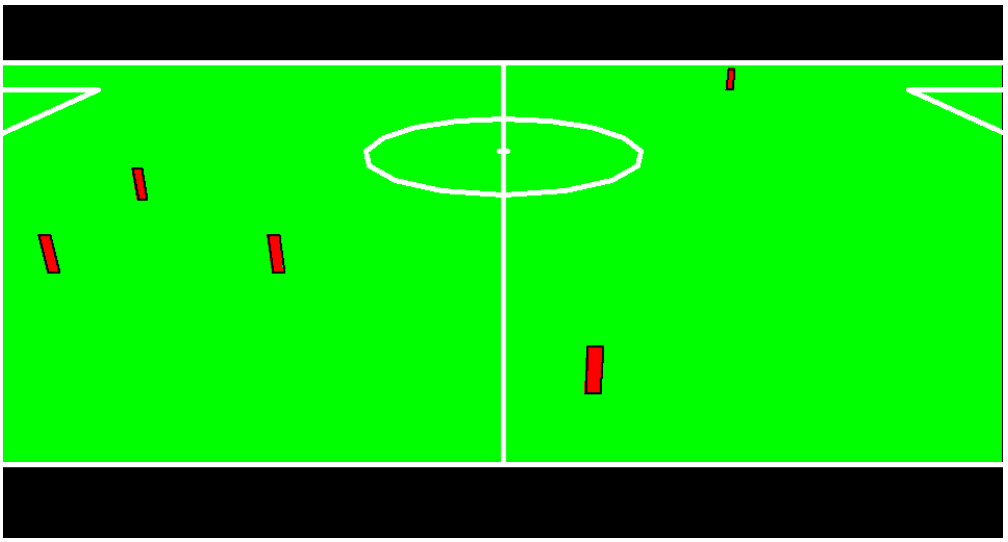


Abbildung 3.2: Bild einer simulierten Kamera mit Aufsicht auf ein Fußballfeld mit Rechtecken in ungefährer Personengröße (Breite 0,50 m und Höhe 1,80 m)

projiziert. Es werden nur Abtastungen berücksichtigt, bei denen beide Punkte komplett im Sichtbereich der Kamera liegen. In diesem Beispiel sind das $n := 4531$ von insgesamt 7245 Abtastungen. Aus den Koordinaten werden nun die unabhängige Variable $y_i := y_i^0$ (y-Koordinate) und die abhängige Variable $\hat{h}(y_i) := y_i^0 - y_i^{1,8}$ (Höhe) abgeleitet. Der Grad des linearen Zusammenhangs zwischen y-Koordinate und Höhe im Bild lässt sich am Korrelationskoeffizient (oder auch Pearson-Korrelation) ablesen, der sich wie folgt bestimmt:

$$r_{y,h} := \frac{\sum_{i=1}^{n}(y_i - \overline{y})(\hat{h}(y_i) - \overline{h})}{\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2 \sum_{i=1}^{n}(\hat{h}(y_i) - \overline{h})^2}} \tag{3.3}$$

wobei $\overline{y} := \frac{1}{n}\sum_{i=1}^{n} y_i$ und $\overline{h} := \frac{1}{n}\sum_{i=1}^{n} \hat{h}(y_i)$ die empirischen Mittelwerte sind und $r_{y,h} \in [-1; 1]$. Dabei stehen Werte von 1 und -1 für einen vollständigen positiven bzw. negativen linearen Zusammenhang und ein Wert von 0 für keinerlei linearen Zusammenhang. Im Beispiel ist $r_{y,h} = 0{,}991894$ und unterstützt damit die Wahl des linearen Modells aus Gleichung 3.1. Ähnliche Korrelationswerte werde auch für andere plausible Kamerakonfigurationen erzielt.

Schätzung der Größe

Seien n Objekte als eine Menge von Bounding-Boxen $B := \{B_1, \ldots, B_n\}$ gegeben (beispielsweise bereits im Bild erkannte Spieler). Ziel ist es, die Koeffizienten c_0 und c_1 aus Gleichung 3.1 so zu bestimmen, dass die Summe der Fehlerquadrate minimiert wird, das heißt:

$$\sum_{i=1}^{n} \|\hat{h}((y_i + h_i - 1) - h_i\|_2^2 \rightarrow min \tag{3.4}$$

Dabei ist $y_i + h_i - 1$ die y-Koordinate der Unterkante von B_i und h_i die Höhe von B_i. Für $n \geq 3$ stellt Gleichung 3.4 ein Standardproblem der Regressionsanalyse dar und kann über einen Maximum-Likelihood-Ansatz mit der Methode der kleinsten Quadrate effizient und analytisch gelöst werden. Dabei gibt es allerdings folgende Punkte zu beachten:

- Da die gegebene Objektmenge in der Regel ein Ergebnis von Verfahren der Objekterkennung bzw. -verfolgung ist, ist diese Menge in der Regel fehlerbehaftet. Beispielsweise kann diese Menge fehlerhafte Erkennungen enthalten, die nicht der tatsächlichen Objektgröße entsprechen. Für eine zuverlässige Schätzung mit einer reduzierten Sensibilität gegenüber Ausreißern muss hier auf Methoden der robusten Statistik (Huber und Ronchetti 2009) zurückgegriffen werden.
- Es ist bekannt, dass bei einer perspektivischen Abbildung Objekte im Vordergrund größer abgebildet werden als gleich große Objekte im Hintergrund. Formal bedeutet das $y_j \geq y_i \implies h_j \geq h_i$ und damit $c_0 \geq 0$. Zudem können für gebräuchliche Kamerakonfiguration obere und untere Grenzen für c_0 und c_1 bestimmt werden. Für eine robuste Schätzung sollten diese Randbedingungen ausgenutzt werden.

Der Nachteil von Maximum-Likelihood-Methoden ist, dass der Einfluss von Ausreißern mit der Größe ihrer Abweichung steigt. So kann beispielsweise eine einzelne extreme fehlerhafte Messung die Modellschätzung in großem Maße verfälschen. Sogenannte M-Schätzer können als eine Verallgemeinerung betrachtet werden. Dabei wird versucht, den Einfluss von Ausreißern zu reduzieren. Dies geschieht in der Regel über eine iterative gewichtete Anpassung, bei der in jeder Iteration die Gewichte in Abhängigkeit von der Abweichung zum Modell aus der letzten Iteration bestimmt werden. Dieses Vorgehen wird auch als iterativ-neugewichtete kleinste Quadrate (*Iteratively Reweighted Least Squares*) bezeichnet. Siehe dazu auch die Arbeit von Z. Zhang (Z. Zhang 1997).

Sei $r_i := |\hat{h}(y_i) - h_i|$ die Differenz zwischen geschätzter und tatsächlicher Höhe und sei $\tilde{r}$ der Median der Menge $\{r_1, \ldots, r_n\}$. Dann ist $\sigma_{MAD} := 1{,}4826 \cdot \tilde{r}$ eine robuste Schätzung der Standardabweichung (siehe (Huber und Ronchetti 2009, S. 106)). Sei $z_i := \frac{r_i}{\sigma_{MAD}}$ die normierte Differenz. Dann ist die Gewichtsfunktion des Schätzers *Tukey's biweight* wie folgt definiert (Z. Zhang 1997):

$$w(z_i) := \begin{cases} (1 - \frac{z_i^2}{a^2})^2 & \text{für } |z_i| \leq a \\ 0 & \text{für } |z_i| > a \end{cases} \tag{3.5}$$

Die sogenannte Tuningkonstante a legt als Vielfaches der mittleren Abweichung den Bereich fest, in dem Ausreißer noch akzeptiert werden und wird, wie von Press u. a. (Press u. a. 2007) vorgeschlagen, mit $a := 6$ gewählt.

Algorithmus 3.1 skizziert das Vorgehen bei der Anpassung der Objekthöhe. Neben den Bounding-Boxen $B_1, \ldots, B_n$ erhält der Algorithmus als Eingabe die zugehörigen Qualitätsmaße $q_1, \ldots, q_n$ mit $q_i \in [0; 1]$, die eine Aussage über die Güte (Konfidenz) der gemessenen Bounding-Boxen treffen. Die Funktion `median`$(r_1, \ldots, r_n)$ berechnet den Median des Tupels $(r_1, \ldots, r_n)$. Die Funktion `boundedFit`$(B, \vec{w}^{fit}, \vec{l}, \vec{u})$ in Zeile 7 berechnet die Koeffizienten $\vec{c} := (c_0, c_1)^T$, welche Gleichung 3.4 unter den gegebenen Randbedingungen $\vec{l} \leq \vec{c} \leq \vec{u}$ erfüllen. Es gibt Verfahren, die dieses lineare Problem direkt lösen ((Stark und Parker 1995)). Alternativ kann es auch mit einer Optimierung für nicht-lineare Probleme mit Randbedingungen gelöst werden (z.B. Levenberg-Marquard, siehe (Nocedal und Wright 2006a) (Bochkanov 2014)). Da das unbeschränkte lineare Problem konvex ist, sollten dadurch bei kleinem n keine Performance- und Genauigkeitsprobleme entstehen.

Die Wahl der Parameter für Algorithmus 3.1 kann im Anhang B aus Tabelle B.1 entnommen werden.

3.5 Extraktion der Spielersilhouetten

Das Ergebnis aus Abschnitt 3.3.2 ist ein Bild I_{Grass}, bei dem ein Pixelwert aus $[0; 1]$ angibt, mit welcher Wahrscheinlichkeit das Pixel ein Graspixel ist oder nicht (ein Wert von 0 bedeutet, dass das Pixel mit hoher Wahrscheinlichkeit ein Graspixel ist; ein Wert von 1 bedeutet, dass das Pixel mit hoher Wahrscheinlichkeit ein Vordergrundpixel ist). Um die einzelnen Spieler zu segmentieren liegt es nahe, eine einfache Schwellwertoperation auszuführen. Dabei ergeben sich folgende Probleme:

- Die Feldlinien werden bei einem naiven Schwellwertverfahren segmentiert und ebenfalls als Vordergrund erkannt.
- Durch starke Inhomogenitäten im Rasen (wie starke Beleuchtungskontraste, Rasenlöcher, Matsch oder Werbeflächen auf dem Boden) kann es vorkommen, dass große Bereiche, die zum Rasen gehören, als Vordergrund segmentiert werden.
- Es kann vorkommen, dass es für I_{Grass} keinen globalen Schwellwert gibt, um das Gras vom Vordergrund zu trennen und somit die Qualität des Ergebnisses eines naiven Schwellwertverfahrens lokal unterschiedlich ist.

Algorithmus 3.1 : Iterativ-neugewichtete kleinste Quadrate mit Randbedingungen

Input : Menge von Bounding-Boxen $B := \{B_1, \ldots, B_n\}$

Qualitätsmaße $q_1, \ldots, q_n$

Untergrenzen $\vec{l} := (l_0, l_1)^T$

Obergrenzen $\vec{u} := (u_0, u_1)^T$

Iterationslimit d

Konvergenzlimit ϵ

Tuningkonstante a

Output : Koeffizienten $\vec{c} := (c_0, c_1)^T$ aus Gleichung 3.1 mit $\vec{l} \leq \vec{c} \leq \vec{u}$

$\vec{c} \leftarrow \vec{0},\ cnt \leftarrow 0,\ diff \leftarrow \infty$ `/* Initialisierung */`

foreach $i \in \{1, \ldots, n\}$ **do** $w_i \leftarrow 1$ `/* Initialisierung der Gewichte */`

while $(cnt < d) \wedge (diff > \epsilon)$ **do** `/* Iteration bis Erreichen der Limits */`

- $\vec{c}_{prev} \leftarrow \vec{c}$
- **foreach** $i \in \{1, \ldots, n\}$ **do** `/* Gewichtung mit Qualitätsmaßen */`
  - $w_i^{fit} \leftarrow q_i \cdot w_i$
- $\vec{c} \leftarrow \texttt{boundedFit}(B, \vec{w}^{fit}, \vec{l}, \vec{u})$ `/* Berechnung der Koeffizienten */`
- **foreach** $i \in \{1, \ldots, n\}$ **do** `/* Residuen (Gleichung 3.4) */`
  - $r_i \leftarrow |\hat{h}(y_i + h_i - 1) - h_i|$
- $\sigma_{MAD} \leftarrow 1.4826 \cdot \texttt{median}(r_1, \ldots, r_n)$ `/* Robuste Standardabweichung */`
- **foreach** $i \in \{1, \ldots, n\}$ **do**
  - $w_i \leftarrow 0$
  - $z_i \leftarrow r_i / \sigma_{MAD}$
  - **if** $|z_i| \leq a$ **then**
    - $w_i \leftarrow (1 - \frac{z^2}{a^2})^2$ `/* Tukey's biweight */`
- $cnt \leftarrow cnt + 1$
- $diff \leftarrow \|\vec{c} - \vec{c}_{prev}\|_2$

return $\vec{c}$

- Spielerregionen und andere Vordergrundregionen (z.B. Feldlinien) überschneiden sich häufig. Daher ist es nicht ausreichend, nur zusammenhängende Regionen zu entfernen. In solchen Fällen müssen nur Teilbereiche einer Region entfernt werden.

Um diese Probleme zu umgehen, werden nach der Schwellwertoperation noch verschiedene Nachbearbeitungsschritte (wie morphologische Operationen, Analyse von Regionenmerkmalen etc.) durchgeführt, um ausschließlich Regionen, die ähnliche Form und

Größe von Spielern haben, zu extrahieren. Da Inhomogenitäten in der Regel selten lokal auftreten, werden in der Umgebung dieser spielerähnlichen Regionen lokale Schwellwertoperationen durchgeführt. Das Ziel ist es, nur die Regionen im Bild zu identifizieren, die Spieler enthalten. Das Vorgehen ist in Algorithmus 3.2 dargestellt.

Algorithmus 3.2 : Extraktion der Spielersilhouetten

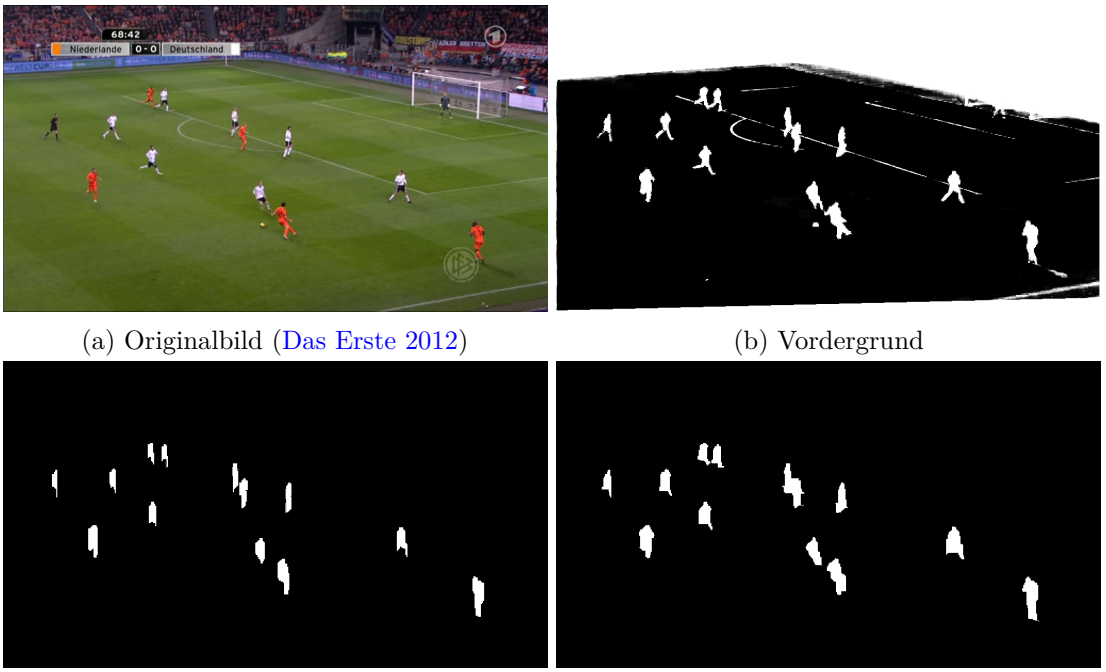
Input : Bild I mit zugehöriger Spielfeldhülle H und Grasmaske I_{Grass}
Größenschätzung Koeffizienten $\vec{c} := (c_0, c_1)$
Schwellwert τ_{Grass}
Größe des Strukturelements r
Toleranzfaktoren $\vec{s} := \left(s_x^l, s_x^u, s_y^l, s_y^u\right)^T$
Dilatationsfaktor s_{dil}
Output : Spielerregion $\overline{FG}$

1 $BG \leftarrow \{(x,y) \mid I_{Grass}(x,y) \leq \tau_{Grass}\}$ `/* Segmentierung des Grases */`
2 $\widetilde{H} \leftarrow$ `erodeRect`(H, r, r) `/* Erosion der Feldhülle */`
3 $FG \leftarrow \widetilde{H} \setminus BG$ `/* Bestimmung des Vordergrunds */`
4 $\widetilde{FG} \leftarrow$ `filterRuns`$(FG, \vec{c}, \vec{s})$ `/* Entfernen von Feldlinien */`
5 $CC \leftarrow$ `connection`$(\widetilde{FG})$ `/* Zusammenhangskomponenten */`
6 $\widetilde{CC} \leftarrow$ `localThreshold`(CC, r, s_{dil}) `/* Segmentierung mit lokalen Schwellwerten */`
7 $\overline{CC} \leftarrow$ `filterSize`$(\widetilde{CC}, \vec{c}, \vec{s})$ `/* Filterung anhand der Größe */`
8 $\overline{FG} \leftarrow \bigcup \overline{CC}$
9 **return** $\overline{FG}$

Dabei wird in Zeile 1 eine einfache Schwellwertoperation durchgeführt, um die Region der potentiellen Graspixel zu extrahieren.

Durch optische Verzeichnung und andere Artefakte kann die berechnete Feldhülle kleinere Bereiche außerhalb des Spielfelds umfassen. Diese Bereiche können einen störenden Einfluss haben, beispielsweise durch Boden- und Bandenwerbung.
Der Aufruf `erodeRect`(R, w, h) in Zeile 2 führt daher eine morphologische Erosion (Steger u. a. 2008, S. 130) auf der Bildregion R mit einem rechteckigen Strukturelement mit Breite w und Höhe h durch und gibt die resultierende Region zurück. Anschließend wird der Vordergrund innerhalb des Spielfelds als Mengendifferenz von erodierter Feldhülle und Grasregion in Zeile 3 bestimmt (siehe Abbildung 3.3b).

Die Spieler im Bild haben ein relativ festes Seitenverhältnis und können unter Einbezug der Größenschätzung meistens sehr gut von horizontalen und vertikalen Linien unterschieden werden. Der Aufruf `filterRuns`$(R, \vec{c}, \vec{s})$ in Zeile 4 dient dazu, schmale und



(a) Originalbild (Das Erste 2012)

(b) Vordergrund

(c) Nach Entfernen der Feldlinien

(d) Spielersilhouetten

ABBILDUNG 3.3: Zwischenergebnisse der Extraktion der Spielersilhouetten

ausgedehnte Bereiche der segmentierten Region herauszufiltern. Dazu wird die Region R in eine Lauflängenkodierung (Steger u. a. 2008, S. 68) transformiert und zwar sowohl zeilenweise (zur Filterung der Breite) als auch spaltenweise (zur Filterung der Höhe). Mit Hilfe der geschätzten Breite bzw. Höhe durch die Größenkoeffizienten $\vec{c}$ werden Lauflängen, die diesbezüglich außerhalb eines Toleranzbandes liegen, entfernt. Das Toleranzband ergibt sich aus der geschätzten Größe an der jeweiligen Stelle. Dabei geben die Faktoren s_x^l und s_x^u, sowie s_y^l und s_y^u die relativen Ober- und Untergrenzen für Breite bzw. Höhe, mit $\vec{s} := \left(s_x^l, s_x^u, s_y^l, s_y^u\right)^T$ vor. Die übrigen Lauflängen werden wieder zu einer Region vereinigt. Als resultierende Region wird die Schnittmenge der zwei Regionen, die bezüglich Zeilen und Spalten gefiltert wurden, zurückgegeben. In diesem Schritt sollen insbesondere Feldlinien, die sich mit Spielerregionen überschneiden, entfernt werden (siehe Abbildung 3.3c).

Da sich die Spieler auf dem Feld verteilen, ist die bisher generierte Region in der Regel nicht zusammenhängend. Mit dem Aufruf `connection`(R) werden die Zusammenhangskomponenten (Steger u. a. 2008, S. 111 ff.) der Region R bestimmt und als Menge von Regionen zurückgegeben. Demnach werden in Zeile 5 die Zusammenhangskomponenten der gefilterten Vordergrundregion bestimmt, wobei eine Komponente in der Regel einen einzelnen Spieler beziehungsweise eine überlappende Gruppe von Spielern repräsentiert.

Der Aufruf `localThreshold`(CC, r, s_{dil}) durchläuft alle Zusammenhangskomponenten

in der Menge CC. In der Umgebung von jeder Zusammenhangskomponente wird eine lokale Schwellwertoperation durchgeführt. Hierbei werden keine fixen Schwellwerte verwendet, sondern eine automatische Schwellwertanalyse durchgeführt. Diese basiert auf der Annahme, dass das Histogramm der Grauwerte bimodal ist, also zwei Maxima besitzt (in diesem Fall Gras und Vordergrund). Als Schwellwert wird ein Minimum dazwischen bestimmt. Ist ein solches nicht vorhanden, wird das Histogramm solange mit einem Gauß-Filter geglättet, bis ein Minimum bestimmt werden kann (Steger u. a. 2008, S. 102 ff.). Die oben genannte Umgebung einer Zusammenhangskomponente bestimmt sich aus deren Bounding-Box mit Höhe h und Breite w. Diese Bounding-Box wird entlang der Y-Koordinate auf beiden Seiten um $s_{dil} \cdot (h + r)$ und entlang der X-Koordinate auf beiden Seiten um $s_{dil} \cdot (w + r)$ erweitert. In den resultierenden lokalen Regionen werden durch ein *Closing* (Steger u. a. 2008, S. 135) mit einem Rechteck der Größe $r \times r$ möglicherweise vorhandene Löcher geschlossen. Diese lokalen Regionen werden zu einer Region zusammengefügt und die Zusammenhangskomponenten dieser Region als Resultat zurückgegeben.

Die in Zeile 6 bestimmten Zusammenhangskomponenten werden in Zeile 7 nach ihrer Größe entsprechend gefiltert. Mit dem Aufruf `filterSize` $\left(\widetilde{CC}, \vec{c}, \vec{s}\right)$ werden die Komponenten aus CC entfernt, die nicht innerhalb einer Toleranz bezüglich der mit den Koeffizienten $\vec{c}$ geschätzten Größe liegen. Dazu werden für jede Zusammenhangskomponente die Orientierung und die Radien der äquivalenten Ellipse berechnet. Abhängig von der Orientierung der Ellipse werden die bestimmten Radien zur Schätzung von Höhe und Breite der Region genutzt. Bei der Aussortierung stellen die Toleranzfaktoren $\vec{s}$ ebenfalls die relativen Unter- und Obergrenzen bezüglich Höhe und Breite dar. Die nicht entfernten Komponenten werden zurückgegeben.

Das Resultat ist die Bildregion $\overline{FG}$, die im Idealfall alle Spielerregionen umfasst und keine anderen potentiellen Vordergrundregionen einschließt. (siehe Abbildung 3.3d) Die Wahl der Parameter für Algorithmus 3.2 kann im Anhang B aus Tabelle B.2 entnommen werden.

3.6 Farbtemplates

„Beide Teams tragen Farben, durch die sie sich klar voneinander sowie vom Schiedsrichter und von den Schiedsrichterassistenten unterscheiden. Jeder Torwart unterscheidet sich in der Farbe der Sportkleidung von den anderen Spielern, vom Schiedsrichter und von den Schiedsrichterassistenten.“ (Fédération Internationale de Football Association (FIFA) 2015b, S. 22)

Wie schon die Spielregeln andeuten, sind Farben ein sehr gutes Unterscheidungsmerkmal bei der optischen Verfolgung von Fußballspielern. Alle Spieler heben sich in der Regel farblich sehr gut vom Spielfeld ab, um vom Betrachter gut wahrgenommen zu werden, selbst wenn die Trikots grün sind. Für eine robuste automatische Verfolgung und Unterscheidung der Spieler ist die Berücksichtigung der Farbinformation nahezu unumgänglich. Zudem bestätigen zahlreiche Studien, dass Farbe ein aussagekräftiges Merkmal bei der Objekterkennung und -verfolgung darstellt, wie beispielsweise die Artikel von Kviatkovsky u. a. (Kviatkovsky u. a. 2013) oder Sebastian u. a. (Sebastian u. a. 2008) demonstrieren.

Bei der Benutzung von Templates zur Verfolgung von Objekten gibt es verschiedene Herangehensweisen:

- **Globales Template:** Für alle Objekte einer Klasse (beispielsweise Spieler) gibt es ein globales Template, in dem möglichst viele Varianten des Aussehens modelliert sind. Das hat den Vorteil, dass zum einen nur ein Template erstellt und gespeichert werden muss und zum anderen durch den Mittelungseffekt eine gewisse Verallgemeinerung und Robustheit gegenüber Schwankungen im Aussehen erzielt wird. Der Nachteil ist, dass im Anschluss an mehrdeutige Situationen (wie bei einer Überdeckung zweier Spieler) eine Auflösung (d.h. eine richtige Identifizierung der individuellen Objekte) erschwert wird. Zudem ist die modellierte Variabilität stark abhängig von der Variabilität innerhalb der Trainingsmenge.

- **Individuelles Template:** Für jedes individuelle Objekt wird ein eigenes Template generiert. Dies setzt meist eine initiale Erkennung des Objektes voraus. Dabei werden (falls vorhanden) fehlerhafte Erkennungen in das Modell integriert, welche die Robustheit des Verfahrens negativ beeinflussen können. Das heißt, die Güte der erstellten Templates ist direkt abhängig von der Güte der Objekterkennung. Auf der anderen Seite können individuelle Templates die korrekte Identifizierung von Objekten erleichtern.

- **Fixiertes Template:** Das Template wird initial erstellt und bleibt für die komplette Zeitspanne der Objektverfolgung unverändert. Veränderungen des Aussehens des Objektes (beispielsweise, wenn der Spieler aus dem Schatten in die Sonne läuft) können dazu führen, dass eine korrekte Verfolgung nicht mehr sicher gestellt ist.

- **Adaptives Template:** Das Template wird initial erstellt und von Zeit zu Zeit angepasst (beispielsweise in jedem Frame einer Videosequenz). Dadurch können Veränderungen des Aussehens abgebildet werden. Werden bei der Anpassung der

Templates vermehrt Regionen außerhalb des Objektes berücksichtigt, so kommt es häufig zu einem Abdriften.

In dieser Arbeit wird ein hybrider Ansatz bezüglich globaler / individueller Farbtemplates verfolgt. Das heißt, es wird versucht für jedes Outfit im Rahmen einer Videosequenz ein Template zu erstellen. Das motiviert sich aus den folgenden Gründen:

- Die Outfits beider Teams, der Schiedsrichter und der Torhüter lassen sich farblich in der Regel sehr gut voneinander unterscheiden.
- Die einzelnen Spieler eines Teams bzw. die Schieds- und Linienrichter lassen sich farblich eher schlecht unterscheiden. Es gibt zwar Merkmale wie Haut-, Haar- und Schuhfarbe. Diese sind insbesondere bei Aufnahmen schlechterer Qualität nicht für eine Unterscheidung geeignet, da sie meist nur durch wenige Pixel beschrieben werden.
- Die Farbtemplates, die Teamoutfits repräsentieren, werden anhand mehrerer Objekte erstellt und modellieren somit eine gewisse Verallgemeinerung bezüglich Schwankungen des Aussehens.

In der Regel sind in einem Fußballspiel fünf Outfits zu sehen (Spieler Heimmannschaft, Torwart Heimmannschaft, Spieler Gastmannschaft, Torwart Gastmannschaft, Schieds- und Linienrichter). Allerdings kann die Anzahl in einer Videosequenz variabel sein. Zum einen können in einer kurzen Videosequenz weniger als fünf Outfits zu sehen sein, da beispielsweise ein Torwart nie ins Blickfeld der Kamera kommt. Zum anderen kann es vorkommen, dass Trainer, Kameramänner oder andere Personen am Spielfeldrand sichtbar sind.

Zudem wird in dieser Arbeit ein hybrider Ansatz bezüglich fixierter / adaptiver Farbtemplates verfolgt. Um das Problem des Abdriftens zu umgehen, ist ein Farbtemplate prinzipiell fixiert und wird einmalig erstellt und danach nicht mehr geändert. Allerdings kann bei der Erstellung des Templates die Information mehrerer Bilder am Anfang einer Videosequenz (beispielsweise aus der ersten Minute) genutzt werden. Das soll zwei Punkten dienen:

- Durch die Mittelung über mehrere Bilder soll eine Robustheit gegenüber Variationen im Aussehen in das Template integriert werden.
- Es sollen möglichst alle sichtbaren Outfits bei der Erstellung der Templates verfügbar sein.

Seien nun die Bilder $I^1, \ldots, I^m$ aus eine Videosequenz mit $m \in \mathbb{N}^*$ gegeben sowie die zugehörigen Grashüllen $H^1, \ldots, H^m$ und Grasmasken $I^1_{Grass}, \ldots, I^m_{Grass}$ und sei für jedes Bild I^t eine Menge B^t von n_t Objekten (in Form von Bounding-Boxen) gegeben mit $n_t \in \mathbb{N}$ und $B^t := \{B^t_1, \ldots, B^t_{n_t}\}$ für $n_t \geq 1$. Die Erstellung der Farbtemplates erfolgt in den folgenden Schritten:

1. Ausschluss von überdeckten Objekten.
2. Bestimmung der dominanten Farben zur Quantisierung.
3. Erzeugung von Farbtemplates für individuelle Objekte.
4. Gruppierung der individuellen Objekttemplates zu Outfittemplates.

3.6.1 Ausschluss von überdeckten Objekten

Wird ein Objekt von einem anderen Objekt vollständig oder teilweise überdeckt, so kann dieses Objekt die Erstellung eines Templates verfälschen, da zum einen ein Bereich des vorderen Objektes mit in die Berechnung einfließt und zum anderen nicht die volle Farbinformation des Objektes genutzt werden kann (beispielsweise sind die Beine und Hose eines Spielers nicht zu sehen). Aus diesem Grund ist es wichtig, stark überdeckte Objekte vor der Templateerstellung herauszufiltern.

Eine Überdeckung der Objekte B_1 und B_2 liegt genau dann vor, wenn gilt:

$$o_J(B_1, B_2) > \tau_{ovlp}, \tag{3.6}$$

wobei $\tau_{ovlp} \in [0; 1]$ den Überdeckungsschwellwert darstellt und o_J der Jaccard-Koeffizient aus Gleichung 2.3 ist. Ein Objekt B_i aus einer Menge $B := \{B_1, \ldots, B_n\}$ ist genau dann überdeckungsfrei in Bezug auf B, wenn gilt:

$$\nexists B_j \in B \setminus \{B_i\} : (o_J(B_i, B_j) > \tau_{ovlp}) \wedge (y_i + h_i < y_j + h_j) \tag{3.7}$$

Dem liegt die Annahme zu Grunde, dass die Unterkante von Objekten im Vordergrund eine größere y-Koordinate hat als die Unterkante von Objekten die sich aus Sicht der Kamera dahinter befinden.

Um zu vermeiden, dass Farbinformation außerhalb des Spielfelds mit in die Templateberechnung (beispielsweise Bandenwerbung) einfließt, werden nur Objekte B_i berücksichtigt, die im Bild I weitestgehend innerhalb der Feldhülle H liegen, das heißt für die gilt:

$$\frac{|B_i \cap H|}{|B_i|} > \tau_{ovlp}. \tag{3.8}$$

Das Resultat dieses ersten Schrittes ist für jedes Bild I^t mit $t \in \{1, \ldots, m\}$ eine Menge von Bounding-Boxen $\overline{B}^t \subseteq B^t$ für die für jedes $B \in \overline{B}^t$ die Gleichungen 3.7 und 3.8 bezüglich B^t und H^t erfüllt sind.

3.6.2 Bestimmung der dominanten Farben

Im Folgenden wird davon ausgegangen, dass sich ein Einzelbild einer Videosequenz aus drei Farbkanälen zusammensetzt, das heißt $c_I := 3$ in Gleichung 2.1. Ein Farbraum mit drei Kanälen (wie beispielsweise RGB oder LAB) und einer 8-bit Quantisierung hat $256^3 \approx 16{,}8 \times 10^6$ verschiedene Farbwerte. Kommen histogrammbasierte Templates zum Einsatz, so sind diese Histogramme zum einen speicherintensiv (ein Histogramm hat bei 32-bit Zählern einen Verbrauch von ca. 64 MB, was bei fünf Outfits mit je 3 Körperregionen einen Speicheraufwand von ca. 1 GB entspricht). Zum anderen sind diese Histogramme in der Regel dünn besetzt (bei einem Full-HD-Bild stehen $1920 \times 1024 \approx 2 \times 10^6$ Pixel den $16{,}8 \times 10^6$ Histogrammklassen gegenüber; da nur die Objektregionen betrachtet werden, ist dieses Verhältnis in der Regel noch unausgeglichener).

Zudem ist die Anzahl der Farben bei Fußballspielern normalerweise sehr begrenzt. Jedes Outfit hat als farblich hervorstechendes Merkmale die Trikotfarbe, die Farbe der Hose und die Farbe der Stutzenstrümpfe. Wenn man annimmt, dass die Kleidungsstücke jeweils maximal zweifarbig sind, kommt man bei fünf Outfits auf eine Anzahl von $5 \cdot 3 \cdot 2 = 30$ verschiedenen Farben. Das legt nahe, den Farbraum in geeigneter Weise zu quantisieren.

Eine naheliegende Möglichkeit ist es, den Farbraum in gleich große Würfel aufzuteilen. Das würde voraussetzen, dass die Farben im Farbraum gleichmäßig verteilt sind. Diese Voraussetzung ist im Allgemeinen nicht erfüllt. Da in den Trikots nur wenige Farbtöne vorherrschen, sind die einzelnen Farben der Outfits in der Regel alles andere als gleichmäßig verteilt. Die vorliegende Arbeit begegnet dieser Problematik mit dem Ansatz, den Farbraum mit Hilfe einer Clusteranalyse zu quantisieren.

Seien nun die Bilder $I^1, \ldots, I^m$ sowie die zugehörigen Grashüllen $H^1, \ldots, H^m$ und Grasmasken $I^1_{Grass}, \ldots, I^m_{Grass}$ und für jedes Bild I^t mit $t \in \{1, \ldots, m\}$ die gefilterten Objekte $\overline{B}^t := \{\overline{B}^t_1, \ldots, \overline{B}^t_{\bar{n}_t}\}$ aus Abschnitt 3.6.1 und die Vordergrundregion $\overline{FG}^t$ aus Abschnitt 3.5 gegeben. Die Menge der Vordergrundfarbwerte (Farbvektoren) die zur Clusteranalyse genutzt werden, ist

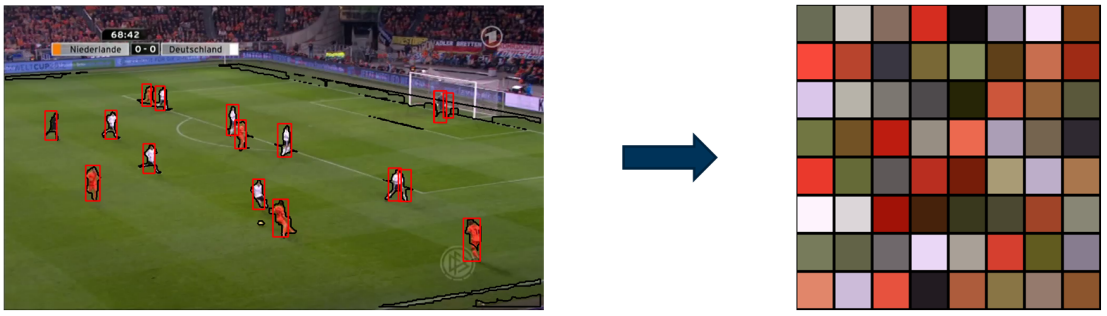


ABBILDUNG 3.4: Ein Beispiel für die Bestimmung der dominanten Farben mit $k_{FG} := 64$. Originalbild aus (Das Erste 2012).

$$C_{FG} := \bigcup_{t=1}^{m} \{I^t(\vec{x}) \mid \vec{x} \in \bigcup_{j=1}^{\bar{n}_t} \overline{B}_j^t \bigwedge I_{Grass}^t(\vec{x}) = 0\}. \tag{3.9}$$

Hierbei ist $I(\vec{x}) := I(x_1, x_2)$ mit $\vec{x} := (x_1, x_2)^T \in \mathbb{R}^2$. Die Menge C_{FG} enthält demnach die Farbwerte aller Pixel innerhalb der gegebenen Objekte, die mit sehr hoher Wahrscheinlichkeit kein Gras sind.

Da aufgrund von Ungenauigkeiten bei der Grassegmentierung (insbesondere am Rand der Objekte) die Menge C_{FG} auch Graspixel oder grasähnliche Pixel enthalten kann, wird in einem iterativen Verfahren versucht, diese Pixel aus der Menge C_{FG} herauszufiltern. Dazu werden die dominanten Farben des Grases bestimmt, das heißt, es wird eine Clusteranalyse auf der Menge C_G durchgeführt, die wie folgt definiert ist:

$$C_G := \bigcup_{t=1}^{m} \{I^t(\vec{x}) | \vec{x} \in H^t \setminus \overline{FG}^t \bigwedge I_{Grass}^t(\vec{x}) = 1\} \tag{3.10}$$

Zur Clusteranalyse kommt das Verfahren *k-means++* (Arthur und Vassilvitskii 2007) zum Einsatz, welches eine Erweiterung des *k-means*-Algorithmus (Lloyd 1982) darstellt und mit einer intelligenten Initialisierung in der Praxis die Effizienz und Genauigkeit des Verfahrens erhöht. Als Distanzmaß kommt hier wie bei *k-means* üblich, die euklidische Distanz zum Einsatz. Da die Initialisierung mit Hilfe eines Zufallsgenerators erfolgt, wird das Clusterverfahren mehrfach durchgeführt und die beste Lösung übernommen. Ein Beispiel für die Bestimmung der dominanten Farben mit $k_{FG} := 64$ ist in Abbildung 3.4 dargestellt.

Aus Algorithmus 3.3 ist das genaue Vorgehen zu entnehmen. Dabei ist `kmeanspp`(C, k) der Aufruf von *k-means++* auf der Menge C mit k Clustern und `kmeans`(C, k, M) der Aufruf des herkömmlichen *k-means* auf der Menge C mit k Clustern und die Menge der initialen Clustermittelpunkte M mit $|M| = k$. Nach dem Clustering der Grasfarben und der Vordergrundfarben werden grasähnliche Farben aussortiert. In Zeile 14 werden die

Farbvektoren des Vordergrunds ermittelt, die näher an einer der dominanten Grasfarben sind als an einer der bis dahin bestimmten dominanten Vordergrundfarben.

Algorithmus 3.3 : Bestimmung der dominanten Farben

Input : Farbvektoren Vordergrund C_{FG}
Farbvektoren Gras C_G
Anzahl Vordergrundfarben k_{FG}
Anzahl Iterationen Vordergrund i_{FG}
Anzahl Grasfarben k_G
Anzahl Iterationen Gras i_G
Output : Menge von Vordergrundfarben $\overline{C}_{FG}$ mit $|\overline{C}_{FG}| = k_{FG}$
Menge von Grasfarben $\overline{C}_G$ mit $|\overline{C}_G| = k_G$

$\phi_G \leftarrow \infty,\ \phi_{FG} \leftarrow \infty,\ C'_{FG} \leftarrow C_{FG}$ `/* Initialisierung */`
foreach $i \in \{1, \ldots, i_G\}$ **do** `/* Clustering der Grasfarben */`
  $(\overline{C}, \phi) \leftarrow$ `kmeanspp`(C_G, k_g)
  if $\phi < \phi_G$ **then**
    $\overline{C}_G \leftarrow \overline{C}$
    $\phi_G \leftarrow \phi$
foreach $i \in \{1, \ldots, i_{FG}\}$ **do** `/* Clustering der Vordergrundfarben */`
  $(\overline{C}, \phi) \leftarrow$ `kmeanspp`(C_{FG}, k_{FG})
  if $\phi < \phi_{FG}$ **then**
    $\overline{C}_{FG} \leftarrow \overline{C}$
    $\phi_{FG} \leftarrow \phi$
while $C'_{FG} \neq \emptyset$ **do** `/* Verfeinerung der Vordergrundfarben */`
  $C \leftarrow \overline{C}_G \bigcup \overline{C}_{FG}$
  $C'_{FG} \leftarrow \{\vec{c}_{FG} \in C_{FG} \mid \arg\min_{\vec{c} \in C} \|\vec{c} - \vec{c}_{FG}\|_2 \in \overline{C}_G\}$ `/* Grasähnliche Farben */`
  $C_{FG} \leftarrow C_{FG} \setminus C'_{FG}$
  $(\overline{C}_{FG}, \phi_{FG}) \leftarrow$ `kmeans`$(C_{FG}, k_{FG}, \overline{C}_{FG})$
return $(\overline{C}_{FG}, \overline{C}_G)$

Die Wahl der Parameter für Algorithmus 3.3 kann im Anhang B aus Tabelle B.3 entnommen werden.

3.6.3 Die Wahl des Farbraums

Bei der Verarbeitung von Farbbildern stehen zahlreiche Farbräume zur Auswahl, die alle spezielle Eigenschaften vorweisen (Plataniotis und Venetsanopoulos 2000). Aus den in dieser Arbeit analysierten Videosequenzen, werden die Einzelbilder standardmäßig im RGB-Farbraum extrahiert. Ein Problem des RGB-Farbraums ist es, dass euklidische Distanzen in diesem Raum nicht der menschlichen Farbwahrnehmung entsprechen. Der *k-means*-Algorithmus, der in Algorithmus 3.3 zur Bestimmung der dominanten Farben genutzt wird, basiert jedoch auf der euklidischen Distanz. Aus diesem Grund wird für die Erstellung und Verwendung der Farbtemplates auf den CIE L*a*b-Farbraum (Plataniotis und Venetsanopoulos 2000) zurückgegriffen und die Eingagnsbilder von RGB nach L*a*b konvertiert. Dieser Farbraum wurde gezielt so entworfen, dass die euklidischen Distanzen proportional zur menschlichen Unterscheidung von Farben ist.

3.6.4 Erzeugung von Farbtemplates für individuelle Objekte

Analog zu der Arbeit von Comaniciu u. a. (Comaniciu u. a. 2003) werden in der vorliegenden Arbeit normierte Farbhistogramme als Templates genutzt. Ein normiertes Histogramm mit m Klassen kann als Vektor $\vec{h} \in [0;1]^m$ mit $\vec{h} := (h_1, \ldots, h_m)^T$ und $\sum_{i=1}^{m} h_i = 1$ aufgefasst werden. Als Maß der Ähnlichkeit zweier Histogramme $\vec{h}^1$ und $\vec{h}^2$ wird in dieser Arbeit (ebenfalls analog zu (Comaniciu u. a. 2003)) die Hellinger Distanz $d_H \in [0;1]$ verwendet, die wie folgt definiert ist (siehe (Cha 2007) und (E. Deza und M.-M. Deza 2006)):

$$d_H(\vec{h}^1, \vec{h}^2) := \sqrt{1 - s_{BC}(\vec{h}^1, \vec{h}^2)}. \tag{3.11}$$

Dabei ist s_{BC} der Bhattacharyya-Koeffizient mit

$$s_{BC}(\vec{h}^1, \vec{h}^2) := \sum_{i=1}^{m} \sqrt{h_i^1 \cdot h_i^2}. \tag{3.12}$$

Für die Menge der dominanten Farben $\overline{C}_{FG}$ aus Abschnitt 3.6.2 mit $|\overline{C}_{FG}| = k_{FG}$ ist ein zugehöriges Farbhistogramm ein Vektor $\vec{h}^c := (h_1^c, \ldots, h_{k_{FG}}^c)$ mit k_{FG} Einträgen. Jeder Eintrag repräsentiert eine zugehörige Farbe aus $\overline{C}_{FG} := \{\vec{c}_1, \ldots, \vec{c}_{k_{FG}}\}$ und ein beliebiger Farbvektor $\vec{c} \in \mathbb{R}^3$ wird dem Histogrammeintrag ($\hat{=}$ Vektoreintrag) zugeordnet, zu dessen entsprechender Farbe er den geringsten Abstand hat.

Sei $\overline{C} := \overline{C}_G \bigcup \overline{C}_{FG}$. $\texttt{bin}(\vec{x}) : \mathbb{R}^c \to \mathbb{Z}$ ist eine Abbildung, die einem Farbvektor den Index der zugehörigen Histogrammklasse zuordnet und ist definiert als

$$\texttt{bin}(\vec{x}) := \begin{cases} \arg\min_{\vec{c} \in \overline{C}_{FG}} \|\vec{c}_i - \vec{x}\|_2 & \text{für } \arg\min_{\vec{c} \in \overline{C}} \|\vec{c} - \vec{x}\|_2 \in \overline{C}_{FG} \\ -1 & \text{sonst} \end{cases} \tag{3.13}$$

Einem Farbvektor wird also der Index der Histogrammklasse zugeordnet, die der nächstliegenden dominanten Farbe entspricht. Ist die nächstliegende dominante Farbe eine Grasfarbe, wird der Farbvektor für die Histogrammberechnung verworfen ($\texttt{bin}(\vec{x}) < 0$). Für die Histogrammberechnung für eine Menge von Farbvektoren wird diese Menge durchlaufen und für jeden Vektor $\vec{x}$ für den $\texttt{bin}(\vec{x}) > 0$ gilt, wird der jeweilige Histogrammeintrag $h^c_{\texttt{bin}(\vec{x})}$ hochgezählt. Am Ende wird das Histogramm normiert.

Im Anlehnung an den Ansatz von Beetz u. a. (Beetz u. a. 2007) wird für die Erstellung eines Templates eine Bounding-Box von oben nach unten in mehrere Teile aufgeteilt. Dies soll die unterschiedliche Farbverteilung in unterschiedlichen Teilen des Spielers modellieren (beispielsweise Trikot und Hose). Wird eine Bounding-Box $B_i := B(x_i, y_i, w_i, h_i)$ in k Teile aufgeteilt, sind die einzelnen Teile ${}^pB_i$ mit $p \in \{1, \ldots, k\}$ definiert als

$${}^pB_i := \begin{cases} B(x_i, y_i + (p-1) \cdot h_i/k, w_i, h_i/k) & \text{für } p < k \\ B(x_i, y_i + (p-1) \cdot h_i/k, w_i, h_i - (p-1) \cdot h_i) & \text{für } p = k \end{cases} \tag{3.14}$$

Das zugehörige Histogramm $\vec{h}^{cp}_i$ besteht demnach aus $k \cdot k_{FG}$ Histogrammklassen mit $\vec{h}^{cp}_i := ({}^1\vec{h}^c_i, \ldots, {}^k\vec{h}^c_i) := (h^{cp}_{i_1}, \ldots, h^{cp}_{i_{k \cdot k_{FG}}})$ und $\sum_{j=1}^{k \cdot k_{FG}} h^{cp}_{i_j} = 1$. Ein Beispiel für ein Spielerhistogramm mit $k_{FG} := 64$ und $k := 3$ ist in Abbildung 3.5 dargestellt.

3.6.5 Gruppierung der individuellen Objekttemplates

Seien die Bilder $I^1, \ldots, I^m$ und für jedes Bild I^t jeweils die gefilterten Objekte $\overline{B}^t := \{\overline{B}^t_1, \ldots, \overline{B}^t_{n_t}\}$ aus 3.6.1 und die dazugehörigen zusammengesetzten Histogramme $H^t := \{\vec{h}^t_1, \ldots, \vec{h}^t_{n_t}\}$ aus 3.6.4 gegeben. Um die Objekte anhand der Farbhistogramme zu verschiedenen Outfits zu gruppieren, bietet sich ebenfalls der *k-means*-Algorithmus an. Allerdings stößt man dabei auf folgende Probleme:

1. Bei *k-means* wird die Anzahl der Cluster als Parameter vorgegeben. Allerdings ist, wie schon erwähnt, die Anzahl der verschiedenen Outfits nicht bekannt. Zum einen können in einer kürzeren Sequenz einige Outfits nicht vorkommen (oder zumindest nicht erkennbar sein). Zum anderen können unbeteiligte Personen, wie Trainer oder Kameramänner am Spielfeldrand sichtbar sein.

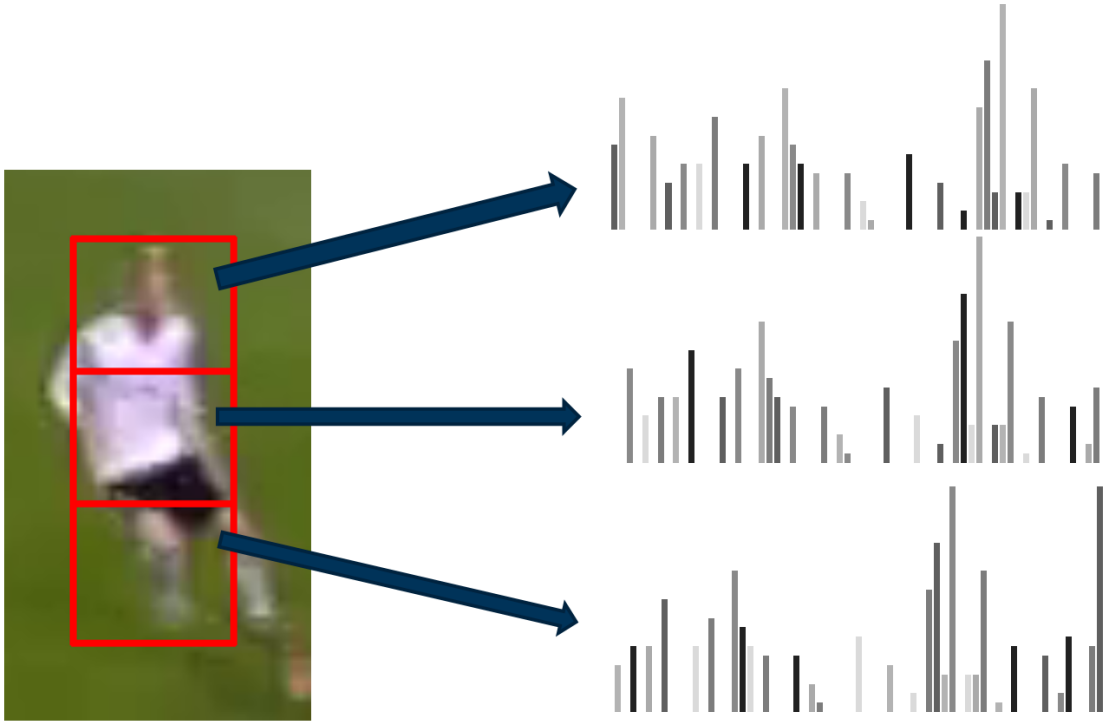


ABBILDUNG 3.5: Ein Beispiel für das zusammengesetzte Farbhistogramm eines Spielers mit $k_{FG} := 64$ und $k := 3$. Originalbild aus (Das Erste 2012).

2. *k-means* und die Berechnung des arithmetischen Mittels als Clusterzentrum basieren auf der quadratischen euklidischen Distanz. Der Einsatz von *k-means* ist somit für Merkmalsräume fragwürdig, in denen die euklidische Distanz keine semantische Interpretation zulässt. Zudem ist das arithmetische Mittel anfällig gegenüber Ausreißern (siehe auch (Bishop 2006, S. 424 ff.)).

Der Problematik in Punkt 1 wird begegnet, indem zunächst mit $k = 1$ begonnen wird und iterativ anhand eines Qualitätsmaßes entschieden wird, ob eine neue Clusteranalyse durchgeführt werden soll, bei der k um eins erhöht wird. Das Qualitätsmaß wird wie folgt bestimmt: Sei $H := \bigcup_{t=1}^{m} H^t := \{\vec{h}_1, \ldots, \vec{h}_n\}$ die Menge der Farbhistogramme, sei $H^C := \{\vec{h}_1^C, \ldots, \vec{h}_k^C\}$ die Menge der Clusterzentren und sei jedes Histogramm $\vec{h}_i$ genau einem Cluster zugeordnet durch $l(\vec{h}_i) \in \{1, \ldots, k\}$. Dann ist $d_{mm}(H, H^C) \in [0; 1]$ definiert durch

$$d_{mm}(H, H^C) := \max_{\vec{h}_j \in H} d_H(\vec{h}^C_{l(\vec{h}_j)}, \vec{h}_j). \tag{3.15}$$

In Worten ist das über alle Cluster die maximale Distanz, die ein Vektor zu seinem Clusterzentrum aufweist.

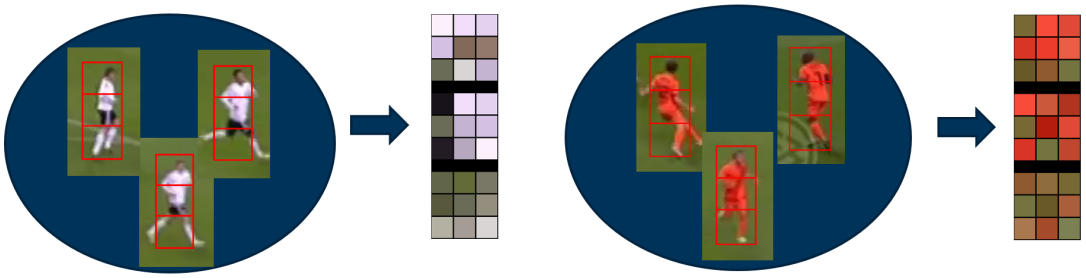


ABBILDUNG 3.6: Ein Beispiel für Teamhistogramme mit $k_F G := 9$ und $k := 3$. Originalbild aus (Das Erste 2012).

Die Problematik in Punkt 2 wird durch eine Variante des *k-means*-Verfahren versucht zu umgehen: das *k-medoids*-Verfahren, bei dem immer ein Punkt aus dem Cluster als Clusterzentrum genommen wird. In dieser Arbeit ist dies das Histogramm, das die Distanz d_H zu allen anderen Histogrammen im Cluster minimiert. Dabei wird das Verfahren analog zu *k-means++* initialisiert. Algorithmus 3.4 veranschaulicht das detaillierte Vorgehen. Aus Abbildung 3.6 ist ein Beispiel für zwei Teamhistogramme mit $k_{FG} := 9$ und $k := 3$ zu entnehmen.

Algorithmus 3.4 : Bestimmung der verschiedenen Outfit-Templates

Input : Farbhistogramme $H := \{\vec{h}_1, \ldots, \vec{h}_n\}$
Anzahl Iterationen n_i
Maximale Anzahl Outfits k_{max}
Abbruchkriterium t
Output : Outfithistogramm $H^C := \{\vec{h}_1^C, \ldots, \vec{h}_k^C\}$

$\phi' \leftarrow \infty,\ k \leftarrow 2$ `/* Initialisierung */`
foreach $i \in \{1, \ldots, n_i\}$ **do** `/* Initiales Clustering */`
  $(H'^C, \phi') \leftarrow$ `kmedoidspp`(H, k)
  if $\phi' < \phi$ **then**
    $H^C \leftarrow H'^C$
    $\phi \leftarrow \phi'$
while $d_{mm}(H, H^C) > t \wedge k < k_{max}$ **do** `/* Bestimmung von` k `*/`
  $k \leftarrow k + 1$
  $\vec{h}_{max} \leftarrow \arg\max_{\vec{h}_j \in H} d_H(\vec{h}^C_{l(\vec{h}_j)}, \vec{h}_j)$
  $H^C \leftarrow H^C \cup \{\vec{h}_{max}\}$
  $(H^C, \phi) \leftarrow$ `kmedoids`(H, k, H^C)
return H^C

Die Wahl der Parameter für Algorithmus 3.4 kann im Anhang B aus Tabelle B.4 entnommen werden.

Sind die Outfithistogramme H^C bestimmt, so wird einer beliebigen Bounding-Box B_i mit dem zugehörigen (aus k Teilen) zusammengesetzten Farbhistogramm $\vec{h}(B_i)$ ein Outfit durch die Funktion $o(B_i) \in \mathbb{R}^{k \cdot k_{FG}}$ zugewiesen mit

$$o(B_i) := \arg\max_{\vec{h} \in H^C} d_H(\vec{h}, \vec{h}(B_i)). \tag{3.16}$$

Daraus ergibt sich eine Zuweisung von einem Label zu jeder Bounding-Box B_i. Mit $H^C := \{\vec{h}_1^C, \ldots, \vec{h}_k^C\}$ gibt es also k Outfitklassen mit den Labels $\{1, \ldots, k\}$. Sei $o(B_i) = \vec{h}_j^C$, dann ist das Label der Bounding-Box B_i definiert als

$$lbl_i := lbl(B_i) := j. \tag{3.17}$$

Nach der Segmentierung der Spielerregionen und der Ermittlung der Teamoutfits müssen diese Merkmale geeignet verwertet werden, um für die Spielererkennung und -verfolgung genutzt werden zu können. Im den folgenden Abschnitten werden die Details hierzu erläutert.

3.7 Die Konfidenzkarte (*Confidence Map*)

Abbildung 3.7: Bildausschnitte (oben) und die zugehörigen Konfidenzkarten (unten). Originalbild aus (Das Erste 2012).

In einer Konfidenzkarte (*Confidence Map*, siehe beispielsweise (Poiesi u. a. 2013)) werden die Informationen der Messungen aus verschiedenen Quellen zusammengetragen und

zu einem einheitlichen Intensitätswert verrechnet. Die verschiedenen Quellen können beispielsweise verschiedene Sensoren sein oder, wie in der vorliegenden Arbeit, unterschiedliche Merkmale, die aus den Daten eines Sensors ermittelt werden. Im Falle der Objekterkennung bzw. -verfolgung wird die Intensität für jedes Pixel im Bild berechnet (*Map*). Diese signalisiert, zu welchem Grad das Messverfahren an dieser Stelle im Bild eines der gesuchten Objekte vermutet (*Confidence*). Wie schon Eingangs im Abschnitt 3.1 erwähnt, setzt sich die hier vorgestellte Konfidenzkarte aus verschiedenen Merkmalen zusammen: silhouettenbasierte und überlappungsbasierte Merkmale, farbbasierte Merkmale, texturbasierte Merkmale ((*Histogram Of Oriented Gradients* (Dalal und Triggs 2005)) sowie vorhersagebasierte Merkmale. Die Konfidenzkarten von drei Bildausschnitten sind in Abbildung 3.7 dargestellt. Die einzelnen Merkmale werden im Folgenden im Detail vorgestellt.

Seien die Outfithistogramme H^C mit der Outfitfunktion $o(\cdot)$ aus Abschnitt 3.6.5, ein Bild I sowie die zugehörige Vordergrundregion $\overline{FG}$ aus Abschnitt 3.5 und eine Menge von Bounding-Boxen $B := \{B_1, \ldots, B_{n_t}\}$ zu einem Zeitpunkt t gegeben (der Index t entfällt im Folgenden aus Übersichtsgründen). Seien $C := \{C_1, \ldots, C_m\}$ die Zusammenhangskomponenten von $\overline{FG}$. Jede Bounding-Box wird derjenigen Zusammenhangskomponente zugeordnet, mit der sie die größte Überschneidung hat:

$$c(B_i) := \begin{cases} \arg\max_{C_j \in C} |C_j \cap B_i| & \text{für } B_i \cap \overline{FG} \neq \emptyset \\ R_i^{\emptyset} & \text{sonst} \end{cases}, \tag{3.18}$$

wobei $R_i^{\emptyset}$ eine Bildregion mit der Fläche Null, also $|R_i^{\emptyset}| = 0$ ist. Sei nun

$$C^*(B) := \{c(B_i) \mid B_i \in B\} \tag{3.19}$$

die Menge aller Zusammenhangskomponenten, die eine maximale Überschneidung mit einer Bounding-Box haben, vereint mit der Menge von Bildregionen, die für jede Bounding-Box B_i mit $B_i \cap \overline{FG} = \emptyset$ eine Bildregion $R_i^{\emptyset}$ mit $|R_i^{\emptyset}| = 0$ enthält. Jeder Region $R_j \in C^*(B)$ ist somit eine Menge ${}_jB$ von Bounding-Boxen zugeordnet mit ${}_jB := \{B_i \mid c(B_i) = R_j\} := \{{}_j\widetilde{B}_1^t, \ldots, {}_j\widetilde{B}_{n_R}^t\}$.

Im den folgenden Abschnitten werden Divisionen nur berechnet, wenn der Divisor ungleich 0 ist. Ansonsten ist das Ergebnis der Division 0. Alle Berechnungen beziehen sich auf eine Region $R_j \in C^*(B^t)$ und den jeweils zugeordneten Bounding-Boxen. Zugunsten der Übersichtlichkeit wird dabei der Index j weggelassen und $n_R := |{}_jB| \in \mathbb{N}^*$ als die Anzahl der zugeordneten Bounding-Boxen definiert. In den meisten Fällen gilt $n_R = 1$.

$n_R > 1$ tritt auf, wenn sich mehrere Objekte überlappen und gemeinsam zu einer großen Vordergrundkomponente verschmelzen.

3.7.1 Silhouettenbasierte Konfidenz

Die silhouettenbasierten Merkmale sind inspiriert von den Kompaktheits- (*Compactness Constraints*) und Größenbedingungen (*Height Constraints*) aus (Beetz u. a. 2007) und (Gedikli u. a. 2007). Sie basieren auf der Annahme, dass bei einer korrekten Verfolgung, die segmentierte Vordergrundkomponente eines Objektes vollständig von der zugehörigen Bounding-Box überlagert ist. Diese Idee wird hier aufgegriffen und durch Sensitivität, positiven Vorhersagewert und F_1-Maß formalisiert (siehe dazu beispielsweise (Powers 2011)).

Das Merkmal der Sensitivität (*Recall*) $c_{rec} \in [0;1]$ beschreibt den Grad der Überdeckung der Zusammenhangskomponente R durch die Vereinigung der Bounding-Boxen $\{\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}\}$:

$$c_{rec}(\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}, R) := \frac{|(\bigcup_{i=1}^{n_R} \widetilde{B}_i) \cap R|}{|R|}. \tag{3.20}$$

Das Merkmal des positiven Vorhersagewerts (*Precision*) $c_{pre} \in [0;1]$ bestraft Bereiche innerhalb der Bounding-Boxen, die nicht von der Vordergrundregion abgedeckt sind. Um Positionen mit einer gleichmäßigen Verteilung entlang von horizontaler und vertikaler Achse zu bevorzugen, wird jede Bounding-Box $\widetilde{B}_i$ in vier gleich große Teile aufgeteilt. Das resultiert in den Bounding-Boxen für das linke obere Viertel ${}^{tl}\widetilde{B}_i$, das rechte obere Viertel ${}^{tr}\widetilde{B}_i$, das linke untere Viertel ${}^{bl}\widetilde{B}_i$ und das rechte untere Viertel ${}^{br}\widetilde{B}_i$. c_{pre} ist als das Minimum über alle Rechteckteile wie folgt definiert:

$$c_{pre}(\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}, R) := \min_{\widetilde{B}_i} \min_{o \in \{tl,tr,bl,br\}} \frac{|{}^{o}\widetilde{B}_i \cap R|}{|{}^{o}\widetilde{B}_i|} \tag{3.21}$$

Die Wahl des Minimums sorgt dafür, dass eine ausreichende Überdeckung aller Teile aller Bounding-Boxen gewährleistet ist. Das geometrische Mittel aus Sensitivität und positiver Vorhersage wird auch als F_1-Maß bezeichnet und bildet die silhouettenbasierte Konfidenz c_1 mit

$$c_1(\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}, R) := 2 \cdot \frac{c_{rec} \cdot c_{pre}}{c_{rec} + c_{pre}} \tag{3.22}$$

3.7.2 Überlappungsbasierte Konfidenz

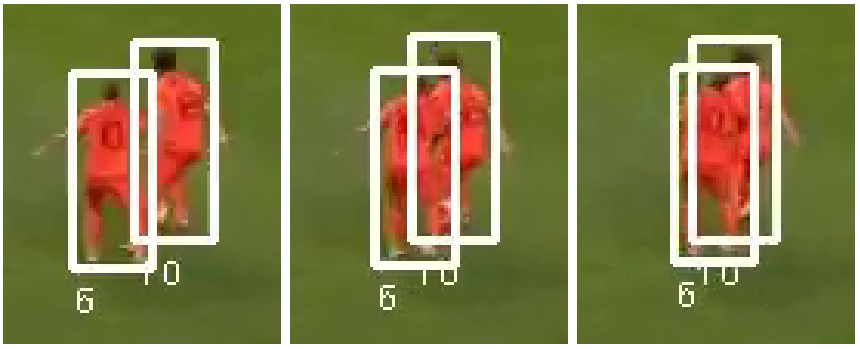


Abbildung 3.8: Beispiel einer Situation in der die überlappungsbasierte Konfidenz dafür sorgt, dass die beiden Spieler weiter einzeln verfolgt werden. Originalbild aus (Das Erste 2012).

Es kommt sehr selten vor, dass sich zwei Spieler komplett überlappen. Meistens ist noch ein Teil des hinteren Spielers zu sehen. Dennoch hat in der Regel der vordere Spieler, da er komplett sichtbar ist, die höhere optische Evidenz (insbesondere wenn die zwei Spieler aus dem gleichen Team sind). In Abbildung 3.8 ist eine solche Situation mit drei aufeinander folgenden Einzelbildern dargestellt. Um dies auszugleichen, bestraft die überlappungsbasierte Konfidenz c_2 Bounding-Boxen, die sich stark überlappen und sorgt so für eine Art Abstoßungseffekt:

$$c_2(\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}, R) := 1 - \max_{i,j} \frac{|\widetilde{B}_i \cap \widetilde{B}_j|}{|\widetilde{B}_i \cup \widetilde{B}_j|} \tag{3.23}$$

Für $n_R > 2$ werden durch die Maximumoperation die zwei Bounding-Boxen bewertet, die die größte Überschneidung innerhalb der Menge haben.

3.7.3 Farbbasierte Konfidenz

Die farbbasierte Konfidenz c_3 basiert auf den Outfit-Farbhistogrammen aus Abschnitt 3.6.5 und der Outfitfunktion $o(B_i)$, die jeder Bounding-Box B_i das Farbhistogramm des zugehörigen Outfits zuordnet:

$$c_3(\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}, R|I) := 1 - \sqrt[n_R]{\prod_{i=1}^{n_R} d_H(\vec{h}(\widetilde{B}_i), o(\widetilde{B}_i))} \tag{3.24}$$

3.7.4 Texturbasierte Konfidenz

HOG-Merkmale (*Histogram of Oriented Gradients*, siehe (Dalal und Triggs 2005)) haben sich als Hilfsmittel bei der Erkennung von Personen hervorgehoben. Dabei wird das Bild in überlappende Rechtecke aufgeteilt und die Merkmale grob in den folgenden drei Schritten berechnet: Gradientenberechnung, Histogrammberechnung (bezüglich der Gradientenrichtung) und Kontrastnormalisierung. Mit den berechneten Merkmalen eines Trainingsdatensatzes wird ein Klassifikator trainiert, welcher in einem *Sliding Window* Ansatz mit Bildpyramiden zur Objekterkennung genutzt wird. In der vorliegenden Arbeit werden die Standardparameter aus (Dalal und Triggs 2005) sowie eine lineare *Support Vector Machine* (siehe (Fan u. a. 2008) und (Bishop 2006, S. 325 ff.)) als Klassifikator genutzt. Für die Konfidenzkarte wird der Klassifikator für die berechneten Merkmale innerhalb einer Bounding-Box wie folgt angewandt: Für jede Bounding-Box $\widetilde{B}_i$ wird ein in der Größe angepasstes Teilbild der Größe von 64×128 Pixel erzeugt, so dass

- die Höhe der Bounding-Box im Teilbild 100 Pixel entspricht,
- das Teilbild den gleichen Mittelpunkt wie die Bounding-Box besitzt und
- das Seitenverhältnis der Pixel nicht verändert wird.

Der Personendetektor berechnet die HOG-Merkmale für dieses Teilbild und gibt den Entscheidungswert $d_B(\widetilde{B}_i) \in \mathbb{R}$ der linearen *Support Vector Machine* zurück. Dieser Wert wird mit den Normalisierungsparametern u_D und v_D auf

$$d'_D(\widetilde{B}_i|I, u_D, v_D) := \begin{cases} 1 & \text{für } d_B(\widetilde{B}_i) \geq v_D \\ \frac{d_B(\widetilde{B}_i) - u_D}{v_D - u_D} & \text{für } u_D < d_B(\widetilde{B}_i) < v_D \\ 0 & \text{sonst} \end{cases} \tag{3.25}$$

abgebildet.

Die texturbasierte Konfidenz c_4 ist über das geometrische Mittel definiert:

$$c_4(\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}, R|I, u_D, v_D) := \sqrt[n_R]{\prod_{i=1}^{n_R} d'_D(\widetilde{B}_i|u_D, v_D)} \tag{3.26}$$

3.7.5 Vorhersagebasierten Konfidenz

Bei der Objektverfolgung wird in der Regel der zeitliche Kontext genutzt und die Position des verfolgten Objekts in einer Videosequenz von einem Bild zum nächsten Bild geschätzt (beispielsweise durch die Annahme eines Bewegungsmodells, siehe Abschnitt 4.3.2). Liegt eine solche geschätzte Vorhersage vor, können Messungen aufgrund der Übereinstimmung mit der Vorhersage (falls vorhanden) bewertet oder ausgeschlossen werden. Das heißt, der Suchraum wird durch Einbezug von zeitlicher Information eingeschränkt und gewichtet. In der Literatur wird dabei von einer Messungsselektion (*Gating*) gesprochen (siehe zum Beispiel (Blackman und Popoli 1999)). Seien nun eine Bounding-Box B und die zugehörige Vorhersage B_p mit Breite w_p und Höhe h_p sowie die jeweiligen Schwerpunkte $\vec{c}$ und $\vec{c}_p$ gegeben. Dann ist die Vorhersagedistanz $d_p(B) \in [0;1]$ definiert als der euklidische Abstand der Schwerpunkte im Verhältnis zur Länge der Diagonalen der Vorhersage:

$$d_p(B) := \min(1, \frac{\|\vec{c} - \vec{c}_p\|_2}{\sqrt{w_p^2 + h_p^2}}). \tag{3.27}$$

Die vorhersagebasierte Konfidenz c_5 ist der Mittelwert der Vorhersagedistanzen der Bounding-Boxen:

$$c_5(\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}, R) := \frac{\sum_{i=1}^{n_R} d_p(\widetilde{B}_i)}{n_r} \tag{3.28}$$

3.7.6 Ensemble Averaging

In Anlehnung an die Daten- und Sensorfusion auf Merkmalsebene (siehe Klein 2004, S. 92 ff.) werden die einzelnen Konfidenzen $c_i, i \in \{1, \ldots, 5\}$ mit den Gewichten $w_i \in [0;1], i \in \{1, \ldots, 5\}$ zur Gesamtkonfidenz c verrechnet:

$$c(\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}, R|I, u_D, v_D, w_1, \ldots, w_5) := \frac{\sum_{i=1}^{5} w_i \cdot c_i(\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}, R|I, u_D, v_D)}{\sum_{i=1}^{5} w_i} \tag{3.29}$$

Wegen $c_i(\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}, R) \in [0;1]$ gilt auch $c(\widetilde{B}_1, \ldots, \widetilde{B}_{n_R}, R|w_1, \ldots, w_5) \in [0;1]$. Für $|R^t| = 0$ ist $w_1 = 0$ und für $n_r = 1$ ist $w_2 = 0$. Falls keine Vorhersagen vorhanden sind gilt $w_5 = 0$.

Die Wahl der Parameter für die Berechnung der Konfidenzkarte kann im Anhang B aus Tabelle B.5 entnommen werden.

3.8 Auffinden lokaler Konfidenzmaxima

Wie bereits in Abschnitt 3.7.5 erwähnt, wird bei der Objektverfolgung versucht, die aktuelle Position eines oder mehrerer Objekte zu detektieren, ausgehend von einer gegebenen Position des / dieser Objekte (beispielsweise die Position im letzten Frame oder die vorhergesagte Position). Dabei werden lokale Maxima der Konfidenz aus 3.7.6 gesucht, die sich in der Umgebung der gegebenen Position befinden.

Eine naive Möglichkeit ist es, die Konfidenz für alle möglichen Positionen in der Umgebung der gegebenen Position zu evaluieren und die Position mit der höchsten Konfidenz zu ermitteln. Alternativ können auch gradientenbasierte Verfahren wie Verfahren des steilsten Abstiegs mit Liniensuche (Nocedal und Wright 2006a) oder das *Mean Shift*-Verfahren (Comaniciu u. a. 2003) angewendet werden. Bei all diesen Verfahren kann es in Abhängigkeit von n_R und der Größe der gewählten Umgebung zu einer großen Anzahl von notwendigen Berechnungen kommen.

Um die notwendige Anzahl der Berechnungen zu reduzieren, wird in dieser Arbeit in Analogie zum Bergsteigeralgorithmus (*Hill Climbing* (siehe Russell und Norvig 2003, S. 325 ff.)) auf eine lokale Suche zurückgegriffen. Dabei werden zum einen relativ große Schrittweiten gewählt. Dies hat den Vorteil, dass mit wenigen Schritten ein Maximum erreicht werden kann. Es birgt allerdings die Gefahr, dass nicht die exakte Position des Maximums ermittelt wird. Um diesen Nachteil abzumildern, werden zum anderen bei der lokalen Suche auch Schritte in Richtung des Gradienten untersucht.

Im Folgenden repräsentiert ein Vektor $\vec{x} \in \mathbb{R}^{2n_R}$ die linken oberen Ecken von n_R Bounding-Boxen $B_1, \ldots, B_{n_R}$ mit $\vec{x} := (x_1, y_1, \ldots, x_{n_R}, y_{n_R})^T$. Zudem ist eine Funktion $f : \mathbb{R}^{2n_R} \to \mathbb{R}$ gegeben, die über die Konfidenz c aus 3.7.6 definiert ist. Für die Berechnung von $f(\vec{x})$ wird $c(B_1, \ldots, B_{n_R})$ ausgewertet, indem Breite und Höhe der Bounding-Boxen gleich bleiben und die linken oberen Ecken durch $\vec{x}$ festgelegt werden.

Sei nun eine initiale Position $\vec{x}_0 \in \mathbb{R}^{2n_R}$ und eine Schrittweite $\Delta s \in \mathbb{N}^*$ gegeben. Das detaillierte Vorgehen ist in Algorithmus 3.5 veranschaulicht. In den Zeilen 6 bis 8 wird die partielle Ableitung $\frac{\partial f}{\partial x_i}(\vec{x})$ nach dem i-ten Eintrag von $\vec{x}$ und somit der i-te Eintrag des Gradienten $\vec{\nabla} \in \mathbb{R}^{2n_R}$ durch einen zentralen Differenzenquotient angenähert. Dabei ist $\vec{e}_i$ der i-te kanonische Einheitsvektor. Die Positionen, an denen f für den Differenzenquotient ausgewertet wird, werden zusätzlich bei der lokalen Suche nach einem Maximum in den Zeilen 9 bis 12 berücksichtigt. Nach der Normierung des approximierten Gradienten in Zeile 13 wird für die lokale Suche ein Schritt in Gradientenrichtung vollzogen und f an dieser Position ausgewertet. Gibt es eine Steigerung von f an einer der $2 \cdot n_R + 1$ ausgewerteten Positionen, so wird die Position mit dem maximalen Wert übernommen.

Dieses Vorgehen wird solange iterativ wiederholt, bis keine Position mit einem höheren Wert von f erreicht werden kann.

Die gewählte Schrittweite Δs hat einen maßgeblich Einfluss zum einen auf die Genauigkeit mit der die Position eines lokalen Maximums gefunden wird und zum anderen auf die Anzahl der Iterationen bis zur Konvergenz. Bei kleinen Schrittweiten können die Positionen von Maxima auf Kosten einer großen Anzahl Iterationen genauer bestimmt werden. Bei großen Schrittweiten tritt genau das Gegenteil ein und es besteht die Gefahr, dass ein Maximum übersprungen wird. Allerdings können mit größeren Schrittweiten Plateaus besser überwunden werden.

Algorithmus 3.5 : Auffinden lokaler Konfidenzmaxima

Input : Initiale Position $\vec{x}_0 \in \mathbb{R}^{2n_R}$

Zu maximierende Zielfunktion $f : \mathbb{R}^{2n_R} \to \mathbb{R}$

Schrittweite $\Delta s \in \mathbb{N}^*$

Output : Position $\vec{x}_{max} \in \mathbb{R}^{2n_R}$

1 $\vec{x}_{max} \leftarrow \vec{x}_0,\ v_{max} \leftarrow f(\vec{x}_0)\ v_{old} \leftarrow 0$ `/* Initialisierung */`

2 **while** $(v_{max} - v_{old}) > 0$ **do**

3 $\quad \vec{x} \leftarrow \vec{x}_{max}$

4 $\quad v_{old} \leftarrow v_{max}$

5 $\quad$ **foreach** $i \in \{1, \ldots, n_R\}$ **do** `/* Iterations bis keine Veränderung mehr */`

6 $\quad\quad \vec{x}_f \leftarrow \vec{x} + \frac{\Delta s}{2} \cdot \vec{e}_i$ `/* Schritt vorwärts in Richtung` i `*/`

7 $\quad\quad \vec{x}_b \leftarrow \vec{x} - \frac{\Delta s}{2} \cdot \vec{e}_i$ `/* Schritt rückwärts in Richtung` i `*/`

8 $\quad\quad \vec{\nabla}[i] \leftarrow \frac{f(\vec{x}_f) - f(\vec{x}_b)}{\Delta s}$ `/*` i`-ter Eintrag Gradient */`

9 $\quad\quad \vec{x}_m \leftarrow \arg\max_{\vec{x} \in \{\vec{x}_f, \vec{x}_b\}} f(\vec{x})$

10 $\quad\quad$ **if** $f(\vec{x}_m) > v_{max}$ **then** `/* Aktuelles Maximum */`

11 $\quad\quad\quad v_{max} \leftarrow f(\vec{x}_m)$

12 $\quad\quad\quad \vec{x}_{max} \leftarrow \vec{x}_m$

13 $\quad \vec{\nabla} \leftarrow \frac{\vec{\nabla}}{|\vec{\nabla}|}$ `/* Normierung des Gradienten */`

14 $\quad$ **if** $f(\vec{x} + \Delta s \cdot \vec{\nabla}) > v_{max}$ **then** `/* Lokale Suche in Gradientenrichtung */`

15 $\quad\quad v_{max} \leftarrow f(\vec{x} + \Delta s \cdot \vec{\nabla})$

16 $\quad\quad \vec{x}_{max} \leftarrow \vec{x} + \Delta s \cdot \vec{\nabla}$

17 **return** $\vec{x}_{max}$

3.9 Diskussion und Ausblick

In diesem Kapitel wurde eine Kombination von Merkmalen vorgestellt, die auf die robuste Erkennung und Verfolgung von Fußballspielern zugeschnitten ist. Bei der Erstellung dieser Konfidenzkarte wird vorhandenes Domänenwissen gezielt ausgenutzt, um die Silhouetten der Spieler zu extrahieren und die Trikotfarben der Mannschaften zu bestimmen und als Farbvorlagen zu benutzen. Im Gegensatz zu vergleichbaren Verfahren werden zusätzlich allgemeingültige Merkmale eingebunden, wie die Personendetektion, die Schätzung der Personengrößen ohne Kameraparameter oder die überlappungsbasierte Konfidenz. Dadurch ist der Ansatz in der Lage, Situationen zu beherrschen, in denen die domänenspezifischen Annahmen nicht zutreffen oder nicht ausreichend modelliert sind. Das vorgestellte Optimierungsverfahren ermöglicht als Kombination von zufallsbasierter Suche und Gradientenaufstieg eine effiziente lokale Suche in der Konfidenzkarte und bereitet somit den Weg zu einer echtzeitfähigen Lösung.

Durch die explizite Trennung von domänenspezifischen und allgemeingültigen Merkmalen, kann der vorgestellte Ansatz leicht für andere Domänen angepasst werden. So können die Extraktion der Spielersilhouetten und die Bestimmung der Trikottemplates mit wenigen Änderungen auch für andere feldbasierte Sportspiele angewendet werden. In Kombination mit einer geeigneten Hintergrundschätzung und der Vorgabe von dominanten Farben der Kleidung ist auch eine Anwendung im Überwachungsbereich denkbar.

Die allgemeingültigen Merkmale haben eine unterstützende Funktion und sind in der Regel für sich alleinstehend nicht für eine robuste Erkennung und Verfolgung geeignet. So wird das System scheitern, wenn das modellierte Domänenwissen deutlich von den tatsächlichen Gegebenheiten abweicht. Denkbare Beispiele sind ein Spielfeld mit einer komplett anderen Farbe als grün oder eine Videosequenz, in der in den ersten Bildern, die zur Extraktion der Mannschaftsfarben genutzt werden, nur ein Torwart zu sehen ist. Solche Situationen können durch eine manuelle Nachbesserung, wie die Vorgabe des ungefähren Farbtons des Felds oder der Markierung von Spielertrikots im späteren Verlauf der Sequenz, aufgelöst werden.

Die Leistungsfähigkeit von maschinellen Lernverfahren, insbesondere von tiefen neuronalen Netzen, hat sich in den letzten Jahren drastisch verbessert und die Bild- und Objekterkennung revolutioniert (siehe He u. a. (He u. a. 2015)) Diese Techniken sind möglicherweise mittlerweile in der Lage, einige der genutzten Merkmale zu verbessern, beispielsweise die Personenerkennung oder die Graskennung. Doch einige Merkmale lassen spezifisches Kontextwissen einfließen, wie beispielsweise die Schätzung der Spielergröße oder die überlappungsbasierte Konfidenz, welche schwierig in einem neuronalen

Netz integriert werden können. Ein weiteres Problem von solchen Verfahren ist die benötigte Anzahl an Trainingsbeispielen, selbst bei einer Anpassung von bereits trainierten Netzwerken, die einen enormen Aufwand an manueller Annotation erfordert, um eine Überanpassung zu vermeiden.

Die Berechnung der Merkmale ist der Flaschenhals bei der Erkennung und Verfolgung der Spieler. Doch neben der lokalen Suche in der Konfidenzkarte, bei der sich Werte an verschiedenen Positionen unabhängig voneinander berechnen lassen, lässt sich auch die Bestimmung der einzelnen Merkmale, wie die Erstellung von Farbhistogrammen oder die Personendetektion, sehr gut auf entsprechender Hardware parallelisieren. Dies ist eine wesentliche Voraussetzung für die potentielle Echtzeitfähigkeit des gesamten Systems.

Kapitel 4

Spielerverfolgung

Die Inhalte dieses Kapitels basieren zu Teilen auf folgender Veröffentlichung:

Herrmann, M., Hoernig, M. und Radig, B. (2014). „Online Multi-player Tracking in Monocular Soccer Videos“. In: *AASRI Procedia* 8, S. 30–37.

4.1 Einleitung

4.1.1 Online 2D-Spielerverfolgung

Im ersten Teil dieses Kapitels wird ein Online-Verfahren vorgestellt, das auf Basis der Merkmale aus Kapitel 3 die Verfolgung der Spieler im Bildraum innerhalb einer Videosequenz vollautomatisch vornimmt. Im Gegensatz zu Offline-Verfahren, bei denen die gesamten Bilddaten einer Videosequenz zur Erzeugung der Ergebnisse genutzt werden, werden bei Online-Verfahren zur Berechnung der Ergebnisse zur Zeit t auch nur die Bilddaten bis zur Zeit t genutzt (Laplante 2001, Seite 343). Das bedeutet für die Spielerverfolgung, dass die ermittelten Spielerpositionen prinzipiell in Echtzeit ausgegeben werden können. Die Eigenständigkeit der 2D-Spielerverfolgung ist von Bedeutung, da in bestimmten Situation eine Transformation der Bildkoordinaten in 3D-Feldkoordinaten nicht möglich oder sehr fehlerbehaftet ist, wie beispielsweise bei starken Zooms. Der Input des Systems besteht lediglich aus einer Videosequenz und als Output stellt das Verfahren die Position (als Pixelkoordinaten), die Größe (in Form von Bounding-Boxen in Pixel) und die Mannschaftszugehörigkeit der Spieler in jedem Einzelbild zur Verfügung. Dabei kommen folgende Techniken zum Einsatz:

- **Stochastische Filterung:** Mit Hilfe eines Kalman-Filters (Kálmán 1960) werden die Trajektorien der Spieler im Bild behandelt. Das dient zum einen dazu,

Messungenauigkeiten auszugleichen und glatte Trajektorien zu generieren. Zum anderen wird das inkorporierte Bewegungsmodell genutzt, um den Suchraum im nachfolgenden Einzelbild zu beschränken (siehe dazu auch Abschnitt 3.7.5).

- **Messvorgang:** Die Messung ist der zentrale Vorgang bei der 2D-Spielerverfolgung. Neben der Vorhersage der stochastischen Filterung, die genutzt wird, um den Suchraum einzuschränken, finden hier in erster Linie die Konfidenzkarte und die Suche nach lokalen Maxima aus Kapitel 3 ihren Einsatz. Die Kombination der verschiedenen Merkmale erlaubt ein robustes Verfahren, das den Ausfall einzelner Diskriminierungsmerkmale kompensieren kann.

- **Automatische initiale Spielererkennung:** Zu Beginn einer Videosequenz ist nichts über die Position von Spielern bekannt. Die automatische Erkennung nutzt eine Personendetektion und die Merkmale des Messvorgangs, um die initialen Spielerpositionen im ersten Einzelbild zu bestimmen. Dabei wird ein iteratives Vorgehen angewandt, das durch die Einschränkung des Suchraums (abhängig von der Messung in der vorherigen Iteration) die Robustheit steigert.

- **Bewegungsmodell:** Dem Bewegungsmodell kommt in diesem Ansatz eine wichtige Rolle zu, da neben der Filterung der Spielertrajektorien auch der Suchraum der Messung davon abhängt. Zudem müssen schnelle Kamerabewegungen von dem Modell entweder modelliert oder zumindest toleriert werden können. Im Sinne von *Ockham's Razor* wird hier bewusst ein einfaches lineares Modell gewählt. Neben der reduzierten Zahl an Freiheitsgraden, bietet es vor allem die Möglichkeit einer optimalen und effizienten stochastischen Filterung. Bewegungen der Kamera werden durch eine hohe Amplitude des Prozessrauschens modelliert.

- **2D-Spielerverfolgung:** Die eigentliche Spielerverfolgung besteht nicht nur darin, die Messergebnisse und die gefilterten Trajektorien zu verwalten. Ebenso wichtig ist es, den Verlust und die Neuerkennung von Spielern (beispielsweise bei Verlassen des Bildbereichs beziehungsweise bei Eintritt in den Bildbereich) zu modellieren. Dabei kommen plausibilitätsbasierte Regeln zum Einsatz.

4.1.2 Verfeinerung der 3D-Trajektorien

Aus den Trajektorien im 2D-Bildraum kann vor allem für eine statistische oder taktische Auswertung einer Fußballpartie noch nicht viel Nutzen gezogen werden. Daher ist ein entscheidender Schritt, mit Hilfe einer Abbildung die Trajektorien vom 2D-Bildraum in den 3D-Spielfeldraum abzubilden. Im Dreidimensionalen können die Trajektorien durch Plausibilitätsannahmen in der realen Welt (maximale physikalische Geschwindigkeit und

Ähnliches) weiter verbessert werden und Mehrdeutigkeiten im Bild aufgelöst werden. Dies geschieht in den folgenden Schritten:

- **Transformation vom Bild ins Spielfeld:** Der entscheidende Schritt ist die Bestimmung einer Abbildung (Homographie) von der Bildebene in die Spielfeldebene. Hierzu werden vor allem die Feldlinien als Landmarken genutzt. Falls diese unzureichend sichtbar sind (beispielsweise bei einem hohen Zoomfaktor) können auch auffällige Texturen im Rasen (falls vorhanden) verfolgt werden. Zu diesem Zweck wird auf das Verfahren von Hoernig (Hoernig 2016) zurückgegriffen.

- **Auftrennung und Glättung der Trajektorien:** Vor einer Weiterverarbeitung werden die Spielertrajektorien in unterbrechungsfreie Abschnitte aufgeteilt. Der Vorteil einer stochastischen Glättung (beispielsweise der Rauch-Tung-Striebel-Glättung (Rauch u. a. 1965), einer Erweiterung des Kalman-Filters) ist es, dass sie nicht nur für jeden Zeitpunkt die vergangenen Messungen berücksichtigt (wie bei der Filterung), sondern auch die nachfolgenden Messungen einfließen lässt. Zudem ist es möglich, durch die bereits im ersten Schritt „entfernten“ Kamerabewegungen und die physikalischen Grenzen in der realen Welt (beispielsweise die maximale Beschleunigung), die Trajektorien sinnvoller zu glätten als das im Zweidimensionalen möglich ist.

- **Entfernung falsch-positiver Erkennungen:** Ein weiterer Vorteil der Trajektorien in der realen Welt ist es, dass mit einfachen Regeln falsch-positive Erkennungen ausgeschlossen werden. Es ist beispielsweise im normalen Spielbetrieb sehr unwahrscheinlich, dass sich ein Spieler während der Dauer einer kompletten Videosequenz permanent an der Mittellinie aufhält.

- **Wiedervereiningung der Trajektorien:** Für weitere Analysen (beispielsweise die Ermittlung des Ballbesitzes) ist es wichtig, dass Trajektorien eindeutig einer Spieleridentität zugeordnet sind. Durch Überdeckungen und durch das Betreten und Verlassen des Bildbereiches können (eigentlich) zusammengehörige Trajektorien Unterbrechungen enthalten. Ziel in diesem Schritt ist es, über räumlich-zeitliche Bedingungen die unterbrochenen Trajektorien einzelner Spieler richtig zusammenzuführen.

4.2 Stand der Forschung

Stochastische Filterung und Bewegungsmodell

Als Alternativen zum Kalman-Filter (Kálmán 1960) können beispielsweise sequenzielle

Monte-Carlo-Methoden (Partikelfilter) (Arulampalam u. a. 2002; Doucet und Johansen 2011) oder die *Unscented Transform* (Julier und Uhlmann 1997) aufgeführt werden. Der Vorteil dieser Verfahren liegt vor allem in der Möglichkeit, nicht-lineare Bewegungsmodelle zu verwenden. Allerdings sind die nicht-linearen Elemente (beispielsweise Kamerabewegung oder abrupte Spielerbewegungen) sehr schwierig zu modellieren. Komplexere Modelle laufen immer Gefahr, durch die größere Anzahl an Freiheitsgraden, das Verfahren zu destabilisieren. Die Wahl fällt in dieser Arbeit auf ein lineares Bewegungsmodell mit normal-verteiltem Prozessrauschen, das einfach und effizient ist und dennoch in den meisten Fällen gute Resultate liefert. Hierbei sind der Kalman-Filter beziehungsweise sein Pendant die Rauch-Tung-Striebel -Glättung (Rauch u. a. 1965) die geeigneten Methoden. Zum einen, da er im Sinne der Minimierung des quadratischen Fehlers optimal ist und zum anderen, da der Aufwand für die Berechnung deutlich geringer als bei anderen Verfahren, wie dem Partikelfilter, ist.

Messvorgang

Die silhouettenbasierten Konfidenzmerkmale sind inspiriert von Beetz u. a. (Beetz u. a. 2007; Gedikli u. a. 2007). Der vorgestellte Ansatz ist ähnlich zu der Arbeit von Gerke u. a. (Gerke u. a. 2013), die ein reines Detektionssystem vorstellen. Eine Verfolgung der Spieler über die Zeit wurde dabei allerdings nicht behandelt. Viele Verfahren zur Personenverfolgung basieren auf einer vorausgehenden Personendetektion (*Tracking-by-Detection*), wie beispielsweise die Ansätze von Breitenstein u. a. (Breitenstein u. a. 2011) oder von Izadinia u. a. (Izadinia u. a. 2012). Die Erkennungen in den einzelnen Bildern werden dabei zumeist durch ein stapelbasiertes Verfahren (*Batch Processing*) zu Trajektorien verbunden. Bei kleinen Objekten weisen solche Detektoren, wie beispielsweise von P. Felzenszwalb u. a. (P. Felzenszwalb u. a. 2008) oder von Dollár, Wojek u. a. (Dollár u. a. 2009; Dollár, Wojek u. a. 2012; Dollár, Appel und Kienzle 2012; Dollár u. a. 2014; Nam u. a. 2014), meist schlechte Genauigkeiten und einen hohen Berechnungsaufwand auf. Bei Fußballaufnahmen sind Spieler mit einer Größe von 40 Pixeln und weniger keine Seltenheit, insbesondere bei geringen Auflösungen oder Szenen mit Weitwinkel. Zudem haben die stapelbasierten Verfahren den Nachteil, dass sie nicht online durchgeführt werden können, sondern erst nach Durchführung der Personendetektion auf der gesamten Sequenz. Bei einer naiven Anwendung von Kernel-basierten Ansätzen, wie dem *Mean-Shift*-Verfahren (Comaniciu u. a. 2003), wird eine Form der Initialisierung benötigt und die Verfahren laufen immer Gefahr, vom eigentlichen Ziel abzudriften. Zudem ist dazu pro Zeitschritt eine große Anzahl an Merkmalsberechnungen notwendig. Der hier vorgestellte Ansatz versucht die Vorteile beider Ansätze zu verbinden. Das Verfahren von J. Zhang u. a. (J. Zhang u. a. 2012) konnte mit einer ähnlichen Idee Ergebnisse erzielen, die den Stand der Technik übertroffen haben. Sie benutzen ebenfalls Kalman-Filter

für die Verfolgung der einzelnen Objekte und eine Kombination aus Personendetektor, Farbhistogrammen und der geschätzten Personengröße als Merkmal. Im Gegensatz zu dem in der vorliegenden Arbeit vorgestellten Ansatz, wird zum einen für jedes Bild eine vollständige Personensuche durchgeführt. Zum anderen wird im Messvorgang das *Mean-Shift*-Verfahren (Comaniciu u. a. 2003) genutzt, bei dem die Anzahl der Stellen an denen die Merkmale berechnet werden sehr groß ist. Beides führt zu einem vergleichsweise deutlich höheren Berechnungsaufwand, wie es in Tabelle 6.9 in Abschnitt 6.5.4 abzulesen ist.

Initiale Spielererkennung

Das Ziel der initialen Spielererkennung ist es, im ersten Bild (oder in den ersten Bildern) einer Sequenz die Position der sichtbaren Spieler zu lokalisieren. In der Literatur wird häufig zur Erkennung von Personen ein Detektor wie von Dalal und Triggs (Dalal und Triggs 2005; Dalal 2006), P. Felzenszwalb u. a. (P. Felzenszwalb u. a. 2008) oder Dollár, Wojek u. a. (Dollár, Wojek u. a. 2012) genutzt. Wenn die verwendeten Klassifikatoren mit einem repräsentativen Trainingsdatensatz eingelernt werden, weisen sie eine Allgemeingültigkeit auf und kommen mit unterschiedlichen Bedingungen gut zurecht. Auf der anderen Seite sind sie nicht für kleine Objekte optimiert, wie sie in Fußballaufzeichnungen häufig vorkommen, und weisen dabei eine schlechte Erkennungsrate sowie einen hohen Berechnungsaufwand auf. Zudem ist es schwierig, formelles Kontextwissen in diese Verfahren zu integrieren, obwohl das den Untersuchungen von Divvala u. a. (Divvala u. a. 2009) zu Folge, die Erkennungsleistung signifikant steigern kann. In der Domäne Fußball wird Kontextwissen bei der Erkennung von Spielern genutzt, indem die Unterscheidbarkeit von Vordergrund und Hintergrund durch die Grasfarbe zur binären Segmentierung eingesetzt wird, wie beispielsweise in den Arbeiten von Figueroa u. a. (Figueroa u. a. 2006b), Miura und Kubo (Miura und Kubo 2008), Pallavi u. a. (Pallavi u. a. 2008), Hoyningen-Huene (Hoyningen-Huene 2011), Siles Canales (Siles Canales 2014) oder Sabirin u. a. (Sabirin u. a. 2015). Natürlich können durch morphologische Untersuchungen, wie in Abschnitt 3.5 beschrieben, die Spielersilhouetten von anderen Vordergrundregionen getrennt werden. Dennoch treten bei solchen Ansätzen in der Regel Probleme auf, wenn das Gras starke Inhomogenitäten aufweist, mehrere Spieler zusammen eine Silhouette bilden oder Spieler nur teilweise das Gras überdecken (zum Beispiel wenn sie sich mit der Bandenwerbung überlappen). In der vorliegenden Arbeit wird deshalb ein kombinierter Ansatz vorgestellt, bei dem, nach der Bestimmung des Vordergrunds, anhand von als sicher eingestuften Spielersilhouetten, die Spielergrößen geschätzt werden. Diese Information wird dazu genutzt, einen Personendetektor (Dalal und Triggs 2005) auf einem eingeschränkten Bildbereich und einer lokal eingeschränkten Skalierung

einzusetzen. Dadurch wird das aufwendige Scannen des kompletten Bildes in verschiedenen Skalierungsstufen vermieden. Dieses Vorgehen wird iterativ angewendet, wodurch mit Hilfe einer verfeinerten Größenschätzung in jeder Iteration das Erkennungsergebnis verbessert wird.

Online 2D-Spielerverfolgung

Die Anzahl an Veröffentlichungen im Bereich der Verfolgung mehrerer Objekte und im Speziellen im Bereich der Personenverfolgung ist immens. Wenn man noch die Verfolgung einzelner Objekte miteinbezieht, so vervielfacht sich dieser Wert, wie die Übersichtsartikel von Yilmaz u. a. (Yilmaz u. a. 2006), H. Yang u. a. (H. Yang u. a. 2011) oder H. Yang u. a. (Jalal und Singh 2012) zeigen.

Viele Ansätze basieren auf *Tracking-by-Detection*. Dabei liegen für jedes Bild die Ergebnisse eines Objekt- / Personendetektors vor und werden durch geeignete Verfahren zu Trajektorien verbunden. Der entscheidende Schritt dabei ist die sogenannte Datenvereinigung (*Data Association*), bei der entschieden wird, welche Erkennung welcher Trajektorie zugeordnet wird. Häufig wird ein stapelbasiertes Verfahren (*Batch Processing*) vorgeschlagen, welches auf den Erkennungen von großen Teilen oder gar der kompletten Sequenz arbeitet.

Dabei werden in der Regel graphentheoretische Ansätze verfolgt (Maggio und Cavallaro 2011, S. 136 ff.), bei denen kürzeste Wege in Netzwerken aus Erkennungen gesucht werden, wie es in den Arbeiten von L. Zhang u. a. (L. Zhang u. a. 2008), Berclaz u. a.(Berclaz u. a. 2011), Pirsiavash u. a. (Pirsiavash u. a. 2011) oder Izadinia u. a.(Izadinia u. a. 2012) vorgeschlagen wird. Diese Ansätze haben neben einer hohen Berechnungskomplexität und der Schwierigkeit, dass sie inhärent nicht online tauglich sind, den Nachteil, dass es nur mit erhöhtem Aufwand möglich ist, einfache Bewegungsmodelle zu integrieren (Collins 2012; Butt und Collins 2013).

Eine andere Herangehensweise ist die Multi-Hypothesen-Verfolgung (*Multiple-Hypothesis Tracking (MHT)*, siehe (D. Reid 1979) und (Maggio und Cavallaro 2011, S. 139 ff.)). Bei der Zuordnung von Erkennungen zu Trajektorien wird ein Hypothesenraum aufgebaut und am Ende werden die plausibelsten Hypothesen gewählt. Dieser Ansatz wird beispielsweise in der Arbeit von Beetz u. a. (Beetz u. a. 2007) bei der Verfolgung von Fußballspielern gewählt. Da bei einem globalen Ansatz der Raum der Hypothesen exponentiell wächst, sind solche Verfahren mit steigender Anzahl von verfolgten Objekten und einer hohen Überlappungswahrscheinlichkeit für längere Sequenzen nicht praktikabel. Um diese Komplexität zu beherrschen, werden in der Literatur Monte-Carlo-Methoden vorgeschlagen, bei denen der untersuchte Hypothesenraum durch zufallsbasierte Stichproben eingeschränkt wird. So nutzen zum Beispiel die Arbeiten von Benfold und I.

Reid (Benfold und I. Reid 2011) und W. Choi u. a. (W. Choi u. a. 2013) diese Technik zur Verfolgung von Fußgängern.

Generell kann man aber zusammenfassen, dass diese globalen Methoden eine hohe Komplexität aufweisen und theoretische Hypothesen im Raum betrachten, die in der Praxis nicht plausibel sind. Denn das Problem der Datenvereinigung ist (insbesondere in der Domäne Fußball) in den meisten Fällen inhärent lokal, sowohl räumlich als auch zeitlich. Wird zum Beispiel ein Spieler überdeckt oder verlässt er durch einen Kameraschwenk das Blickfeld, so tritt er meist kurze Zeit später an ähnlicher Stelle im Bild wieder auf, wenn die Überdeckung beendet ist oder die Kamera wieder zurückschwenkt. Ist dies nicht der Fall, das heißt, wird der Spieler erst nach langer Zeit wieder sichtbar, so ist es aufgrund des ähnlichen Aussehens von Spielern eines Teams meist optisch schwierig, diesen Spieler wieder richtig zu identifizieren. Aus diesem Grund wird in dieser Arbeit ein gieriger (*Greedy*) Ansatz gewählt, wie er auch von Wu und Nevatia (Wu und Nevatia 2006) oder Breitenstein u. a. (Breitenstein u. a. 2011) vorgeschlagen wird. Dabei werden einem Spieler die lokal und zeitlich nächste Erkennung zugeordnet.

4.3 Stochastische Filterung und Glättung

Verfahren zur stochastischen Filterung und Glättung (insbesondere der Kalman-Filter) sind ein gängiges Hilfsmittel, um die Robustheit von Anwendungen im Bereich der Objektverfolgung zu gewährleisten. Die folgenden Abschnitte bilden eine kurze Einführung in die Grundlagen dieser Techniken.

4.3.1 Zustands- und Messmodell

Für die bayessche Filterung bzw. Glättung wird von folgenden Annahmen ausgegangen:

- $\vec{x}_t \in \mathbb{R}^n$ ist der Zustand zum Zeitpunkt t.
- $\vec{y}_t \in \mathbb{R}^m$ ist die Messung zum Zeitpunkt t.
- Die Zustände bilden eine Markov-Kette, das heißt, dass $\vec{x}_t$, wenn $\vec{x}_{t-1}$ gegeben ist, unabhängig von allen $\vec{x}_k$ mit $k < t-1$ ist und es gilt folgender Zusammenhang:

$$\vec{x}_t := A\vec{x}_{t-1} + \vec{q}_{t-1}. \tag{4.1}$$

  Dabei ist $A \in \mathbb{R}^{n\times n}$ die Übergangsmatrix und $\vec{q}_{t-1} \sim \mathcal{N}(0,Q)$ das normalverteilte Prozessrauschen mit Kovarianzmatrix $Q \in \mathbb{R}^{n\times n}$.

- Die Messung $\vec{y}_t$ ist, wenn $\vec{x}_t$ gegeben ist, unabhängig (*Conditionally Independent*) von allen anderen Messungen $\vec{y}_k$ und Zuständen $\vec{x}_k$ mit $k < t$ und es gilt folgender Zusammenhang:

$$\vec{y}_t := H\vec{x}_t + \vec{r}_t. \tag{4.2}$$

  Dabei ist $H \in \mathbb{R}^{m \times n}$ die Messmodellmatrix und $\vec{r}_t \sim \mathcal{N}(0,R)$ das normalverteilte Messrauschen mit Kovarianzmatrix $R \in \mathbb{R}^{m \times m}$.

- Der Zustand $\vec{x}_t \sim \mathcal{N}(\vec{m}_t, P_t)$ zum Zeitpunkt t ist normalverteilt mit dem Mittelwert $\vec{m}_t \in \mathbb{R}^n$ und der Kovarianzmatrix $P_t \in \mathbb{R}^{n \times n}$ und die initiale Verteilung ist gegeben mit $\vec{m}_0$ und P_0.

Insbesondere das lineare Bewegungsmodell 4.1 und die Normalverteilung von Prozess- und Messrauschen stellen theoretisch eine starke Einschränkung dar. Es zeigt sich allerdings (beispielsweise an den Ergebnisse der vorliegenden Arbeit), dass die Vorteile wie die Einfachheit des Modells und die effiziente Berechnung diese Nachteile mit Hilfe einer geeigneten Parameterwahl in der Praxis deutlich überwiegen.

4.3.2 Kalman-Filter

Eine stochastische Filterung ist ein Mittel, zur Behebung von Störungen, die durch unpräzise Messungen entstehen. Eine der wohl bekanntesten Filtertechniken ist der Kalman-Filter (Kálmán 1960). Dieser kann angewendet werden, wenn die Annahmen aus Abschnitt 4.3.1 (näherungsweise) zutreffen. Der Kalman-Filter schätzt die Verteilung der Vorhersage und des neuen Zustands zur Zeit t in folgenden rekursiven Berechnungen:

- Die Vorhersage (*Prediction Step*):

$$\vec{m}_t^- := A\vec{m}_{t-1} \tag{4.3}$$

$$P_t^- := AP_{t-1}A^T + Q \tag{4.4}$$

- Die Korrektur (*Update Step*):

$$\vec{m}_t := \vec{m}_t^- + K_t\left(\vec{y}_t - H\vec{m}_t^-\right) \tag{4.5}$$

$$P_t := P_t^- - K_t S_t K_t^T \tag{4.6}$$

  mit

$$S_t := HP_t^- H^T + R \tag{4.7}$$

und

$$K_t := P_t^- H^T S_t^{-1} \tag{4.8}$$

Details und Herleitungen zum Kalman-Filter sind beispielsweise in (Särkkä 2013) zu finden.

4.3.3 Stochastische Glättung

Während bei der stochastischen Filterung versucht wird, den aktuellen Zustand anhand der Messungen bis zum aktuellen Zeitpunkt zu schätzen, wird bei der stochastischen Glättung ein vergangener Zustand anhand der Messungen bis zum aktuellen Zustand bestimmt. In die Schätzung eines Zustands gehen demnach nicht nur Informationen aus der Vergangenheit ein, sondern auch Informationen aus der Zukunft. Eine Glättung auf Basis des Kalman-Filters stellt die Rauch-Tung-Striebel -Glättung (*RTS Smoother*) (Rauch u. a. 1965) dar. Dabei wird die komplette Sequenz der Messungen zweimal durchlaufen: einmal vorwärts und einmal rückwärts. Beim Vorwärtsdurchlauf wird der Kalman-Filter angewendet und für jeden Zeitpunkt t werden der Mittelwert $\vec{m}_t$ und die Kovarianzmatrix P_t gespeichert. Die Rückwärts-Rekursion wird mit folgenden Schritten berechnet:

- Die Vorhersage:

$$\vec{m}_{t+1}^- := A\vec{m}_t \tag{4.9}$$

$$P_{t+1}^- := AP_t^- A^T + Q \tag{4.10}$$

- Die Rückwärtskorrektur:

$$\vec{m}_t^S := \vec{m}_t + G_t \left(\vec{m}_{t+1}^S - \vec{m}_{t+1}^-\right) \tag{4.11}$$

$$P_t^S := P_t + G_t \left(P_{t+1}^S - P_{t+1}^-\right) G_t^T \tag{4.12}$$

mit

$$G_t := P_t A^T \left(P_{t+1}^-\right)^{-1}. \tag{4.13}$$

Die Rückwärts-Rekursion beginnt beim letzten Zeitpunkt t_E mit $\vec{m}_{t_E}^S := \vec{m}_{t_E}$ und $P_{t_E}^S := P_{t_E}$. Weitere Details und Herleitungen sind ebenfalls aus (Särkkä 2013) zu entnehmen.

4.3.4 Zustandsmodelle für die Spielerverfolgung

Um die Filter- und Glättungsverfahren aus den Abschnitten 4.3.2 und 4.3.3 erfolgreich anzuwenden, ist es erforderlich, geeignete Zustandsmodelle zu definieren. Hierfür sind A, H, Q und R, sowie $\vec{m}_0$ und P_0 sinnvoll zu wählen.

4.4 2D-Spielerverfolgung

4.4.1 Messvorgang für die Spielerpositionen

Ein wichtiger und kritischer Schritt bei der Verfolgung von Objekten ist die Durchführung der Messung. Dabei soll die Position der zu verfolgenden Objekte im aktuellen Einzelbild bestimmt werden. Bei detektionsbasierten Verfahren (wie beispielsweise (Breitenstein u. a. 2011)) geschieht das meist ohne Ausnutzung des zeitlichen Kontexts. Prinzipiell kann die Konfidenzkarte aus Abschnitt 3.7 für das komplette Bild berechnet werden und dadurch auch für eine detektionsbasierte Objektverfolgung genutzt werden. Diese Ansätze sind allerdings sehr rechenaufwendig, insbesondere bei kleinen Objektgrößen, wie sie bei der Spielerverfolgung häufig vorkommen. Daher nutzt das Vorgehen der vorliegenden Arbeit im Gegensatz dazu die zeitliche Information auf zwei Arten:

- Zum einen fließt durch die vorhersagebasierte Konfidenz (siehe Abschnitt 3.7.5) die Abweichung einer Messung von der vorhergesagten Position mit in die Bewertung der Messung ein. Dadurch wird die Auswahl der besten Messung robuster.
- Zum anderen wird durch die Suche der lokalen Maxima in der Konfidenzkarte in der Umgebung der vorhergesagten Position die Suchregion deutlich eingeschränkt und ermöglicht so eine effiziente Messung auch bei kleinen Objektgrößen.

Sei $B := \{B_1, \dots B_n\}$ die Menge der vorgegebenen Bounding-Boxen und I das aktuelle Einzelbild. Voraussetzung für die Durchführung des Messvorgangs ist, dass geeignete Outfithistogramme H^C aus Abschnitt 3.6.5 verfügbar sind. Im Folgenden sind die einzelnen Schritte des Messvorgangs beschrieben und aus Algorithmus 4.1 ist der detaillierte Ablauf zu entnehmen.

Die Wahl der Parameter für Algorithmus 4.1 kann im Anhang B aus Tabelle B.6 entnommen werden.

Algorithmus 4.1 : Messvorgang

Input : Bild I

Bounding-Boxen $B := \{B_1, \dots B_n\}$

Outfithistogramme H^C

Faktor für die Residualfläche s_a

Größentoleranzen für Neu-Detektionen s_w und s_h

Skalenwerte für die Größenoptimierung s_{min}, s_{max} und s_{step}

Output : Gemessene Spielerpositionen als Bounding-Boxen $\overline{B}$

Outfitlabels der gemessenen Spielerpositionen LBL

Messqualitätswerte der gemessenen Spielerpositionen Q

1 $\vec{c} \leftarrow$ `initSizeEstimation` (B) `/* siehe Algorithmus 3.1 */`

2 $B \leftarrow$ `resize` $(B, \vec{c})$ `/* Anpassung der Größe */`

3 $FG \leftarrow$ `detectFG` $(I, \vec{c})$ `/* siehe Algorithmus 3.2 */`

4 $C \leftarrow$ `connection`(FG) `/* Zusammnhangskomponenten */`

5 $\left[\overline{B}, LBL, Q\right] \leftarrow$ `optimizeConfidence` $\left(I, B, C, H^C, s_a, s_w, s_h, s_{min}, s_{max}, s_{step}\right)$

6 **return** $\left[\overline{B}, LBL, Q\right]$

Größenschätzung und Anpassung der Bounding-Boxen

Zunächst werden die Koeffizienten $\vec{c}$ der Größenschätzung neu initialisiert, wie in Abschnitt 3.4 beschrieben (Aufruf `initSizeEstimation` in Zeile 1, siehe Algorithmus 3.1).

Bounding-Boxen, die zu mehr als der Hälfte ihrer Fläche nicht im Bildbereich sind, bekommen die Messqualität $q = 0$ zugewiesen und werden im Folgenden nicht weiter beachtet. Die Größen der restlichen Bounding-Boxen werden anhand der neu ermittelten Größenkoeffizienten angepasst. Beide Operationen sind in dem Aufruf `resize` in Zeile 2 gebündelt.

Vordergrunddetektion und Zuordnung

Im nächsten Schritt wird der Vordergrund ermittelt wie in Abschnitt 3.5 beschrieben, wobei die Größenkoeffizienten $\vec{c}$ eine robustere Detektion ermöglichen (Aufruf `detectFG` in Zeile 3, siehe Algorithmus 3.2). Die Zusammenhangskomponenten des Vordergrunds werden in Zeile 4 bestimmt.

Optimierung der Konfidenz

In Zeile 5 wird mit dem Aufruf `optimizeConfidence` die zentrale Operation des Messvorgangs durchgeführt. Dabei wird, wie in Abschnitt 3.7 beschrieben, jede Bounding-Box

der Zusammenhangskomponente zugeordnet, zu welcher sie die größte Überlappung aufweist. Für die so gebildeten Gruppen von Bounding-Boxen werden die Positionen bezüglich der Konfidenz (siehe Gleichung 3.29, Abschnitt 3.8 und Algorithmus 3.5) optimiert.

Sei B_{C_i} die Menge der Bounding-Boxen, die bei der Berechnung der Konfidenz der Zusammenhangskomponente $C_i \in C$ zugeordnet sind. Die Bestimmung der optimalen Positionen resultiert in den Bounding-Boxen der Messung $\overline{B}_{C_i}$. Diese werden zur Menge $\overline{B} := \bigcup \overline{B}_{C_i}$ vereinigt.

Nach der Optimierung der Positionen bezüglich der Konfidenz werden die zwei folgenden Schritte durchgeführt, um Spieler nach Überdeckungen oder nach Eintritt in das Blickfeld zu erkennen:

1. **Auflösung von potentiellen Überdeckungen:**

   Bei einer vollständigen Überdeckung eines Spielers kann es vorkommen, dass dieser Spieler während der Verfolgung verloren geht. Das heißt, er ist nicht durch eine Bounding-Box in der Menge B repräsentiert. Wenn der Spieler wieder sichtbar wird, ergibt sich eine Vordergrundregion, die mit einem anderen Spieler vereinigt ist, aber durch keine Bounding-Box abgedeckt ist. Um Bereiche von C_i zu berücksichtigen, die nicht von $\overline{B}_{C_i}$ abgedeckt sind, wird daher das Residual $R_{C_i} := C_i \setminus \bigcup \overline{B}_{C_i}$ gebildet. Die Zusammenhangskomponenten von R_{C_i} werden bestimmt. Diese Komponenten repräsentieren potentielle Kandidaten für Spieler nach einer vollständigen Überdeckung. Eine Residualkomponente $R_j \in R_{C_i}$ wird dann für die weitere Verarbeitung zu C hinzugefügt, wenn folgende Bedingung für die Fläche $a(R_j)$ erfüllt ist:

   $$a(R_j) \geq s_a \cdot \overline{a}_{C_i}, \tag{4.14}$$

   wobei $\overline{a}$ die durchschnittliche Fläche der Bounding-Boxen in $\overline{B}_{C_i}$ ist. Dieser Schwellwert ist wichtig, um nicht beliebige Residualregionen (beispielsweise erzeugt durch einen ausgestreckten Arm) als potentielle Spielerkandidaten zu betrachten.

2. **Neuerkennung von Spielern:**

   Zusammenhangskomponenten R ohne zugeordnete Bounding-Boxen werden als potentielle neue Spieler aufgefasst. Solche Komponenten können nach einer Überdeckung in Schritt 1 hinzugefügt worden sein oder sie wurden in Zeile 3 in Algorithmus 4.1 detektiert. Dies ist beispielsweise bei Spielern, die das Blickfeld der Kamera betreten, der Fall. An ihrer Position wird mit den Größenkoeffizienten Höhe $\overline{h}$ und Breite $\overline{w}$ geschätzt und mit der tatsächlichen Höhe h und Breite w (der Bounding-Box von R) verglichen. Es wird überprüft, ob sich der Unterschied

innerhalb einer gewissen Toleranz bewegt, das heißt, ob gilt:

$$\frac{|h - \overline{h}|}{\overline{h}} \leq s_h \tag{4.15}$$

und

$$\frac{|w - \overline{w}|}{\overline{w}} \leq s_w. \tag{4.16}$$

Sind diese Bedingungen erfüllt, wird eine neue Bounding-Box mit gleichem Schwerpunkt wie R erzeugt, der Komponente R zugeordnet und mit dieser Gruppe ein Optimierung der Konfidenz durchgeführt. Die resultierende Bounding-Box wird der Menge $\overline{B}$ hinzugefügt. Dieser Schritt ist wichtig, um Spieler, die bisher nicht verfolgt wurden (weil sie beispielsweise bisher nicht im Blickfeld der Kamera waren) zu erkennen und mit in die Verfolgung einzuschließen.

Bestimmung der Outfitklassen

Zuletzt wird für alle Bounding-Boxen in $\overline{B}$ die zugehörige Outfitklasse aus H^C und das entsprechende Label bestimmt. Dies resultiert für $\overline{B} := \{\overline{B}_1, \ldots, \overline{B}_m\}$ in einer Menge von Labels $LBL := \{lbl_1, \ldots, lbl_m\}$ mit $lbl_i := lbl(\overline{B}_i)$.

Ermittlung von Änderungen der Größe

Spieler verändern im Verlauf einer Videosequenz ihre Größe, wenn sie sich beispielsweise vom vorderen Bildbereich in den hinteren Bildbereich bewegen oder die Kamera den Zoomfaktor verändert. Es ist daher wichtig, die von Bild zu Bild leichte Veränderung der Größe in der Messung zu berücksichtigen. Für alle gemessenen Bounding-Boxen in $\overline{B}$ wird jeweils einzeln bezüglich der zugeordneten Zusammenhangskomponente (ohne Berücksichtigung der restlichen zugeordneten Bounding-Boxen) für eine Größenskalierung von s_{min} bis s_{max} mit einer Schrittweite von s_{step} die Konfidenz berechnet. Die Größe mit der höchsten Konfidenz wird als Ergebnis in $\overline{B}$ festgehalten und die Messqualität q als diese berechnete Konfidenz gesetzt. Dies resultiert für $\overline{B} := \{\overline{B}_1, \ldots, \overline{B}_m\}$ in eine Menge von Messqualitätswerten $Q := \{q_1, \ldots, q_m\}$ mit $q_i := q(\overline{B}_i)$.

Dementsprechend gibt der Aufruf `optimizeConfidence` die gemessenen Bounding-Boxen $\overline{B}$ und die zugehörigen Labels LBL und Messqualitätswerte Q zurück.

4.4.2 Initiale Spielererkennung

Die initiale Erkennung der Spieler basiert auf dem in Abschnitt 4.4.1 vorgestellten Messprinzip und hat zwei zentrale Ziele:

- Zum einen werden die initialen Positionen der Spieler zur weiteren Verfolgung bestimmt.
- Zum anderen werden die Outfitklassen und die zugehörigen Farbhistogramme ermittelt (siehe Abschnitt 3.6).

Dementsprechend stellt die initiale Erkennung einen kritischen Schritt bei der Analyse einer Videosequenz dar. Wenn zu Beginn eine schlechte Erkennung von Spielern, eine fehlerhafte Erkennung von Nicht-Spieler-Regionen oder eine ungeeignete Initialisierung der Farbhistogramme vorliegt, kann eine hohe Genauigkeit des gesamten Verfolgungsverfahrens nicht mehr gewährleistet werden. In der Regel wird die initiale Spielererkennung mit dem ersten Einzelbild einer Videosequenz durchgeführt.

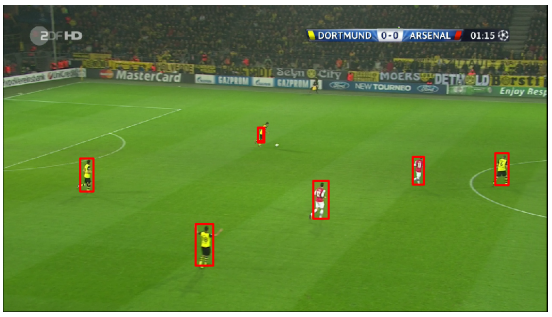
Aus Algorithmus 4.2 ist der detaillierte Ablauf der initialen Spielererkennung zu entnehmen. Zunächst versucht das Verfahren in einem iterativen Vorgehen, mit Hilfe von Vordergrund- und Personendetektion, die Personen auf dem Spielfeld im Bild zu erkennen. Dabei werden am Ende jeder Iteration die erkannten Personen genutzt, um die Größenschätzung zu initialisieren und damit die Erkennungsleistung in der nächsten Iteration zu verfeinern (siehe Zeilen 2 bis 8). Nach der letzten Iteration werden die Farbhistogramme der Outfitklassen ermittelt und das Ergebnis damit weiter verbessert. Abbildung 4.3 zeigt an einem Beispiel die Ergebnisse der initialen Erkennung nach der ersten und der dritten Iteration sowie das Endergebnis.

Vordergrunderkennung

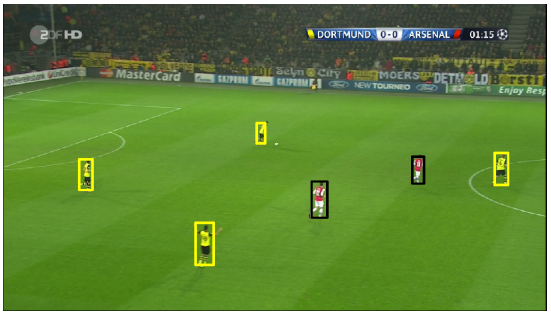
In Zeile 3 wird mit dem Aufruf `detectFG` eine Vordergrunderkennung (siehe Algorithmus 3.2) durchgeführt. Dabei werden das aktuelle Einzelbild I, die Koeffizienten der Größenschätzung $\vec{c}$, der Schwellwert τ_{Grass}, die Größe des Strukturelements r, die Toleranzfaktoren $\vec{s}$ und der Dilatationsfaktor s_{dil} übergeben. Die Zusammenhangskomponenten CC der Vordergrunds werden in Zeile 4 berechnet.

In Zeile 5 werden Erkennungen aus den vorherigen Iterationen aussortiert, da aufgrund der neuen Schätzung der Größenkoeffizienten (siehe Zeile 8) eine verfeinerte Berechnung der Konfidenz möglich ist. Eine Erkennung wird verworfen, wenn neu berechnete Qualitätswert (siehe Gleichung 3.29 in Abschnitt 3.7) kleiner als der Schwellwert τ_{conf} ist. Die Konfidenz wird bezüglich der Region aus CC mit der größten Überlappung berechnet. Da noch keine Farbtemplates generiert wurden, wird sie ohne die farbbasierte Konfidenz in Abschnitt 3.7.3 bestimmt (das heißt $w_3 = 0$). Sie kann sich zur vorherigen Iteration verändern, da durch veränderte Größenkoeffizienten in Zeile 8 auch eine veränderte Vordergrundschätzung resultieren kann.

(a) Ergebnis nach der ersten Iteration

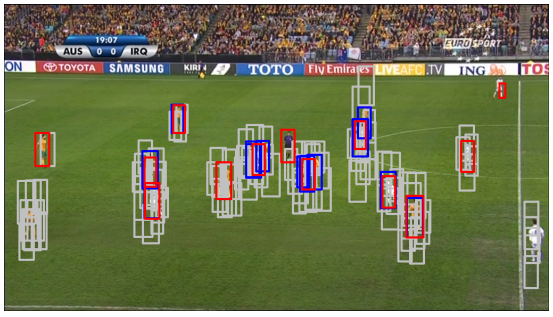


(b) Ergebnis nach der dritten Iteration

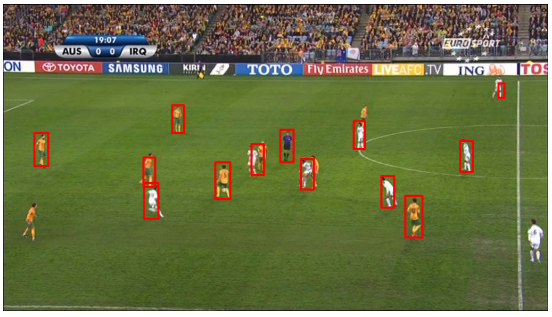


(c) Endergebnis

Abbildung 4.1: Beispiel für die Ergebnisse der initialen Detektion nach der ersten (a) und der dritten (b) Iteration (rote Bounding-Boxen) sowie das Endergebnis (c) (Bounding-Boxen unterschiedlicher Outfitklassen sind in unterschiedlichen Farben dargestellt). Originalbild aus (ZDF 2013a).



(a) Ohne NMS



(b) Mit NMS

Abbildung 4.2: Beispiel für die Ergebnisse der NMS (*Non-Maximum Suppression*). Originalbild aus (Eurosport 2013d).

Ermittlung von Regionen als Spielerkandidaten

Falls noch keine Initialisierung der Größenkoeffizienten stattgefunden hat (in der ersten Iteration mit $k = 1$), das heißt $\vec{c} = 0$, werden mit dem Aufruf `findPlayerCandidates` in Zeile 6 plausible Kandidaten für Personen in der Menge der Zusammenhangskomponenten gesucht. Eine zusammenhängende Bildregion R gilt als plausibler Kandidat für Personen, wenn sie folgende Bedingungen erfüllt:

- Höhe: Da es sich bei den behandelten Videosequenzen um Übersichtsaufnahmen aus der Totalen handelt, können Regionen von sehr großer und sehr kleiner Höhe

Algorithmus 4.2 : Initiale Spielererkennung

Input : Bild I

Parameter für die Vordergrunderkennung $\tau_{Grass}, r, \vec{s}, s_{dil}$

Anzahl an Iterationen n

Konfidenzschwellwert τ_{conf}

Regionenschwellwerte für die Suche von Spielerkandidaten $\tau_{hl}, \tau_{hu}, \tau_{ar}, \tau_{\phi}$

Skalierungsparameter für die Personenerkennung s_{margin}, s_{scale}

Output : Initial erkannte Spieler als Bounding-Boxen B^{best} mit Outfitlabels LBL^{best}

Farbhistogramme der Outfitklassen H^C

$\vec{c} \leftarrow \vec{0}$, $\widetilde{h} \leftarrow 0$, $B^{best} \leftarrow \emptyset$

for $k \leftarrow 1$ **to** n **do**

  $FG \leftarrow$ `detectFG` $(I, \vec{c}, \tau_{Grass}, r, \vec{s}, s_{dil})$

  $CC \leftarrow$ `connection`(I, FG)

  foreach $B_i \in B^{best}$ **with** $q(B_i) < \tau_{conf}$ **do** $B^{best} \leftarrow B^{best} \setminus B_i$

  if $\vec{c} = \vec{0}$ **then** $\left[CC, \widetilde{h}\right] \leftarrow$ `findPlayerCandidates` $(CC, \tau_{hl}, \tau_{hu}, \tau_{ar}, \tau_{\phi})$

  $B^{best} \leftarrow$ `detectHumanNMS` $\left(I, CC, B^{best}, s_{margin}, s_{scale}, \tau_{conf}, \widetilde{h}\right)$

  $\vec{c} \leftarrow$ `initSizeEstimation`(B^{best})

$H^C \leftarrow$ `initColorTemplates`(B^{best})

$\left[B^{best}, LBL^{best}\right] \leftarrow$ `measureConfidenceNMS`$(B^{best}, \vec{c}, H^C, \tau_{conf})$

return $\left[B^{best}, LBL^{best}, H^C\right]$

im Bild ausgeschlossen werden. Daher gilt R als plausibler Kandidat, wenn für die Höhe h der Bounding-Box $\tau_{hl} \leq h \leq \tau_{hu}$ gilt, mit den Schwellwerten $\tau_{hl} \in \mathbb{R}^{\geq 0}$, $\tau_{hu} \in \mathbb{R}^{\geq 0}$ und $\tau_{hl} \leq \tau_{hu}$.

- Anisometrie: Wie schon bei der Vordergrundsegmentierung in Abschnitt 3.5 wird für R die äquivalente Ellipse berechnet, die die gleiche Orientierung und das gleiche Seitenverhältnis hat. Die Anisometrie ist definiert als das Verhältnis $\frac{r_a}{r_b}$ von der Länge der großen Halbachse r_a zur Länge der kleinen Halbachse r_b mit $r_a \geq r_b$ und $\frac{r_a}{r_b} \geq 1$. Um sehr schmale Regionen auszuschließen, gilt R als plausibler Kandidat, wenn $\frac{r_a}{r_b} \leq \tau_{ar}$ für einen Schwellwert $\tau_{ar} \in \mathbb{R}$ mit $\tau_{ar} \geq 1$ gilt.

- Orientierung: Die Personen in einer Aufzeichnung eines Fußballspiels stellen in der Regel vertikale Strukturen dar. Daher gilt R als plausibler Kandidat, wenn der Winkel der großen Halbachse der äquivalenten Ellipse nicht mehr als τ_{ϕ} von der Vertikalen abweicht.

Zusätzlich zu den Bildregionen, die als plausible Kandidaten gelten, gibt der Aufruf `findPlayerCandidates` auch den Median $\widetilde{h}$ der Höhen dieser Bildregionen zurück.

Erkennung von Personen

Mit dem Aufruf `detectHumanNMS` in Zeile 7 wird in der Umgebung jeder Zusammenhangskomponente $R \in CC$ eine Personenerkennung durchgeführt.

Zunächst wird die Bounding-Box von R mit Höhe h und Breite w bestimmt. Diese Bounding-Box wird vertikal auf beiden Seiten um $s_{margin} \cdot h$ und horizontal auf beiden Seiten um $s_{margin} \cdot w$ erweitert. In dem Teilbild, das von dieser erweiterten Bounding-Box mit Höhe h_{ext} und Breite w_{ext} beschrieben wird, wird im Anschluss eine Personendetektion durchgeführt (siehe Abschnitt 3.7.4). Die Faktoren für die unterste und oberste Ebene der durchlaufenen Bildpyramide werden durch die Größenschätzungen an Ober- und Unterkante der Bounding-Box bestimmt. Sei h_{Det} die Personengröße des Detektors (bei Skalierung 1) und seien h_u und h_l mit $h_u \leq h_l$ die Größenschätzungen an Ober- und Unterkante der Bounding-Box. Dabei wird die maximale Verkleinerung des Bildes mit $s_{min} := \frac{h_{Det}}{s_{scale} \cdot h_l}$ und die maximale Vergrößerung des Bildes mit $s_{max} := \frac{s_{scale} \cdot h_{Det}}{h_u}$ mit dem Skalierungsfaktor $s_{scale} \in \mathbb{R}$ festgelegt. Falls noch keine Initialisierung der Größenkoeffizienten stattgefunden hat (in der ersten Iteration mit $k = 1$), das heißt $\vec{c} = 0$, ist die maximale Verkleinerung auf die Größe des Suchfensters beschränkt. Die maximale Vergrößerung ist durch die Höhe h_{ext} der erweiterten Bounding-Box und durch den Höhenmedian $\widetilde{h}$ mit $s_{max} := \frac{h_{Det}}{\min\left(h_{ext}, \widetilde{h}\right)}$ beschränkt.

Für jede detektierte Bounding-Box B_i wird die Konfidenz q_i berechnet (siehe Gleichung 3.29 in Abschnitt 3.7). Die Konfidenz wird bezüglich der Region R und ebenfalls ohne farbbasierte Konfidenz (also $w_3 = 0$) berechnet. Es werden nur Bounding-Boxen B_i betrachtet, für die $q_i \geq \tau_{conf}$ gilt. Um Erkennungen zu unterdrücken, die mit hoher Wahrscheinlichkeit kein lokales Maximum der Konfidenz repräsentieren (*Non-Maximum Suppression*), werden die Erkennungen bezüglich ihrer Konfidenz aufsteigend sortiert und der Reihe nach zur Menge B^{best} hinzugefügt. Liegt eine signifikante Überlappung ($o_J > 0{,}35$) mit einem Element der Menge B^{best} vor, wird die Erkennung mit der besseren Konfidenz übernommen und die andere Erkennung verworfen. `detectHumanNMS` liefert die Menge der bisher besten Erkennungen B^{best} zurück. Ein Beispiel für die Wirkweise der *Non-Maximum Suppression* ist in Abbildung 4.2 dargestellt.

Größenschätzung

Wie bereits erwähnt, werden am Ende jeder Iteration in Zeile 8 die Koeffizienten $\vec{c}$

der Größenschätzung anhand der besten Erkennungen B^{best} neu initialisiert. Durch eine verfeinerte Größenschätzung soll die Erkennungsleistung in der nächsten Iteration verbessert werden.

Ermittlung der Farbhistogramme

Nach der letzten Iteration werden in Zeile 9 die Farbhistogramme der Outfitklassen anhand der besten Erkennung B^{best} ermittelt (siehe Algorithmus 3.4).

Durchführung eines Messvorgangs

Zum Schluss werden die bisherigen Erkennungen durch eine lokale Optimierung der Konfidenz (Messvorgang; siehe Abschnitt 4.4.1) mit dem Aufruf `measureConfidenceNMS` in Zeile 10 verfeinert. Dabei dienen die bisher besten Erkennungen B^{best} als Vorhersagen. Von den resultierenden Messungen B_i mit Konfidenz q_i werden diejenigen verworfen, für die $q_i < \tau_{conf}$ gilt. Zusätzlich wird eine gierige *Non-maximum Suppression* bei signifikanten Überlappungen ($o_J > 0{,}35$) durchgeführt und die übrigen Messungen zurückgegeben. Für jede Messung wird zusätzlich das näheste Teamoutfit bestimmt und das zugehörige Outfitlabel zurückgegeben.

Der Algorithmus liefert als Ergebnis die besten Messungen B^{best}, die zugehörigen Outfitlabels LBL^{best} und die ermittelten Farbhistogramme H^C zurück.

4.4.3 Zustandsmodell für die 2D-Spielerverfolgung

Bei der 2D-Spielerverfolgung wird ein einfaches Bewegungsmodell mit konstanter Geschwindigkeit angewendet und es gilt $n := 5$ mit

$$\vec{x} := \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ h \end{pmatrix}, \tag{4.17}$$

wobei x und y die Position und h die Höhe der Bounding-Box repräsentieren (die Breite wird von h abgeleitet, siehe Gleichung 3.2) und $\dot{x}$ und $\dot{y}$ die aktuelle Geschwindigkeit in

x und y darstellen. Die Übergangsmatrix $A \in \mathbb{R}^{5\times5}$ sieht dann wie folgt aus:

$$A := \begin{pmatrix} 1 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.18}$$

Veränderungen in der Geschwindigkeit (Beschleunigung) und Veränderungen in der Höhe werden demnach über das Prozessrauschen modelliert. Da bei der Messung der aktuellen Spielerposition nur die Position und Größe der Bounding-Box gemessen wird, gilt $m := 3$ und die Messmatrix $H \in \mathbb{R}^{3\times5}$ wird wie folgt definiert:

$$H := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.19}$$

Die Kovarianzmatrix $Q \in \mathbb{R}^{5\times5}$ des Prozessrauschens und die Kovarianzmatrix $R \in \mathbb{R}^{3\times3}$ werden in Anlehnung an (Särkkä 2013) gewählt:

$$Q := \begin{pmatrix} q_p^2\frac{\Delta t^3}{3} & 0 & q_p^2\frac{\Delta t^2}{2} & 0 & 0 \\ 0 & q_p^2\frac{\Delta t^3}{3} & 0 & q_p^2\frac{\Delta t^2}{2} & 0 \\ q_p^2\frac{\Delta t^2}{2} & 0 & q_p^2\Delta t & 0 & 0 \\ 0 & q_p^2\frac{\Delta t^2}{2} & 0 & q_p^2\Delta t & 0 \\ 0 & 0 & 0 & 0 & q_s^2 \end{pmatrix} \tag{4.20}$$

und

$$R := \begin{pmatrix} r_p^2 & 0 & 0 \\ 0 & r_p^2 & 0 \\ 0 & 0 & r_s^2 \end{pmatrix}, \tag{4.21}$$

wobei $q_p^2 \in \mathbb{R}$ und $q_s^2 \in \mathbb{R}$ die Varianzen des Prozessrauschens bezüglich der Position (spektrale Dichte, zeit-kontinuierliche Varianz) und der Größe darstellen sowie $r_p^2 \in \mathbb{R}$ und $r_s^2 \in \mathbb{R}$ die Varianzen des Messrauschens bezüglich Position und Größe.

4.4.4 Online-Spielerverfolgung in 2D

In diesem Abschnitt wird ein Online-Verfahren vorgestellt, das die Ergebnisse der initialen Spielererkennung (Abschnitt 4.4.2) und des Messvorgangs (Abschnitt 4.4.1) nutzt, um die Positionen der Spieler über die Zeit zu ermitteln und zu Trajektorien zusammenzufassen.

Für die Spielerverfolgung wird eine Menge von n_t Spielern zur Zeit t als Menge von Bounding-Boxen $B^t := \{B_1^t, \dots B_{n_t}^t\}$ mit zugeordneten Identitäten $ID := \{id_1^t, \dots, id_{n_t}^t\}$ mit $i \neq j \Rightarrow id_i^t \neq id_j^t$ und zugeordneten Outfitlabels $LBL := \{lbl_1^t, \dots, lbl_{n_t}^t\}$ betrachtet.

Während der Verfolgung wird jedem Spieler einer der drei folgenden Zustände zugeordnet:

- `verfolgt`: Der Spieler wird vom Algorithmus verfolgt (d.h. es wird eine Prädiktion und eine Messung der neuen Position im nächsten Frame durchgeführt) und seine Position wird als Ergebnis ausgegeben.
- `verloren`: Der Spieler gilt als verloren (beispielsweise nach Verlassen des sichtbaren Bereichs) und wird vom Algorithmus nicht weiter verfolgt. Seine Position wird nicht als Ergebnis ausgegeben. Verlorene Spieler dienen bei der Neuerkennung von Spielern zur Aufrechterhaltung der Identität (beispielsweise bei Wiedereintritt in den sichtbaren Bereich).
- `neu`: Die Bounding-Box repräsentiert einen potentiell neu erkannten Spieler. Der Spieler wird nicht verfolgt und seine Position wird nicht als Ergebnis ausgegeben. Potentiell neue Spieler müssen erst in nachfolgenden Frames durch Messungen bestätigt werden, bevor sie in den Status `verfolgt` übergehen.

Die Initialisierung zu Beginn der Videosequenz ($t = 0$) erfolgt durch die initiale Spielerdetektion, wie sie in Algorithmus 4.2 in Abschnitt 4.4.2 beschrieben ist. Die resultierenden Bounding-Boxen B^{best} werden als Vorhersagen $\hat{B}^0$ gesetzt. Die zugehörigen Labels und die Outfithistogramme H^C werden gespeichert.

Bei der Durchführung des Messvorgangs wird zusätzlich zu den gemessenen Positionen für die Spieler, die bisher im Zustand `verfolgt` waren, eine Menge von neuen Messungen bestimmt (siehe Algorithmus 4.1). Diese Messungen werden genutzt, um neue Spieler zu erkennen und der Menge der verfolgten Spieler hinzuzufügen.

Die Verfolgung der Spieler zur Zeit t mit Einzelbild I^t läuft in folgenden Schritten ab:

1. Zunächst wird ein Messvorgang auf Basis des Bildes I^t, der Kalman-Vorhersagen $\hat{B}^{t-1} := \{\hat{B}_1^{t-1}, \dots, \hat{B}_{n_{t-1}}^{t-1}\}$ der Spieler im Zustand `verfolgt` und der Outfithistogramme H^C durchgeführt (siehe Algorithmus 4.1). Dies resultiert in der Menge der gemessenen Bounding-Boxen $\overline{B}^t := \{\overline{B}_1^t, \dots, \overline{B}_{n_{t-1}}^t, \overline{B}_{n_{t-1}+1}^t, \dots \overline{B}_{m_t}^t\}$, wobei die ersten n_{t-1} Messungen den Spielern im Zustand `verfolgt` zugeordnet werden. Die restlichen Messungen sind potentiell neu erkannte Spieler. Zusätzliche Resultate sind die Menge der gemessenen Labels $\overline{LBL}^t$ und die Menge der Messqualitätswerte Q^t.

2. Als nächstes wird für jeden Spieler i im Zustand `verfolgt` die Vorhersage (siehe Gleichungen 4.3 und 4.4) und Korrektur (siehe Gleichungen 4.5 und 4.6) des Kalman-Filters durchgeführt.
   Das Ergebnis ist zunächst die Menge der Vorhersagen $\hat{B}^t := \{\hat{B}_1^t, \ldots, \hat{B}_{n_{t-1}}^t\}$ und die Menge der aktuellen Zustände $B^t := \{B_1^t, \ldots B_{n_{t-1}}^t\}$.

3. Für jeden Spieler i im Zustand `verfolgt` wird überprüft, ob die gemessene Konfidenz q_i^t unterhalb eines Schwellwertes τ_{reject} liegt, also $q_i^t \leq \tau_{reject}$. Ist das der Fall, so wird der Zähler $n_{reject}(i)$ um eins erhöht. Falls $n_{reject}(i) \geq N_{reject}$, wird der Status des Spielers von `verfolgt` auf `verloren` gesetzt. Falls $q_i^t > \tau_{reject}$, setze $n_{reject}(i) := 0$.

4. Für jeden Spieler i im Zustand `verfolgt` wird überprüft, ob das Label des Spielers lbl_i dem gemessenen Label $\overline{lbl}_i$ entspricht. Falls $lbl_i \neq \overline{lbl}_i$, wird der Zähler $n_{label}(i)$ um eins erhöht. Falls $n_{label}(i) \geq N_{label}$, wird der Status des Spielers von `verfolgt` auf `verloren` gesetzt. Die Messung wird als neue Messung zu $\overline{B}^t$ hinzugefügt (und ggf. später als neu gemessenes Objekt erkannt).

5. Für jedes Spielerpaar i, j mit $i < j$ wird die Überlappung der Bounding-Boxen o_{ij} berechnet. Sind beide Spieler im Zustand `verfolgt` und ist die Überlappung größer als ein Schwellwert τ_{ovlp}, also $o_{ij} > \tau_{ovlp}$, so wird der Zähler n_{ovlp}^{ij} um eins erhöht. Falls $n_{ovlp}^{ij} \geq N_{ovlp}$ gilt, wird der Status des Spielers mit der niedrigeren gemessenen Konfidenz von `verfolgt` auf `verloren` gesetzt. Falls $o_{ij} \leq \tau_{ovlp}$ oder einer der beiden Spieler im Zustand `verloren` ist, setze $n_{ovlp}^{ij} := 0$.

6. Für jeden Spieler j im Zustand `verloren` werden die Überlappungszähler zurückgesetzt, das heißt $n_{ovlp}^{ij} := 0$ bzw. $n_{ovlp}^{jk} := 0$.

7. Für jede neue Messung wird geprüft, ob es eine Überlappung o mit einem Spieler im Zustand `verfolgt` gibt, die größer als τ_{meas} ist, also $o > \tau_{meas}$. Ist das der Fall, wird diese neue Messung als Duplikat verworfen.

8. Für jede neue Messung j, die kein Duplikat ist, wird der Spieler i im Zustand `neu` mit der größten Überlappung o bestimmt. Falls $o > \tau_{meas}$ gilt und die Messqualität groß genug ist, also $q_j^t \geq 0{,}9 \cdot \tau_{accept}^j$, und $lbl_i = \overline{LABEL}_J$ gilt, wird der Zustand auf `verfolgt` gesetzt und der Kalman-Filter entsprechend initialisiert. Dabei ist $\tau_{accept}^j := \tau_{accept} \cdot \sqrt{\frac{h_j}{\bar{h}}}$. Falls die Messung in der Nähe eines Spielers k im Zustand `verloren` mit dem gleichen Label liegt (d.h. die Mittelpunkte liegen weniger als ein Viertel der Bilddiagonale auseinander), so wird der Spieler k in den Zustand `verfolgt` gesetzt und der neue Spieler i wird gelöscht. Falls sich die Messung mit keinem Spieler im Zustand `neu` ausreichend überschneidet, aber die Messqualität $q_j^t \geq \tau_{accept}^j$, so wird die Messung als Spieler mit Status `neu` für den nächsten Frame

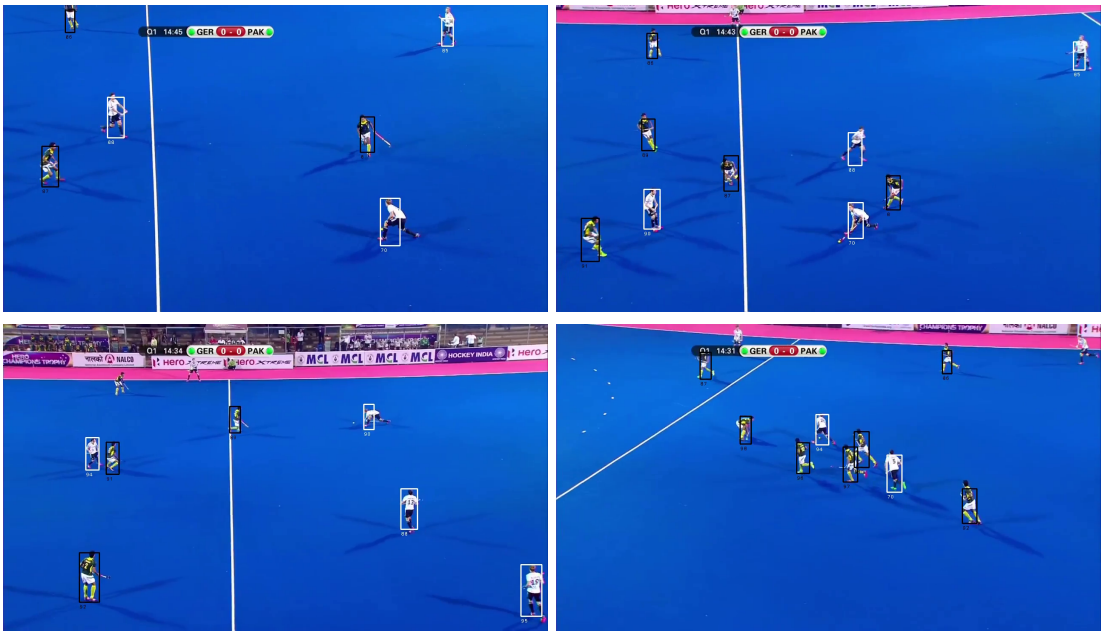


Abbildung 4.3: Anwendung der 2D-Spielerverfolgung für eine Übertragung einer Feldhockey-Begegnung (Fédération Internationale de Hockey (FIH) 2014)

hinzugefügt. Bei der Umstellung des Status von `neu` bzw. `verloren` auf `verfolgt`, werden die Zählvariablen $n_{reject}(i)$ und $n_{label}(i)$ auf 0 zurückgesetzt.

9. Alle Spieler mit dem Status `neu` (aus dem letzten Einzelbild), denen in diesem Frame keine Messung zugeordnet werden konnte, werden gelöscht.

4.4.5 Spielerverfolgung bei anderen Sportarten

Die Verfolgung von Spielern in Videos mit dem hier vorgestellten Verfahren ist prinzipiell auch bei anderen Sportarten anwendbar. Voraussetzung dafür ist es, dass sich die Spieler vor einem mehr oder weniger homogenen Hintergrund abgrenzen. In Abbildung 4.3 sind Ausschnitte der Ergebnisse des Verfolgungsverfahrens für eine Aufnahme einer Feldhockeybegegnung abgebildet. Hierfür war lediglich die Anpassung der Schwellwerte der „Grasfarbe“ für die Bestimmung von Feldhülle (siehe Abschnitte 3.3.1 und 3.3.2) notwendig.

4.5 Verfeinerung der 3D-Trajektorien

4.5.1 Einleitung

Die Positionen der Spieler im Bild sind für eine weitere Analyse hilfreich. Um Auswertungen auf sportlicher Ebene durchführen zu können (wie beispielsweise Statistiken

zu Ballbesitz und Laufleistung), ist es notwendig die Bildkoordinaten auf Koordinaten im Koordinatensystem des Spielfelds abzubilden. Dazu ist es für Spieler ausreichend, die X- und Y-Koordinaten auf dem Spielfeld zu ermitteln. Sind diese Koordinaten bekannt, können mit Hilfe von Gesetzen und Regeln in der "realen Welt"(wie z.B. durch eine maximale Laufgeschwindigkeit) die Ergebnisse der Spielerverfolgung verfeinert und verbessert werden. Die Bearbeitung der 3D-Daten läuft in folgenden Schritten ab:

1. Transformation der Bildpositionen zu Positionen im Spielfeld.
2. Auftrennung der Trajektorien in unterbrechungsfreie Stücke.
3. Glättung der Trajektorien (Rauch-Tung-Striebel).
4. Filterung von Objekten außerhalb des Spielfelds und „falsch-positiven“ Erkennungen in der Nähe Feldlinien.
5. Identifizierung der Trajektorien.
6. Rückprojektion ins Bild.

4.5.2 Transformation der Bildpositionen zu Positionen im Spielfeld

Die reelle projektive Ebene $\mathbb{R}P^2$ ist ein geometrisches Konstrukt, welches eine Erweiterung der euklidischen Ebene $\mathbb{R}^2$ darstellt. Im Gegensatz zur euklidischen Ebene, in der sich parallele Geraden nicht schneiden, ist eine projektive Ebene so konstruiert, dass sich zwei (nicht-identische) Geraden in einem (eindeutigen) Punkt schneiden. Die Punkte der projektiven Ebene werden durch ihre sogenannten homogenen Koordinaten aus $\mathbb{R}^3$ dargestellt, wobei ein Punkt $(x,y) \in \mathbb{R}^2$ aus der euklidischen Ebene durch die homogenen Koordinaten $(x_1,x_2,x_3) \in \mathbb{R}^3$ mit $x = \frac{x_1}{x_3}$ und $y = \frac{x_2}{x_3}$ repräsentiert werden kann. Das heißt, dass zwei Vektoren aus $\mathbb{R}^3$ (mit $x_3 \neq 0$), die sich nur durch einen Faktor ($\neq 0$) unterscheiden, den gleichen Punkt auf der euklidischen Ebene repräsentieren. Eine Homographie H (Projektivität / projektive Kollineation) ist eine umkehrbare Abbildung $H : \mathbb{R}P^2 \rightarrow \mathbb{R}P^2$, die im $\mathbb{R}^3$ eine nicht-singuläre lineare Abbildung darstellt. Eine Homographie bildet drei Punkte, die auf einer Linie liegen, so ab, dass ihre Bilder auch auf einer Linie liegen. Im Bereich der Bildverarbeitung spielen Homographien eine wichtige Rolle. So kann beispielsweise bei der Darstellung einer planaren Szene (beispielsweise einer Häuserfassade) mit einer Lochkamera die Abbildung der Punkte von der Ebene in der Szene auf die Bildebene mit einer Homographie modelliert werden. Diese Eigenschaft wird hier genutzt, um die Transformation von Bildkoordinaten in Koordinaten auf dem Spielfeld durch eine Homographie zu modellieren (siehe Sonka u. a. 2008, S. 556).

Homographien werden in der Regel über Punktkorrespondenzen bestimmt. Um eine Homographie eindeutig zu bestimmen, sind mindestens vier Punktepaare notwendig, wobei jeweils höchstens nur zwei Punkte auf einer Gerade liegen dürfen. In den meisten Anwendungen stehen mehr als vier Korrespondenzen zur Verfügung, die durch Messungenauigkeiten oder Ähnliches verunreinigt sind. Dadurch hat das entstehende überbestimmte Gleichungssystem keine eindeutige Lösung und kann über eine nicht-lineare Methode der kleinsten Quadrate, wie beispielsweise der Levenberg-Marquard-Algorithmus (siehe Nocedal und Wright 2006a), mit einer geeigneten Initialisierung gelöst werden. Ein gutes Verfahren zur Initialisierung bietet die *Direct Linear Transformation* (Hartley und Zisserman 2003), bei der versucht wird, die algebraische Distanz zu minimieren. Häufig liefert dieses Verfahren auch schon zufriedenstellende Ergebnisse (Sonka u. a. 2008, S. 559).

Wie schon erwähnt, werden Positionen der Spieler im Bild mittels Homographien zu Positionen der Spieler im Spielfeld transformiert. Seien die Position eines Spielers im Bild als Bounding-Box $B_i := B(x_i, y_i, w_i, h_i)$ und eine geeignete Homographie $H \in \mathbb{R}^{3\times 3}$ gegeben, dann berechnen sich die Koordinaten $\vec{x}_i \in \mathbb{R}^2$ des Spielers im Spielfeld in zwei Schritten:

$$\begin{pmatrix} x_i^h \\ y_i^h \\ z_i^h \end{pmatrix} := H \begin{pmatrix} x_i + 0{,}5 \cdot w_i \\ y_i + h_i \\ 1 \end{pmatrix} \tag{4.22}$$

und

$$\vec{x}_i := \begin{cases} \begin{pmatrix} \frac{x_i^h}{z_i^h} \\ \frac{y_i^h}{z_i^h} \end{pmatrix} & \text{für } z_i^h \neq 0 \\ \vec{0} & \text{sonst} \end{cases} \tag{4.23}$$

Das entscheidende Problem ist es, eine geeignete Homographie zu finden. Prinzipiell ist es möglich, durch das manuelle Festlegen von Punktkorrespondenzen (zum Beispiel markante Eckpunkte der Spielfeldmarkierungen, die im Bild sichtbar sind) für jedes Einzelbild eine homographische Abbildung zu schätzen. Handelt es sich um eine statische Kamera, bleibt die Homographie konstant über die Zeit und müsste nur einmalig geschätzt werden. Bei einer bewegten Kamera ist dieser Ansatz allerdings nicht praktikabel (bei einer Halbzeit mit 45 min und einer Bildrate von 50 FPS wären das 135000 Einzelbilder). Da ein robuster vollautomatischer Ansatz in der Praxis schwer umzusetzen ist, wird für die Bestimmung der Homographien folgender Ansatz verfolgt (siehe Hoernig 2016):

1. Zuerst werden die Punktkorrespondenzen für eine kleine Anzahl weniger Einzelbilder (z.B. nur das erste Bild einer Videosequenz) manuell bestimmt.

2. Danach werden die Homographien für die annotierten Bilder geschätzt, unter Zuhilfenahme eines Liniendetektors, der sichtbare Feldlinien im Bild erkennt.

3. Zuletzt wird ausgehend von den Einzelbildern mit manuell bestimmten Homographien versucht, Feldlinien und markante Punkte im Gras (z.B. Löcher oder Dreck) in den folgenden Bildern zu verfolgen und die Änderung der Homographie anzupassen.

Nach der Bestimmung der Homographien, kann man für jedes Einzelbild und jeden verfolgten Spieler die entsprechenden Positionen im Spielfeld berechnen. Das heißt, für jeden Spieler i mit Identität id_i, der im Bild zur Zeit t erkannt wurde, gibt es die zugehörigen Feldkoordinaten $\vec{x}_i^t \in \mathbb{R}^2$.

4.5.3 Auftrennung der Trajektorien in unterbrechungsfreie Stücke

Die in Abschnitt 4.5.2 bestimmten Positionsdaten eines Spielers können zeitliche Unterbrechungen enthalten. Dies ist beispielsweise der Fall, wenn ein Spieler das Blickfeld der Kamera verlässt, einige Zeit später wieder betritt und von Verfolgungsverfahren wieder erkannt und richtig identifiziert wird. Da Trajektorien mit zeitlichen Unterbrechungen in den nachfolgenden Schritten schwierig zu behandeln sind, werden Positionsdaten in diesem Schritt aufgesplittet. Seien $\{\vec{x}_i^{t_1}, \vec{x}_i^{t_2}, \dots \vec{x}_i^{t_n}\}$ die zeitlich sortierten Positionsdaten von Spieler i mit $t_1 < t_2 < \cdots < t_n$ und sei Δt die Zeitdifferenz zwischen zwei Einzelbildern. Eine zeitliche Unterbrechung zum Zeitpunkt t_k mit $k < n$ liegt vor, wenn $t_{k+1} - t_k > \Delta t$. Zum Zeitpunkt t_n liegt in diesem Zusammenhang auch eine Unterbrechung vor. Für eine Unterbrechung zum Zeitpunkt t_u wird eine neue Trajektorie $\{\vec{x}_i^{t_{u_0}}, \dots \vec{x}_i^{t_u}\}$ ohne Unterbrechungen gebildet. t_{u_0} ist der Zeitpunkt der vorherigen Unterbrechung oder, falls nicht vorhanden, t_1. Jede neue Trajektorie wird als eigener Spieler j mit einer neuen eindeutigen Identität id_j und dem gleichbleibenden Label $lbl_j := lbl_i$ angesehen.

4.5.4 Glättung der Trajektorien

4.5.4.1 Zustandsmodell für 3D-Trajektorien

Für die 3D-Analyse wird mit Koordinaten (x,y) im Spielfeld gearbeitet. Es wird ebenfalls ein einfaches Bewegungsmodell mit konstanter Geschwindigkeit angewendet. Die

Modellbildung erfolgt somit analog zum 2D-Szenario ohne die Schätzung der Höhe. Es gilt dementsprechend:

$$\vec{x} := \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix}, \tag{4.24}$$

$$A := \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.25}$$

$$H := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{4.26}$$

$$Q := \begin{pmatrix} q_p^2 \frac{\Delta t^3}{3} & 0 & q_p^2 \frac{\Delta t^2}{2} & 0 \\ 0 & q_p^2 \frac{\Delta t^3}{3} & 0 & q_p^2 \frac{\Delta t^2}{2} \\ q_p^2 \frac{\Delta t^2}{2} & 0 & q_p^2 \Delta t & 0 \\ 0 & q_p^2 \frac{\Delta t^2}{2} & 0 & q_p^2 \Delta t \end{pmatrix} \tag{4.27}$$

und

$$R := \begin{pmatrix} r_p^2 & 0 \\ 0 & r_p^2 \end{pmatrix} \tag{4.28}$$

4.5.5 Entfernung potentiell falsch-positiver Erkennungen

Für die Analyse der 3D-Spielerpositionen sind in erster Linie die Positionen der Spieler (und vielleicht noch die des Schiedsrichters) von Bedeutung. Allerdings liefert die 2D-Spielerverfolgung unter Umständen auch Trajektorien von Objekten, die für die 3D-Analyse nicht interessant oder sogar störend sind, wie beispielsweise:

- Objekte außerhalb des Spielfeldes (Trainer, Zuschauer etc.)
- Falsch-positive Erkennungen, die insbesondere an den Seitenlinien, der Mittellinie und dem Mittelkreis oder außerhalb des Spielfeldes (Bandenwerbung) auftreten

Aus diesem Grund wird in einem Nachbearbeitungsschritt versucht, solche Trajektorien zu entfernen. Dazu werden einer Trajektorie zu jedem Zeitpunkt folgende Zustände zugeordnet:

- Außerhalb des Spielfelds: Eine Position hat den Status `InsideOfField`, wenn die Position innerhalb des Rechtecks liegt, das von den zwei Torlinien und den zwei

Seitenlinien gebildet wird (in Koordinaten $|x| < 52{,}5 \wedge |y| < 34$, siehe die Spielfeldmaße in Abschnitt 2.6). Ansonsten hat die Position den Status `OutOfField`.

- In der Nähe einer Spielfeldmarkierung: Eine Position hat den Status `NearLine`, wenn die Position einen Abstand von maximal 2,5 m zu einer der Torlinien, zu einer der Seitenlinien oder zur Mittellinie hat. Des Weiteren hat eine Position den Status `NearLine`, wenn sie einen maximalen Abstand von 1,0 m zum Mittelkreis hat. Ansonsten hat die Position den Status `FarOffLine`.

Werden nun die Zustände einer Trajektorie über ihre Dauer gezählt, so erhält man die Werte #`InsideOfField`, #`OutOfField`, #`NearLine` und #`FarOffLine`. Eine Trajektorie wird entfernt, wenn gilt:

$$\frac{\#\texttt{OutOfField}}{\#\texttt{OutOfField} + \#\texttt{InsideOfField}} \geq 0{,}9 \tag{4.29}$$

oder

$$\frac{\#\texttt{NearLine}}{\#\texttt{NearLine} + \#\texttt{FarOffLine}} \geq 0{,}9 \tag{4.30}$$

4.5.6 Vereinigung der Trajektorien

Durch den Bearbeitungsschritt in Abschnitt 4.5.3 werden die Spielerpositionen in unterbrechungsfreie Trajektorien aufgeteilt. Bei Verlassen und Wiederbetreten des Blickbereichs der Kamera oder durch kurze Aussetzer der Verfolgung (z.B. bei während Überdeckung) kann es sein, dass die Positionen eines Spielers auf mehrere Trajektorien verteilt sind. Ziel ist es in diesem Schritt, die Trajektorien zu identifizieren, die zu einem Spieler gehören.

Seien t_S und t_E der Anfang und das Ende der Sequenz. Jede Trajektorie X_i ist eine Menge von zeitlich sortierten Positionen $\{\vec{x}_i^{t_S^i}, \ldots, \vec{x}_i^{t_E^i}\}$, wobei t_S^i und t_E^i Start- und Endzeitpunkt der Trajektorie X_i sind. Zu jedem diskreten Zeitpunkt der Videosequenz enthält eine Trajektorie höchstens eine Position. Eine Trajektorie kann auch Unterbrechungen enthalten. $X := \{X_1, \ldots X_n\}$ ist initial die Menge der geglätteten Trajektorien, wobei alle Trajektorien in X zunächst unterbrechungsfrei sind. Jede Trajektorie X_i hat die Identität id_i und das Label lbl_i. Algorithmus 4.3 veranschaulicht das Vorgehen. Es werden zu jedem Zeitpunkt t die Trajektorien X_i untersucht, für die $t_E^j = t$ gilt.

Der Aufruf `getNearest`(X_i,X) in Zeile 3 sucht für X_i eine Trajektorie zur Vereinigung. Dabei werden die Trajektorien $X_j \in X$ betrachtet, die folgende Bedingungen erfüllen:

1. X_j endet, bevor X_i beginnt, das heißt $t_E^j < t_S^i$.

2. X_j hat das gleiche Label wie X_i, das heißt $lbl_j = lbl_i$.

3. Die notwendige Geschwindigkeit, um startend von Endpunkt von X_j zur Zeit t_E^j, den Startpunkt von X_i zur Zeit t_S^i zu erreichen, ist theoretisch von einem Menschen zu erreichen. Sei $\delta_p := \|\vec{x}_i^{t_S^i} - \vec{x}_j^{t_E^j}\|_2$ die örtliche Distanz und $\delta_t := t_S^i - t_E^j$ die zeitliche Distanz. Folgende Bedingung muss erfüllt sein:

$$\frac{\delta_p}{\delta_t} \leq 8{,}0\frac{\mathrm{m}}{\mathrm{s}} \tag{4.31}$$

4. Sowohl zeitliche Nähe als auch örtliche Nähe sprechen dafür, dass zwei Trajektorien zu demselben Spieler gehören. Daher wird von den Trajektorien, die obige Bedingungen erfüllen, die Trajektorie zurückgegeben, die $\delta_{tp} := \sqrt{\delta_t \cdot \delta_p}$ minimiert, also das geometrische Mittel aus räumlicher und zeitlicher Distanz. Kann keine geeignete Trajektorie gefunden werden, so liefert `getNearest`(X_i,X) als Ergebnis $\varnothing$ zurück.

Der Aufruf `union`(X_i, X_j) in Zeile 6 vereinigt die zwei übergeben Trajektorien und gibt die Vereinigung $X_k := X_i \cup X_j$ mit $id_k := id_i$ und $lbl_k := lbl_i$ zurück.

Algorithmus 4.3 : Vereinigung der Trajektorien

Input : X, Menge unterbrechungsfreier Trajektorien

Output : X, Menge der vereinigten Trajektorien

```
1 for t ← t_S to t_E do
2     foreach X_i ∈ X with t_S^i = t do
3         X_j ← getNearest(X_i, X)
4         if X_j ≠ ∅ then
5             X ← X \ {X_i, X_j}
6             X ← X ∪ union(X_i, X_j)
7 return X
```

4.5.7 Rückprojektion ins Bild

Die geglätteten und vereinigten Trajektorien bezüglich der Feldkoordinaten können mit Hilfe der invertierten Homographien aus Abschnitt 4.5.2 wieder in das Bild zurück transformiert werden. Ein projizierter Punkt im Bild repräsentiert den Mittelpunkt der Unterkante einer Bounding-Box. Die Größe der Bounding-Box wird mit Hilfe des Verfahrens aus Abschnitt 3.4 geschätzt. Zur Initialisierung der Koeffizienten werden die ursprünglichen Erkennungen verwendet. Dies resultiert in der Regel in einem glatten Ergebnis

ohne große Sprünge von Bild zu Bild. Die Voraussetzung dafür ist eine korrekt bestimmte Homographie. Ist diese fehlerbehaftet, kommt es dennoch zu Sprüngen u.ä., die alle Objekte im Bild betreffen.

4.6 Diskussion und Ausblick

Im ersten Teil dieses Kapitels wurde ein Verfahren zur Online-Spielerverfolgung in 2D-Bildkoordinaten vorgestellt. Die initialen Spielerpositionen sowie die Farbverteilungen der Mannschaftstrikots werden mit einer iterativen Methode robust erkannt, wobei die verfeinerte Größenschätzung in jeder Iteration die Erkennungsleistung steigert. Der effiziente Messvorgang nutzt die aussagekräftigen Merkmale aus Kapitel 3 für eine lokale Suche in der Konfidenzkarte. So wird in dem regelbasierten Ansatz das Problem der Datenvereinigung mit einem gierigen Vorgehen gelöst sowie der Verlust und die Wiedererkennung von verfolgten Spielern modelliert.

Die initiale Spielererkennung ist ein kritischer Schritt, da sie die Basis der Extraktion der Mannschaftsfarben darstellt. Werden hier fehlerhafte Regionen detektiert, kann sich das negativ auf die komplette Prozessierung einer Sequenz auswirken. Diesem Problem kann entgegengewirkt werden, indem die Spielererkennung in regelmäßigen Abständen auf dem vollständigen Bild durchgeführt wird. So können die Farbvorlagen im Laufe der Zeit korrigiert werden. Allerdings bedeutet dies auch eine deutliche Zunahme des Berechnungsaufwands (abhängig von der Dauer zwischen zwei Erkennungsvorgängen).

Der Vorteil der gierigen, lokalen Datenvereinigung ist vor allem die einfache und effiziente Berechnung. Sie begründet sich dadurch, dass ein Großteil der Mehrdeutigkeiten in lokaler Umgebung wieder aufgelöst werden kann. Da jeder Spieler als individuelles Objekt verfolgt wird, besteht die Möglichkeit einer parallelen Ausführung und einer echtzeitfähigen Performanz. Bei langen Überdeckungen, einem großen zeitlichen Abstand zwischen dem Verlassen des Bildes und dem Wiedereintritt sowie bei einer Überdeckung von sehr vielen Spielern, besteht allerdings eine hohe Wahrscheinlichkeit, dass die Spieler einer Mannschaft nicht wieder korrekt zugeordnet werden. Dieses Problem ist genereller Natur und besteht auch bei komplexeren Verfahren, die beispielsweise auf globalen, graphentheoretischen Optimierungsverfahren basieren. Abhilfe könnten hier die gezielte Extraktion von individuellen optischen Merkmalen schaffen, wie die Farbe der Schuhe oder der Haare, wobei hier die kleinen Strukturen zu erheblichen Ungenauigkeiten führen können. Eine weitere Möglichkeit ist es, die 3D-Feldkoordinaten zu nutzen und in Anlehnung an den Ansatz von Hoyningen-Huene (Hoyningen-Huene 2011), Spieler über die taktische Position zu identifizieren. Dieses Vorgehen wird aufgrund der unvollständigen Information durch ein eingeschränktes Blickfeld signifikant erschwert.

Im zweiten Teil des Kapitels wurde ein Verfahren vorgestellt, dass die 3D-Spielfeldkoordinaten nutzt, um die Trajektorien der Spieler anhand von realen physikalischen Randbedingungen zu verfeinern und mit Hilfe von einfachen Regeln falsch-positive Erkennungen zu entfernen. Durch ihre Einfachheit ist diese Methode sehr effizient und kann nach Ablauf einer Videosequenz die nachträglich verbesserten Ergebnisse annähernd in Echtzeit zur Verfügung stellen. Dafür ist natürlich ein echtzeitfähiges Verfahren zur Bestimmung der Koordinatentransformation notwendig. Die schnelle Berechnung würde auch eine Einbettung in das Online-Verfahren erlauben, die in bestimmten Abständen die Ergebnisse zur Laufzeit verfeinert. Die Methode ist nicht in der Lage langanhaltende Überdeckungen und Ähnliches aufzulösen. Hierzu müssten komplexere Regeln und das Ausnutzen der Bildinformation integriert werden, wodurch sich die Berechnungszeit um ein Vielfaches vergrößern würde.

Eine Voraussetzung für die Methode ist eine einigermaßen fehlerfreie Transformation vom 2D-Bildraum in den 3D-Spielfeldraum. Dies ist vor allem bei Szenen mit stärkerem Zoomfaktor und wenig sichtbaren Feldlinien, wie es bei Fernsehübertragungen häufig vorkommt, nicht immer gegeben. Solche Szenen können in vielen Situationen erkannt werden, beispielsweise anhand der Anzahl der sichtbaren Spieler oder anhand von starken Änderungen der Koordinatentransformation, und die 3D-Trajektorienverfeinerung in diesem Zeitraum ausgesetzt werden.

Wenn die extrahierten Bildmerkmale entsprechend angepasst werden, sind die Verfahren in diesem Kapitel nicht auf eine bestimmte Anwendung beschränkt und können leicht in anderen Domänen angewendet werden.

Kapitel 5

Automatisierte Bestimmung von Parametern

Die Ideen dieses Kapitels basieren zu Teilen auf folgender Veröffentlichung:

Herrmann, M., Mayer, C. und Radig, B. (2014). „Automatic Generation of Image Analysis Programs“. In: *Pattern Recognition and Image Analysis* 24 (3), S. 400–408.

5.1 Einleitung

In diesem Kapitel wird ein Verfahren vorgestellt, um die Parameter für die Spielererkennung aus den Kapiteln 3 und 4 automatisch zu bestimmen. Ein entscheidender Faktor für die Performanz und die Robustheit eines Systems ist die Wahl geeigneter Parameter. Ziel der Verfahrensentwicklung ist die Minimierung der Anzahl der Parameter. Dies ist bei komplexen Systemen in der Regel nicht oder nur bis zu einem bestimmten Grad möglich.

Eine Möglichkeit der Parameterbestimmung ist die manuelle Festlegung anhand empirischer Evidenzen und Plausibilitäten. Zwar können der gesunde Menschenverstand und die Erfahrung eines Entwicklers dabei helfen, den Parameterraum signifikant einzuschränken. Allerdings ist die Repräsentativität und Fähigkeit zur Generalisierung der gewählten Parameter gemessen am zeitlichen Aufwand für die Auswertung verschiedener Parameterkombinationen eher gering. Zudem müssen gewählte und getestete Parameterwerte akribisch dokumentiert werden, um geeignete Anpassungen für neue Daten zu ermöglichen. Dennoch ist es in einigen Fällen erforderlich, Parameter per Hand zu bestimmen, beispielsweise wenn nicht ausreichend repräsentatives annotiertes Material zur Verfügung steht. So wurden die Parameter für die zeitliche Verfolgung der Spieler

im Rahmen dieser Arbeit empirisch bestimmt. Die Wahl der Parameter wird in Abschnitt 5.2.1 vorgestellt.

Eine Alternative dazu besteht in der automatischen Bestimmung der Parameter. Dies kann einerseits adaptiv in Echtzeit erfolgen. Das heißt die Parameter passen sich während der Berechnung an die „Bedürfnisse“ der Daten an, was jedoch leicht zu einer Überanpassung beziehungsweise einer ungewollten Anpassung führen kann. Beispielsweise birgt ein adaptives Template bei der Verfolgung die Gefahr, sich an ein falsches Zielobjekt anzupassen. Andererseits gibt es die gängige Methode des maschinellen Lernens: die automatische Bestimmung der Parameter anhand von repräsentativen Trainingsdaten. Dabei sind für das zugrundeliegende Optimierungsproblem drei zentrale Punkte zu beachten:

- **Zielfunktion**: Um Parameter zu bestimmen wird eine objektive Zielfunktion benötigt, die bei gewünschtem Verhalten des Verfahrens minimale oder maximale Werte annimmt. Die Wahl einer geeigneten Zielfunktion ist kritisch für die Robustheit und Performanz des Systems.
- **Optimierungsverfahren**: Ziel ist es, den Punkt im Parameterraum zu finden, an dem die Zielfunktion ihren minimalen / maximalen Wert annimmt. Dazu wird ein geeignetes Optimierungsverfahren benötigt. Aufgrund komplexer Parameterräume enden solche Verfahren meist in einem lokalen Optimum. Entscheidend ist es dabei, ein gutes lokales Optimum zu erreichen, das möglichst nahe am globalen Optimum liegt.
- **Trainingsdaten**: Die Trainingsdaten sollten repräsentativ sein, das heißt, sie sollten möglichst vollständig die Breite des gewünschten Anwendungsfelds abdecken. Im Kontext der Spielerverfolgung bedeutet dies, möglichst viele verschiedenartige Aufnahmebedingung wie Auflösung, Wetterbedingung, Zoomfaktoren und Ähnliches zu beinhalten.

Da der Parameterraum für die Spielererkennung teilweise diskrete und teilweise kontinuierliche Wertebereiche aufweist und über seine Struktur sehr wenig bekannt ist, sind herkömmliche Optimierungsverfahren (beispielsweise Gradientenabstieg) nicht geeignet. Eine vollständige Suche muss aufgrund des hohen Berechnungsaufwands ebenfalls ausgeschlossen werden, wie es im Artikel von Herrmann, Mayer u. a. (Herrmann, Mayer u. a. 2014) aufgezeigt wird. Der Parameterraum wächst mit jedem zusätzlichen Parameter exponentiell. Um dem zu begegnen, wird zum einem mit einer randomisierten Suche, wie von Bergstra und Bengio (Bergstra und Bengio 2012) vorgeschlagen, gearbeitet, die zusätzlich iterativ den Suchbereich einschränkt. Zum anderen werden die Parameter in

einem hierarchischen Ansatz zunächst für die Extraktion der Spielersilhouetten und die Outfitbestimmung separat ermittelt. Als Trainingsdaten dient ein Satz von annotierten Einzelbildern aus einer heterogenen Menge von Videosequenzen.

5.2 Parameter und Parameterbestimmung

5.2.1 Parameter für die 2D-Spielerverfolgung

Wie eingangs schon erwähnt, wurden die Parameter für die 2D-Spielerverfolgung aus Abschnitt 4.4.4 empirisch ermittelt. Dies ist der Tatsache geschuldet, dass die Parameter im zeitlichen Kontext angewandt werden. Daher müssten sie bei einer automatischen Bestimmung mit Bezug auf ganze Videosequenzen optimiert werden. Das bedeutet, dass zum einen der Aufwand für die Annotation eines umfangreichen und repräsentativen Trainingsdatensatzes immens hoch ist. Zum anderen ist der Berechnungsaufwand für eine solche Trainingsmenge nicht in einem vernünftigen Rahmen zu erbringen. Im Folgenden werden daher die empirisch ermittelten Parameterwerte aufgeführt und kurz erläutert.

5.2.1.1 Kalman-Filter

Bei der Festlegung der Varianzen für den Kalman-Filter muss die Größe des verfolgten Objekts im Bild berücksichtigt werden. Ein großes Objekt im Vordergrund wird in der Regel eine größere Beschleunigung in Bildkoordinaten aufweisen als ein kleines Objekt im Hintergrund. Das gleiche gilt bei unterschiedlichen Auflösungen / Zoomfaktoren der Kamera. Zu diesem Zweck wird eine durchschnittliche Objekthöhe (in Pixel) von

$$\overline{h} := 61 \tag{5.1}$$

angenommen. Die Varianz des Prozessrauschens bezüglich der Position q_p^2 (spektrale Dichte, zeit-kontinuierliche Varianz) spiegelt im Prinzip die potentiell mögliche Beschleunigung eines Objekts im Bild wieder. Mit Berücksichtigung der Objektgröße h ist sie wie folgt gewählt:

$$q_p := 100 \cdot \frac{h}{\overline{h}} \tag{5.2}$$

Das Modell nimmt an, dass die Änderung der Geschwindigkeit von Bild zu Bild normalverteilt mit einem Mittelwert von 0 ist. Bei einem Objekt mit durchschnittlicher Größe entspricht bei der Wahl von q_p nach Gleichung 5.2 die Standardabweichung der Verteilung grob 20 Pixel (wenn man nur die Diagonale der Kovarianzmatrix betrachtet). Die Wahl von q_p^2 sollte nicht zu groß ausfallen, da ansonsten die gewünschte Filterwirkung ausbleibt und die Vorhersagen weniger Aussagekraft haben. Auf der anderen Seite

können bei einer zu kleinen Wahl plötzliche Positionsänderungen durch einen schnellen Kameraschwenk nicht abgebildet werden. Die Varianz des Prozessrauschens bezüglich der Größe q_s^2 ist gewählt als

$$q_s := 7{,}5 \cdot \frac{h}{\overline{h}}. \tag{5.3}$$

Die Varianzen der Messungen der Position und der Größe sind nicht nur abhängig von der Größe des Objekts, sondern auch von der Qualität der Messung q und sind wie folgt gewählt:

$$r_p := 3 \cdot \frac{1}{q} \cdot \frac{h}{\overline{h}} \tag{5.4}$$

und

$$r_s := 20 \cdot \frac{1}{q} \cdot \frac{h}{\overline{h}}. \tag{5.5}$$

Die Gewichtung mit der Messqualität q dient dazu, dass Messungen mit höherer Qualität als vertrauenswürdiger eingestuft werden.

5.2.1.2 Statusbehandlung

Zur Behandlung des Status verfolgter und neuer Spieler werden folgende Parameterwerte gewählt:

$$\tau_{accept} := 0{,}75; \quad \tau_{reject} := 0{,}35; \quad N_{reject} := 2 \tag{5.6}$$

und

$$N_{label} := 10 \tag{5.7}$$

und

$$\tau_{ovlp} := 0{,}7; \quad N_{ovlp} := 5; \quad \tau_{meas} := 0{,}15. \tag{5.8}$$

5.2.2 Automatische Parameterbestimmung

Für die Elemente des Systems, die statische Einzelbilder auswerten, liegt es nahe, die Parameter automatisch zu ermitteln. Dazu gehören die Bestimmung der Spielersilhouetten (siehe Abschnitt 3.5), die Outfiterkennung (siehe Abschnitt 3.6) und die initiale Spielererkennung (siehe Abschnitt 4.4.2). Ein repräsentativer Trainingsdatensatz aus Einzelbildern kann mit deutlich weniger Aufwand erstellt werden und auch die Berechnungszeit ist beherrschbar. Im Folgenden werden die Details zur automatischen Optimierung der Parameter vorgestellt.

Abbildung 5.1: Eine kleine Auswahl an Bildern zum Einlernen der Parameter. Bildquellen oben: (MDR 2013), unten links: (Eurosport 2013c), unten rechts: (ZDF 2013b).

5.2.2.1 Trainingsdatensatz

Zum automatischen Einlernen wichtiger Parameter wurde ein Datensatz aus 125 Bildern erstellt und alle Personen auf der Grasfläche nach den Vorgaben aus Abschnitt 6.2 annotiert. Die Bilder wurden nach folgenden Kriterien ausgewählt:

1. Die Aufnahmen sind aus einer totalen Kameraposition gemacht, das heißt, das Spielfeld nimmt einen Großteil des Bildes ein und es ist eine ausreichend große Anzahl an Spielern (> 5) im Bild zu sehen. Nahaufnahmen, Aufnahmen von Hintertorkameras und Ähnliches sind nicht enthalten.

2. Unter Berücksichtigung von Punkt 1 sind eine Vielzahl von verschiedenen Kameraeinstellungen bezüglich Winkel, Zoom und Position abgedeckt.

3. Mit den Aufnahmen ist eine große Diversität an Wettbewerben (International, National, Jugend, Frauen etc.), an Quellen (verschiedene Fernsehsender, Amateuraufnahmen, DFL-Scoutingfeeds) und an Aufnahmeformaten (SD, HD, Full-HD) abgedeckt.

In Abbildung 5.1 sind exemplarisch einige Bilder inklusive Annotationen aus dem Trainingsdatensatz dargestellt.

5.2.2.2 Optimierung der Parameter

Die Optimierung der Parameter mithilfe der Referenzdaten stellt kein triviales Problem dar. Zum einen ist es in der Regel aufwendig eine Parameterkonfiguration auszuwerten, da das entsprechende Verfahren auf allen Referenzdaten ausgeführt werden muss. Zum anderen steigt die Anzahl der möglichen Parameterkonfigurationen exponentiell mit der Dimension des Parameterraums (*Curse of Dimensionality*, siehe (Bellman 1961) und (Bishop 2006, S. 180 ff.)). Prinzipiell gibt es verschiedene Möglichkeiten die Parameter eines Verfahrens zu optimieren:

- Rastersuche (*Grid Search*): Ein häufig genutzter Brute-Force-Ansatz der Parameteroptimierung ist die Rastersuche. Sie wird häufig bei der Optimierung der Hyperparameter einer *Support Vector Machine* angewendet (Hsu u. a. 2003). Dabei wird für jeden Parameter ein geeigneter diskreter Wertebereich festgelegt und systematisch alle möglichen Wertekombinationen ausgewertet. Die Wertekombination, die das beste Bewertungsmaß liefert, wird zurückgegeben. Die Güte und der zeitliche Berechnungsaufwand sind in direktem Maße abhängig von der Anzahl der Parameter und der Wahl der Wertebereiche. Stellen im Parameterraum mit einem optimalen Bewertungsmaß sollten von den Wertebereichen abgedeckt sein. Allerdings können umfangreiche Wertebereiche die Auswertezeit drastisch erhöhen, so dass eine praktikable Optimierung nicht mehr möglich ist (siehe dazu das Beispiel in (Herrmann, Mayer u. a. 2014)). Sinnvoll anwendbar ist die Rastersuche nur bei einer geringen Anzahl an Parametern mit kleinen Wertebereichen.

- Gradientenbasierte Suche: Bei gradientenbasierten Verfahren oder anderen gierigen Verfahren (wie beispielsweise ein Bergsteigerverfahren, *Hill Climbing* (siehe Russell und Norvig 2003, S. 325 ff.)) ist ein entscheidendes Kriterium, einen geeigneten Startpunkt zu wählen. Bei einer ungeeigneten Wahl läuft man Gefahr, in einem schlechten lokalen Optimum zu landen. Um dieses Problem zu umgehen, kann man mehrere Startpunkte wählen. Dies erhöht den Berechnungsaufwand erheblich. Zudem ist in der Regel über die Topographie des Parameterraums sehr wenig bekannt, so dass eine geeignete Wahl der Startpunkte nicht trivial ist. Da die Theorie von gradientenbasierten Verfahren auf kontinuierlichen Wertebereichen aufbaut, ist die Berücksichtigung von Parametern mit diskreten Wertebereichen ebenfalls nicht trivial (im Gegensatz zu Raster- und zufallsbasierter Suche).

- Zufallsbasierte Suche (*Random Search*): Bei der zufallsbasierten Suche werden Parameterkonfigurationen erzeugt, in dem die Werte für einen Parameter aus einer unabhängigen und gleichverteilten Wahrscheinlichkeitsverteilung gezogen werden. Die zufällig gezogenen Parameterkombinationen werden ausgewertet und die mit

dem besten Bewertungsmaß wird zurückgegeben. Bergstra und Bengio (Bergstra und Bengio 2012) zeigten anhand von empirischen Experimenten, dass die zufallsbasierte Suche bei gleichem Berechnungsaufwand gleich gut oder besser abschneidet als die Rastersuche. Die zufallsbasierte Suche kann auch noch effizient in einem Parameterraum mit hoher Dimensionalität durchgeführt werden. Für einen praktikablen Berechnungsaufwand weist bei großen Parameterräumen die zufallsbasierte Suche allerdings eine Unterabtastung auf.

5.2.2.3 Zufallsbasierte Suche mit adaptiven Wertebereichen

Um den Problemen der zufallsbasierten Suche zu begegnen, werden im Rahmen dieser Arbeit die folgenden zusätzlichen Maßnahmen durchgeführt:

1. Es werden eine oder mehrere empirisch ermittelte Parameterkonfigurationen mit ausreichend guten Ergebnissen vorgegeben.
2. Die Wertebereiche der Parameter werden als symmetrische Intervalle um die in Punkt 1 vorgegebenen initialen Werte definiert.
3. Die Parameterkonfigurationen werden für jeden Parameter unabhängig und gleichverteilt aus den in Punkt 2 definierten Wertebereichen gezogen.
4. Die Referenzdaten werden Bild für Bild durchlaufen und für jedes Bild wird die komplette Menge der Parameterkonfigurationen ausgewertet. So können regelmäßig nach einer bestimmten Anzahl von Bildern die schlechtesten Konfigurationen entfernt werden und so der Berechnungsaufwand für die folgenden Bilder reduziert werden.
5. Die Punkte 1 bis 4 werden iterativ wiederholt. Dabei werden die besten Parameterkonfigurationen der letzten Iteration als Startpunkt für die nächste Iteration in Punkt 1 verwendet und die Intervalle in Punkt 2 von Iteration zu Iteration enger gewählt. In späteren Iterationen wird demnach in der näheren Umgebung von Parameterkonfigurationen gesucht, die sich in der vorigen Iteration bereits als gut herausgestellt haben.

5.2.2.4 Bewertungsmaße

Um die Güte einer Parameterkonfiguration zu bewerten, sind geeignete Maße notwendig. Diese Maße vergleichen das Ergebnis eines Verfahrens mit den gegebenen annotierten Referenzobjekten. Für jedes Bild I_i ist eine Referenzmenge von Bounding-Boxen $G^i :=$

$\{G_1^i, \ldots, G_{n_i}^i\}$ gegeben, jeweils mit den zugehörigen Labels (Outfits) $lbl_1^i, \ldots, lbl_{n_i}^i$. In Abbildung 5.1 sind unterschiedliche Labels mit unterschiedlichen Farben der Bounding-Boxen dargestellt. Die Bewertungsmaße sind für das jeweilige Modul des Verfahrens anzupassen. So benötigt die Erkennung der Outfits eine andere Bewertungsfunktion als die Spielersegmentierung.

Da die Outfitbestimmung auf der Spielersegmentierung basiert und die initiale Spielererkennung wiederum von der Segmentierung der Spieler und der Ermittlung der Outfits abhängt, werden die Parameter in folgender Reihenfolge bestimmt:

1. Zuerst werden die Parameter für die Extraktion der Spielersilhouetten bestimmt.
2. Die in Punkt 1 bestimmten Parameter für die Spielererkennung werden danach festgesetzt und nur die Parameter der Outfitbestimmung optimiert.
3. Die Parameter der initialen Spielererkennung werden zum Schluss optimiert, wobei auch hier die Parameter aus den Schritten 1 und 2 nicht mehr verändert werden.

Im Folgenden werden die Bewertungsmaße der einzelnen Module vorgestellt:

Bewertungsmaß für die Bestimmung der Spielersilhouetten Der Algorithmus 3.2 gibt eine Vordergrundregion FG zurück, die im besten Falle ausschließlich die Bereiche aller Spieler umfasst. Zum Vergleich mit den Referenzdaten werden für jedes Bild I_i folgende Schritte durchgeführt:

1. Wie in den Abschnitten 3.3.1 und 3.3.2 beschrieben, werden die Feldhülle H und die Grasmaske I_{Grass} bestimmt.
2. Wie in Abschnitt 3.4 beschrieben, werden die Koeffizienten $\vec{c}$ für die Größenschätzung bestimmt, mit der Menge G^i als Eingabe.
3. Der Algorithmus 3.2 wird mit der Eingabe I_i, H, I_{Grass}, $\vec{c}$ und der zu bewertenden Parameterkonfiguration ausgeführt. Das Verfahren gibt als Ergebnis die Bildregion FG_i zurück.
4. Die Zusammenhangskomponenten CC_{FG_i} der Vordergrundregion werden ermittelt.
5. Die Bounding-Boxen der Zusammenhangskomponenten CC_{FG_i} werden ermittelt und zu einer Menge $B^i := \{B_1^i, \ldots, B_{m_i}^i\}$ zusammengefasst.

6. Da eine Zusammenhangskomponente des Ergebnisses mehrere Personen umfassen kann (z.B. bei einer Überdeckung), ist der Vergleich von einzelnen Bounding-Boxen nicht zielführend. Daher wird jeweils die Vereinigung der Bounding-Boxen gebildet, das heißt $G := \bigcup_{j=1}^{n_i} G_j^i$ und $B := \bigcup_{j=1}^{m_i} B_j^i$.

7. Es werden folgende Werte ermittelt: $\texttt{TP} := |G \bigcap B|$, $\texttt{FP} := |B \backslash G|$ und $\texttt{FN} := |G \backslash B|$, sowie $\texttt{F}_1 := \frac{2 \cdot \texttt{TP}}{2 \cdot \texttt{TP} + \texttt{FP} + \texttt{FN}}$.

Für eine Parameterkonfiguration wird das mittlere $\texttt{F}_1$-Maß über alle Bilder berechnet.

Bewertungsmaß für die Outfiterkennung Ziel der Outfiterkennung ist es, die verschiedenen Outfits, der im Bild zu sehenden Personen, farblich zu erkennen und zu unterscheiden. Die in Abschnitt 3.6 beschriebenen Verfahren ermitteln für eine Menge von Bounding-Boxen die dominanten Farben und die Farbhistogramme, welche die verschiedenen Outfits repräsentieren. Um die Übereinstimmung mit den Referenzdaten für jedes Bild I_i zu bewerten, werden folgende Schritte durchgeführt:

1. Wie in den Abschnitten 3.3.1 und 3.3.2 beschrieben, werden die Feldhülle H und die Grasmaske I_{Grass} bestimmt.

2. Wie in Abschnitt 3.4 beschrieben, werden die Koeffizienten $\vec{c}$ für die Größenschätzung bestimmt, mit der Menge G^i als Eingabe.

3. Es wird die Vordergrundregion bestimmt, wie in Algorithmus 3.2 beschrieben. Die Ausgabe ist die Bildregion FG_i.

4. Wie in Abschnitt 3.6 beschrieben, wird die Bestimmung der Outfithistogramme durchgeführt. Die Eingabe besteht aus I_i, G^i, I_{Grass} sowie der zu bestimmenden Parameterkonfiguration. Als Ergebnis wird für jede Bounding-Box ein zugehöriges Label $\hat{lbl}_1^i, \ldots, \hat{lbl}_{n_i}^i$ ausgegeben.

5. Ein intuitiver Vergleich der Referenzlabel und der Ergebnislabel ist nicht möglich. So können sich zum einen die absoluten Zahlenwerte der Labels sowie die Anzahl verschiedener Label unterscheiden. Zum anderen können einzelne Personen einer falschen Outfitgruppe zugeordnet sein. Es wird zunächst eine (optimale) Zuordnung von Referenzlabeln zu Ergebnislabeln ermittelt, bevor eine Bewertung vorgenommen werden kann. Seien $\gamma_1, \ldots, \gamma_k$ die verschiedenen Label-Werte der Referenzdaten und seien $\hat{\gamma}_1, \ldots, \hat{\gamma}_l$ die verschiedenen Label-Werte der Ergebnisdaten. Eine Zuordnung bestimmt $\min(k,l)$ Referenz-Ergebnis-Paare. Für jedes Paar $(\gamma_i, \hat{\gamma}_j)$ ist die Bewertung durch $|\{lbl | lbl = \gamma_i \wedge \hat{lbl} = \hat{\gamma}_j\}|$ gegeben. Das ist die Anzahl der Bounding-Boxen in G^i, die als Referenz das Label γ_i und im Ergebnis

das Label $\hat{\gamma}_j$ haben. Die Bewertung einer Zuordnung ist die Summe ihrer Paar-Bewertungen. Die Suche einer optimalen Zuordnung lässt sich damit als maximales Matching in einem vollständigen bipartiten Graphen formulieren und wird mit der ungarischen Methode (Kuhn-Munkres-Algorithmus (Kuhn 1955; Munkres 1957)) gelöst.

6. Die Bewertung des Ergebnisses ergibt sich aus der Bewertung der optimalen Zuordnung im Verhältnis zur Anzahl aller Referenzobjekte n_i.

Für eine Parameterkonfiguration wird die durchschnittliche Bewertung der Zuordnungen über alle Bilder berechnet.

Bewertungsmaß für die Spielererkennung Das Bewertungsmaß der Spielererkennung bezieht sowohl die initiale Spielererkennung aus Abschnitt 4.4.2 als auch einen Messvorgang für vorgegebene Spielerpositionen (Vorhersagen) aus Abschnitt 4.4.1 mit ein. Zunächst wird die Spielerverfolgung mit dem Eingabebild I_i initialisiert und im zweiten Schritt wird ein zusätzlicher Messvorgang mit demselben Bild durchgeführt, wobei als Vorhersagen die künstlich verrauschten Referenzpositionen genutzt werden. Die Bewertung wird in folgenden Schritten durchgeführt:

1. Wie in Abschnitt 4.4.2 wird die initiale Spielerdetektion mit dem Eingabebild I_i durchgeführt. Das Ergebnis ist eine Menge von Bounding-Boxen $B^i := \{B^i_1, \ldots, B^i_{m_i}\}$. inklusive der jeweiligen Konfidenzwerte $q^i_1, \ldots, q^i_{m_i}$.

2. Die Referenzdaten $G^i := \{G^i_1, \ldots, G^i_{n_i}\}$ werden zu $\overline{G}^i := \{\overline{G}^i_1, \ldots, \overline{G}^i_{n_i}\}$, indem sie mit künstlichem, zufällig generiertem Rauschen versetzt werden. Dazu werden für jede Bounding-Box x gleichverteilt aus dem Intervall $\pm 0{,}3 \cdot w$ verändert sowie y und h jeweils gleichverteilt aus dem Intervall $\pm 0{,}3 \cdot h$. Die Breite wird wie gehabt auf $0{,}41 \cdot h$ gesetzt.

3. Mit der in Punkt 1 initialisierten Spielerdetektion wird eine Messung der Spielerpositionen durchgeführt, wobei als Vorhersage die verrauschten Referenzdaten $\overline{G}^i$ genutzt werden. Das Ergebnis ist eine Menge von Bounding-Boxen $\overline{B}^i := \{\overline{B}^i_1, \ldots, \overline{B}^i_{o_i}\}$. inklusive der jeweiligen Konfidenzwerte $\overline{q}^i_1, \ldots, \overline{q}^i_{o_i}$

4. Die erkannten Bounding-Boxen B^i werden den Referenzdaten G^i bezüglich des Jaccard-Koeffizient (siehe Gleichung 2.3) zugeordnet. Dies geschieht über ein maximales Matching in einem vollständigen bipartiten Graphen (siehe dazu auch Abschnitt 6.3.2.1). Es werden nur Bounding-Box-Paare zugeordnet, deren Jaccard-Koeffizient größer als 0,7 ist.

5. Analog zu Punkt 4 werden auch die Ergebnisse $\overline{G}^i$ den Referenzdaten G^i zugeordnet. Dabei werden nur Bounding-Box-Paare zugeordnet, deren Jaccard-Koeffizient größer als 0,5 ist.

6. Nach den Zuordnungen in den Punkten 4 und 5 werden nun alle zugeordneten Referenz-Bounding-Boxen (TP), alle nicht zugeordneten Referenz-Bounding-Boxen (FN) und alle nicht zugeordneten Ergebnis-Bounding-Boxen (FP) gezählt und das $\mathtt{F}_1$-Maß berechnet, mit $\mathtt{F}_1 := \frac{2 \cdot \mathtt{TP}}{2 \cdot \mathtt{TP} + \mathtt{FP} + \mathtt{FN}}$.

Für eine Parameterkonfiguration wird das durchschnittliche $\mathtt{F}_1$-Maß über alle Bilder berechnet.

Sowohl die automatisch als auch die empirisch bestimmten Parameter sind im Anhang B aufgeführt.

5.3 Diskussion und Ausblick

Das in diesem Kapitel vorgestellte Verfahren zur automatischen Ermittlung der Parameter für die Spielererkennung und -verfolgung basiert auf einer zufallsbasierten Suche. Um die Suche zu leiten, wird der Suchraum iterativ auf Intervalle um vielversprechende Stellen herum eingeschränkt. Die Länge dieser Intervalle wird von Iteration zu Iteration reduziert und die Methode dadurch zu einer Art Konvergenz geleitet. Die gefundenen optimalen Stellen sind weder garantierte globale noch lokale Optima. Dennoch kann durch das iterative Vorgehen angenommen werden, dass sie einer globalen optimalen Stelle sehr nahe kommen. Da der multidimensionale Parameterraum exponentiell mit der Anzahl der Parameter wächst, ist aufgrund des enormen Berechnungsaufwands eine erschöpfende Rastersuche nicht zielführend. Des Weiteren sind durch die gemischte diskret-kontinuierliche Struktur gradientenbasierte Methoden nicht direkt anwendbar. Zudem verhindert die zufallsbasierte Suche das Hängenbleiben in einem lokalen Optimum oder auf einem Plateau und reduziert die Gefahr einer Überanpassung. Die Methode ermöglicht außerdem das Einbinden von Expertenwissen, da die initialen Intervalle anhand von plausiblen Werten gewählt werden können.

Entscheidend für den Erfolg des Verfahrens ist die Wahl der Zielfunktion. Zum einen hat sie direkten Einfluss auf die tatsächliche Güte der ermittelten Parameter. Zum anderen ist der Berechnungsaufwand der Funktion ein wichtiger Faktor, der beeinflusst, ob die Suche in einem multidimensionalen Parameterraum praktikabel ist. In diesem Kapitel wurden Zielfunktionen für die Bestimmung der Spielersilhouetten, die Outfiterkennung

und die initiale Spielererkennung vorgestellt. Diese Bewertungsmaße zeichnen sich dadurch aus, dass sie zum einen Versuchen, die Gewichtung von Sensitivität und positiven Vorhersagewert zu balancieren. Zum anderen sind sie effizient in praktikabler Laufzeit zu berechnen.

Die bestimmten Parameter sind natürlich immer nur so repräsentativ wie der genutzte Datensatz. Daher wurde zur automatischen Parameterbestimmung ein Menge von Bildern genutzt, die eine Vielfalt von äußeren Rahmenbedingungen abdeckt und so für eine Generalisierbarkeit sorgt.

In diesem Kapitel wurden die Parameter für die statischen, einzelbildbasierten Operationen der Spielerverfolgung bestimmt. Beide oben genannten Punkte, die Berechnungszeit und die Anforderung an die Trainingsdaten, erschweren dasselbe Vorgehen für die Operationen, die den zeitlichen Kontext integrieren. Zum einen ist es deutlich aufwendiger ganze Videosequenzen zu annotieren. Zum anderen ist die Prozessierungsdauer pro annotierte Sequenz länger als für ein Einzelbild. Deshalb wurden diese Parameter im Rahmen dieser Arbeit mit dem traditionellen manuellen Verfahren empirisch anhand von Sichtkontrollen bestimmt. Abhilfe könnte ein semi-automatisches Verfahren zur Annotierung schaffen, bei dem automatisch erzeugte Verfolgungsresultate per Hand nachgebessert und korrigiert werden. Zudem kann eine Konzentration auf sehr kurze Sequenzen mit schwierigen Szenen den Annotationsaufwand erträglich gestalten.

Die Anwendbarkeit der vorgestellten Methode ist nicht auf eine Domäne beschränkt. Für andere Anwendungen müssen lediglich die Zielfunktion und der Trainingsdatensatz geeignet angepasst werden. Die vorgestellten Bewertungsmaße für die Silhouettenbestimmung und die Spielererkennung sind zudem ohne weitere Schwierigkeit für andere Fragestellungen der Objekterkennung einsetzbar.

Kapitel 6

Evaluierung

6.1 Einleitung

Für die Evaluierung von Verfahren zur Verfolgung von mehreren Objekten, sind Methoden erforderlich, mit denen man objektiv bewerten und vergleichen kann. In der Regel werden dazu sogenannte Softwarequalitätsmetriken nach dem IEEE Standard 1061-1998 (IEEE 1998) benutzt. Diese Metriken bilden Softwareeinheiten auf einen Zahlenwert ab, welcher aussagt, zu welchem Grad die Software eine bestimmte Eigenschaft besitzt.

Die Evaluierung der Objektverfolgung bedient sich dabei bei den Konzepten der Evaluierung im Bereich der Objekterkennung. Dazu werden in der Regel Referenzdaten, die sogenannte *Ground Truth*, erzeugt, in dem die Positionen von zu erkennenden Objekten in Bildern manuell markiert werden. Dieser Vorgang wird auch manuelle Annotation genannt. Die Ergebnisse der zu bewertenden Verfahren werden mit den vorgegebenen Referenzen verglichen, um verschiedene Maßzahlen zu berechnen, wie zum Beispiel den positiven Vorhersagewert (*Recall*) oder die Sensitivität (*Sensitivity*, *Precision*) (siehe zum Beispiel (Powers 2011)).

Der Sachverhalt bei der Objektverfolgung ist etwas komplexer als bei der reinen Objekterkennung, da hier nicht nur die räumliche Korrektheit betrachtet, sondern auch der zeitliche Verlauf bewertet werden muss. Das Ziel ist es, Objekte über den kompletten Zeitraum hinweg korrekt zu lokalisieren und zu identifizieren. Fehlerhaft zugeordnete Identifizierungen müssen als Fehler gewertet werden. Die Bewertungsmetriken der Objektverfolgung stellen diesbezüglich Erweiterungen der Metriken für die Objekterkennung dar. Da solche Erweiterungen viele mögliche Freiheitsgrade aufweisen, gibt es in der Forschungslandschaft leider keinen etablierten und anerkannten Standard. Am weitesten akzeptiert sind wohl die CLEAR MOT Metriken (Bernardin und Stiefelhagen 2008).

Mit der jüngst angelaufenen Initiative des *Multiple Object Tracking Benchmark* (Milan, Leal-Taixé, Schindler u. a. 2015b; Leal-Taixé u. a. 2015; Milan u. a. 2016) sollen Datensätze und Auswertungsmetriken zur Evaluierung vereinheitlicht werden und somit die Vergleichbarkeit neuer Verfahren verbessert werden. Obwohl die CLEAR MOT Metriken mehr oder weniger etabliert sind, mangelt es bei den meisten Veröffentlichungen zu diesem Thema an der Eindeutigkeit. Dies führt zu diversen Freiheitsgraden bei der Implementierung. Das kann auch an den zahlreichen frei verfügbaren Softwarepaketen (teilweise quelloffen) abgelesen werden, die (im Detail) Unterschiede bei ein und derselben Metrik aufweisen, die häufig nur durch Experimente nachzuvollziehen sind (Milan u. a. 2013). Hierzu sei auf eine kleine Fallstudie im Anhang C verwiesen. Dieses Kapitel versucht daher, die wichtigsten und am meisten etablierten Metriken formal und eindeutig zu definieren und die wichtigsten Freiheitsgrade zu benennen. Dies kann als wichtiger wissenschaftlicher Beitrag aufgefasst werden, da es bisher an einer eindeutigen und formalen Definition mangelt, auch wenn die Initiative des *Multiple Object Tracking Benchmark* (Milan u. a. 2016) in die richtige Richtung geht.

Bei Sportspielen wie Fußball, ist neben der Identifizierung einzelner Spieler, auch die Erkennung der Mannschaftszugehörigkeit wichtig. Die gängigen Bewertungsmaße für die Verfolgung mehrerer Objekte sind nur für die individuelle Identifizierung (beispielsweise von Fußgängern) ausgelegt und berücksichtigen diesen Aspekt nicht. Zudem wird bei vielen Implementierungen die Verfolgung der Objekte nicht über die Bildgrenze hinweg bewertet. Das heißt, ein Spieler, der den Bildbereich verlässt und wieder betritt wird als neues Objekt betrachtet. Dabei ist bei der Analyse von Sportspielen häufig die korrekte Identifizierung der Spieler über die komplette Videosequenz hinweg von Nöten. In diesem Kapitel werden daher die etablierten Metriken für die Verfolgung mehrerer Objekte an die Bedürfnisse der Fußballanalyse angepasst und erweitert.

Nach der formellen Einführung werden in diesem Kapitel die Metriken genutzt, um die Verfahren zur Spielerverfolgung in 2D und 3D aus Kapitel 4 zu evaluieren. Die 2D-Spielerverfolgung wird anhand einer Menge von Videosequenzen untersucht. Da es im Bereich der Verfolgung von Spielern in Fußballvideos (nach Wissen des Autors) keine breit angelegten, öffentlich verfügbaren Testdatensätze gibt, wurde im Rahmen dieser Arbeit eine repräsentative Menge von Videosequenzen ausgewählt. Dabei wurde darauf geachtet, eine Heterogenität in Bezug auf Videoauflösung, Videoqualität, Zoomfaktor, Aufnahmemodalität und ähnlichen Kriterien zu generieren. Für die meisten Videosequenzen war keine (oder nur teilweise) *Ground Truth* verfügbar und wurde im Rahmen dieser Arbeit manuell annotiert. Der entstandene Datensatz kann in Zukunft auch anderen Wissenschaftlern für die Bewertung und den Vergleich von Verfahren dienen. Um die erzielten Ergebnisse des in dieser Arbeit vorgestellten Verfahrens besser einordnen zu können, werden sie mit den Ergebnissen von anderen frei zugänglichen Verfahren

zur Mehrpersonenverfolgung, die den Stand der Technik repräsentieren, verglichen. Dabei werden neben der schnelleren Auswertezeit in den meisten Bereichen auch deutlich bessere Ergebnisse bei der Verfolgung erzielt.

Für die 3D-Spielerverfolgung werden Aufzeichnungen von vier Bundesligapartien (jeweils ein Halbzeit) genutzt. Diese Aufnahmen dienen den Vereinen zur nachträglichen Analyse des Spiels und sind so angefertigt, dass die meiste Zeit alle Feldspieler im Blickfeld sind (sogenannte *Scoutingfeeds*). Für die vier Aufnahmen stehen sowohl 3D-Positionsdaten der Spieler von einem professionellen System (TRACAB 2015) als auch Transformationen von Bild- zu Spielfeldebenen für jedes Einzelbild, ermittelt mit dem Verfahren von Hoernig (Hoernig 2016), zur Verfügung. Diese Daten sind ebenso wie die 3D-Positionsdaten fehlerbehaftet. Diese Tatsache muss bei der Einordnung der Evaluierungsergebnisse der Spielerverfolgung gesondert berücksichtigt werden.

6.1.1 Stand der Forschung

Wie eingangs schon erwähnt, war die Evaluierung von Verfahren zur Verfolgung mehrerer Objekte bis vor kurzer Zeit sehr heterogen und wenig vergleichbar. Häufig haben die Autoren neuer Veröffentlichungen eigene Datensätze und Metriken oder unterschiedliche Implementierungen und Annotationen genutzt. Der Grund dafür war in erster Linie der Mangel an einem zentral koordinierten Datensatz inklusive Evaluierungsmetriken, öffentlichen Vergleichsportalen und Wettbewerben, wie sie beispielsweise für die Objekterkennung (Everingham u. a. 2015), die semantische Annotation von Bildern (Russakovsky u. a. 2015) oder die Rekonstruktion von Stereoaufnahmen (Scharstein u. a. 2014) verfügbar sind.

Eine Übersicht über die am häufigsten genutzten Metriken haben Milan u. a. (Milan u. a. 2013) zusammengestellt. Dabei gehen die Autoren auch auf generelle Herausforderungen bei der Evaluierung ein. So konnten sie durch Experimente herausstellen, dass verschiedene, öffentlich verfügbare Annotationen für eine Videosequenz mit unterschiedlichen Kriterien erstellt wurden und teilweise deutliche Abweichungen aufweisen. Zudem haben die Autoren festgestellt, dass die Definitionen der Metriken häufig unscharf sind und diverse Freiheitsgrade aufweisen. Daher können anhand der öffentlich verfügbaren Implementierungen leichte Unterschiede ausgemacht werden. Da viele Trackingansätze als Input eine Menge von Objekterkennung übernehmen, hängt die Qualität des Verfahrens auch stark vom verwendeten Erkennungsverfahren ab. Auch dies schmälert die Vergleichbarkeit. Daher stellen Milan u. a. (Milan u. a. 2013) die Forderung nach einer einheitlichen Evaluierungsplattform für die Mehrpersonenverfolgung auf, die Videosequenzen, zugehörige Erkennungen, *Ground Truth* und Evaluierungsskripte einheitlich zur Verfügung stellt.

Diese Forderung wurde dann auch mit dem *Multiple Object Tracking Benchmark* (Milan, Leal-Taixé, Schindler u. a. 2015b; Leal-Taixé u. a. 2015; Milan u. a. 2016) in die Tat umgesetzt. Leider liegt der Fokus dabei auf dem Szenario der Fußgängerverfolgung und umfasst keine Datensätze für Sportspiele. Dennoch werden die Erkenntnisse der Studie im Rahmen dieser Arbeit berücksichtigt. Daher werden zum einen die benutzten Richtlinien der Annotation detailliert aufgeführt (siehe Abschnitt 6.2) und zum anderen ist die Implementierung der Metriken formell und eindeutig wiedergegeben(siehe Abschnitt 6.3).

Die am weitesten anerkannten Metriken sind wohl die CLEAR MOT Metriken (Stiefelhagen u. a. 2007; Bernardin und Stiefelhagen 2008). Dabei werden in jedem Einzelbild die Ergebnisse des zu evaluierenden Verfahrens anhand des Jaccard-Koeffizienten (siehe Gleichung 2.3) den Daten der *Ground Truth* zugeordnet. Dabei wird auch ermittelt, wie gut sich Zuordnungen über die Zeit aufrecht erhalten lassen. Dabei wird für die MOTA (*Multiple Object Tracking Accuracy*) die Anzahl der falsch-negativen, der falsch-positiven und der Identitätswechsel zu einer normierten Zahl über die komplette Videosequenz verrechnet. Zur Bewertung der örtlichen Genauigkeit wird für die MOTP (*Multiple Object Tracking Precision*) die mittlere Abweichung aller richtig-negativen Ergebnisse zu den korrespondierenden *Ground Truth* Objekten ermittelt. Die Metriken wurden im Laufe der Zeit von anderen Autoren angepasst, wie beispielsweise durch Gewichtung der Summanden der MOTA (Kasturi u. a. 2009; Ellis und Ferryman 2010) oder durch strengere Bewertung von Identitätswechseln bei Sportspielen (Ben Shitrit u. a. 2011). Die CLEAR MOT Metriken kommen auch in dieser Arbeit zum Einsatz, werden allerdings für die Anwendung bei Sportspielen erweitert.

Ebenfalls häufig genutzt werden die von Li, Huang u. a. (Li, Huang u. a. 2009) vorgestellten trajektorienbasierten Maßzahlen, die angelehnt an die Veröffentlichung von Wu und Nevatia (Wu und Nevatia 2006) und L. Zhang u. a. (L. Zhang u. a. 2008) sind. Dabei wird die Verfolgung von ganzen *Ground Truth* Trajektorien als Ganzes bewertet und als größtenteils verfolgt (*Mostly Tracked*), größtenteils nicht verfolgt (*Mostly Lost*) oder teilweise verfolgt (*Partially Tracked*) eingestuft. Zudem wird die Fragmentierung einer Trajektorie ermittelt, das heißt, es wird die Anzahl der Unterbrechungen gezählt. Auch diese Metriken kommen im Rahmen dieser Arbeit zum Einsatz.

Neben den bereits erwähnten Metriken wurden auch noch zahlreiche andere Methoden zur Evaluierung von Verfahren zur Verfolgung mehrerer Objekte vorgestellt. Da sie sich in der Forschergemeinde nicht weiter durchgesetzt haben, werden sie hier kurz erwähnt, aber im Folgenden nicht weiter genutzt. Ein wirklich umfangreicher Satz an Kennzahlen

wurde von Smith u. a. (Smith u. a. 2005) vorgestellt. Dabei werden diverse Konfigurationsfehler (Falsch-positiv, Falsch-negativ, u.ä.), Zuordnungsfehler und trajektorienbasierte Fehler ermittelt. Allerdings stehen diese Kennzahlen für sich alleine und werden nicht (wie beispielsweise bei der MOTA) zu einer Metrik verrechnet. Dieselbe Problematik trifft auch auf die von Nghiem u. a. (Nghiem u. a. 2007) vorgestellten Metriken zu. Dabei werden zahlreiche Kennzahlen für die Objekterkennung, die Objektlokalisation und die Objektverfolgung genutzt.

Metriken zur Evaluierung von Systemen zur Verfolgung von Fußballspielern wurden von Li u. a. (Li u. a. 2005) vorgestellt. Diese Metriken zielen in erster Linie auf Multikamerasysteme mit Kalibrierung ab. Es werden jedoch auch Fehler bei Zuordnung der Identität und der Mannschaftszugehörigkeit bewertet. Die letztere Idee wird in dieser Arbeit aufgegriffen und mit den CLEAR MOT Metriken kombiniert.

6.2 Annotation

Die Annotation der Videosequenzen wurde mit VATIC (Vondrick u. a. 2012) und in Anlehnung der Vorgaben der *PASCAL VOC Challenge* (Everingham u. a. 2010) (Everingham u. a. 2007) durchgeführt:

- Die Position eines Objektes (Spieler, Schiedsrichter usw.) wird durch eine Bounding-Box beschrieben.
- Als Objekte werden alle Spieler, der Schiedsrichter und die Linienrichter im Bild annotiert sowie Personen, deren Kontur sich mit der Grasfläche des Spielfeldes überlappen. Personen außerhalb des Spielfeldes (zum Beispiel auf einer Grasfläche hinter der Bande) werden nicht annotiert.
- Ein Objekt wird nicht annotiert, wenn das Objekt im Bild sehr klein ist ($< 10-20$ Pixel) oder weniger als $10-20\%$ des Objektes sichtbar sind.
- Die Bounding-Box sollte alle sichtbaren Pixel umschließen, es sei denn die Bounding-Box müsste übermäßig vergrößert werden, um einige wenige zusätzliche Pixel ($< 5\%$) zu umfassen.
- Ist das Objekt überdeckt, so wird die Bounding-Box in der geschätzten Größe des überdeckten Objektes definiert (im Unterschied zu (Everingham u. a. 2007)).
- Jedem Objekt wird eine eindeutige Identifizierungsnummer (*id*) und ein Label (*lbl*) aus den sechs Kategorien (Spieler Mannschaft 1, Torwart Mannschaft 1, Spieler Mannschaft 2, Torwart Mannschaft 2, Schiedsrichter / Linienrichter, Sonstiges)

zugeordnet. Beides ist für eine ganze Videosequenz gültig (auch wenn das Objekt zeitweise nicht sichtbar ist).

- Sollte ein Objekt während der Videosequenz den Bildbereich verlassen und zu einem anderen Zeitpunkt wieder in den sichtbaren Bereich kommen, so sind diesem Objekt die Identifizierungsnummer und Kategorie zuzuordnen, die dem Objekt vor dem Verlassen des Bildbereiches zugeordnet waren (falls eine eindeutige Zuordnung möglich ist).

Dollar et al. (Dollár, Wojek u. a. 2012) schlagen eine nachträgliche Normierung des Seitenverhältnisses Breite zu Höhe von 0,41 : 1 von Referenzannotationen und Ergebnissen vor. Dies wurde im Rahmen dieser Arbeit ebenfalls gemacht. Insbesondere wird dabei der Fokus auf die entscheidende Verfolgung des Körpers gelegt und ausgebreitete Posen mit Beinen oder Armen vernachlässigt, wie in Abbildung 6.1 dargestellt ist.

Abbildung 6.1: Normierung des Seitenverhältnis. Die gezeichneten Rechtecke haben ein normiertes Seitenverhältnis von 0,41 : 1. Bildquellen: links (Eurosport 2013b); rechts (ZDF 2012)

6.3 Metriken zur Evaluierung von Verfahren zur Verfolgung mehrerer Ziele

6.3.1 Eingabedaten

Eine Methode zur Evaluierung der Objektverfolgung erhält als Eingabe sowohl die manuell erstellten Referenzdaten (*Ground Truth*) als auch die vom Verfolgungsverfahren automatisch generierten Ergebnisdaten. Diese Daten sind jeweils zu diskreten Zeitpunkten (z.B. Einzelbilder einer Videosequenz) in einem Zeitabschnitt $[t_S; t_E] \subset \mathbb{N}$ gegeben.

Ein Objekt o zum Zeitpunkt t wird dabei repräsentiert durch ein Positionstupel $p(o,t) \in \mathbb{R}^n$, einer Identifizierungsnummer $b(o) \in \mathbb{N}$ und einer Kategorie $c(o,t) \in \mathbb{N}$. Im Falle der

Referenzdaten ist $c(o,t)$ über die Zeit konstant, während die Zuordnung in den Ergebnisdaten prinzipiell variabel ist (zum Beispiel durch eine Korrektur der Mannschaftszuordnung nach einer bestimmten Zeit der Verfolgung). Im Falle von 2D-Daten ist $n := 4$ (Bounding-Box) und von 3D-Daten ist $n := 2$ (siehe unten).

Für jeden Zeitpunkt t ist also eine Menge von Referenzobjekten $G^t := \{g_1^t, \ldots, g_{m^t}^t\}$ und eine Menge von Ergebnisobjekten $R^t := \{r_1^t, \ldots, r_{n^t}^t\}$ mit $n^t, m^t \in \mathbb{N}$ gegeben. Es kann durchaus vorkommen, dass für ein Objekt g gilt $g \in G^{t_1}$ und $g \in G^{t_3}$, aber $g \notin G^{t_2}$ mit $t_1 < t_2 < t_3$, beispielsweise, wenn das Objekt für eine gewisse Zeit das Blickfeld der Kamera verlässt und es später wieder betritt. Das gilt natürlich ebenso für Ergebnisobjekte.

Des Weiteren ist eine Distanzfunktion $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{\geq 0}$ definiert, die den Abstand zweier Objekte angibt. Für zwei Objekte o_i und o_j wird die Schreibweise $d(o_i, o_j, t) := d(p(o_i, t), p(o_j, t))$ verwendet.

Prinzipiell sind die hier vorgestellten Metriken für eine Vielzahl von Anwendungen anwendbar. Es müssen nur die Positionstupel und die Distanzfunktion entsprechend modelliert werden. Bei der Verfolgung von Personen in einer Videosequenz gibt es in der Regel folgende zwei Szenarien, die auch in dieser Arbeit zum Einsatz kommen:

1. **2D:** Die Personen werden als 2D-Objekte im Bild verfolgt. Sie werden als Bounding-Boxen repräsentiert (siehe Gleichung 2.2). Dabei gilt $n := 4$, das heißt, für das Positionstupel eines Objektes o zur Zeit t gilt $p(o,t) \in \mathbb{R}^4$. Es repräsentiert die linke obere Ecke sowie Breite und Höhe der Bounding-Box $B(o,t) := B_o^t$. Analog zu Everingham u. a. (Everingham u. a. 2010) wird als Distanzmaß die sogenannte Jaccard-Distanz d_J verwendet, die sich über den Jaccard-Index o_J (siehe Gleichung 2.3) definiert:
$$d(o_i, o_j, t) := d_J(B(o_i, t), B(o_j, t)) \tag{6.1}$$
mit
$$d_J(B_i, B_j) := 1 - o_J(B_i, B_j) \tag{6.2}$$
.

2. **3D:** Die Personen werden als 3D-Objekte in der Ebene (z.B. das Spielfeld) verfolgt. Dabei werden die Positionen der erkannten Objekte im Bild über eine Transformation (z.B. mit den inneren und äußeren Kameraparameter) auf Punkte auf einer Ebene im Weltkoordinatensystem projiziert. Somit gilt $n := 2$, da ein Positionstupel für Objekt o zur Zeit t die Koordinaten in der Ebene repräsentiert. Als

Distanzfunktion kommt hierbei die euklidische Distanz zum Einsatz, das heißt:

$$d(o_i, o_j, t) := \|p(o_i, t) - p(o_j, t)\|_2 \tag{6.3}$$

.

6.3.2 Zuordnung

Die Basis für die weitere Auswertung ist eine Zuordnung von Ergebnisobjekten zu Referenzobjekten innerhalb eines Einzelbildes. In vielen Fällen ist diese Zuordnung eindeutig. Allerdings können durch Überdeckungen, Mehrfachverfolgungen und den zeitlichen Verlauf Mehrdeutigkeiten entstehen, die durch verschiedene Möglichkeiten aufgelöst werden können. Die Zuordnung wird über eine Funktion $M^t : G^t \times R^t \rightarrow \{0, 1\}$ definiert, mit $\forall r_j^t \in R^t : \sum_{i=1}^{m^t} M^t(g_i^t, r_j^t) \leq 1$ und $\forall g_i^t \in G^t : \sum_{j=1}^{n^t} M^t(g_i^t, r_j^t) \leq 1$. Das heißt, dass jedem Referenzobjekt maximal ein Ergebnisobjekt und umgekehrt jedem Ergebnisobjekt maximal ein Referenzobjekt zugeordnet ist.

Alle Zuordnungsmethoden haben als Parameter die Distanzfunktion d (siehe Gleichungen 6.1 und 6.3) und einen Distanzschwellwert $\tau_d \in \mathbb{R}$. Dieser Schwellwert legt die maximale Distanz fest, bis zu der zwei Objekte zugeordnet werden können. Im Rahmen eines Verfolgungsverfahrens gibt es prinzipiell zwei Möglichkeiten der Zuordnung, die in den folgenden Abschnitten erläutert werden: die Zuordnung ohne zeitliche Komponente und die Zuordnung unter Einbezug der zeitlichen Komponente.

6.3.2.1 Zuordnung ohne zeitliche Komponente

Ohne Berücksichtigung des zeitlichen Verlaufs wird die Zuordnung in jedem Einzelbild einer Videosequenz neu vorgenommen. Dabei gibt es prinzipiell zwei Möglichkeiten:

1. **Gierig (*Greedy*):** Wie aus Algorithmus 6.1 zu entnehmen ist, werden die Referenzobjekte nacheinander durchlaufen und jedem Referenzobjekt wird das ihm nächstliegende (noch nicht zugeordnete) Ergebnisobjekt zugeordnet, falls die Distanz unterhalb des Schwellwertes τ_d liegt. Jedes Ergebnisobjekt kann dabei höchstens einmal zugeordnet werden. Das Ergebnis dieser Zuordnung ist hier abhängig von der Sortierung von G^t und R^t. Zudem ist es auch möglich innere und äußere Schleife zu vertauschen, also zuerst die Menge der Ergebnisobjekte zu durchlaufen und die Referenzobjekte zuzuordnen.

2. **Optimal:** Die global optimale Zuordnung stellt ein gewichtetes Zuordnungsproblem in einem vollständigen bipartiten Graphen dar und kann mit der ungarischen

Methode (Kuhn-Munkres-Algorithmus (Kuhn 1955; Munkres 1957))) in polynomieller Laufzeit von $\mathcal{O}(n^3)$ gelöst werden. Distanzen, die größer als der Schwellwert τ_d sind, werden vor der Berechnung auf ∞ gesetzt. Die Lösung ist weitestgehend unabhängig von der Sortierung der Objektmengen (nur wenn mehrere Objekte der einen Menge identische Distanzen zu einem Objekt der anderen Menge aufweisen, kann es potentiell mehrere Lösungen geben).

Algorithmus 6.1 : Gierige (*Greedy*) Zuordnung von Referenz- und Ergebnisobjekten

Input : Zeit t
Menge der Referenzobjekte G^t
Menge der Ergebnisobjekte R^t
Distanzfunktion d
Distanzschwellwert $\tau_d \in \mathbb{R}$
Output : Zuordnungsfunktion M^t

```
1  R ← R^t                                              /* Initialisierung */
2  foreach g ∈ G^t do                  /* Iteration über alle Referenzobjekte */
3      d_min ← ∞
4      foreach r ∈ R do                /* Iteration über alle Ergebnisobjekte */
5          M^t(g, r) ← 0
6          if d(g, r, t) < d_min then  /* Referenzobjekt mit minimaler Distanz */
7              r_min ← r
8              d_min ← d(g, r_min, t)
9      if d_min ≤ τ_d then                        /* Überprüfung mit Schwellwert */
10         M^t(g, r_min) ← 1
11         R ← R \ r_min
12 return M^t
```

6.3.2.2 Zuordnung mit zeitlicher Komponente

Wenn der zeitliche Verlauf berücksichtigt wird, wird zuerst versucht, Zuordnungen aus früheren Zeitpunkten wiederherzustellen. Dazu wird eine zweite Zuordnungsfunktion $M^t_{\texttt{last}}$ mit den jeweils letzten Zuordnungen in der Vergangenheit aufrechterhalten. Seien $G^{t_1 \to t_2} := \bigcup_{t_1 \leq t \leq t_2} G^t$ und $R^{t_1 \to t_2} := \bigcup_{t_1 \leq t \leq t_2} R^t$ und seien t_S der erste Zeitpunkt und t_E der letzte Zeitpunkt der Videosequenz. Dann gilt zur Zeit t mit $t_S \leq t \leq t_E$:

$$M^t_{\texttt{last}} : G^{t_S \to t_E} \times R^{t_S \to t_E} \to \{0, 1\} \tag{6.4}$$

mit

$$\forall g_i \in G^{t_S \to t} : \sum_{r_j \in R^{t_S \to t}} M^t_{\texttt{last}}(g_i, r_j) \leq 1 \tag{6.5}$$

und

$$\forall r_j \in R^{t_S \to t} : \sum_{g_i \in G^{t_S \to t}} M^t_{\texttt{last}}(g_i, r_j) \leq 1 \tag{6.6}$$

und

$$\forall g_i \in G^{t_S \to t_E} \setminus G^{t_S \to t} : \sum_{r_j \in R^{t_S \to t_E}} M^t_{\texttt{last}}(g_i, r_j) = 0 \tag{6.7}$$

und

$$\forall r_j \in R^{t_S \to t_E} \setminus R^{t_S \to t} : \sum_{g_i \in G^{t_S \to t_E}} M^t_{\texttt{last}}(g_i, r_j) = 0 \tag{6.8}$$

.

Aus Gleichung 6.4 geht hervor, dass der Definitionsbereich von $M^t_{\texttt{last}}$ das kartesische Produkt der Menge aller Referenzobjekte und der Menge aller Ergebnisobjekte der gesamten Videosequenz ist. Es werden demnach nicht nur Objekte des aktuellen Zeitpunktes t zugeordnet. Die Gleichungen 6.5 und 6.6 besagen, dass jedem Referenzobjekt höchstens ein Ergebnisobjekt und jedem Ergebnisobjekt höchstens ein Referenzobjekt zugeordnet ist. Die Gleichungen 6.7 und 6.8 besagen, dass zum Zeitpunkt t keine Referenz- und Ergebnisobjekte zugeordnet sind, die erst nach dem Zeitpunkt t auftreten.

Die Zuordnung zum Zeitpunk t erfolgt nun in folgenden Schritten (analog zu (Bernardin und Stiefelhagen 2008)):

1. **Initialisierung M^t:** Setze $\forall g \in G^t, \forall r \in R^t : M^t(g, r) := 0$.

2. **Herstellen alter Zuordnungen für $t > t_S$:** Setze $M^t(g, r) := 1$ für jedes Paar (g, r) für das $M^{t-1}_{\texttt{last}}(g, r) = 1$, $g \in G^t$, $r \in R^t$ und $d(g, r, t) \leq \tau_d$ gilt.

3. **Herstellen neuer Zuordnungen:** Wende ein Zuordnungsverfahren aus Abschnitt 6.3.2.1 an für alle $g_i \in G^t$ mit $\sum_{j=1}^{n^t} M^t(g_i, r_j) = 0$ und für alle $r_j \in R^t$ mit $\sum_{i=1}^{m^t} M^t(g_i, r_j) = 0$. Übernehme die ermittelten Zuordnungen nach M^t.

4. **Übernahme der alten Zuordnung** Für alle Paare $(g, r) \in G^{t_S \to t_E} \times R^{t_S \to t_E}$, initialisiere $M^t_{\texttt{last}}$ wie folgt:

$$M^t_{\texttt{last}}(g, r) := \begin{cases} M^{t-1}_{\texttt{last}}(g, r) & \text{für } t > t_S \\ 0 & \text{für } t = t_S \end{cases} \tag{6.9}$$

5. **Update der letzten Zuordnungen:** Für alle Paare $(g, r) \in G^t \times R^t$ mit $M^t(g, r) = 1$ setze

$$M^t_{\texttt{last}}(g, r_j) := 0, \forall r_j \in R^{t_S \to t_E} \setminus r, \tag{6.10}$$

$$M^t_{\texttt{last}}(g_i, r) := 0, \forall g_i \in G^{t_S \to t_E} \setminus g \text{ und} \tag{6.11}$$

$$M^t_{\texttt{last}}(g, r) := 1. \tag{6.12}$$

6.3.2.3 Varianten

Das Zuordnungsverfahren bietet zahlreiche Möglichkeiten der Variation, die im Folgenden erläutert werden:

- **Gierig kontra optimal:** Im Schritt 3 können beide Verfahren aus Abschnitt 6.3.2.1 zur Anwendung kommen.
- **Einzelbildweise Zuordnung:** Schritt 2 wird nicht durchgeführt. Die Zuordnung erfolgt wie in Abschnitt 6.3.2.1.
- **Ein-Zeitschritt Zuordnung:** Schritt 2 wird für $t > t_S$ abgeändert zu: Setze $M^t(g, r)) := 1$ für jedes Paar (g, r) für das $g \in G^{t-1}$, $r \in R^{t-1}$, $M^{t-1}(g, r) = 1$, $g \in G^t$, $r \in R^t$ und $d(g, r, t) \leq \tau_d$ gilt. Es werden also nur Zuordnungen aus dem letzten Zeitpunkt überprüft und wieder hergestellt.
- **Referenzbetonte Zuordnung:** Die Bedingung aus Gleichung 6.6 fällt weg und beim Update der letzten Zuordnung wird der Schritt aus Gleichung 6.11 nicht durchgeführt. Das bedeutet, dass für jedes Referenzobjekt die letzte Zuordnung gespeichert wird, unabhängig ob das entsprechende Ergebnisobjekt danach anderweitig zugeordnet wurde. Das kann dazu führen, dass zwei (oder auch mehrere) Referenzobjekte g_1 und g_2 zuletzt demselben Ergebnisobjekt r zugeordnet waren, also $M^{t-1}_{\texttt{last}}(g_1, r) = 1$ und $M^{t-1}_{\texttt{last}}(g_2, r) = 1$. Tritt nun der Fall ein, dass beide Objekte nah genug an r liegen, also $d(g_1, r, t) \leq \tau_d$ und $d(g_2, r, t) \leq \tau_d$, dann ist nicht eindeutig, welche von beiden Zuordnungen für den Zeitpunkt wieder hergestellt wird. Das kann in Abhängigkeit von der Sortierung von R^t geschehen (was nicht besonders intuitiv ist) oder in Abhängigkeit davon, welche der beiden Zuordnungen (zeitlich) zuletzt hergestellt wurde.
- **Ergebnisbetonte Zuordnung:** Analog zur referenzbetonten Zuordnung, nur dass die Bedingung aus Gleichung 6.5 und der Schritt in Gleichung 6.10 wegfallen.
- **Global optimal:** Es werden alle möglichen Zuordnungsmöglichkeiten über alle Zeitpunkte hinweg ermittelt und die *beste* Zuordnungsabfolge (bezüglich einer Metrik, z.B. MOTA) bestimmt. Anhand dieser werden die Endergebnisse berechnet. Die Anzahl der Zuordnungsmöglichkeiten steigt im ungünstigsten Fall exponentiell.

6.3.3 Richtig-positiv, falsch-positiv und falsch-negativ

Zur Evaluierung ist es interessant für jeden Frame zu zählen, wie viele Objekte richtig erkannt wurden (richtig-positiv), wie viele Objekte nicht erkannt wurden (falsch-negativ) und wie viele Fehlalarme (falsch-positiv) erzeugt wurden und diese Zahlen in Relation zu setzen.

Zum Zeitpunkt t sind diese Kennziffern demnach wie folgt definiert:

- Anzahl Referenzobjekte: $\mathtt{GT}^t := |G^t|$ und $\mathtt{GT} := \sum_{t=t_S}^{t_E} \mathtt{GT}^t$.
- Richtig-positiv: $\mathtt{TP}^t := |\{g \in G^t | \sum_{r \in R^t} M^t(g,r) = 1\}|$ und $\mathtt{TP} := \sum_{t=t_S}^{t_E} \mathtt{TP}^t$:
- Falsch-negativ: $\mathtt{FN}^t := |\{g \in G^t | \sum_{r \in R^t} M^t(g,r) = 0\}|$ und $\mathtt{FN} := \sum_{t=t_S}^{t_E} \mathtt{FN}^t$.
- Falsch-positiv: $\mathtt{FP}^t := |\{r \in R^t | \sum_{g \in G^t} M^t(g,r) = 0\}|$ und $\mathtt{FP} := \sum_{t=t_S}^{t_E} \mathtt{FP}^t$.

6.3.4 Genauigkeit, Trefferquote und F-Maße

Mit den Kennziffern aus 6.3.3 lassen sich nun klassische Bewertungsmaße berechnen, die beispielsweise bei der Beurteilung eines Klassifikators zum Einsatz kommen:

- Genauigkeit (auch positiver Vorhersagewert oder *Precision*): $\mathtt{PR} := \frac{\mathtt{TP}}{\mathtt{TP}+\mathtt{FP}}$
- Trefferquote (auch Sensitivität oder *Recall*): $\mathtt{RC} := \frac{\mathtt{TP}}{\mathtt{TP}+\mathtt{FN}}$
- F-Maß: $\mathtt{F}_\alpha := (1+\alpha^2) \cdot \frac{\mathtt{PR} \cdot \mathtt{RC}}{\alpha^2 \cdot \mathtt{PR} + \mathtt{RC}}$

In Worten beschreibt die Genauigkeit den Anteil der richtig erkannten Objekte an der Menge aller erkannten Objekte (Ergebnisobjekte). Die Trefferquote hingegen beschreibt den Anteil der richtig erkannten Objekte an der Menge aller zu erkennenden Objekte (Referenzobjekte). Beide Werte liegen in dem Intervall $[0;1]$ und höhere Werte spiegeln eine bessere Performance wider. Bei der Optimierung von Verfahren resultiert häufig eine Verbesserung in der Genauigkeit zu einer Verschlechterung in der Trefferquote (und umgekehrt). Aus diesem Grund wird meistens auch ein F-Maß betrachtet, das beide Werte (gewichtet) zusammenfasst. In der Regel wird hier das F_1-Maß benutzt, bei dem beide Werte gleich gewichtet sind und das somit das harmonische Mittel aus Genauigkeit und Trefferquote darstellt:

$$\mathtt{F}_1 := 2 \cdot \frac{\mathtt{PR} \cdot \mathtt{RC}}{\mathtt{PR} + \mathtt{RC}} \tag{6.13}$$

.

Mit $\mathtt{N} := \sum_{t=t_S}^{t_E} 1$, als die Anzahl der diskreten Zeitpunkte (Einzelbilder), wird die Anzahl falscher Alarme pro Einzelbild (*False Alarms per Frame*) wie folgt berechnet:

$$\mathtt{FAF} := \frac{\sum_{t=t_S}^{t_E} \mathtt{FP}}{\mathtt{N}} \tag{6.14}$$

.

6.3.5 Verwechslungen der Identität

Wie schon erwähnt ist es bei der Verfolgung von Objekten nicht nur wichtig ein Objekt zu erkennen, sondern es auch über die Zeit hinweg richtig zu identifizieren. So kann es beispielsweise nach einer Überdeckung zweier Objekte oder dem Aus- und Wiedereintritt eines Objektes in den Bildbereich leicht zu fehlerhaften Identifizierungen (*Id Switch* oder *Mismatch Error*) kommen. Aus diesem Grund wird versucht, Fehler in der Identifizierung in der Bewertungsmetrik zu berücksichtigen.

Die Anzahl der Verwechslungen zur Zeit $t > t_S$ wird nach der Zuordnung mit zeitlicher Komponente wie in 6.3.2.2 beschrieben berechnet (genauer gesagt nach Schritt 5).

Sei $\overline{G}^t$ die Menge der Referenzobjekte, die in Frame t erkannt wurden, also:

$$\overline{G}^t := \{g \in G^t | \sum_{r \in R^t} M^t(g,r) = 1\} \tag{6.15}$$

.

Und sei $\overline{G}^{t-1}_{\mathtt{last}}$ die Menge der Referenzobjekte, für die zum Zeitpunkt $t-1$ eine letzte Zuordnung gespeichert ist, das heißt:

$$\overline{G}^{t-1}_{\mathtt{last}} := \{g \in G^{t_S \to t_E} | \sum_{r \in R^{t_S \to t_E}} M^{t-1}_{\mathtt{last}}(g,r) = 1\} \tag{6.16}$$

.

Für die Elemente beider Mengen existiert dann jeweils eine eindeutige Zuordnung auf die zugeordneten Ergebnisobjekte, das heißt:

$$\overline{r}^t(g) = r \iff M^t(g,r) = 1 \tag{6.17}$$

und

$$\overline{r}^{t-1}_{\mathtt{last}}(g) = r \iff M^{t-1}_{\mathtt{last}}(g,r) = 1 \tag{6.18}$$

Die Menge der Referenzobjekte, bei denen es zum Zeitpunkt t einen Wechsel des zugeordneten Ergebnisobjekts gibt, ist

$$G^t_{\texttt{MME}} := \{g \in (\overline{G}^t \cap \overline{G}^{t-1}_{\texttt{last}}) | \overline{r}^t(g) \neq \overline{r}^{t-1}_{\texttt{last}}(g)\} \tag{6.19}$$

Analog sind die Mengen $\overline{R}^t$, $\overline{R}^{t-1}_{\texttt{last}}$, die Zuordnungen $\overline{g}^t$, $\overline{g}^{t-1}_{\texttt{last}}$ und die Menge $R^t_{\texttt{MME}}$ für die Ergebnisobjekte definiert.

Die Anzahl der Verwechslungen der Identität zur Zeit t ist im einfachsten Fall definiert als

$$\texttt{MME}^t_1 := |G^t_{\texttt{MME}}| \tag{6.20}$$

.

Alternativ werden auch die Wechsel der Zuordnungen bei Ergebnisobjekten miteinbezogen, das heißt:

$$\texttt{MME}^t_2 := |G^t_{\texttt{MME}}| + |R^t_{\texttt{MME}}| \tag{6.21}$$

.

In einer dritten Variante werden Wechsel der Zuordnung nur einfach gezählt, das heißt:

$$\texttt{MME}^t_3 := |G^t_{\texttt{MME}}| + |R^t_{\texttt{MME}} \setminus \{r \in R^t_{\texttt{MME}} | \overline{g}^t(r) \in G^t_{\texttt{MME}}\}| \tag{6.22}$$

.

Summiert über alle Einzelbilder ist der `MME` (*Mismatch Error*) dann

$$\texttt{MME} := \sum_{t=t_S}^{t_E} \texttt{MME}^t_1 \tag{6.23}$$

.

Ben Shitrit u. a. (Ben Shitrit u. a. 2011) stellten zurecht heraus, dass der `MME` für Anwendungen nicht geeignet ist, bei denen die richtige Identifizierung der Objekte wichtig ist (wie bei Sportspielen), da Verwechslungen der Identität nur einmalig gezählt werden. Daher schlagen sie den `GMME` (*Global Mismatch Error*) vor. Jedem Referenzobjekt wird dabei die Identität des ersten zugeordneten Ergebnisobjekts eindeutig zugeordnet. Für jeden diskreten Zeitpunkt, an dem die zugeordnete Identität eine andere ist, wird der `GMME` hochgezählt. Ändert sich beispielsweise die zugeordnete Identität zur Hälfte einer Trajektorie, wird beim `MME` nur ein Fehler gezählt, während beim `GMME` die halbe Trajektorie gezählt wird. Ein Problem von `GMME` ist, dass er unter Umständen eine als gut empfundene Verfolgung sehr schlecht bewertet. Gibt es beispielsweise bei einer Trajektorie gleich zum zweiten diskreten Zeitpunkt einen Wechsel der zugeordneten Identität

und danach keinen Wechsel mehr, so wird nahezu die gesamte Trajektorie als Fehler gewertet. Zudem kann es vorkommen, dass zwei verschiedenen Referenztrajektorien die selbe initiale Ergebnisidentität zugeordnet wird.

Um die Idee des GMME weiterzuführen, aber die Nachteile abzumildern, wird in dieser Arbeit mit dem $\texttt{GMME}_{\texttt{OPT}}$ (*Global Mismatch Error with Optimal Mapping*) ein neues Maß eingeführt. Dieses wird mit Bezug auf die komplette Videosequenz bestimmt und wie folgt festgelegt:

1. Bestimme für jedes Paar (g,r) eines Referenzobjekts g und eines Ergebnisobjekts r, die Anzahl der diskreten Zeitpunkte $N_{g,r}$, an denen g und r einander zugeordnet sind.
2. Bestimme über ein gewichtetes, bipartites Zuordnungsproblem (Kuhn 1955; Munkres 1957) eine eineindeutige Zuordnung von Referenzobjekten und Ergebnisobjekten, so dass die gesamte Anzahl aller Fehlzuordnungen minimiert wird.
3. Der $\texttt{GMME}_{\texttt{OPT}}$ bestimmt sich dann aus der gesamten Anzahl an Fehlzuordnungen, die sich aus dieser Zuordnung ergeben.

6.3.6 Verwechslungen der Mannschaftszuordnung

Bei Sportspielen ist es nicht nur wichtig, die Identität der einzelnen Spieler richtig zuzuordnen. Es ist auch wichtig, den einzelnen Spielern die richtige Mannschaft zuzuordnen. Für manche Fragestellung ist dies sogar völlig ausreichend, beispielsweise bei der Bestimmung der Ballbesitzphasen einer Mannschaft. Analog zu den Maßen bezüglich der Identitätszuordnung in Abschnitt 6.3.5 werden in dieser Arbeit neue Maße vorgestellt, die die Richtigkeit der Mannschaftszuordnung bewerten sollen:

- LMME (*Label Mismatch Error*): Ein LMME wird gezählt, wenn sich für ein Referenzobjekt das Label des zugeordneten Ergebnisobjekts von dem Label des zuletzt zugeordneten Ergebnisobjekts unterscheidet.
- GLMME (*Global Label Mismatch Error*): Jedem Referenzobjekt wird das Label des ersten ihm zugeordneten Ereignisobjekts zugeordnet. Unterscheidet sich im Folgenden das Label eines zugeordneten Ereignisobjekts von dem initial zugeordneten Label, so wird für jeden diskreten Zeitpunkt ein GLMME gezählt.
- $\texttt{GLMME}_{\texttt{OPT}}$ (*Global Label Mismatch Error with Optimal Mapping*): Über ein gewichtetes, bipartites Zuordnungsproblem wird eine global optimale Zuordnung von Referenzlabels zu Ereignislabels ermittelt, so dass die Anzahl der Fehlzuordnungen minimiert wird. Eben diese Anzahl an Fehlzuordnungen ist dann der $\texttt{GLMME}_{\texttt{OPT}}$.

6.3.7 Multiple Object Tracking Accuracy (MOTA)

Die *Multiple Object Tracking Accuracy* (MOTA) (Bernardin und Stiefelhagen 2008) ist definiert als

$$\texttt{MOTA} := 1 - \frac{\sum_{t=t_S}^{t_E}(w_{\texttt{FN}}(\texttt{FN}^t) + w_{\texttt{FP}}(\texttt{FP}^t) + w_{\texttt{MME}}(\texttt{MME}^t))}{\texttt{GT}} \tag{6.24}$$

.

Neben den falsch-positiven und falsch-negativen Erkennungen, gehen auch die Verwechslungen von Identitäten in die MOTA mit ein. Es gilt $\texttt{MOTA} \in [-\infty; 1]$. Negative Werte sind möglich, wenn beispielsweise eine große Anzahl an falsch-positiven Erkennungen vorliegt. Als Gewichtungsfunktionen $w_{\texttt{FN}}$, $w_{\texttt{FP}}$ und $w_{\texttt{MME}}$ werden in dieser Arbeit standardmäßig die identischen Abbildungen gewählt. In manchen Fällen wird auch $w_{\texttt{MME}} := \log_{10}$ gewählt (siehe (Kasturi u. a. 2009; Ben Shitrit u. a. 2011)).

6.3.8 Multiple Object Tracking Precision (MOTP)

Die Definition der *Multiple Object Tracking Precision* (MOTP) (Bernardin und Stiefelhagen 2008) ist abhängig vom Szenario der jeweiligen Anwendung.

Im 2D-Szenario unter Anwendung der Jaccard-Distanz (siehe Gleichung 6.2) ist sie wie folgt definiert:

$$\texttt{MOTP}_{2D} := \frac{\sum_{t=t_S}^{t_E}\sum_{g\in\overline{G}^t} 1 - d_J(g, \overline{r}^t(g))}{\sum_{t=t_S}^{t_E}|\overline{G}^t|} \tag{6.25}$$

Es gilt $\texttt{MOTP}_{2D} \in [0; 1]$, wobei 0 für ein sehr schlechtes und 1 für ein sehr gutes Ergebnis steht.

Um ein ähnliches Verhalten im 3D-Fall zu erzielen, schlagen (Leal-Taixé u. a. 2015) folgende Definition vor:

$$\texttt{MOTP}_{3D_1} := 1 - \frac{\sum_{t=t_S}^{t_E}\sum_{g\in\overline{G}^t} d(g, \overline{r}^t(g))}{\tau_d \cdot \sum_{t=t_S}^{t_E}|\overline{G}^t|} \tag{6.26}$$

Da für $g \in \overline{G}^t$ immer gilt $d(g, \overline{r}^t(g)) \leq \tau_d$, gilt durch die Normierung mit dem Schwellwert τ_d ebenfalls $MOTP_{3D_1} \in [0; 1]$ und durch die Invertierung steht 0 ebenfalls für ein schlechtes und 1 für ein gutes Ergebnis. Das hilft einen Vergleich verschiedener Verfahren übersichtlicher zu gestalten.

In manchen Anwendungen, wie bei der Verfolgung von Fußballspielern, ist die Angabe der Präzision in tatsächlichen Einheiten im Weltkoordinatensystem (Meter) in der Regel aussagekräftiger. Deswegen wird in dieser Arbeit die folgende Definition genutzt:

$$\texttt{MOTP}_{3D_2} := \frac{\sum_{t=t_S}^{t_E} \sum_{g \in \overline{G}^t} d(g, \overline{r}^t(g))}{\sum_{t=t_S}^{t_E} |\overline{G}^t|} \tag{6.27}$$

mit $\texttt{MOTP}_{3D_2} \in [0; \infty]$.

So erhält man eine direkte Aussage über die mittlere Abweichung von erkannten Referenzobjekten zu den zugeordneten Ergebnisobjekten in Metern.

6.3.9 Trajektorienbasierte Maße

Sei $N_g := |\{t | g \in G^t\}|$ die Anzahl der diskreten Zeitpunkte (Einzelbilder), in denen das Referenzobjekt g vorkommt. Das entspricht der Dauer der Trajektorie von g ohne Unterbrechungen. Diese Trajektorie kann unterbrochen sein, wenn das Objekt beispielsweise den Sichtbereich der Kamera verlässt und zu einem späteren Zeitpunkt wieder betritt. Ein gutes Verfahren zur Objektverfolgung sollte einen möglichst großen Anteil einer Trajektorie richtig verfolgen. Das heißt $T_g := |\{t | g \in \overline{G}^t\}|$, die Anzahl der diskreten Zeitpunkte in denen g richtig verfolgt wurde, sollte möglichst groß sein.

Aus diesem Grund führten Wu und Nevatia (Wu und Nevatia 2006) die trajektorienbasierten Metriken `MT` (*Mostly Tracked*) und `ML` (*Mostly Lost*) ein:

$$\texttt{MT} := \frac{|\{g \in G^{t_S \to t_E} | \frac{T_g}{N_g} > 0{,}8\}|}{|G^{t_S \to t_E}|} \tag{6.28}$$

und

$$\texttt{ML} := \frac{|\{g \in G^{t_S \to t_E} | \frac{T_g}{N_g} < 0{,}2\}|}{|G^{t_S \to t_E}|} \tag{6.29}$$

Das heißt, `MT` ist die relative Anzahl der Trajektorien, die zu mehr als 80% richtig verfolgt wurden. Und `ML` ist die relative Anzahl der Trajektorien, die zu weniger als 20% richtig verfolgt wurden.

Wu und Nevatia (Wu und Nevatia 2006) führten auch den Begriff des Fragments ein. Demnach ist ein Fragment eine Ergebnistrajektorie, die weniger als 80% einer Referenztrajektorie entspricht. Dabei wird nicht klar, was passiert, wenn eine Ergebnistrajektorie zwei verschiedene Referenztrajektorien verfolgt (beispielsweise nach einer Identitätsverwechslung). Wu und Nevatia (Li, Huang u. a. 2009) griffen daher diese Idee auf und änderten die Definition etwas ab. Nach (Li, Huang u. a. 2009) ist die Anzahl der Fragmente

definiert als die Anzahl der Unterbrechungen einer Referenztrajektorie. Auch diese Definition ist nicht eindeutig, da nicht hervorgeht, wann genau eine Unterbrechung vorliegt. Erst Leal-Taixé u. a. (Leal-Taixé u. a. 2015) sorgten für Klarheit mit einer Definition, auf welche auch in dieser Arbeit zurückgegriffen wird: Ein Fragment liegt vor, wenn die Verfolgung einer Referenztrajektorie unterbrochen wird und zu einem späteren Zeitpunkt wieder aufgenommen wird. Die Anzahl der Fragmente wird mit `FRAG` abgekürzt.

6.4 Implementierungsdetails

6.4.1 Implementierung

Die vorgestellten Verfahren für die Erkennung und Verfolgung der Spieler, sowie für die Evaluierung wurden ausschließlich mit C++ (Stroustrup 2013) implementiert. Dabei wurde in erster Linie Wert auf Modularität und Design gelegt und weniger auf eine effiziente Codeausführung. Daher bietet die entstandene Software viel Raum zur Laufzeitoptimierung insbesondere durch Ausnutzung spezieller Rechenprozessoren (wie GPUs) und Parallelisierung. Neben den gängigen C++-Bibliotheken, wie die C++-Standardbiblithek und Boost (Dawes u. a. 2012), kam für die Basisoperationen der Bildverarbeitung (wie Glättungsfilter und morphologische Operationen) die kommerzielle Software Halcon (MVTec Software GmbH 2012) zum Einsatz. Für Matrixberechnung wurde Eigen (Jacob und Guennebaud 2013) genutzt sowie für das Laden und Schreiben von Videosequenzen libav (Biurrun und Barbato 2015).

6.4.2 Auswahl der Bilder für die Outfitbestimmung

Die Erstellung der Farbtemplates ist prinzipiell so ausgelegt, dass in mehreren Bildern der Sequenz die Spieler erkannt werden und die dominanten Farben und Teamoutfits mit dieser Information bestimmt werden. Dies hilft dabei, gegebenenfalls auch Spieler und Torhüter mit einzubeziehen, die im ersten Bild nicht sichtbar sind. Bei kürzeren Sequenzen hat dies nichts mehr mit einem Online-Verfahren zu tun, da für die Initialisierung des Verfahrens bereits zukünftige Daten genutzt werden.

Aus diesem Grund und aus Gründen der Vergleichbarkeit mit den anderen Verfahren, wird für die Ergebnisse der 2D-Spielerverfolgung die Initialisierung nur mit dem ersten Bild einer Sequenz durchgeführt. Für die Ergebnisse der 3D-Spielerverfolgung wurden jeweils aus den ersten fünf Sekunden einer Sequenz die drei am besten geeigneten Bilder automatisch ausgewählt. Für die Bestimmung der Eignung eines Bilds wurden dabei

	Start in Spielzeit	Länge (s)	Frames	FPS	Auflösung	Kamera
ML	9:45	14	713	50	1280 × 720	bewegt (TV)
ND	68:42	37	925	25	1024 × 576	bewegt (TV)
BFG	45:35	44	1101	25	1024 × 576	bewegt (TV)
PS	47:16	60	1500	25	1024 × 576	bewegt (TV)
VS	-	100	2500	25	720 × 576	statisch
ISSIA	-	120	3000	25	1920 × 1080	statisch

Tabelle 6.1: Übersicht Videosequenzen

	Datum	Begegnung	Wettbewerb
ML	07.11.2012 20:45	München - Lille	UEFA Champions League
ND	14.11.2012 20:30	Niederlande - Deutschland	Länderspiel
BFG	06.02.2013 20:30	Burkina Faso - Ghana	Afrikameisterschaft
PS	28.06.2013 14:30	Polen - Schweden	Frauen U-17 EM
VS	-	-	-
ISSIA	-	-	-

Tabelle 6.2: Übersicht Begegnungen

Faktoren wie die geschätzte Anzahl der sichtbaren Spieler im Bild, die Schärfe des Bilds und Unterschiede in den geschätzten Kamerawinkeln innerhalb der drei Bilder bewertet.

6.4.3 Erkennung von dauerhafte Einblendungen

Dauerhafte Einblendungen (wie Senderlogos oder Spielstandsanzeigen) können das Ergebnis des Verfahrens negativ beeinflussen. Solche Störfaktoren können meist gut in einem Vorverarbeitungsschritt automatisch erkannt und entfernt werden, wenn man den zeitlichen Kontext im entsprechenden Umfang berücksichtigt.

Für die Auswertung der 2D-Spielerverfolgung wird aufgrund der Vergleichbarkeit mit den anderen Verfahren auf diesen Schritt verzichtet. Für die 3D-Spielerverfolgung wird eine Erkennung von dauerhaften Einblendungen anhand von 15 gleichmäßig verteilten Bildern aus jeder Sequenz ermittelt und während der Prozessierung von den erkannten Spielersilhouetten subtrahiert. Das Verfahren ermittelt durch paarweise Bildvergleiche Regionen, die sich von Bild zu Bild wenig ändern. Dabei wird die Grasmaske (siehe Abschnitt 3.3.2) berücksichtigt, da Pixel mit Grasfarbe mit sehr hoher Wahrscheinlichkeit nicht zu einer dauerhaften Einblendung gehören.

6.5 Evaluierung der 2D-Spielerverfolgung

6.5.1 Videosequenzen

Zur Evaluierung der 2D-Spielerverfolgung wurden Videosequenzen aus Aufzeichnungen von Fußballspielen mit möglichst unterschiedlichen Rahmenbedingungen verwendet. Dabei wurde sowohl auf Aufnahmen von bewegten Fernsehkameras als auch auf statische Amateuraufnahmen zurückgegriffen. Alle Sequenzen umfassen jeweils eine durchgängige Szene ohne Werbeunterbrechungen, Großaufnahmen, Zeitlupen oder sonstige Schnitte. Die Referenzdaten für die Videos wurden mithilfe der Annotiersoftware VATIC (Vondrick u. a. 2012) erzeugt. Für die Sequenzen **VS** und **ISSIA** waren bereits Annotationen online verfügbar und es wurden nur die fehlenden Daten mit VATIC ergänzt. Die Annotationen wurde gemäß den Vorgaben aus 6.2 durchgeführt. Eine Übersicht der Videosequenzen und der zugehörigen Spielbegegnungen sind aus den Tabellen 6.1 und 6.2 zu entnehmen. Insgesamt wurde eine Laufzeit von 375 Sekunden mit 9739 Einzelbildern annotiert. Die Videosequenzen weisen folgende Eigenschaften auf:

- **ML** (ZDF (ZDF 2012)):
  Eine kurze, qualitativ hochwertige HD-Aufnahme einer UEFA Champions League Partie in der Münchner Allianz Arena mit schwenkender Kamera und einem eher hohen Zoomfaktor. Die Rasenbedingungen sind gut.

- **ND** (Das Erste (Das Erste 2012):
  Eine qualitativ hochwertige SD-Aufnahme einer DFB-Länderspielbegegnung in der Amsterdam Arena mit schwenkender Kamera und einem mittleren Zoomfaktor. Die Rasenbedingungen sind gut.

- **BFG** (Eurosport (Eurosport 2013a)):
  Eine SD-Aufnahme mittlerer Qualität eines Halbfinales der Fußball-Afrikameisterschaft 2013 im Stadion von Mbombela, Südafrika. Die schwenkende Kamera weist einen eher geringen Zoomfaktor auf. Die Rasenbedingungen sind sehr schlecht, da der Rasen matschig und teilweise leicht mit Schnee bedeckt ist.

- **PS** (Eurosport (Eurosport 2013b)):
  Eine SD-Aufnahme guter Qualität des Finales der U17-Fußball-Europameisterschaft der Frauen 2013 im Stadion von Nyon, Schweiz mit schwenkender Kamera und eher geringem Zoomfaktor. Die Rasenbedingungen sind gut, allerdings weist die Sequenz Amateurcharakter auf.

- **VS** (University of Reading (University of Reading 2002) Kamera 3 Test):
  Eine Aufnahme schlechterer Qualität einer unbekannten Partie, die für den ersten

Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance im Rahmen der *IEEE International Conference on Computer Vision 2003* in Nizza, Frankreich zur Verfügung gestellt wurde und online erhältlich ist. Die Kamera ist statisch mit einem mittleren Zoomfaktor. Die Position der Kamera befindet sich hinter der Torauslinie. Die Rasenbedingungen sind gut. Die Sequenz weist allerdings Amateurcharakter auf.

- **ISSIA** (D'Orazio u. a. (D'Orazio u. a. 2009a) Kamera 3):
  Eine Full-HD-Aufnahme guter Qualität einer unbekannten Partie. Die Daten wurden von den Autoren im Rahmen ihrer Veröffentlichung online zugänglich gemacht (D'Orazio u. a. 2009b). Die Kamera ist statisch mit einem mittleren Zoomfaktor und befindet sich auf Höhe der Mittellinie hinter der Seitenlinie. Die Rasenbedingungen sind gut und die Sequenz weist Amateurcharakter auf.

Ein Eindruck zu den Videosequenzen kann durch die Ausschnitte in Abbildung 6.2 gewonnen werden.

6.5.2 Vergleichsverfahren

Eine reine Auflistung der verschiedenen Bewertungsmetriken besitzt meist wenig Aussagekraft und trägt nicht maßgeblich dazu bei, die Leistung eines Verfahren einordnen zu können. Aus diesem Grund ist es gängige Praxis, die Ergebnisse eines Verfahrens mit den Ergebnissen anderer kontemporärer Verfahren zu vergleichen, die in ähnlichen Szenarien den Stand der Forschung repräsentieren. Dafür gibt es im Prinzip zwei Möglichkeiten:

1. Es gibt einen einheitlichen, öffentlich verfügbaren Datensatz (im besten Falle inklusive Evaluierungsskript) für den die Ergebnisse anderer Verfahren veröffentlicht wurden (wie beispielsweise beim *Multiple Object Tracking Benchmark* (Milan u. a. 2016)).

2. Die Vergleichsverfahren sind öffentlich verfügbar (quelloffen oder zumindest als ausführbare Anwendung) und können somit direkt mit neu eingeführten Datensätzen und Evaluierungsskripte ausgewertet werden.

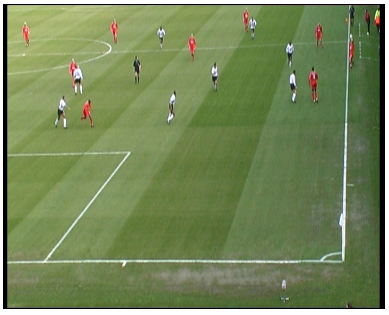
Da es im Bereich Fußball und Spielerverfolgung keine koordinierte Forschungsgemeinschaft gibt, ist die Möglichkeit aus Punkt 1 schwierig umzusetzen. Mit den Datensätzen **VS** und **ISSIA** gibt es zwar öffentlich verfügbare Datensätze. Letztendlich sind diese Daten auf statische Kameras beschränkt und es gibt kaum Studien, die diese Datensätze für eine umfangreiche und einheitliche Evaluierung benutzen. Aus diesem Grund

(a) ML (ZDF 2012)

(b) ND (Das Erste 2012)

(c) BFG (Eurosport 2013a)

(d) PS (Eurosport 2013b)



(e) VS (University of Reading 2002)

(f) ISSIA (D'Orazio u. a. 2009a)

ABBILDUNG 6.2: Ausschnitte aus den Videosequenzen zur Evaluierung der 2D-Spielerverfolgung

wird in dieser Arbeit auf die Möglichkeit aus Punkt 2 zurückgegriffen. Da auch hier im Bereich Fußball wenig öffentliches Material zur Verfügung steht, wird der Vergleich mit Verfahren durchgeführt, die allgemein auf die Erkennung und Verfolgung von Personen (insbesondere von Fußgängern) spezialisiert sind. Die verwendeten Verfahren basieren alle auf dem Prinzip *Tracking-by-Detection* und somit sind die Ergebnisse und vor allem die Ausführungszeit natürlich stark abhängig von der Personenerkennung.

Hinzu kommt, dass Verfahren zur Personenerkennung standardmäßig eine minimale Personengröße im Bild von circa 100 Pixel vorgeben. In Aufnahmen von Fußballspielen kann die Größe eines Spielers deutlich geringer ausfallen. In diesem Fall sollte eher von einer minimalen Größe von circa 20 Pixel ausgegangen werden (beispielsweise bei Spielern die am hinteren Spielfeldrand zu sehen sind). Dies kann gegebenenfalls die Ausführungszeit und die Anzahl an falsch-positiven Erkennungen einer Personenerkennung drastisch

erhöhen.

Zur Einordnung der Ergebnisse der Spielerverfolgung wurden folgende öffentlich verfügbare Verfahren genutzt:

- **GOGA** (Pirsiavash u. a. 2011), *Offline-Verfahren*: Dieses Verfolgungsverfahren nutzt die Personenerkennung von P. F. Felzenszwalb u. a. (P. F. Felzenszwalb u. a. 2010; Girshick u. a. 2009). Auf Basis der Erkennung wird ein Graph konstruiert und das Assoziationsproblem mit geeigneten Kostenfunktionen der Kanten als *Minimum-Cost Flow* Problem formuliert. Dieses Problem wird mit Hilfe von dynamischer Programmierung und *Non-Maximum Suppression* (DP + NMS) gelöst. Die Standardparameter blieben unverändert, nur die minimale Objektgröße wurde auf 24 Pixel gesetzt.

- **TH** (J. Zhang u. a. 2012), *Online-Verfahren*: Dieses Verfahren basiert auf der Personenerkennung der Softwarebibliothek *OpenCV 2.3.1* (itseez 2011). Auf Basis der Erkennungen werden histogrammbasierte Farbtemplates erstellt. Die Verfolgung basiert auf einer Kombination von *Mean Shift* und einem Kalman-Filter. Die Standardparameter blieben unverändert, nur die minimale Objektgröße wurde ebenfalls auf 24 Pixel gesetzt.

- **LP2D** (Leal-Taixé u. a. 2014), *Offline-Verfahren*: Zur Erkennung der Spieler wurde für dieses Verfahren die Personenerkennung von Dollár (Dollár 2014; Dollár u. a. 2014; Nam u. a. 2014) eingesetzt, die derzeit eine der leistungsfähigsten und schnellsten Methoden zur Fußgängererkennung darstellt. Die Erkennungen werden durch das Verfahren zu Trajektorien verbunden, das die Messbasis (*Baseline*) der *MOTChallenge* (Milan, Leal-Taixé, Schindler u. a. 2015b; Leal-Taixé u. a. 2015) darstellt. Ähnlich wie bei **GOGA** wird mit Hilfe der Erkennungen ein Graph konstruiert und ein *Minimum-Cost Flow* Problem gelöst. Dabei kommt der Teil der linearen Programmierung, wie in (Leal-Taixé u. a. 2014) vorgestellt, zum Einsatz. Die Standardparameter blieben unverändert, nur die minimale Objektgröße wurde auf 20 Pixel gesetzt. Der Klassifikator der Personenerkennung wurde auf dem INRIA Datensatz (Dalal und Triggs 2005) trainiert.

6.5.3 Modalitäten der 2D-Evaluierung

Bei der Evaluierung der 2D-Spielerverfolgung werden folgende Besonderheiten angewendet, unter anderem um auf den Fußball-spezifischen Kontext Rücksicht zu nehmen (ähnlich wie es in (Milan u. a. 2016) beispielsweise mit sitzenden Personen oder Personen auf dem Fahrrad gehandhabt wird):

- Das Ziel der Spielerverfolgung ist in erster Linie, die Positionen von Spieler und Schiedsrichter zu bestimmen. Aus diesem Grund werden Personen am Spielfeldrand außerhalb des Spielfeldes (Linienrichter, Trainer, Zuschauer, etc.) als *optionale Objekte* definiert. Das heißt, es spielt keine Rolle, ob das Objekt verfolgt wird oder nicht. Es werden diesbezüglich keine richtig-positiven und keine falsch-negativen Erkennungen sowie keine Identifizierungsfehler gezählt. Wird das Objekt allerdings mehrfach verfolgt, so wird das entsprechend als falsch-positive Erkennung gezählt.
- Referenzobjekte, die sich am Bildrand befinden (das heißt weniger als vier Pixel Distanz), werden ebenfalls als *optionale Objekte* definiert.
- Die Vergleichsverfahren **GOGA**, **TH** und **LP2D** basieren jeweils auf den Ergebnissen eines Personendetektors. Da auf der Tribüne in der Regel Personen zu sehen sind, schlagen diese Detektoren in diesen Bereichen häufig an. Um diese Verfahren nicht zu benachteiligen, werden Erkennungen außerhalb der Feldhülle herausgefiltert.

6.5.4 Ergebnisse der 2D-Evaluierung

Die Ergebnisse der einzelnen Verfahren mit Bezug auf alle Videosequenzen für verschiedene Distanzschwellwerte τ_d können in den Tabellen 6.3, 6.5 und 6.7 entnommen werden. Die `MOTA`-Ergebnisse der Verfahren für einzelne Videosequenzen und verschiedene Schwellwerte sind in Abbildung 6.4 dargestellt. Das in dieser Arbeit vorgestellte Verfahren wird jeweils unter der Bezeichnung **IUKS** aufgeführt.

Es ist zu erkennen, dass in allen Bereichen das vorgestellte Verfahren signifikant besser abschneidet als die Vergleichsverfahren. Laut den Tabellen 6.4, 6.6 und 6.8 schneidet zwar **GOGA** bei `GMME` und $\texttt{GMME}_{\texttt{OPT}}$ auffällig besser ab. Dem liegt jedoch die Tatsache zu Grunde, dass die Erkennungsleistung des Verfahrens deutlich schlechter ist, wie man an den falsch-negativen (`FN`) ablesen kann. Wenn weniger Spieler richtig erkannt und verfolgt werden, kommen dementsprechend auch weniger Identitätswechsel in Betracht.

Wie in Abbildung 6.4 dargestellt, schneidet das vorgestellte Verfahren bei MOTA für Distanzschwellwerte $\tau_d \leq 0{,}6$ in den Sequenzen **ML** und **VS** etwas schlechter ab als die Vergleichsverfahren. Das liegt daran, dass die Erkennung der Spieler tendenziell etwas zu klein ausfällt, das heißt, die Spieler werden zwar richtig erkannt und verfolgt, aber die Bestimmung der Größe ist nicht ganz korrekt, wie in Abbildung 6.3 exemplarisch dargestellt ist. Bei höheren Distanzschwellwerten fällt diese Ungenauigkeit nicht ins Gewicht, was man an den signifikant besseren MOTA-Ergebnisse für $\tau_d > 0{,}6$ erkennen kann.

	F_1	MOTA	MOTP	FAF	MT	ML	FP	FN	FRAG
IUKS	0,73	0,451	0,35	4	0,364	0,168	38806	35724	3882
TH	0,561	-0,00748	0,379	8,88	0,168	0,162	86235	49641	7091
GOGA	0,309	0,164	0,259	0,2	0,0116	0,751	1940	111063	1208
LP2D	0,58	0,173	0,275	5,02	0,139	0,22	48724	60691	4782

Tabelle 6.3: Ergebnisse (MOTA und andere) für Schwellwert $\tau_d := 0{,}5$

	MME	GMME	GMME_{OPT}	LMME	GLMME	$\text{GLMME}_{\text{OPT}}$
IUKS	353	49270	26753	270	6697	6396
TH	1520	65444	36292	-	-	-
GOGA	1011	22837	13966	-	-	-
LP2D	3314	68845	38239	-	-	-

Tabelle 6.4: Ergebnisse (MME und andere) für Schwellwert $\tau_d := 0{,}5$

	F_1	MOTA	MOTP	FAF	MT	ML	FP	FN	FRAG
IUKS	0,899	0,795	0,39	1,49	0,694	0,116	14474	13220	683
TH	0,739	0,4	0,426	5,99	0,491	0,104	58160	22404	2104
GOGA	0,325	0,183	0,275	0,0578	0,0231	0,746	561	109759	907
LP2D	0,758	0,51	0,37	2,57	0,387	0,173	24954	37931	2319

Tabelle 6.5: Ergebnisse (MOTA und andere) für Schwellwert $\tau_d := 0{,}7$

	MME	GMME	GMME_{OPT}	LMME	GLMME	$\text{GLMME}_{\text{OPT}}$
IUKS	259	57022	30650	177	8818	8034
TH	1246	86043	48065	-	-	-
GOGA	1035	24063	14860	-	-	-
LP2D	3967	91011	55846	-	-	-

Tabelle 6.6: Ergebnisse (MME und andere) für Schwellwert $\tau_d := 0{,}7$

	F_1	MOTA	MOTP	FAF	MT	ML	FP	FN	FRAG
IUKS	0,917	0,831	0,398	1,23	0,728	0,116	11908	10927	326
TH	0,763	0,457	0,443	5,57	0,561	0,0925	54119	18850	1307
GOGA	0,328	0,187	0,279	0,0353	0,0289	0,746	343	109542	887
LP2D	0,758	0,51	0,37	2,57	0,387	0,173	24954	37931	2319

Tabelle 6.7: Ergebnisse (MOTA und andere) für Schwellwert $\tau_d := 0{,}9$

	MME	GMME	GMME_{OPT}	LMME	GLMME	$\text{GLMME}_{\text{OPT}}$
IUKS	231	57466	31191	135	8425	7905
TH	1079	88871	50414	-	-	-
GOGA	1042	24263	15023	-	-	-
LP2D	3967	91011	55846	-	-	-

Tabelle 6.8: Ergebnisse (MME und andere) für Schwellwert $\tau_d := 0{,}9$

Abbildung 6.3: Zu klein erkannte Spieler in **ML** (links) und **VS** (rechts).

	Erkennung (s)	Verfolgung (s)	Gesamt (s)	Gesamt (fps)
IUKS	-	-	6703	1,45
GOGA	196475	248	196723	0,05
TH	-	-	338321	0,03
LP2D	16678	9257	25935	0,38

Tabelle 6.9: Laufzeiten und Frameraten der Verfahren in Bezug auf alle sechs Videosequenzen gemeinsam. Alle Experimente wurden auf einem Intel®Core™ i7-2600 CPU mit acht Kernen à 3,40 GHZ und 8 GB Hauptspeicher durchgeführt.

Im Vergleich der Laufzeit bei Verfahren die nach dem Prinzip *Tracking-by-Detection* vorgehen, wird häufig nur die Laufzeit des eigentlichen Trackings angegeben, ohne die Laufzeit der Personenerkennung zu berücksichtigen. Solche Angaben sind mit Vorsicht zu genießen, da die Personenerkennung systematisch das ganze Bild auf verschiedenen Skalen-Niveaus abscannt. Dies kann zu einem erheblichen Anteil an der Laufzeit des Gesamtsystems führen, insbesondere bei großen Bildgrößen und kleinen minimalen Objektgrößen. Die Gesamtlaufzeit wurde vom Beginn der Ausführung bis zum Ende der Ausführung gemessen. Das heißt, dass nicht nur die reine Berechnungszeit, sondern auch Textausgaben oder Ähnliches mitgemessen wurden. Wegen der *Batch*-Bearbeitung konnten für **GOGA** und **LP2D** die Zeiten für Personenerkennung und -verfolgung jeweils getrennt bestimmt werden. Die Gesamtzeit ist somit die Summe der Einzelzeiten. Die Laufzeiten der einzelnen Verfahren sind aus Tabelle 6.9 zu entnehmen.

Es ist deutlich ersichtlich, dass das vorgestellte Verfahren eine signifikant bessere Performanz aufweist. **GOGA** ist zwar bei der Berechnung der Trajektorien sehr effizient, allerdings liegt der Flaschenhals auf der Seite der Erkennung und sorgt für eine unbefriedigende Gesamtlaufzeit.

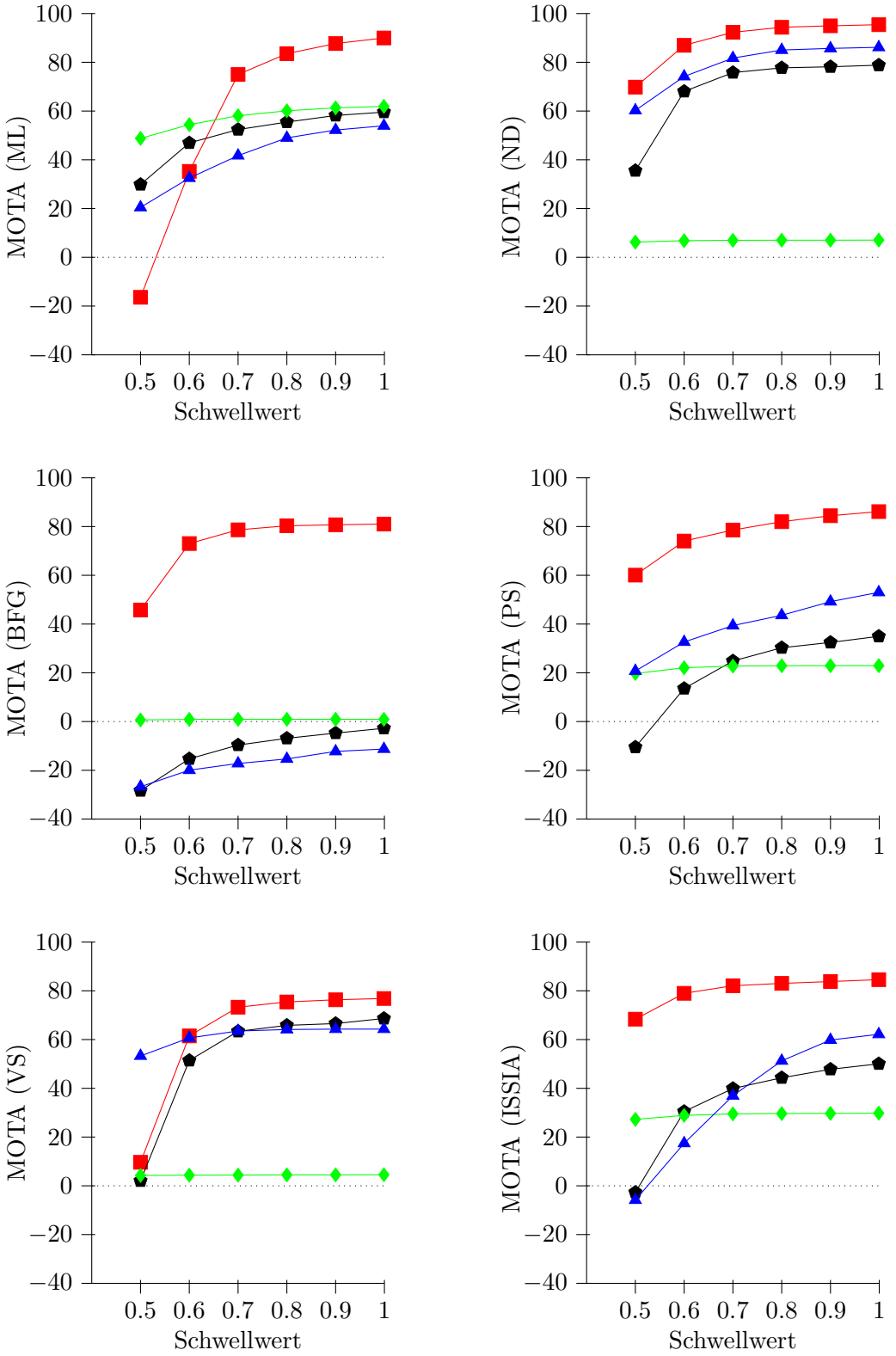


Abbildung 6.4: Ergebnisse für die Videosequenzen **ML**, **ND**, **BFG**, **PS**, **VS** und **ISSIA** (von links nach rechts und von oben nach unten) für die Verfahren IUKS (—■—), GOGA (—♦—), LP2D (—▲—) und TH (—⬟—)

	Datum	Begegnung	Halbzeit
3D-A	01.03.2014 18:30	FC Bayern München - FC Schalke 04	2
3D-B	15.03.2014 18:30	FC Bayern München - Bayer 04 Leverkusen	1
3D-C	10.05.2014 15:30	Hertha BSC - Borussia Dortmund	1
3D-D	08.11.2014 15:30	FC Augsburg - SC Paderborn 07	1

Tabelle 6.10: Übersicht Begegnungen für die 3D-Evaluierung

	Länge (s)	Frames	FPS	Auflösung	Kamera	Quelle
3D-A	2704	135200	50	1280 × 720	bewegt	(DFL 2014c)
3D-B	2693	134650	50	1280 × 720	bewegt	(DFL 2014b)
3D-C	2716	135800	50	1280 × 720	bewegt	(DFL 2014d)
3D-D	2701	135050	50	1280 × 720	bewegt	(DFL 2014a)

Tabelle 6.11: Übersicht Videosequenzen für die 3D-Evaluierung

6.6 Evaluierung der 3D-Spielerverfolgung

6.6.1 Videosequenzen

Für die Evaluierung der 3D-Spielerverfolgung kommen sogenannte *Scoutingfeeds* zum Einsatz. Das sind spezielle Aufnahmen von Spielen in Profiligen mit einer schwenkenden Kamera. Dabei achtet der Kameramann möglichst darauf, dass alle Feldspieler im Blickfeld sind und passt die Kamerawinkel und den Zoom entsprechend an. Die 3D-Referenzdaten wurden von einem professionellen System generiert (TRACAB 2015). Für jede Begegnung wurde eine Halbzeit ausgesucht. Eine Übersicht über die vier Begegnungen und den zugehörigen Videosequenzen ist den Tabellen 6.10 und 6.11 zu entnehmen. Zusammen ergibt sich eine Dauer von 10814 s (180 min und 14 s), was einer Anzahl von 540700 Einzelbildern entspricht.

6.6.2 Modalitäten der 3D-Evaluierung

Ähnlich wie bei der 2D-Evaluierung werden bei der 3D-Evaluierung folgende Besonderheiten angewendet:

- Da die Positionen der Torhüter in den Referenzdaten angegeben sind, die Torhüter sich allerdings die meiste Zeit nicht im Sichtbereich der Kamera befinden, werden die Torhüter beider Mannschaften als *optionale Objekte* (siehe Abschnitt 6.6.2) definiert.
- Die Referenzdaten für die Sequenz **3D-A** enthalten die Positionen des Schiedsrichters und der Linienrichter. Die Linienrichter werden in diesem Fall wie in Abschnitt 6.6.2 als *optionale Objekte* behandelt.

τ_d [m]	F_1	MOTA	MOTP [m]	MT	ML	FP	FN	MME	LMME
0,5	0,45	-0,07	0,72	0	0,14	5449511	6200002	5410	3587
1	0,71	0,44	0,56	0,08	0,09	2632174	3426085	5698	3951
1,5	0,81	0,63	0,47	0,41	0,08	1630823	2444606	5421	3604
2	0,85	0,71	0,40	0,66	0,08	1170664	1992164	5214	3296
2,5	0,87	0,76	0,35	0,75	0,08	914220	1740390	5019	3069
3	0,89	0,79	0,30	0,75	0,08	748949	1577663	4946	2953
3,5	0,90	0,81	0,26	0,77	0,08	637337	1467810	4865	2869

TABELLE 6.12: Ergebnisse der 3D-Evaluierung für verschiedene Schwellwerte bezüglich aller Sequenzen.

- Die Referenzdaten für die Sequenzen **3D-B**, **3D-C** und **3D-D** enthalten nicht die Positionen des Schiedsrichters und der Linienrichter. Das Verfahren erkennt diese Personen sowie andere Personen am Spielfeldrand (Trainer, etc.) richtigerweise. Solche Erkennungen werden vor der Evaluierung aus den Ergebnisdaten entfernt.

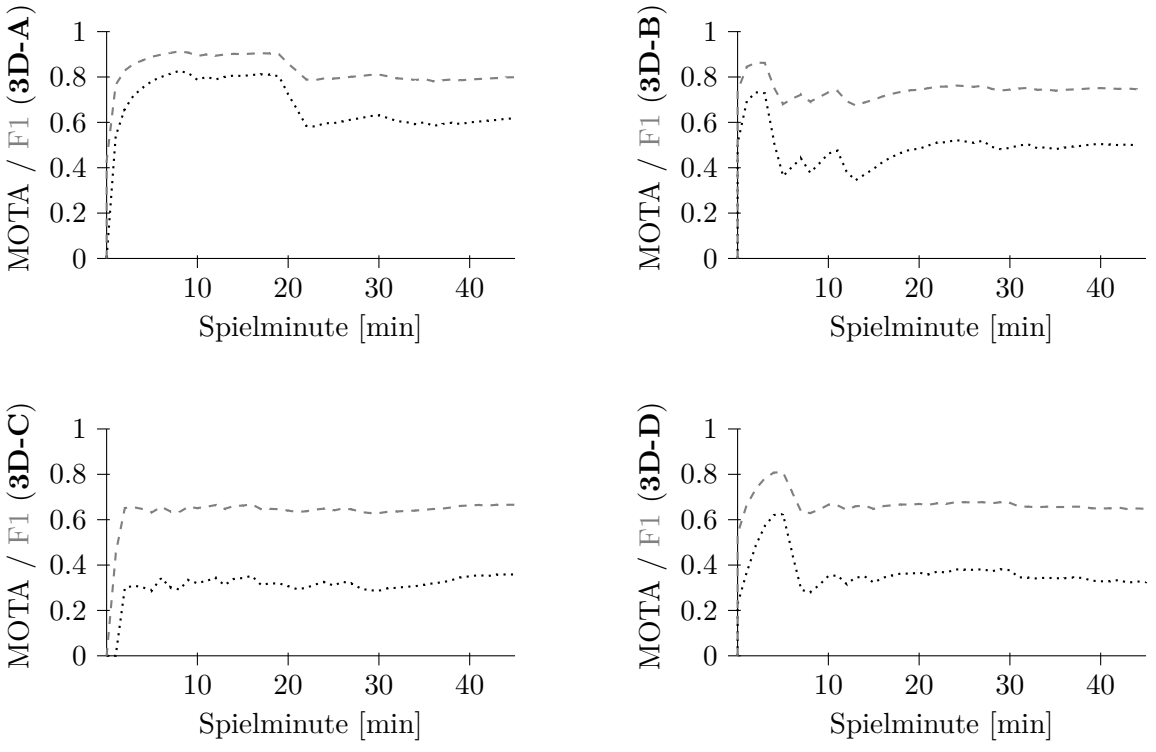


ABBILDUNG 6.5: Ergebnisse für die Videosequenzen **3D-A**, **3D-B**, **3D-C**, **3D-D** (von links nach rechts und von oben nach unten) bezüglich MOTA (·····) und F1 (- - -) als Verlauf über die Spielzeit für den Schwellwert 1,0m.

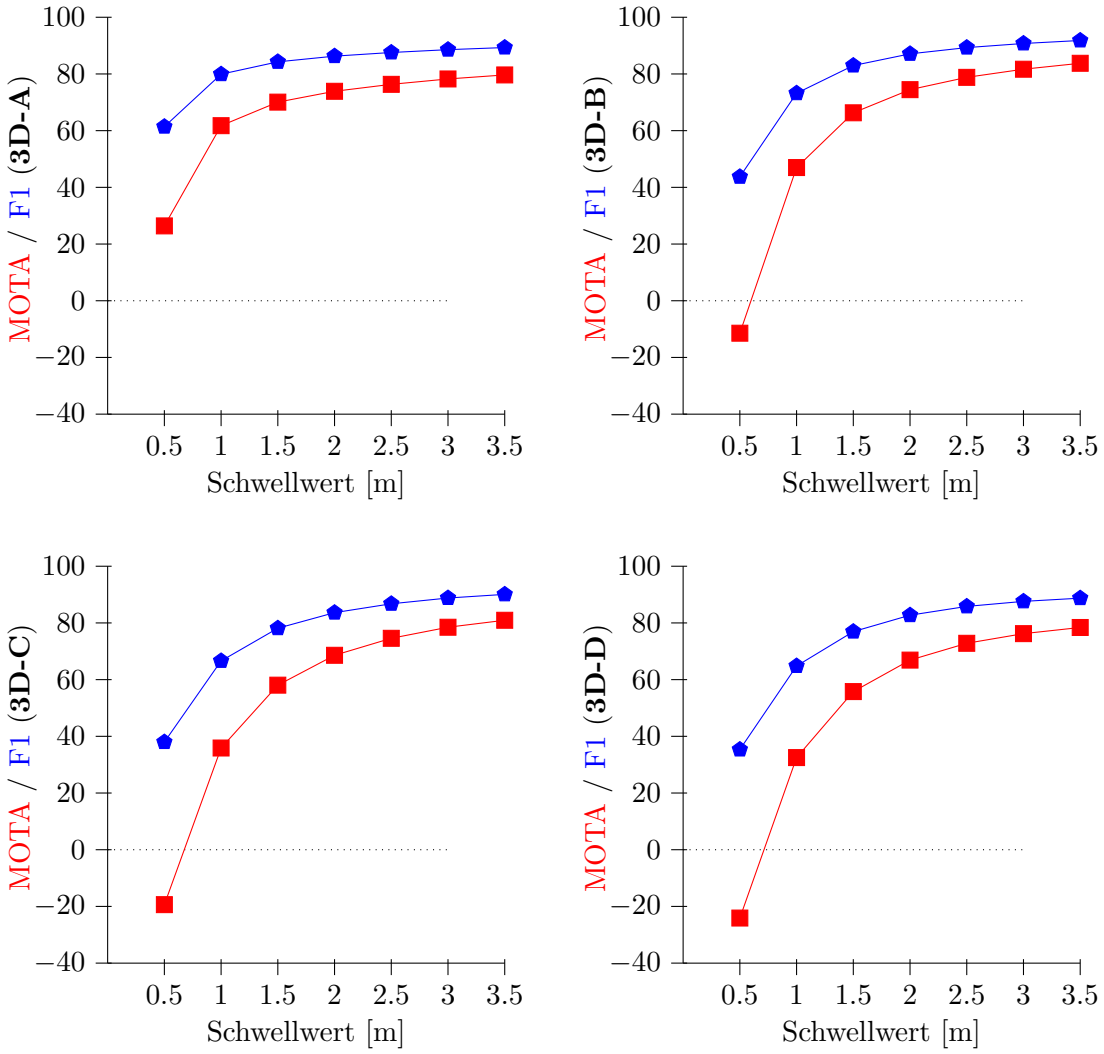


ABBILDUNG 6.6: Ergebnisse für die Videosequenzen **3D-A**, **3D-B**, **3D-C**, **3D-D** (von links nach rechts und von oben nach unten) bezüglich MOTA (—■—) und F1 (—●—) für verschiedene Schwellwerte.

6.6.3 Ergebnisse der 3D-Evaluierung

Die Ergebnisse (über alle Sequenzen) der 3D-Evaluierung für verschiedene Schwellwerte sind aus Tabelle 6.12 zu entnehmen und in Abbildung 6.6 sind die Werte für *MOTA* und *F1* für die vier Videosequenzen in Abhängigkeit des Schwellwertes dargestellt.

In Abbildung 6.5 sind die beiden Werte in Abhängigkeit von der Spielzeit dargestellt für den Schwellwert $\tau_d := 1{,}0\,m$. Hier sind bei allen Sequenzen deutliche Knicke nach unten zu erkennen, von denen sich die Ergebnisse nicht wieder erholen. Diese Knicke entstehen in Phasen, in denen die Kamerakalibrierung für eine gewisse Zeit aus dem Ruder gelaufen ist, beispielsweise bei einem Zoom nach einem Torjubel. In diesen Abschnitten weichen die Transformationen von Bildkoordinaten in Feldkoordinaten deutlich von den

Sollwerten ab. Würde man zwei Auswertungen durchführen, eine vor diesen Aussetzern und eine nach den Aussetzern, so würden diese deutlich bessere Ergebnisse vorweisen.

Insgesamt zeigt das Verfahren ab einem τ_d von 1,5 m sehr gute Ergebnisse ($\texttt{MOTA} > 0{,}6$ und $\texttt{F}_1 > 0{,}8$). Das deutet darauf hin, dass die Spieler richtig erkannt und verfolgt werden, während die Positionsbestimmung Abweichungen von bis zu 3 m aufweist. Dies ist auch an der deutlich abfallenden `MOTP` in Abhängigkeit von τ_d zu erkennen. Hier spielen nicht nur Ungenauigkeiten des vorgestellten Systems hinein. Auch fehlende Präzision der Kamerakalibrierung und der Referenzdaten werden in diesen Zahlen widergespiegelt. Um die Werte ohne Vergleichsverfahren besser einordnen zu können, können die MOTA-Werte *Multiple Object Tracking Benchmark 3D Challenge 2015* (Leal-Taixé u. a. 2015) genutzt werden. Hier weisen die besten Verfahren eine `MOTA` von 0,5 bei einem τ_d von 1 m auf. Dabei muss berücksichtigt werden, dass die dafür benutzten wenig dynamischen Fußgängerszenen mit statischen Kameras aufgenommen wurden und die Kameraparameter somit konstant und ohne größere Fehlerquellen bestimmt sind. Zudem sind die Personen im Bild größer abgebildet als in den Fußballvideos und die Referenzdaten sind manuell aufbereitet und aufgrund der Kürze der Sequenzen deutlich weniger fehlerbehaftet.

Aus Tabelle 6.12 lässt sich ein `MME` von circa 5000 Identitätsfehlern (je nach Schwellwert τ_d) ablesen. Auf die gesamte Laufzeit aller vier Sequenzen betrachtet, bedeutet das, dass alle zwei Sekunden ein Identitätswechsel auftritt. Bei 20 sichtbaren Feldspielern wird die Identität jedes Spielers im Schnitt 40 Sekunden lang richtig verfolgt. Am `LMME` ist zusätzlich ersichtlich, dass ein Großteil (circa 40%) dieser individuellen Identitätsfehler durch Verwechslung von Spielern derselben Mannschaft verursacht werden.

6.7 Diskussion und Ausblick

In diesem Kapitel wurden Softwaremetriken für die Bewertung von Verfahren zur Verfolgung von mehreren Objekten vorgestellt. Neben der formalen Definition von gängigen Bewertungsmaßen wie CLEAR MOT (Bernardin und Stiefelhagen 2008) wurden neue Metriken vorgestellt. Zum einen greift die Bewertung der Zuordnungsfehler mit dem neu eingeführten $\texttt{GMME}_{\texttt{OPT}}$ (*Global Mismatch Error with Optimal Mapping*) als Erweiterung des `MME` (*Mismatch Error*) die Probleme auf, die von Ben Shitrit u. a. (Ben Shitrit u. a. 2011) identifiziert wurden. Dabei bleibt die Metrik fairer gegenüber dem bewerteten Verfahren und damit insgesamt aussagekräftiger als der von Ben Shitrit u. a. vorgeschlagene `GMME` (*Global Mismatch Error*). Zum anderen ermöglichen es die neu vorgestellte Metrik `LMME` (*Label Mismatch Error*) und ihre globalen Erweiterungen, die Verfolgung von

Objekten zusammengehörender Gruppen (wie Mannschaften) zu bewerten. Die definierten und neu eingeführten Bewertungsmaße sind generell für Verfahren zur Verfolgung mehrerer Objekte geeignet, unabhängig von Domäne und Anwendungsbereich.

Die vorgestellten Bewertungsmaße wurden genutzt, um die Verfahren zur Verfolgung von Spielern aus den vorangegangen Kapiteln zu evaluieren. Für das 2D-Verfahren wurde dazu ein Datensatz aus sechs repräsentativen Videosequenzen mit einer Gesamtlaufzeit von mehr als sechs Minuten und 9700 Einzelbildern vorgestellt, der im Rahmen dieser Arbeit manuell annotiert wurde. Die Sequenzen decken bewusst verschiedene Aufnahmebedingungen wie Auflösung, Zoomfaktor und Wetterbedingungen ab. Der Datensatz ist in einem generischen Format gespeichert und kann auf Anfrage vom Autor gerne für andere vergleichende wissenschaftlichen Arbeiten zur Verfügung gestellt werden. Für die 3D-Verfolgung kommt ein Datensatz mit *Scoutingfeed*-Aufnahmen von vier Bundesligapartien mit einer Länge von mehr als 180 Minuten und 540000 Einzelbilder zum Einsatz. Die Referenzdaten stammen dabei von einem professionellen System (TRACAB 2015).

Die Ergebnisse des vorgestellten Verfahrens zur 2D-Spielerverfolgung liegen bei einem Schwellwert τ_d von 0,7 m bei 0,8 (`MOTA`) beziehungsweise 0,9 ($\mathtt{F}_1$). Damit hebt sich das Verfahren deutlich von den vier Vergleichsverfahren ab, die den Stand der Forschung im Bereich Personenverfolgung repräsentieren. Noch deutlicher deklassiert der vorgestellte Ansatz die Vergleichsmethoden mit Blick auf die benötigte Laufzeit. Die ist ein deutliches Indiz für die potentielle Echtzeitfähigkeit, insbesondere weil die gewählte Implementierung noch viel Raum für Optimierungen lässt.

Für die Evaluierung der 3D-Spielerverfolgung standen leider keine Vergleichsverfahren zur Verfügung. Die durchschnittlichen Werte von 0,6 (`MOTA`) und 0,8 (`MOTA`) bei einem τ_d von 1,5 m können anhand der Ergebnisse der *Multiple Object Tracking Benchmark 3D Challenge 2015* (Leal-Taixé u. a. 2015) eingeordnet werden und heben sich von den besten Verfahren ab. Allerdings muss erwähnt bleiben, dass die Vergleichbarkeit hier sehr eingeschränkt und mit Vorsicht zu genießen ist.

An dieser Stelle wäre ein Vergleich mit einem äquivalenten Verfahren für die Verfolgung von Fußballspielern in monokularen Aufnahmen sowohl in 2D als auch in 3D wünschenswert. Dieser scheitert daran, dass keine mit vertretbaren Aufwand implementierbare Anwendung zu diesem Zweck öffentlich verfügbar ist (nach Kenntnis des Autors zur Zeit der Durchführung der Experimente).

Am `MME` lässt sich ablesen, dass die Anzahl der Identitätswechsel generell ein niedriges Niveau hat. Auffällig ist, dass ein Großteil der Identitätsfehler innerhalb einer Mannschaft stattfinden. Das ist nicht weiter verwunderlich, da die farbbasierten Erscheinungsmodelle auf Mannschaftsebene erstellt werden. Für einige Anwendungen ist dies nicht

weiter schlimm, da die Identität des einzelnen Spielers im Gegensatz zur Mannschaftszugehörigkeit eine untergeordnete Rolle spielt (zum Beispiel Spielzuganalysen). Diese Schwachstelle des Verfahrens könnte durch zusätzliche Erscheinungsmodelle für die individuellen Objekte umgangen werden. Hierzu könnten detaillierte Merkmale wie Haar-, Haut- oder Schuhfarbe genutzt werden, wobei die feineren Auflösungen von kommenden Übertragungsformaten (wie Ultra-HD) hilfreich sein kann. Allerdings birgt die dadurch resultierende höhere Anzahl an Freiheitsgraden immer die Gefahr, das Verfahren zu destabilisieren.

Teil III

Erkennung und Verfolgung des Balls

Kapitel 7

Verfolgung des Balls

7.1 Einleitung

Neben der Position der Spieler spielt die Position des Balls bei der Analyse von Fußballspielen eine zentrale Rolle. Die meisten wichtigen und aussagekräftigen Statistiken (wie beispielsweise Ballbesitz) können ohne eine Kenntnis der Ballposition kaum sinnvoll bestimmt werden. Daher ist es für ein System zur videobasierten Analyse notwendig, den Ball (wenn sichtbar) zu erkennen und seine Position in jedem Einzelbild zu bestimmen. Dabei gibt es diverse Faktoren, welche die Erkennung und Verfolgung des Balls erschweren:

- Der Ball ist mit Abstand das kleinste relevante Objekt auf dem Spielfeld. In den untersuchten Aufnahmen hatte der Ball eine Größe von 5×5 bis maximal 30×30 Pixeln. Hinzu kommt, dass sich diese Größe (etwas durch Zoom, Bewegung von hinten nach vorne o.ä.) rasch ändern kann.
- Der Ball ist ein Objekt, das sehr hohe Geschwindigkeiten und Beschleunigungen (beispielsweise bei Schüssen) vorweist.
- Durch die schnellen Bewegungen kommt es zu Veränderungen der Form (rund $\rightarrow$ eiförmig) und starken Bewegungsunschärfen des Balls im Bild.
- Der Ball wird sehr häufig durch Spieler (beim Dribbling o.ä.) verdeckt. In einer solchen Situation kann die Ballposition nur noch geschätzt werden. Hinzu kommen Überlagerungen mit dem Publikum / Hintergrund bei hohen Bällen, bei denen die Position des Balls (selbst für einen menschlichen Zuschauer) schwer auszumachen ist.

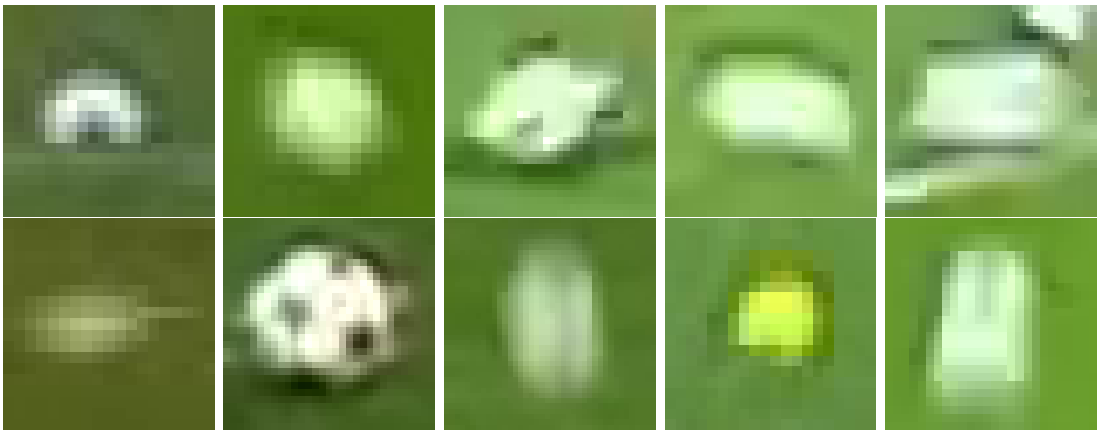


Abbildung 7.1: Verschiedene Erscheinungsformen des Balls

- Der Ball weist häufig verschiedene Muster auf. Allerdings ist in Videoaufnahmen, selbst bei grober Musterung, meist wenig Textur auf dem Ball sichtbar. Zudem gibt es zahlreiche Objekte, die dem Ball in Form (Köpfe) oder Farbe (Füße) sehr ähnlich sind und leicht zu Verwechslungen führen können.
- Zu den spezifischen Schwierigkeiten bei der Erkennung / Verfolgung des Balls gesellen sich natürlich die generellen Schwierigkeiten bei der Auswertung von Fußballaufnahmen, wie beispielsweise Kamerabewegung, inhomogene Ausleuchtung, Störungen im Rasen, schlechte Qualität etc.

In Abbildung 7.1 sind Beispiele für die unterschiedlichen Erscheinungsformen, die der Ball haben kann, dargestellt. In diesem Kapitel wird versucht, diesen Schwierigkeiten bei der 2D-Erkennung und -Verfolgung des Balls in Videosequenzen zu begegnen.

In Abschnitt 7.3 wird eine bildweise Erkennung des Balls vorgestellt, die in erster Linie auf einem Verfahren basiert, das ursprünglich zur Erkennung von Gesichtern entwickelt wurde. Diese Erkennung in Einzelbildern ist dann auch eines von mehreren Merkmalen, die der Verfolgung des Balls in Abschnitt 7.4 dienen.

Das Verfahren und die Ergebnisse aus den Abschnitten 7.3 und 7.4 basieren im Wesentlichen auf den studentischen Arbeiten von Martin Hopper (Hopper 2015) und Trung Hieu Dao (Dao 2015), die unter der Betreuung des Autors der vorliegenden Arbeit entstanden sind und hier mit ihrer freundlichen Genehmigung vorgestellt werden.

7.2 Stand der Forschung

Ballerkennung

In den Veröffentlichungen von Yu u. a. (Yu u. a. 2006; Yu u. a. 2007) sowie von Durus (Durus 2014) werden Ballkandidaten nach einer Bestimmung des Grasbereichs ermittelt. Die Vordergrundobjekte werden anhand von Ballfarbe, geschätzter Ballgröße und Form gefiltert. Die Erstellung des Farbmodells für den Ball ist dabei ein kritischer Schritt und es geht bei beiden Arbeiten nicht klar hervor, wie dieser durchgeführt wird. Ein ähnlicher Ansatz wird von Li, G. Liu u. a. (Li, G. Liu u. a. 2009) vorgestellt, wobei die Vordergrundobjekte lediglich anhand der geschätzten Größe und der Form gefiltert werden. Yu u. a. (Yu u. a. 2007) kommen zu dem Schluss, dass die Schätzung der Ballgröße ohne vorliegende Homographie / Kamerakalibrierung nicht ausreichend genau ist. Da die Farbe des Balls von Spiel zu Spiel unterschiedlich sein kann und bei schnellen Bewegungen deutlich verschwimmt, wird im Ansatz dieser Arbeit auf ein Farbmodell zu Erkennung verzichtet. Stattdessen wird ein texturbasierter Klassifikator genutzt, der mit einer repräsentativen Menge an möglichen optischen Erscheinungen des Balls trainiert wird. Zusätzlich wird diese Erkennung auf unterschiedlichen Skalierungsebenen durchgeführt, wodurch eine exakte Schätzung der Ballgröße nicht notwendig ist.

D'Orazio u. a. (D'Orazio u. a. 2004) führen eine Hough-Transformation zur Erkennung von Kreisen auf einem Kantenbild durch. Die extrahierten Ballkandidaten werden danach mit einem neuronalen Netz klassifiziert, um falsch-positive Erkennungen auszusortieren. Dieser Ansatz schlägt in der Regel fehl, wenn der Ball sich mit anderen Objekten überlagert oder sich mit hoher Geschwindigkeit bewegt, da er keine kreisförmige Struktur im Kantenbild aufweist. Solche Situationen können mit dem in dieser Arbeit vorgestellten Ansatz deutlich besser behandelt werden.

Ballverfolgung

Von den Veröffentlichungen, die sich mit der Verfolgung des Balls in Aufzeichnungen von Fußballspielen beschäftigen, vereinfachen zahlreiche Arbeiten das Problem, indem sie das Szenario deutlich einschränken. So verwenden die Ansätze von Misu u. a. (Misu u. a. 2007), Leo u. a. (Leo u. a. 2008) und Ren u. a. (Ren u. a. 2008) zwei, sechs beziehungsweise acht fixierte Kameras, wodurch die Segmentierung und der Umgang mit Überdeckungen erleichtert wird und die Übertragung auf monokulare Videosequenzen nicht ohne Weiteres möglich ist. Das gleiche gilt für den Einsatz von Stereokameras wie in der Arbeit von Birbach und Frese (Birbach und Frese 2009) oder für ein fixiertes Blickfeld auf den Torbereich wie es D'Orazio u. a. (D'Orazio u. a. 2004; Leo u. a. 2013) vorschlagen.

Die Modellierung und Analyse der Ballbewegung ist ein entscheidender Schritt bei vielen Ansätzen zur Ballverfolgung. So werden für alle generierten Ballkandidaten im Ansatz von Yu u. a. (Yu u. a. 2006; Yu u. a. 2007) die Trajektorien ermittelt und der Ball als Objekt mit der aktivsten Trajektorie bestimmt. Diese Annahme ist nicht immer korrekt und der Ansatz droht fehl zuschlagen, wenn sich der Ball deutlich langsamer bewegt als andere falsch-positiv erkannte Objekte (beispielsweise weiße Schuhe). Einen ähnlichen Ansatz verfolgen Y. Liu u. a. (Y. Liu u. a. 2006). Hier wird mit Hilfe des Viterbi-Algorithmus (Viterbi 1967; Forney 1973) der Kandidat bestimmt, der innerhalb eines Zeitfensters von fünf Einzelbildern die konsistentesten Erkennungen aufweist. Die eigentliche Ballverfolgung wird mit einem Templatematching, welches auf Kreuzkorrelation basiert, und der stochastischen Filterung durch einen Kalman-Filter (Kálmán 1960) durchgeführt. Einen analogen Ansatz verfolgt Durus (Durus 2014). Dieses Vorgehen ähnelt sehr dem Ansatz in der vorliegenden Arbeit, bei dem allerdings das Templatematching durch Merkmale wie morphologische Eigenschaften, der Extraktion der Grasmaske und einem texturbasierten Balldetektor ergänzt wird und der somit eine robustere Verfolgungsleistung vorweisen kann.

Eine der Hauptschwierigkeiten bei der Ballverfolgung ist die häufige Überdeckung des Balls durch Spieler. Dadurch wird kein gutes Ballverfolgungssystem ohne eine geeignete Erkennung und Behandlung von Überdeckungen auskommen. Sowohl der Ansatz von K. Choi und Seo (K. Choi und Seo 2005) als auch der von Shimawaki u. a. (Shimawaki u. a. 2006) nutzen die geschätzten Spielerpositionen, um den Ball zu verfolgen, während er nicht sichtbar ist. Dies ist ähnlich zu dem Vorgehen in der vorliegenden Arbeit, bei dem der ballführende Spieler ermittelt und verfolgt wird, bis der Ball wieder isoliert zum Vorschein kommt. Im Gegensatz zu den anderen Arbeiten, kann durch den sporadischen Einsatz der Balldetektion auf dem ganzen Bild, der Ball auch wieder gefunden werden, wenn ein Spieler fälschlicherweise verfolgt wurde (beispielsweise nach einem Zweikampf). Zudem sind die Ansätze von K. Choi und Seo (K. Choi und Seo 2005) und Shimawaki u. a. (Shimawaki u. a. 2006) nicht online-fähig.

Die Ballerkennung und -verfolgung ist auch für andere Sportarten im Interesse der Forschung, etwa für Tennis (Pingali u. a. 2000; Fei u. a. 2008), Baseball (Theobalt u. a. 2004), Golf (Urtasun u. a. 2005) oder Basketball (Poiesi u. a. 2010; Wang u. a. 2014). Allerdings lassen sich die Erkenntnisse dieser Arbeiten aufgrund der unterschiedlichen Rahmenbedingungen nicht so einfach auf die Domäne Fußball anwenden.

7.3 Erkennung des Balls

Für die Verfolgung des Balls in einer Sequenz von Bildern kann es hilfreich sein, eine Erkennung des Balls auf statischen Einzelbildern durchzuführen (beispielsweise zur Initialisierung oder zur Merkmalsbildung). Da der Ball in Größe und Aussehen (durch Farbe, Bewegungsunschärfe u.ä.) deutlich variieren kann, liegt es nahe, einen Klassifikator zur Objekterkennung anzuwenden, der anhand eines repräsentativen Trainingsdatensatzes eingelernt wird. Ist der Ball verdeckt (etwa durch einen Spieler), ist er außerhalb des Spielfelds oder überdeckt er sich im Flug mit dem Publikum, ist er in statischen Einzelbildern selbst für einen menschlichen Betrachter sehr schwer (wenn überhaupt) auszumachen. In allen anderen Fällen ist die größte Herausforderung, neben den Variationen im Aussehen, die Menge an Objekten vom Ball zu unterscheiden, die eine gewisse Ähnlichkeit zu einem Ball haben, wie beispielsweise helle Köpfe von Spielern und Publikum. In den folgenden Abschnitten werden diese Herausforderungen durch eine klassifikatorenbasierte Objekterkennung angegangen.

Kaskadenbasierte Klassifikatoren sind besonders effizient, wenn die gesuchten Objekte selten auftreten. Da sich in einem Bild an den meisten Positionen kein Ball befindet, bietet sich diese Art der Detektion an dieser Stelle an.

7.3.1 Kaskadenklassifikation mit Boosting

Zur Klassifikation kommt ein Kaskadenklassifikator auf Basis von *AdaBoost* zum Einsatz. Ein ähnliches Verfahren wurde von Viola und Jones (Viola und Jones 2001) mit erstaunlichem Erfolg bei der Erkennung von Gesichtern in Bildern angewendet.

AdaBoost (Freund und Schapire 1996; Freund und Schapire 1997) ist ein Klassifikatoren, welches auf sehr einfachen binären Klassifikationsverfahren (sog. *weak learners*, beispielsweise Entscheidungsbäume der Höhe 1) basiert, die gerade etwas besser sind als eine (gleichverteilte) zufällige Klassifikation. Eine Menge solcher schwachen Klassifikatoren wird in einer gewichteten Summe kombiniert und bildet somit einen komplexen Klassifikator mit gesteigerter Klassifikationsleistung (*Boosting*). Das Training erfolgt in mehreren Runden, wobei in jeder Runde ein schwacher Klassifikator trainiert wird. Dabei wird in der Regel implizit eine Merkmalsselektion durchgeführt. Eine Besonderheit dabei ist, dass die Trainingsbeispiele gewichtet werden, in Abhängigkeit von der Klassifikationsleistung der bisher trainierten schwachen Klassifikatoren. So werden „schwierige“ Beispiele höher gewichtet als „einfache“ und der Fokus in den nächsten Runden auf diese gelenkt. *AdaBoost* gibt einige formale Garantien und hat in zahlreichen praktischen Anwendungen seinen Nutzen gezeigt. *Gentle AdaBoost* (Friedman u. a. 2000) ist eine

Adaption von *AdaBoost*, wobei die wesentlichen Unterschiede darin bestehen, dass die schwachen Klassifikatoren eine Wahrscheinlichkeitsverteilung (anstatt diskreter binärer Werte) zurückgeben und eine andere Fehlerfunktion beim Einlernen minimiert wird. Lienhart und Maydt (Lienhart und Maydt 2002) haben festgestellt, dass *Gentle AdaBoost* in ihren Experimenten zur Gesichtserkennung bessere Resultate erzielte als andere Varianten von *AdaBoost*. Diese Erkenntnis konnte mit Bezug auf die Ballerkennung im Rahmen dieser Arbeit nachvollzogen werden, soll hier aber nicht weiter ausgeführt werden.

Die zugrundeliegende Idee einer Kaskade von Klassifikatoren (Viola und Jones 2001) ist es, dass Hintergrundbereiche eines Bildes durch relativ einfache und effiziente Klassifikatoren aussortiert werden können. Die Kaskade ist eine Art degenerierter Entscheidungsbaum, also eine Aneinanderreihung von mehreren Klassifikatoren (siehe Abbildung 7.2). Wird ein Beispiel von der ersten Stufe als positiv bewertet, so wird die Klassifikation in der zweiten Stufe fortgesetzt. Dies wird bis zur letzten Stufe durchgeführt. Nur wenn ein Beispiel in allen Stufen als positiv bewertet wurde, wird es auch insgesamt als positiv (gesuchtes Objekt) eingestuft. Wird ein Beispiel an irgendeiner Stufe als negativ bewertet, wird es sofort als negativ (kein gesuchtes Objekt / Hintergrund) verworfen. Dies ermöglicht es, eindeutige negative Beispiele durch einfache, effiziente Klassifikatoren in den ersten Stufen auszusortieren, während positive oder schwierige negative Beispiele in komplexeren Klassifikatoren in den späteren Stufen untersucht werden. Jede Stufe der Kaskade wird dabei mit den gleichen Trainingsdaten so trainiert, dass eine maximale Trefferquote und somit eine minimale Rate an falsch-positiven erzielt wird.

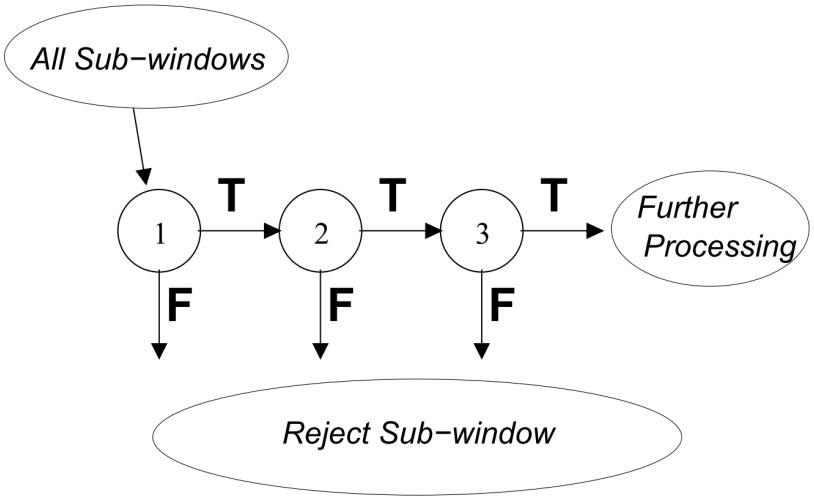


Abbildung 7.2: Das Prinzip einer Kaskade von Klassifikatoren aus (Viola und Jones 2001) ©2001 IEEE.

7.3.2 Local Binary Patterns

Local Binary Patterns (*LPB*) sind einfache und effiziente Merkmale zur Texturanalyse und wurden von Ojala u. a. (Ojala u. a. 1996; Ojala u. a. 2002) erstmalig eingeführt. Bei *LPB* werden die benachbarten Pixel des Zielpixels anhand dessen Wert in 0 (Grauwert ist kleiner) und 1 (Grauwert ist größer gleich) eingeteilt und daraus wird eine binäre Zahl gebildet (klassischerweise acht Bit in der 8er-Nachbarschaft). Mit den binären Zahlen der Pixel (beispielsweise innerhalb eines Suchfensters) wird ein Histogramm erstellt. Der Histogrammvektor kann schließlich als Eingabe für einen Klassifikator dienen. *LPB* zeichnen sich neben der einfachen Berechnung dadurch aus, dass sie eine hohe Unterscheidungsfähigkeit aufweisen und robust gegenüber Helligkeitsunterschieden sind. Der hier vorgestellte Ansatz zur Balldetektion basiert auf den Erkenntnissen von Liao u. a. (Liao u. a. 2007), die *LPB* erfolgreich für eine robuste und effiziente Gesichtserkennung eingesetzt haben. Eine Evaluierung verschiedener Merkmale im Rahmen dieser Arbeit hat ergeben, dass sich *LPB* für die Ballerkennung deutlich besser eignen als andere Merkmale wie Haar-ähnliche Merkmale (Viola und Jones 2001; Lienhart und Maydt 2002) oder HOG (Dalal und Triggs 2005).

7.3.3 Vor- und Nachverarbeitung

Im Folgenden sei ein RGB-Bild I_{RGB} (rot, grün und blau) mit drei Kanälen gegeben, das heißt $I_{RGB} : R_{I_{RGB}} \rightarrow \mathbb{R}^3$. Dabei symbolisieren r, g und b die einzelnen Kanäle, also $I_{RGB}(x,y) := (r(x,y), g(x,y), b(x,y))^T$. Für die Vorverarbeitung wurde die Eignung der folgenden Schritte untersucht:

- Umwandlung in ein Grauwertbild:

$$I_G(x,y) := (0{,}299 \cdot r(x,y) + 0{,}567 \cdot g(x,y) + 0{,}114 \cdot b(x,y)) \tag{7.1}$$

- Schwellwertoperation auf Grauwertbild (*Threshold to Zero*)
- Sobelfilter auf dem Grauwertbild
- Berechnung der Grasmaske (siehe Abschnitt 3.3.2)
- Kombination von Klassifikatoren für jeden RGB-Kanal

Die Versuche haben ergeben, dass die Ballerkennung die besten Ergebnisse lieferte, wenn ein Grauwertbild, das mit einer Schwellwertoperation bearbeitet wurde, als Eingabe

diente. Die Schwellwertoperation $t_z(x)$ (*Threshold to Zero*) bei Eingabe $x \in \mathbb{R}$ und Schwellwert τ_z ist dabei definiert als

$$t_z(x) := \begin{cases} x & \text{für } x \geq \tau_z \\ 0 & \text{sonst} \end{cases} \tag{7.2}$$

Vor der Schwellwertoperation wird eine Histogrammspreizung der Grauwerte des Bildes durchgeführt. Der Schwellwert wird beim Trainieren des Klassifikators für jedes Bild automatisch mit dem Verfahren von Otsu (Otsu 1979) ermittelt. Dabei wird der Schwellwert so optimiert, dass das Histogramm der Grauwerte in zwei Teile unterteilt wird und die Intra-Klassen-Varianz minimiert und die Inter-Klassen-Varianz maximiert wird. Um Störungen im Rasen besser auszublenden, wird bei der Detektion $\tau_z := 245$ gewählt.

In der Umgebung von gesuchten Objekten wird es in der Regel zu mehreren Detektionen kommen, da der Klassifikator auch bei kleinen Schwankungen von Drehung und Verschiebung anschlägt. Aus diesem Grund wird nach der Detektion eine *Non-maximum Suppression* durchgeführt. Die Menge der erkannten Rechtecke wird in Gruppen mit ähnlicher Größe und Position aufgeteilt und das Durchschnittsrechteck jeder Gruppe zurückgegeben. Gruppen mit weniger als N_{Nb} Rechtecken werden verworfen.

Als Nachbearbeitung der Klassifikation werden Erkennungen, die außerhalb der Feldhülle liegen (siehe Abschnitt 3.3.1) nicht beachtet.

7.3.4 Trainings- und Testdaten

Wie auch in den bisherigen Kapiteln dieser Arbeit, wurden für das Einlernen und Testen des Klassifikators Bilder aus totalen Aufnahmen (TV, Amateur u.ä.) von Fußballspielen extrahiert (keine Nahaufnahmen, Wiederholungen etc.). In allen Bildern ist maximal ein Ball enthalten. Die Bilder in denen der Ball zu erkennen ist, wurden als positiv markiert und Position und Ausdehnung des Balls manuell mit Hilfe einer Bounding-Box festgelegt. Bilder in denen kein Ball zu erkennen ist, wurden als negativ markiert. Zusätzlich wurden noch weitere negative Bilder generiert, in dem aus positiven Bildern der Ball mit Hilfe eines Bildbearbeitungsprogramms herausretuschiert wurde. Bilder, in denen sich der Ball mit dem Publikum überlagert, wurden nicht verwendet. Der Bilddatensatz wurde in einen Trainingsdatensatz (288 Positiv, 677 Negativ) und einen Testdatensatz für die Evaluierung (531 Positiv, 502 Negativ) aufgeteilt.

	$N_{Nb} = 3$	$N_{Nb} = 6$
Trefferquote	86,8%	81,8%
⌀ falsch-positiv	11	6
⌀ Jaccard-Koeffizient	0,68	0,69

TABELLE 7.1: Ergebnis der Balldetektion

7.3.5 Implementierung und Auswertung

Das Verfahren wurde mit Hilfe der quelloffenen Bildverarbeitungsbibliothek *OpenCV* (itseez 2015) implementiert, welche das Klassifikationsverfahren (inklusive *Non-maximum Suppression*), die *Local Binary Patterns* sowie die Basisoperationen für die Vor- und Nachverarbeitung zu Verfügung stellt. Die Größe der gesuchten Objekte wurde auf einen Bereich zwischen 8×8 und 28×28 Pixel festgelegt. Der Kaskadenklassifikator wurde mit 17 Stufen trainiert. Dabei wurden pro Stufe eine minimale Trefferquote von 0,995 und eine maximale Rate an falsch-positiven von 0,5 vorgegeben. Die Detektion wurde mit einer (multiplikativen) Skalierungsschrittweite von 1,05 durchgeführt. Ein Ball gilt als erkannt (Treffer), wenn der Jaccard-Koeffizient (siehe Gleichung 2.3) von der Bounding-Box des gefundenen Balls B_D und der Bounding-Box des annotierten Balls B_T größer als 0,3 ist, also $o_J(B_D B_T) > 0{,}3$.

Die Ergebnisse der Balldetektion in Abhängigkeit von der Anzahl benachbarter Rechtecke bei der *Non-maximum Suppression* (N_{Nb}) können aus Tabelle 7.1 entnommen werden. In deutlich über 80% der Bilder wird der Ball korrekt erkannt. Dabei wird eine durchschnittliche (räumliche) Überdeckung von deutlich über 0,5 (Jaccard-Koeffizient) erzielt. Als eher negativ zu bewerten ist die hohe durchschnittliche Anzahl an falsch-positiven Erkennungen. Dennoch kann eine solche Detektion eine sehr hilfreiche Unterstützung einer Ballverfolgung darstellen. Die fehlerhaften Erkennungen können dabei durch Einbezug des zeitlichen Kontextes und der Position der Spieler verworfen werden (siehe den folgenden Abschnitt 7.4).

7.4 Verfolgung des Balls

In diesem Abschnitt wird (im Gegensatz zu vielen Ansätzen in der Literatur) ein *Online*-Verfahren (Laplante 2001, Seite 343) zur Verfolgung des Balls vorgestellt, das heißt, die Position des Balls wird seriell von Einzelbild zu Einzelbild geschätzt. Anders als bei *Offline*-Verfahren ermöglicht ein *Online*-Algorithmus prinzipiell die Bestimmung der Ballposition in Echtzeit. Dies ist für bestimmte Anwendungen eine wichtige Randbedingung, wie beispielsweise bei der taktischen Analyse parallel zum laufenden Spiel.

Das Verfahren benötigt als Eingabe, neben den Bildern einer Videosequenz, auch die initiale Position des Balls im ersten Einzelbild als Bounding-Box. Diese kann manuell oder automatisch (beispielsweise mit einem Detektor wie in Abschnitt 7.3) generiert werden. Optional nutzt das Verfahren auch Grasmasken (siehe Abschnitt 3.3.2) und die Bounding-Boxen der 2D-Spielerpositionen (siehe Abschnitt 4) als Eingabe.

Obwohl das Verfahren mit verschiedensten Aufnahmen von Fußballspielen zurechtkommt, gibt es die folgenden (wenigen) Voraussetzungen an die Bildsequenz:

- Die Szene sollte möglichst von der Seite gefilmt sein.
- Die Szene sollte keine Schnitte oder Übergänge enthalten.
- Der Ball sollte das Spielfeld nicht verlassen. Insbesondere wenn der Ball in der Luft ist und sich mit dem Publikum im Hintergrund überdeckt, kann eine seriöse Erkennung / Verfolgung des Balls nicht gewährleistet werden.
- Die Ballposition zu Beginn muss bekannt sein.

Das *Online*-Verfahren iteriert über die Bildsequenz. Nach der Vorhersage der aktuellen Ballposition mit einem Kalman-Filter (Kálmán 1960) (*Prediction*), wird anhand verschiedener Merkmale die wahrscheinlichste Position des Balls im aktuellen Bild gemessen (*Observation* und *Candidate Selection*). Eine spezielle Behandlung wird in Situationen durchgeführt, in denen der Ball verdeckt ist und somit nicht erkannt wird (*Occlusion Handling*). Dabei wird versucht, den ballführenden Spieler zu verfolgen. Um den Ball nach langen Verdeckungsphasen wieder zu erkennen, werden in regelmäßigen Abständen Hinweise einer aufwendigen Ballerkennungen (siehe Abschnitt 7.3) auf dem ganzen Einzelbild inkorporiert (*Hinting*). Der Algorithmus terminiert, sobald die komplette Videosequenz prozessiert wurde.

7.4.1 Initialisierung

Wie schon erwähnt, ist die Position des Balls im ersten Einzelbild der Videosequenz als Bounding-Box gegeben. Diese Position wird genutzt, um ein Template des Balls zu generieren. Dazu wird einfach das Teilbild I_T in Größe und Position der Bounding-Box extrahiert und gespeichert. Dieses Template wird im Laufe der Zeit mittels guten Erkennungen des Balls verfeinert.

7.4.2 Stochastische Schätzung der Balltrajektorie

Analog zur 2D-Verfolgung der Spieler wird die Trajektorie des Balls mit Hilfe eines Kalman-Filters (Kálmán 1960) geschätzt. Wie bei der Spielerverfolgung, spielt, neben der eigentlichen Filterung der Ballbewegung, der Vorhersageschritt eine zentrale Rolle. Der Ball wird im aktuellen Bild nur in einer eingeschränkten Suchregion um die vorhergesagte Position herum gesucht. Dies ermöglicht eine effiziente Wiedererkennung des Balls von Bild zu Bild und erspart aufwendige, vollständige Suchoperationen auf dem gesamten Bild. Zudem wird analog zur *Gating*-Technik in Abschnitt 3.7.5 die Vorhersage genutzt, um mehrere mögliche Ballpositionen zu bewerten. Es kommt ebenfalls ein einfaches Bewegungsmodell mit konstanter Geschwindigkeit zum Einsatz. Da die Größe des initial erstellten Templates des Balls konstant bleibt, entspricht das Zustandsmodell dem der 3D-Spielerverfolgung mit Position und Geschwindigkeit (siehe Abschnitt 4.5.4.1). Dabei werden die initialen Einträge der Kovarianzmatrix P für die Position sehr klein (0) und für die Geschwindigkeit sehr groß (100) gewählt. Prozess- und Messrauschen werden als

$$Q := E_4 \tag{7.3}$$

und

$$R := \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \tag{7.4}$$

vorgegeben, wobei $E_n \in \mathbb{R}^n$ die n-dimensionale Einheitsmatrix darstellt.

7.4.3 Messung der Ballposition

Um von Bild zu Bild die Position des Balls schätzen zu können, bedarf es aussagekräftiger Merkmale, die im Folgenden erläutert werden.

7.4.3.1 Generierung der Suchregion

Wie bereits erwähnt, wird anhand der vorhergesagten Position des Balls eine Suchregion definiert, in der nach möglichen Ballkandidaten gesucht wird. Die Nutzung des zeitlichen Kontexts ermöglicht es zum einen, aufwendige Suchen auf großen Teilen des Bildes zu vermeiden. Zum anderen wird so auch das Risiko von falsch-positiven Erkennungen reduziert. Seien w_T und h_T die Breite und Höhe des Templates und sei $\vec{m}^-$ die vorhergesagte Position. Dann ist $\vec{m}^-$ der Mittelpunkt der Suchregion und $w_R := f_T \cdot w_T$ und $h_R := f_T \cdot h_T$ die Breite und Höhe der Suchregion R. Der Parameter f_T gibt die Abhängigkeit der Größe der Suchregion von der Templategröße an und wird mit $f_T := 5$

gewählt. Die folgenden Merkmale werden ausschließlich innerhalb der Region R gemessen.

7.4.3.2 Templatematching

Das erste Merkmal innerhalb der Region R wird durch ein einfaches Verfahren zum Templatematching mit dem initial erstellten Template aus Abschnitt 7.4.1 berechnet. Dabei wird das Template I_T über die Suchregion verschoben und an jeder Position die normierte Summe der quadrierten Grauwertdifferenzen zwischen dem Template und dem aktuellen Bild I berechnet (itseez 2015):

$$I_{F_1}(x,y) := 1 - \frac{\sum_{x',y'}(I_T(x',y') - I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'} I_T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}}. \tag{7.5}$$

Die Summen in Gleichung 7.5 werden über alle Farbkanäle iteriert. Für das Ergebnisbild gilt $I_{F_1}(x,y) \in [-\infty, 1]$.

7.4.3.3 Geometrische Suche

Optisch kann der Ball insbesondere bezüglich der Farbe mit Schuhen, Socken oder Köpfen verwechselt werden. Im Gegensatz zu solchen Objekten weist er eine besondere geometrische Eigenschaft auf (außer bei schnellen Bewegungen): er ist klein und rund. Mit dem zweiten Merkmal wird versucht, diese Tatsache zu nutzen. Dazu wird zunächst eine Kantensuche nach Canny (Canny 1986) durchgeführt. Die gefundenen Kanten werden zu Konturen zusammengefügt (Suzuki und Abe 1985) und anschließend werden etwaige Störungen durch ein morphologisches Schließen entfernt. Die gefundenen Konturen werden verworfen, wenn sie die folgenden geometrischen Bedingungen nicht erfüllen (itseez 2015):

- **Fläche:** $a_{min} \leq a < a_{max}$, wobei a die umschlossene Fläche der Kontur darstellt und $a_{min} := (\frac{1}{3} \cdot r)^2 \cdot \pi$ und $a_{max} := r^2 \cdot \pi$ mit $r := \frac{1}{4} \cdot (w_T + h_T)$ gilt.
- **Rundheit:** $c \geq 0{,}5$, wobei $c := \frac{4 \cdot \pi \cdot a}{p^2}$ gilt und p die Länge der Kontur (also den Umfang) darstellt.
- **Inertia:** das Verhältnis von großer zu kleiner Hauptachse $\geq 0{,}2$.
- **Konvexität**: $k := \frac{a}{a_{conv}}$; $k \geq 0{,}5$, wobei a_{conv} die Fläche der konvexen Hülle der Kontur ist.

Die verbleibenden Konturen werden ausgefüllt als Binärbild gespeichert, welches mit einem Mittelwertfilter geglättet wird. Für das resultierende Bild I_{F_2} gilt dann $I_{F_2}(x,y) \in [0;1]$, wobei hohe Werte an der Stelle von Objekten zu finden sind, die im geometrischen Sinne einem Ball nahe kommen.

7.4.3.4 Grasmaske

Die Grasmaske wird berechnet wie in Abschnitt 3.3.2 beschrieben. Das Resultat ist ein Bild I_{F_3} mit $I_{F_3} \in [0;1]$. Hohe Werte sind dabei an der Stelle von Vordergrundobjekten (wie beispielsweise Spieler oder Ball) zu finden.

7.4.3.5 Balldetektor

Das in Abschnitt 7.3 vorgestellte Kaskadenverfahren zur Ballerkennung wird innerhalb der Suchregion R mit einem *Sliding Window*-Ansatz durchgeführt (inklusive der Vor- und Nachbearbeitungsschritte aus Abschnitt 7.3.3). Die erkannten Positionen werden als binäres Rechteck in ein Bild I_{F_4} eingezeichnet und mit einem Mittelwertfilter in der Größe des Templates geglättet. Für das Ergebnis gilt $I_{F_4} \in [0;1]$.

7.4.3.6 Kombination der Merkmale

Die einzelnen Merkmalsbilder werden zu einem Bild kombiniert. Dabei wird analog zu Abschnitt 3.7.6 eine gewichtete Summe berechnet:

$$I_F(x,y) := w_P(x,y) \cdot \sum_i w_i \cdot I_{F_i}(x,y), \tag{7.6}$$

mit $w_1 := 0{,}6$, $w_2 := 0{,}1$, $w_3 := 0{,}2$ und $w_4 := 0{,}1$.

Das Gewicht $w_P(x,y)$ mit $\vec{x} := (x\ y)^T$ ist der *Gating*-Term, der abhängig von der vorhergesagten Position $\vec{m}^-$ ist, mit

$$w_P(\vec{x}) := 1 - \max\left(0, \min\left(1, \frac{\|\vec{m}^- - \vec{x}\|_2 - \tau_P}{k_P}\right)\right). \tag{7.7}$$

Somit werden Positionen in der Nähe der vorhergesagten Position höher bewertet. Innerhalb des Radius $\tau_P := w_T$ werden die Positionen mit 1 gewichtet. Erst außerhalb dieses Radius fällt die Gewichtung linear in Abhängigkeit von $k_P := \sqrt{w_T^2 + h_T^2}$ ab.

Es gilt $I_F(x,y) \in [-\infty; 1]$, wobei höhere Werte ≥ 0 eine Wahrscheinlichkeit repräsentieren, mit der sich ein Ball an der entsprechenden Position befindet.

7.4.3.7 Auswahl des besten Kandidaten

Um die besten Kandidaten für die Ballposition in der Suchregion R zu bestimmen, werden die lokalen Maxima in $I_F(x,y)$ ermittelt. Für diese *Non-maximum Suppression* wird zunächst das Bild mit einem Mittelwertfilter geglättet. Danach wird eine Grauwert-Dilatation (Steger u. a. 2008, S. 140 ff.) ausgeführt, die in dem Bild I_{dil} resultiert. Als Kandidaten für die Ballposition werden die Position (x,y) ausgewählt, für die gilt: $I_F(x,y) \geq I_{dil}$. Aus diesen Kandidaten wird diejenige Position (x',y') mit dem höchsten Wert $I_F(x',y')$ bestimmt.

Der Kandidat mit der höchsten Konfidenz ist nicht notwendigerweise der Ball, denn der Ball ist sehr häufig verdeckt (meistens durch Spieler) und zudem kommt es häufig vor, dass Objekte mit ähnlichem Aussehen besser abschneiden (etwa wenn der Ball optisch unscharf ist). Nur wenn

$$I_F(x',y') \geq \tau_F \tag{7.8}$$

gilt mit $\tau_F := 0{,}72$, wird die Position als Messung übernommen, der Kalman-Filter aktualisiert und das nächste Bild in der Videosequenz prozessiert. Andernfalls wird der Ball als verdeckt angenommen und eine spezielle Behandlung durchgeführt (siehe Abschnitt 7.4.6).

7.4.4 Aktualisierung des Templates

Das Templatebild des Balls I_T, das initial erstellt wird, hat eine zentrale Rolle bei der Merkmalsberechnung. Allerdings weist der Ball eine hohe Variation im Aussehen auf, da er vor allem bei schnellen Bewegungen verwischt und unförmig erscheint. Es besteht das Risiko, dass das initial erstellte Template I_{T_0} nicht sehr repräsentativ ist. Daher wird bei sehr guten Erkennungen des Balls das Template graduell auf die aktuelle Erscheinung angepasst. Gilt $I_F(x',y') < \tau_T$ mit $\tau_T := 0{,}8$, dann wird das Template nicht verändert und es gilt $I_{T_{n+1}} := I_{T_n}$. Andernfalls wird das Template aktualisiert mit

$$I'_{T_{n+1}} := (1 - \alpha_N) \cdot I_{T_n} + \alpha_N \cdot I_{T_N} \tag{7.9}$$

und

$$I_{T_{n+1}} := (1 - \alpha_B) \cdot I'_{T_{n+1}} + \alpha_B \cdot I_{T_0} \tag{7.10}$$

Dabei ist I_{T_N} das Bild der aktuellen Messposition und die Parameter $\alpha_N := 0{,}1$ und $\alpha_B := 0{,}5$ steuern den Grad des Templateupdates. Der Schritt in Gleichung 7.10 verhindert ein zu großes Abweichen vom ursprünglichen Template. So wird das langsame Abdriften vermieden, wenn zum Beispiel fälschlicherweise ein Schuh als Ball erkannt wird.

7.4.5 Bestimmung des ballführenden Spielers

Sind die Positionen der Spieler als Bounding-Boxen gegeben, so wird in jedem Einzelbild, in dem der Ball erkannt wurde, der ballführende Spieler ermittelt. Ballbesitz ist Definitionssache (siehe dazu auch Abschnitt 8.3 und (Hoernig u. a. 2016)): Eine naive Definition ist, dass der Ballbesitz beginnt, wenn der Spieler den Ball berührt und endet, wenn ein anderer Spieler den Ball berührt. Nach der Definition von Yu u. a. (Yu u. a. 2005) endet der Ballbesitz nach der letzten Berührung des Spielers. Diese Definitionen sind geeignet für die taktische Analyse. Im Fall der Behandlung von Überdeckung ist die Definition von Durus (Durus 2014) anhand der Reichweite eines Spielers geeigneter. Sei (x_b,y_b) die Position des Balls (Mitte des Templates) im Bild und sei (x_f,y_f) die geschätzte Position des Fußes des Spielers im Bild. Dabei ist x_f in der Mitte der Bounding-Box $B(x, y, w, h)$, also $x_f := x + 0{,}5 \cdot w$ und etwas oberhalb der Unterkante y_f mit $y_f := y + 0{,}8\overline{3} \cdot h$. Über die Abstände $d_x := x_b - x_f$ und $d_y := y_b - y_f$ zwischen Ball und (geschätzter) Fußposition wird die Reichweite definiert. Die Distanz zwischen Ball und Spieler ist definiert durch

$$d_{bp} := \sqrt{d_x^2 + (e \cdot d_y)^2} \tag{7.11}$$

mit $e := 1{,}5$.

Der Ball ist in Reichweite des Spielers, wenn

$$-1{,}5 \leq \frac{d_x}{w} \leq 1{,}5 \tag{7.12}$$

und

$$-1{,}5 \leq e \cdot \frac{d_y}{h} \leq 0{,}5 \tag{7.13}$$

gilt.

Die größeren Schwellwerte in Gleichung 7.12 basieren auf der Tatsache, dass die meisten schnellen Bewegungen im Fußball quer zur Mittellinie (also in X-Richtung im Bild)

stattfinden. Durch die asymmetrischen Schwellwerte in Gleichung 7.13 wird versucht, die perspektivische Verzeichnung abzufedern. Somit repräsentiert in der Regel ein Pixel hinter dem Spieler ($d_y < 0$) eine größere Fläche auf dem realen Rasen als ein Pixel vor dem Spieler ($d_y > 0$).

Die Regeln zur Bestimmung des ballführenden Spielers lauten dann wie folgt:

1. Befindet sich der Ball innerhalb der Bounding-Box eines oder mehrerer Spieler, so hat derjenige Spieler Ballbesitz, innerhalb dessen Bounding-Box sich der Ball befindet und der die kürzeste Distanz d_{bp} zum Ball nach Gleichung 7.11 hat.
2. Befindet sich der Ball innerhalb der Reichweite eines oder mehrerer Spieler, so hat derjenige Spieler Ballbesitz, innerhalb dessen Reichweite sich der Ball befindet und der die kürzeste Distanz d_{bp} zum Ball nach Gleichung 7.11 hat.
3. Treffen die Punkte 1 und 2 nicht zu, so hat kein Spieler Ballbesitz.

7.4.6 Behandlung von Überdeckungen

Wird der Schwellwert τ_F aus Gleichung 7.8 unterschritten, so gilt der Ball als nicht erkannt und es wird implizit angenommen, dass der Ball durch einen Spieler verdeckt ist. Sind die Positionen der Spieler als Bounding-Boxen gegeben, so wird versucht, den ballführenden Spieler zu verfolgen, bis der Ball wieder sichtbar (erkennbar) wird. Sind keine Spielerpositionen verfügbar, so wird die Ballposition mit Hilfe des Bewegungsmodells geschätzt.

7.4.6.1 Verfolgung des ballführenden Spielers

Ist der Ball verdeckt, so wird die vorhergesagte Position des Balls als Messung übernommen. Diese Position wird allerdings auf einen rechteckigen Bereich in der Mitte (in der Nähe der Hüfte) der Bounding-Box des ballführenden Spielers beschränkt. Das garantiert, dass die Suchregion für die Wiedererkennung (siehe Abschnitt 7.4.6.3) zentral über dem verdeckenden Spieler positioniert ist.

7.4.6.2 Überdeckung ohne ballführenden Spieler

Gibt es keine Spielerinformationen oder wurde im letzten Einzelbild vor der Überdeckung kein ballführender Spieler identifiziert, so wird die Position des Balls anhand des Bewegungsmodells (Kalman-Filter) geschätzt. Dabei wird die geschätzte Geschwindigkeit von Bild zu Bild durch einen Faktor $\alpha \in [0;1]$ mit $\alpha := 0{,}5$ abgemildert. Diesem

Vorgehen liegt die Annahme zu Grunde, dass sich der Lauf des Balls abgebremst hat, da er ansonsten schnell wieder sichtbar werden würde. Es können dadurch kurze Überdeckungen behandelt werden. Bei längeren Überdeckungen werden die Vorhersagen zu ungenau. Dann ist die weitere Verfolgung des Balls abhängig von einer erfolgreichen Wiederkennung des Balls.

7.4.6.3 Wiedererkennung des Balls

Zur Wiedererkennung des Balls wird im Falle einer Überdeckung die Suchregion R in Abhängigkeit von der Dauer der Überdeckung vergrößert. Je länger die Überdeckung andauert, umso unsicherer wird eine Aussage über die Position des Balls und umso größer wird die Suchregion gewählt. Wird nun ein bester Ballkandidat (x',y') gefunden, für den gilt

$$I_F(x',y') \geq \tau_R \tag{7.14}$$

mit dem Wiedererkennungschwellwert $\tau_R := 0{,}76$, so gilt der Ball als wiedererkannt an eben dieser Position. τ_R ist bewusst größer als τ_F aus Gleichung 7.8 gewählt, da bei der Wiedererkennung keine Vorinformation aus dem zeitlichen Kontext vorhanden ist.

7.4.7 Hinweise durch Balldetektion

Der Ansatz der Suche nach Maxima im Merkmalsraum (siehe Abschnitt 7.4.3) innerhalb einer eingeschränkten Suchregion dient in erster Linie einer effizienteren Ballverfolgung und dem Ausschluss von falsch-positiven Erkennungen. Dennoch kann es nützlich sein, von Zeit zu Zeit eine aufwendige Suche mit dem Balldetektor aus Abschnitt 7.3 auf dem kompletten Bild durchzuführen. Für eine erkannte Ballposition (x'',y'') wird schließlich der Wert des Merkmalsbildes $I_F(x'',y'')$ berechnet. Ist dieser Wert besser als jeder andere Kandidat und gilt zusätzlich noch $I_F(x'',y'') \geq \tau_R$, dann wird dieser Hinweis als neue Ballposition übernommen und die Verfolgung inklusive Kalman-Filter neu initialisiert.

7.4.8 Testdaten

Das Verfahren wurde auf acht verschiedenartigen Videosequenzen mit bewegter Kamera getestet, wobei drei davon schon zur Auswertung der 2D-Spielerverfolgung in Abschnitt 6.5 zum Einsatz kamen. Details zu den Sequenzen und den jeweiligen Begegnungen sind aus den Tabellen 7.2 und 7.3 zu entnehmen.

	Start in Spielzeit	Länge (s)	Frames	FPS	Auflösung	Sender
ND	68:42	37	925	25	1024 × 576	ZDF
MLEV	48:14	65	3250	50	1280 × 720	DFL
SPCHI	33:25	45	2250	50	1280 × 720	ARD
BFG	45:35	44	1101	25	1024 × 576	Eurosport
REALATL	28:29	41,5	2075	50	1280 × 720	ZDF
PS	47:16	60	1500	25	1024 × 576	Eurosport
CROGER	83:41	34	1700	50	1280 × 720	ZDF
GERARG	111:59	23,5	1175	50	1280 × 720	ARD

Tabelle 7.2: Übersicht Videosequenzen

	Datum	Begegnung	Wettbewerb
ND	14.11.2012 20:30	Niederlande - Deutschland	Länderspiel
MLEV	15.03.2014 18:30	Bayern München - Leverkusen	Bundesliga
SPCHI	18.06.2014 21:00	Spanien - Chile	Weltmeisterschaft
BFG	06.02.2013 20:30	Burkina Faso - Ghana	Afrikameisterschaft
REALATL	24.05.2014 20:30	Real Madrid - Atletico Madrid	Champions League
PS	28.06.2013 14:30	Polen - Schweden	Frauen U-17 EM
CROGER	27.11.2013 15:00	Kroatien - Deutschland	Frauen Länderspiel
GERARG	13.07.2014 20:30	Deutschland - Argentinien	Weltmeisterschaft

Tabelle 7.3: Übersicht Begegnungen

7.4.9 Implementierung und Auswertung

Das Verfahren wurde mit Hilfe der quelloffenen Bildverarbeitungsbibliothek *OpenCV* (itseez 2015) implementiert. Die Grasmasken wurden mit dem Verfahren aus Abschnitt 3.3.2 erstellt. Für die Sequenzen **ND** und **PS** wurden die manuellen Annotationen der Bounding-Boxen der Spieler aus Abschnitt 6 benutzt. Für alle anderen Sequenzen wurden die Spielerpositionen mit Hilfe der 2D-Spielerverfolgung (siehe Abschnitt 4.4) generiert und sind somit fehlerbehaftet.

Die tatsächlichen Ballpositionen (ohne Ausmaße) im Bild wurden ebenfalls mit der Hand annotiert. Dabei wurde die Position nicht in allen Bildern markiert, sondern zwischen zwei annotierten Bildern interpoliert. Dabei wurde durch Sichtkontrolle sichergestellt, dass die interpolierten Positionen korrekt sind.

Für die Auswertung sind drei verschiedene Zustände des Verfahrens definiert:

1. **Erkannt:** Der Ball wurde erkannt. In diesem Fall, wird in der Auswertung zwischen korrekten und falschen Erkennungen unterschieden.
2. **Verfolgt:** Der Ball ist nicht erkannt und der ballführende Spieler wird verfolgt.
3. **Geschätzt:** Der Ball ist nicht erkannt, kein ballführender Spieler wird verfolgt und die Position des Balls wird anhand des Bewegungsmodells geschätzt.

Video	Erkannt		Geschätzt		Gesamt-
	Anteil	Genauigkeit	Anteil	Genauigkeit	Genauigkeit
ND	67,5%	91,4%	32,5%	38,3%	74,2%
MLEV	87,8%	90,8%	12,2%	40,8%	74,2%
SPCHI	89,4%	95,7%	10,6%	49,7%	90,8%
BFG	54,6%	96,3%	45,4%	21,4%	62,3%
REALATL	65,3%	93,3%	34,7%	31,3%	71,8%
PS	57,0%	67,0%	43,0%	11,1%	43,0%
CROGER	40,3%	84,0%	59,7%	28,5%	50,9%
GERARG	66,9%	96,7%	33,1%	64,4%	86,0%

Tabelle 7.4: Ergebnisse ohne Spielerpositionen (50 Bilder / Sequenz)

Video	Erkannt		Verfolgt		Geschätzt		Gesamt-
	Anteil	Genauigk.	Anteil	Genauigk.	Anteil	Genauigk.	Genauigk.
ND	70,7%	94,2%	23,9%	51,7%	5,3%	12,8%	79,7%
MLEV	88,8%	90,8%	6,2%	71,1%	5,0%	14,2%	85,3%
SPCHI	89,4%	95,2%	9,2%	60,1%	1,4%	65,2%	91,6%
BFG	55,9%	96,5%	29,7%	31,4%	14,4%	20,9%	66,3%
REALATL	62,9%	92,0%	29,0%	33,6%	8,2%	25,8%	69,7%
PS	60,1%	70,4%	17,7%	17,6%	22,3%	6,6%	46,8%
CROGER	42,3%	86,3%	22,3%	45,7%	35,3%	29,6%	57,2%
GERARG	65,4%	96,7%	19,3%	72,6%	15,3%	6,0%	86,5%

Tabelle 7.5: Ergebnisse mit Spielerpositionen (50 Bilder / Sequenz)

Die Position des Balls gilt als richtig bestimmt, wenn die berechnete Position maximal eine Templategröße von der tatsächlichen Position entfernt liegt.

Von den Videosequenzen wurden alle Szenen ausgeschlossen, in denen der Ball das Spielfeld verlassen hat oder sich im Flug mit dem Publikum im Hintergrund überdeckt hat. In diesen Fällen ist die Verfolgung des Balls mit dem vorgestellten Verfahren nicht möglich. Die restlichen Szenen wurden zum einen in Sequenzen mit einer Länge von 50 Bildern und zum anderen in Sequenzen mit einer Länge von 100 Bildern aufgeteilt, wobei sich diese Sequenzen jeweils um 50 Bilder überlappen. Diese beliebige Aufteilung soll vermeiden, dass von Hand bevorzugt gutartige Sequenzen und Startpositionen ausgewählt werden. Mit den sich überlappenden Sequenzen der Länge von 100 Bildern soll vor allem der Nutzen der Hinweise des Detektors ausgewertet werden. Als Startposition wird für jede Sequenz die annotierte, tatsächliche Position genutzt.

Die Ergebnisse für die Sequenzen mit einer Länge von 50 Bildern sind ohne Einbezug von Spielerpositionen in Tabelle 7.4 dargestellt. Die Ergebnisse unter Einbezug der Spielerpositionen sind aus Tabelle 7.5 zu entnehmen. Hierbei fällt auf, dass teilweise ein erheblicher Anstieg der Genauigkeit mit Hilfe der Spielerpositionen erreichbar ist. Zudem wird ersichtlich, dass ein enger Zusammenhang zwischen dem Anteil der Phasen, in denen der Ball erkennbar ist (*Erkannt*) und der Genauigkeit vorliegt.

Video	Genauigkeit	
	mit Hinweise	ohne Hinweise
ND	79,0%	73,6%
MLEV	82,3%	76,3%
SPCHI	90,1%	86,9%
BFG	63,0%	50,8%
REALATL	69,9%	63,3%
PS	44,4%	38,6%
CROGER	56,1%	40,9%
GERARG	85,4%	77,3%

TABELLE 7.6: Ergebnisse mit und ohne Hinweise der Balldetektion (100 Bilder / Sequenz)

In Tabelle 7.6 sind die Ergebnisse für die Sequenzen mit einer Länge von 100 Bilder aufgelistet. Dabei werden die Werte mit und ohne Einbezug der Hinweise des Balldetektors aus Abschnitt 7.3 verglichen. Es wird deutlich, dass die Gesamtperformanz durch die Nutzung der Detektorhinweise deutlich gesteigert werden kann.

7.5 Diskussion und Ausblick

Wie eingangs erwähnt, ist der Ball das kleinste, und sich am schnellsten bewegende relevante Objekt in monokularen Aufnahmen von Fußballspielen. Dementsprechend ist die Erkennung und Verfolgung des Balls eine schwierige Aufgabe. Dieser Herausforderung stellen sich die Methoden, die in diesem Kapitel vorgestellt wurden.

Im ersten Teil wurde ein Verfahren zur automatischen Erkennung des Balls in statischen Einzelbildern vorgestellt. Der Ball kommt in den meisten Fällen darin genau einmal vor. Daher kommt ein kaskadenbasierter Klassifikator zum Einsatz, der den überwiegenden Teil der vielen negativen Beispiele (kein Ball) im Bild auf der untersten Ebene aussortieren kann. Nur die positiven Beispiele und die schwierigen negativen Beispiele werden zusätzlich auf den komplexeren höheren Kaskadenebenen ausgewertet. Zusammen mit den einfach zu berechnenden *Local Binary Patterns* (*LPB*) ergibt sich daraus ein effizientes Verfahren, welches in dem umfangreichen und schwierigen Evaluierungsdatensatz in weit über 80% den Ball richtig erkannt hat. Das Verfahren ist prinzipiell auch zur Erkennung des Balls in anderen Sportarten geeignet. Dazu müsste in erster Linie ein anderer Datensatz für das Training des Klassifikators genutzt werden. Voraussetzung dafür ist allerdings eine entsprechend große Erscheinung des Balls im Bild. Bei Sportarten wie Feldhockey, Eishockey oder Baseball ist das Sportgerät sehr klein und schnell, so dass der Ansatz dort an seine Grenzen kommen würde. Ein weiterer Schwachpunkt

der Erkennung in statischen Einzelbildern ist die hohen Anzahl an falsch-positiven Detektionen. Die ist der Tatsache geschuldet, dass es zahlreiche andere Objekte im Bild gibt, die dem Ball ähnlich sind, wie etwas Füße oder Köpfe. Hier können der Einbezug des zeitlichen Kontexts und der Erkennung der Spieler die Ergebnisse noch verbessern.

Im zweiten Teil des Kapitels wurde daher ein Verfahren zur Verfolgung des Balls vorgestellt, dass ebendiese Informationen ausnutzt. Dabei kommt wie bei der Spielerverfolgung ein Kalman-Filter (Kálmán 1960) zum Einsatz, zum einen um die Bewegungstrajektorie des Balls zu glätten. Zum anderen kann mit der Vorhersage die Suchregion für das nächste Einzelbild eingeschränkt werden. In dieser wird der Ball mit Hilfe einer Kombination von Merkmalen wie Templatematching, geometrischen Bedingungen, Spielersilhouetten und der Ballerkennung durch ein Optimierungsverfahren gesucht. Durch die Positionen der Spieler können Überdeckungen behandelt werden, indem bei einer Verdeckung der ballführende Spieler verfolgt wird. Bei einem temporären Verlust des Balls helfen gelegentlich Erkennungsvorgänge auf dem gesamten Bild bei einer Wiedererkennung. Die Evaluierung auf dem umfangreichen und schwierigen Datensatz von acht repräsentativen Videosequenzen ergibt eine durchschnittliche Genauigkeit von über 70%. Eine Einschränkung des Verfahrens ist die notwendige manuelle Initialisierung zur Templateerstellung und Festlegung der Startposition des Balls. Hier könnte der Einsatz des automatischen Erkennungsklassifikators und eines Multi-Hypothesen-Ansatz in den ersten Einzelbildern hilfreich sein, bei dem die falsch-positiven Hypothesen über den zeitlichen Kontext und den Spielererkennung nach und nach verworfen werden. Das Verfahren ist prinzipiell auch für die Ballverfolgung in anderen Sportarten geeignet. Neben dem Einlernen des Klassifikators mit einem anderen Datensatz, ist hierfür vor allem eine Anpassung der Erkennung der Spielersilhouetten notwendig.

Teil IV

Anwendungen

Kapitel 8

Anwendungen

8.1 Einleitung

In diesem Kapitel werden potentielle Einsatzmöglichkeiten für die beschriebenen Verfahren aus den vorangegangenen Kapiteln vorgestellt. Dabei sind insbesondere die 3D-Positionsdaten von Spieler und Ball auf dem Spielfeld relevant.

Die in dieser Arbeit vorgestellten Verfahren zur Erkennung und Verfolgung von Spielern und dem Ball sind robust gegenüber Änderungen der äußeren Bedingungen und können auch mit qualitativ schlechteren Aufnahmen umgehen. Somit sind sie für eine breitere Benutzergruppe ausgelegt, wie etwa Spieler und Trainer von Amateurvereinen, Fußballfans und Lokal-Journalisten, die meist nicht über Hochglanz-Videomaterial verfügen. Dementsprechend wurden die Methoden dieser Arbeit und der Arbeit von Hoernig u. a. (Hoernig u. a. 2015) in eine Webanwendung eingebettet, mit der im Prinzip jeder weltweit seine Videosequenz hochladen und analysieren lassen kann. Diese Anwendung wird im Abschnitt 8.2 vorgestellt.

Die generierten 3D-Daten können genutzt werden, um sport- und trainingswissenschaftliche Analysen durchzuführen. Hier ist vor allem der Ballbesitz von Interesse, wobei herkömmliche manuelle Methoden lediglich den Mannschaftsballbesitz mit einer grobkörnigen Auflösung bestimmen. Für detaillierte Untersuchungen ist jedoch eine individuelle Bestimmung des Ballbesitzes unter Berücksichtigung von Spielunterbrechungen und der Einstufung des Grades der Ballkontrolle entscheidend. Diesbezüglich wird in Abschnitt 8.3 kurz auf die Ergebnisse der Arbeit von Hoernig u. a. (Hoernig u. a. 2016) eingegangen.

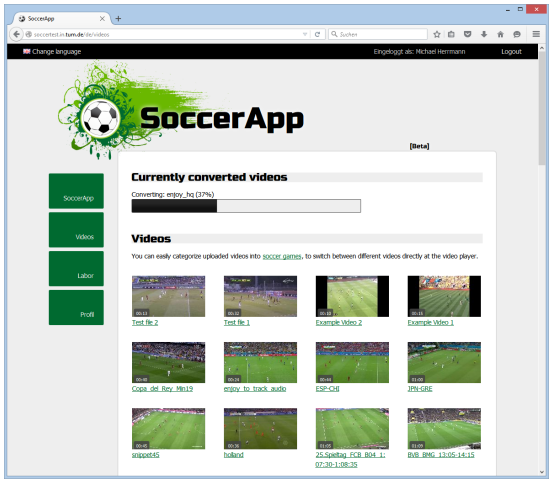
Neben der reinen Auswertung der Positionsdaten, können diese auch zur (virtuellen) Visualisierung von Fußballspielen nützlich sein. So wird beispielsweise in Wettbüros der

Beweis für relevante Spielszenen (Tore o.ä.) häufig durch die Anzeige von Fernsehbildern geführt. Um die Lizenzgebühren für das Bildmaterial zu sparen, könnten in Echtzeit generierte Positionsdaten für solche Szenen in gerenderter Form übertragen werden. Oder die Daten können zur Realisierung eines *Virtual Reality* Erlebnisses genutzt werden (Virtually Live US, Inc. 2016; LiveLike VR 2016). Für beide Anwendungen ist eine Transformation von den reinen Koordinaten auf dem Spielfeld zu realitätsgetreuen Darstellungen mit Hilfe einer Computer-Grafik-Engine notwendig.

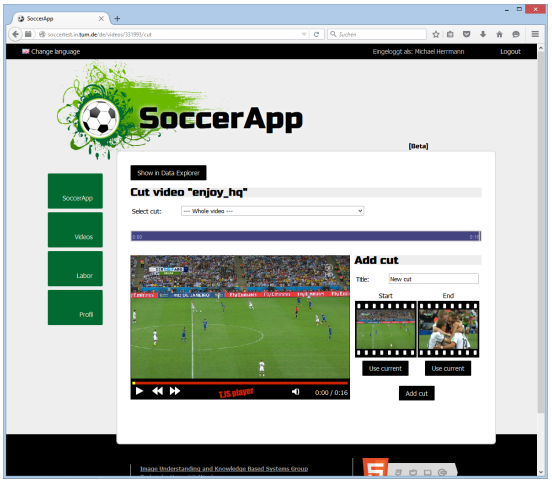
Um noch realistischere Darstellungen der Spieler zu gewährleisten, muss die genaue Pose der Spieler aus dem 2D-Bild rekonstruiert werden. Dabei werden die relativen Positionen der einzelnen Körperteile eines Spielers bestimmt. Das Vorliegen einer exakten Pose kann auch bei der Erstellung von Statistiken hilfreich sein. So kann beispielsweise ein Schuss mit dem linken Fuß von einem Schuss mit dem rechten Fuß oder einem Kopfball unterschieden werden. Für die Bestimmung der Posen ist es sehr nützlich, die Position und Größe der Spieler im 2D-Bild (siehe Abschnitt 4.4) zu kennen. Ein Verfahren zur Bestimmung der Posen der Spieler auf Basis von *Random Forests* wird in Abschnitt 8.4 vorgestellt.

8.2 Spielerverfolgung mit einer Webanwendung

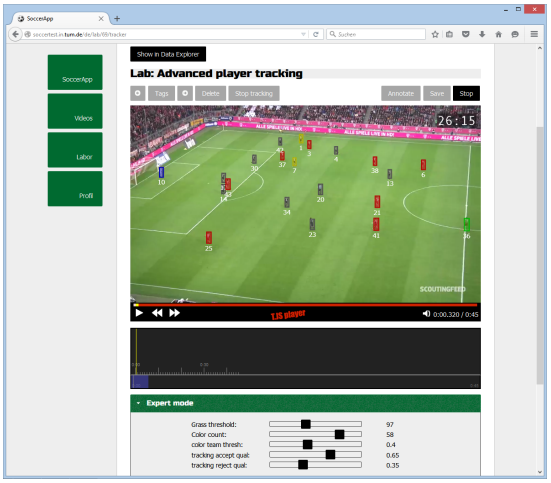
Mit der entwickelten Webanwendung können Benutzer ihre Aufnahmen von Fußballspielen analysieren und die 3D-Positionen der Spieler generieren lassen. Diese Aufnahmen können beispielsweise von einem Camcorder bei Amateurbegegnungen oder von der Handykamera im Stadion stammen oder alte Fernsehaufzeichnungen sein. Der erste Schritt für den Benutzer ist das Hochladen der Videosequenz auf den Analyseserver, wobei eine Vielzahl von Formaten unterstützt wird (siehe Abbildung 8.1a). Hierbei wandelt die Anwendung alle Sequenzen in ein einheitliches Format mit einer Komprimierung in ausreichender Qualität um. Als Nächstes kann der Benutzer die relevanten Teile der Sequenz definieren und unwichtige Abschnitte, wie etwa Werbeeinblendungen oder Aufnahmen der Halbzeitpause, entfernen (siehe Abbildung 8.1b). Danach wird die Bestimmung der 2D-Spielerpositionen (siehe Abbildung 8.1c) sowie der Kameraparameter (siehe Abbildung 8.1d) für die jeweiligen relevanten Abschnitte einer Sequenz gestartet. Die Berechnungen laufen serverseitig, so dass der Benutzer keinerlei Rechenressourcen zur Verfügung stellen muss. Die ermittelten Ergebnisse der Online-Verfahren werden bildweise visualisiert, sobald sie zur Verfügung stehen. So kann der Benutzer zur Laufzeit die Ergebnisse überwachen. Zusätzlich ist es möglich einige wichtige Parameter der Verfahren über die Webseite zu ändern. Wurden die 2D-Spielerpositionen und Kameraparameter



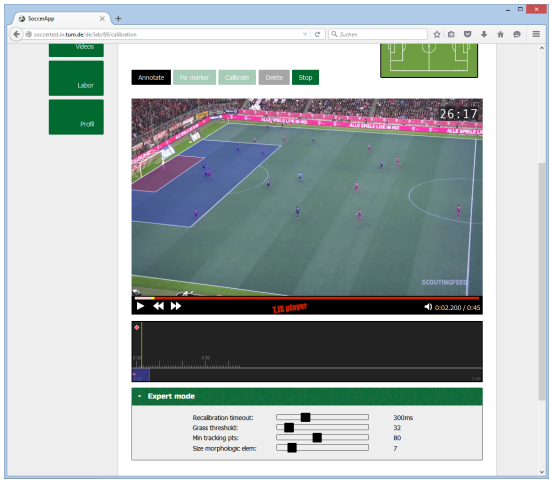
(a) Hochladen von Videos



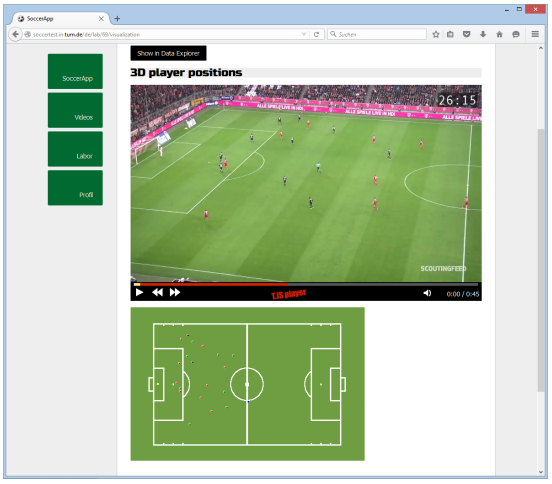
(b) Schneiden von Videos



(c) Spielerverfolgung



(d) Kameraverfolgung



(e) Visualisierung der 3D-Positionen

Abbildung 8.1: Eine Webanwendung zur Analyse von Fußballvideos. Bildquellen b: (Das Erste 2014); c, d und e: (DFL 2014b).

für eine Sequenz bestimmt, so können die transformierten Positionen der Spieler im Spielfeld angezeigt und als Textdatei exportiert werden (siehe Abbildung 8.1e).

8.3 Bestimmung des Ballbesitzes

Der Ballbesitz ist eine wichtige Maßzahl in der analytischen Bewertung von Fußballbegegnungen. Auch wenn er nicht immer direkt mit Erfolg und Misserfolg korreliert, so ist er doch ein wichtiger Hinweis auf die Dominanz einer Mannschaft, da ohne den Besitz des Balls ein Torerfolg sehr unwahrscheinlich ist. Aufgrund des manuellen Aufwands wird häufig nur der Ballbesitz je Mannschaft registriert, was dazu führt, dass Forschungsstudien sich bislang in erster Linie auf den Mannschaftsballbesitz fokussieren. In der Arbeit von Hoernig u. a. (Hoernig u. a. 2016) wird ein Verfahren vorgestellt, das automatisiert den individuellen Ballbesitz von Spielern bestimmt. Hierzu werden die 3D-Spielfeldpositionen von Spieler und Ball genutzt. Der Begriff des Ballbesitzes ist im Allgemeinen nicht eindeutig bestimmt, weswegen in der genannten Arbeit folgende Formen des individuellen Ballbesitzes definiert werden (siehe (Hoernig u. a. 2016)):

- Individueller Ballbesitz (IBP) beginnt in dem Moment, in dem der Spieler eine Aktion mit dem Ball ausführen kann und endet, wenn eine anderer Spieler IBP hat.
- Individuelle Ballaktion (IBA) beginnt in dem Moment, in dem der Spieler eine Aktion mit dem Ball ausführen kann und endet, wenn er keine Aktion mehr durchführen kann.
- Individuelle Ballkontrolle (IBC) ist analog zu IBA, nur dass der Spieler die Möglichkeit haben muss, zwischen verschiedenen taktischen Aktionen zu wählen.

Entsprechend dieser Definition hat beispielsweise ein Spieler noch individuellen Ballbesitz (IBP) nach einem Pass, solange kein anderer Spieler am Ball ist, obwohl er keine Aktion mehr durchführen kann und somit keine individuelle Ballaktion (IBA) mehr hat. Köpft ein Abwehrspieler den Ball nach einer Ecke aus dem Strafraum, so hat er für eine kurze Zeit eine individuelle Ballaktion (IBA), aber keine individuelle Ballkontrolle (IBC). Die Grundlage für erfolgreiche Spielzüge ist es diese individuelle Ballkontrolle innerhalb einer Mannschaft zu halten.

Die automatisierte Erkennung von IBP, IBA und IBC, wie sie in der Arbeit von Hoernig u. a. (Hoernig u. a. 2016) vorgestellt wird, bereinigt und glättet zunächst die Positionsdaten von Spieler und Ball. Im nächsten Schritt werden Schüsse über die Extremwerte in

der Ableitung der Ballgeschwindigkeit detektiert. Befindet sich ein Spieler in der Nähe des Balls, wenn dieser stark seine Geschwindigkeit ändert, so beginnt für ihn der IBP. Während die IBA über einen Distanzschwellwert bestimmt wird, kommt für die IBC ein Bayes'sche Netz (Cooper und Herskovits 1992) zum Einsatz, welches unter anderem mit Merkmalen wie die Dauer der Ballaktion, Geschwindigkeit und Beschleunigung des Balls die IBA-Intervalle klassifiziert. Eine Evaluierung anhand der Daten einer Partie in einer europäischen Liga ergaben eine Sensitivität und einen positiven Vorhersagewert für IBP und IBA von 0,8 bis 0,9. Für weitere Details sei auf (Hoernig u. a. 2016) verwiesen.

8.4 Bestimmung der Posen

In diesem Abschnitt wird ein neuartiges Verfahren zur Bestimmung von Posen von Spielern in monokularen Bildern vorgestellt. Der Ansatz basiert auf einer pixelweisen Klassifizierung der Körperteile mit Hilfe von *Random Forests* und auf inverser Kinematik, um die Skelettteile zu rekonstruieren. Da die Methode ursprünglich für Tiefenbilder entwickelt wurde, wird eine umfangreiche Untersuchung bezüglich der Eignung von gängigen 2D-Merkmalen durchgeführt. Zusätzlich wird die Orientierung einer Person geschätzt, um die Klassifizierung der Körperteile und die Anpassung des Skeletts zu unterstützen. Für die Schätzung der Posen werden die erkannten Körperteile an die Gelenke eines 3D-Skeletts angepasst. Das Ergebnis ist eine 3D-Pose, die in den 2D-Raum zurückprojiziert wird. Die Evaluierung anhand eines anspruchsvollen Datensatzes resultiert in Ergebnissen, die auf und über der Augenhöhe des Stands der Forschung liegen. Dieser Abschnitt basiert im Wesentlichen auf den Ergebnissen der Masterdarbeit (Bigontina 2014), des Artikels (Bigontina u. a. 2015) und bisher unveröffentlichten Materials, die unter Betreuung und Mitwirkung des Autors der vorliegenden Arbeit entstanden sind und die mit der freundlichen Genehmigung von Andreas Bigontina hier vorgestellt werden.

8.4.1 Stand der Forschung

Wie man beispielsweise an den Übersichtsartikeln von Moeslund u. a. (Moeslund u. a. 2006) und Poppe (Poppe 2007) erkennen kann, ist die Bestimmung der Posen von Personen eine intensiv untersuchte Anwendung. Ein beliebter Ansatz für monokulare Bilder ist die Verwendung des Modells der *Pictorial Structures* (Fischler und Elschlager 1973), welches zuerst von P. F. Felzenszwalb und Huttenlocher (P. F. Felzenszwalb und Huttenlocher 2005) für die Posenbestimmung benutzt wurde. Viele Erweiterungen dieser Technik wurden vorgeschlagen, wobei einige immer noch den gegenwärtigen Stand der Technik repräsentieren, wie beispielsweise der von Andriluka u. a. (Andriluka u. a. 2012). Die prinzipielle Idee des Modells der *Pictorial Structures* ist es, ein verformbares Objekt

durch eine Menge von Bildausschnitten zu repräsentieren, die von flexiblen Verbindungen zusammengehalten werden. Dabei können sich die einzelnen Teile eines Objekts bewegen, allerdings werden Abweichungen von der eingelernten relativen Position bestraft. Für diese Art der Modellierung wurden diverse Varianten vorgeschlagen, beispielsweise von Y. Yang und Ramanan (Y. Yang und Ramanan 2011; Y. Yang und Ramanan 2013).

Andriluka u. a. (Andriluka u. a. 2008) haben die Idee der *Pictorial Structures* angepasst und das verformbare Modell (*Deformable Model*) dazu genutzt, um Personen in schwierigen Szenen zu erkennen und zu verfolgen. Dieser Ansatz wurde noch erweitert, um besonders ausgeprägte Posen im Sportbereich zu bestimmen (Andriluka u. a. 2009) und um 3D-Posen in Videosequenzen zu erkennen (Andriluka u. a. 2010). Letzteres geschieht in drei Stufen: die Erkennung von Körperteilen auf Basis von *Shape Context* Merkmalen (Belongie u. a. 2001) und *AdaBoost* (Freund und Schapire 1996; Freund und Schapire 1997), die Zusammenführung zeitlicher Information für kurze Bildfolgen und die Rekonstruktion von 3D-Posen auf Basis von 2D-Beobachtungen. Da das Aussehen von Personen abhängig von der Blickrichtung ist, wird dabei eine spezielle Detektion der Blickrichtung durchgeführt. Diese Idee wird in dem hier vorgestellten Ansatz aufgenommen, indem versucht wird, die Orientierung von Personen im Bild zu klassifizieren. Andriluka u. a. nutzen die Technik von Mori und Malik (Mori und Malik 2002; Mori und Malik 2006) um eine Transformation von 2D-Posen zu 3D-Posen anhand von Beispielbildern zu trainieren. Im Gegensatz dazu, wird in dem hier vorgestellten Ansatz ein 3D-Skelett an das Ergebnis der 2D-Körperteilerkennung angepasst.

Die Erkennung der Posen kann deutlich robuster gestaltet werden, wenn Tiefeninformation hinzugenommen wird. Shotton u. a. (Shotton u. a. 2011) erzielten überzeugende Resultate, indem jedes (Vordergrund-)Pixel einzeln anhand der Tiefeninformation mit Hilfe eines *Random Forest* einem Körperteil zugeordnet wurde. Da dieses Vorgehen im höchsten Maße parallelisierbar ist, kann die Klassifizierung in Echtzeit erfolgen (beispielsweise mit einer GPU (Sharp u. a. 2008)).

Der hier vorgestellte Ansatz ist eine Erweiterung des Ansatzes von Shotton u. a., um auf monokularen Bildern ohne Tiefeninformation zu arbeiten. Zusätzlich werden Ideen von Wei u. a. (Wei u. a. 2012) übernommen, die ein 3D-Skelett an die klassifizierten Körperteile anpassen.

Kazemi u. a. (Kazemi u. a. 2013) habe den Ansatz von Shotton u. a. ebenfalls für monokulare Bilder genutzt. Dabei kamen HOG Merkmale zum Einsatz, ohne dass die Eignung verschiedener Merkmale untersucht wurde. Zudem wurden keine Körperteile klassifiziert, sondern die Position der Gelenke. Daraus wurden über einen probabilistischen Ansatz mit Hilfe des *Pictorial Structures* Modell 2D-Posen rekonstruiert.

In jüngster Zeit haben die Ergebnisse mit Ansätzen des *Deep Learnings* wie von Tompson u. a. (Tompson u. a. 2014) oder Chen und Yuille (Chen und Yuille 2014) die Posenbestimmung auf ein neues Niveau gehoben. Allerdings werden dabei nur 2D-Posen bestimmt.

8.4.2 Klassifikation von Körperteilen und Orientierung

Im Folgenden wird angenommen, dass die Position einer Person im Bild (in Form einer Bounding-Box) gegeben ist. In dem hier vorgestellten Ansatz werden zunächst zwei Probleme angegangen:

- Die Bestimmung der Orientierung der (ganzen) Person.
- Die Einteilung aller Pixel in Körperteilklassen.

In beiden Fällen werden *Random Forest*-Klassifikatoren eingesetzt. In diesem Abschnitt wird nach einer kurzen Einführung untersucht, wie gut sich unterschiedliche Bildmerkmale für die jeweilige Aufgabe eignen. Eine ausführliche Beschreibung von *Random Forests* findet sich beispielsweise in der Veröffentlichung von Criminisi u. a. (Criminisi u. a. 2011).

8.4.2.1 Random Forests

Ein *Random Forest* ist eine Menge von *Random Trees*. *Random Trees* sind im Prinzip Entscheidungsbäume, die durch einen randomisierten Lernprozess trainiert wurden. Jedem inneren Knoten eines Entscheidungsbaums wird eine einfache binäre Testfunktion zugeordnet, ein sogenannter *Weak Learner*. Die Klassifikation eines einzelnen Datenpunkts beginnt am Wurzelknoten. Auf Basis der Entscheidungen der *Weak Learners* an jedem inneren Knoten wird der Baum durchlaufen, bis ein Blatt des Baums erreicht wird. Die Blätter enthalten schließlich die Information für das Klassifikationsergebnis.

Jeder *Weak Learner* teilt den Eingaberaum in zwei Teilregionen auf und wird über eine binäre Funktion h definiert. Dabei ist das Ergebnis in diesem Falle abhängig von einer Bildkoordinate $\vec{u} \in \mathbb{R}^2$ und einem Vektor von Entscheidungsparametern $\vec{\rho} \in \mathbb{R}^p$:

$$\mathrm{h}(\vec{u}, \vec{\rho}) \in \{0,1\} \tag{8.1}$$

Für den hier vorgestellten Ansatz kommen ausschließlich einfache *Weak Learner* zum Einsatz, die den Eingaberaum anhand einer achsenparallelen Hyperebene aufteilen, das

heißt:

$$\mathrm{h}(\vec{u}, \vec{\rho} = (\rho_1, \ldots, \rho_{p-1}, \tau)) := \mathbb{I}\big[\, \phi(\vec{u}, \vec{\rho}) > \tau \big] \tag{8.2}$$

Dabei ist $\mathbb{I}[e]$ die charakteristische Funktion, die 1 zurückgibt, wenn der Ausdruck e wahr ist und 0, wenn e nicht wahr ist. Die Funktion ϕ wählt abhängig von $\vec{\rho}$ ein Merkmal des Pixels $\vec{u}$ aus. Das Ergebnis von ϕ wird schließlich mit dem Schwellwert $\tau \in \mathbb{R}$ verglichen, wobei τ den letzten Eintrag des Vektors $\vec{\rho}$ darstellt. Die Funktion ϕ und der Parametervektor $\vec{\rho}$ werden während des Trainings eingelernt. Solche einfachen *Weak Learner* bieten den Vorteil eines geringen Berechnungsaufwands und ermöglichen so eine effiziente Klassifikation.

8.4.2.2 Training

In der Trainingsphase werden die Entscheidungsbäume generiert und für jeden Knoten die „besten“ Parameter ermittelt. Um die Eignung eines Parametersatzes zu bewerten, wird häufig (wie auch in diesem Ansatz) der Informationsgewinn (*Information Gain*) I betrachtet. Sei S eine Menge von Datenpunkten, wobei jeder Datenpunkt einer Klasse c zugeordnet ist. S sei in die Teilmengen $S^L \subseteq S$ and $S^R = S \setminus S^L$ aufgeteilt. Dann bestimmt sich der Informationsgewinn der Aufteilung wie folgt:

$$\mathrm{I}(S, \vec{\rho}) := \mathrm{H}(S) - \sum_{X \in \{R,L\}} \frac{|S^X|}{|S|} \cdot \mathrm{H}(S^X) \tag{8.3}$$

Dabei ist H der mittlere Informationsgehalt (Entropie nach Shannon (Shannon 2001)):

$$\mathrm{H}(S) := -\sum_{c \in C(S)} \mathrm{p(c)} \cdot \log_2(\mathrm{p(c)}) \tag{8.4}$$

$C(S)$ ist die Menge der verschiedenen Klassen an Datenpunkten in S und $\mathrm{p}(c)$ die relative Häufigkeit der Klasse c in S.

Gesucht werden die Parameter $\vec{\rho} \in M \subseteq \mathbb{R}^p$, die den Informationsgewinn maximieren, das heißt:

$$\arg\max_{\vec{\rho} \in M} \mathrm{I}(S, \vec{\rho}) \tag{8.5}$$

Da M beliebig mächtig sein kann, wird nur eine zufällig ausgewählte Teilmenge von M betrachtet (daher der Name *Random Forest*).

(a) (b) (c) (d)

ABBILDUNG 8.2: Darstellung einiger untersuchter Merkmale (Bigontina 2014): (a) Silhouette, (b) Gradient, (c) Haar-ähnliche Merkmale (d) HOG (Dalal und Triggs 2005)

Das Training eines Baums wird beendet, wenn eine vorgegebene maximale Tiefe erreicht ist oder wenn die Anzahl der Trainingsbeispiele im aktuellen Knoten einen Schwellwert unterschreitet. Die relativen Häufigkeiten jeder Klasse werden im entstehenden Blattknoten gespeichert.

8.4.2.3 Klassifikation

Ein Eingabevektor wird klassifiziert, indem in jedem Baum des *Random Forest* beim Wurzelknoten gestartet wird und der Baum durchlaufen wird, bis ein Blatt erreicht ist. In den Blättern ist jeweils für jede Klasse eine Wahrscheinlichkeit gespeichert. Die Gesamtwahrscheinlichkeit einer Klasse wird durch die Kombination der Ergebnisse der einzelnen Bäume ermittelt. Dabei wird für n Bäume mit den jeweiligen Wahrscheinlichkeiten $\mathrm{p}_t(c) \in [0;1]$, $t \in \{1,\ldots,n\}$ für Klasse c das arithmetische Mittel bestimmt:

$$\mathrm{p}(c) := \frac{1}{n} \cdot \sum_{t=1}^{n} \mathrm{p}_t(c). \tag{8.6}$$

mit $\mathrm{p}(c) \in [0;1]$.

8.4.2.4 Merkmale

Um die nicht vorhandene Tiefeninformation wie von Shotton u. a. (Shotton u. a. 2011) zu kompensieren, werden im Folgenden Bildmerkmale untersucht, die direkt aus den monokularen Bilddaten gewonnen werden können. In Abbildung 8.2 sind einige Merkmale bildlich dargestellt. Die meisten Merkmale beziehen sich auf ein RGB-Bild I_{RGB}, wobei die Symbole r, g und b für die jeweiligen Kanäle stehen.

Silhouette Zur Berechnung der Silhouette wird der Ansatz von Hoernig u. a. (Hoernig u. a. 2015) angewendet (siehe auch Abschnitt 3.3.2). Das Ergebnis gibt an mit welcher (geschätzten) Wahrscheinlichkeit ein Pixel zum Vordergrund gehört. Die Entscheidungsfunktion untersucht die Pixel in der Umgebung des zu klassifizierenden Pixels mit den

Koordinaten $\vec{u} \in \mathbb{R}^2$. Die Grasmaske $I_{Grass} : \mathbb{R}^2 \to [0;1]$ gibt den Wahrscheinlichkeitswert der Silhouette für die gegebenen Pixelkoordinaten zurück. Der Vektor $\vec{\delta} \in \mathbb{R}^2$ bestimmt, welche Pixel in der Umgebung betrachtet werden. Die Entscheidungsfunktion für die Silhouette ist definiert durch

$$\mathrm{h_{sil}}(\vec{u}, \vec{\rho} = (\vec{\delta}, \tau)) := \mathbb{I}\big[I_{Grass}(\vec{u} + \vec{\delta}) > \tau\big] \tag{8.7}$$

.

Hautfarbe Ein Nachteil der silhouettenbasierten Klassifikation ist, dass gehäuft Mehrdeutigkeiten auftreten. So kann beispielsweise eine Hand, die sich vor dem Oberkörper befindet, über die Silhouette nicht erkannt werden. Allerdings ermöglicht die Hautfarbe der Hand eine Unterscheidung vom Oberkörper. Auch wenn es Ausnahmen gibt (beispielsweise wenn eine Person Handschuhe trägt), kann die Farbe der Haut in vielen Fällen wichtige Hinweise geben, insbesondere für die Extremitäten und das Gesicht. Hautfarbe für sich alleine bietet dennoch nicht genug Information und muss immer mit anderen Merkmalen kombiniert werden.

Hautfarbe ist ein subjektiver Begriff und in dieser Untersuchung wird die einfache Klassifikationsregel von Gomez und Morales (Gomez und Morales 2002) angewendet mit

$$\begin{aligned} \mathrm{skin}(\vec{u}) := & \\ \Big(\tfrac{\mathrm{b}(\vec{u})}{\mathrm{g}(\vec{u})} &< 1.249\Big) \ \wedge \\ \Big(\tfrac{\mathrm{r}(\vec{u})+\mathrm{g}(\vec{u})+\mathrm{b}(\vec{u})}{3\,\mathrm{r}(\vec{u})} &> 0.696\Big) \ \wedge \\ \Big(\tfrac{1}{3} - \tfrac{\mathrm{b}(\vec{u})}{\mathrm{r}(\vec{u})+\mathrm{g}(\vec{u})+\mathrm{b}(\vec{u})} &> 0.014\Big) \ \wedge \\ \Big(\tfrac{\mathrm{g}(\vec{u})}{3(\mathrm{r}(\vec{u})+\mathrm{g}(\vec{u})+\mathrm{b}(\vec{u}))} &< 0.108\Big) \end{aligned} \tag{8.8}$$

Dabei ist die binäre Funktion $\mathrm{skin}(\vec{u})$ genau dann wahr, wenn das Pixel mit den Koordinaten $\vec{u}$ als Haut klassifiziert wird und $\mathrm{r}(\vec{u}), \mathrm{g}(\vec{u}), \mathrm{b}(\vec{u}) \in [0{,}1]$ stellen die Werte der RGB-Kanäle dar. Die Entscheidungsfunktion für die Hautfarbe ist definiert als

$$\mathrm{h_{ski}}(\vec{u}, \vec{\rho} = \vec{\delta}) := \mathbb{I}\big[\,\mathrm{skin}(\vec{u} + \vec{\delta})\big] \tag{8.9}$$

RGB-Daten Die Farbe kann sehr hilfreich bei der Unterscheidung der Körperteile sein (beispielsweise durch Farben der Kleidung, Hautfarbe und unterschiedliche Ausleuchtung in der 3D-Szene). Daher liegt eine Klassifikation anhand der RGB-Rohdaten nahe. Zumal

ein mächtiger Farbklassifikator theoretisch einen Hautfarbenklassifikator beinhaltet, wie auch Khan u. a. (Khan u. a. 2010) bereits demonstriert haben. Um sowohl die Farbe einzelner Pixel als auch den Unterschied zweier Pixel als Merkmal zuzulassen, ist die Entscheidungsfunktion für die RGB-Daten wie folgt definiert:

$$\mathrm{h}_{\mathrm{col}}(\vec{u}, \vec{\rho} = (\vec{\delta}_1, \vec{\delta}_2, i, b, \tau)) := \mathbb{I}\big[\,\mathrm{c}(\vec{u} + \vec{\delta}_1, i) - b \cdot \mathrm{c}(\vec{u} + \vec{\delta}_2, i) > \tau\big] \tag{8.10}$$

Dabei ist $\mathrm{c}(\vec{u}, i)$ eine Funktion, die den Farbwert in Kanal i an der Position $\vec{u}$ abfragt. Der Parametervektor $\vec{\rho}$ beinhaltet die Versatzvektoren $\vec{\delta}_1$ und $\vec{\delta}_2$, den Kanalindex i, den Schwellwert τ und den Parameter $b \in \{0{,}1\}$. Dieser macht für $b := 0$ aus der Differenzoperation eine einfache Farbkanalabfrage (analog zu Shotton u. a. (Shotton u. a. 2013)).

Gradienten Gradienten im Bild geben Auskunft über ausgeprägte Farbänderungen zwischen benachbarten Pixeln. Dahinter steht die Annahme, dass im Übergangsbereich zwischen verschiedenen Körperteilen starke Änderungen in der Farbe vorliegen. Die Norm des Gradienten $|\vec{\nabla}(\vec{u})|$ an der Stelle $\vec{u}$ wird mit Hilfe des zentralen Differenzenquotienten auf Basis der euklidischen Distanz zwischen zwei RGB-Vektoren angenähert. Die Entscheidungsfunktion ist dann wie folgt definiert:

$$\mathrm{h}_{\mathrm{gra}}(\vec{u}, \vec{\rho} = (\vec{\delta}, \tau)) := \mathbb{I}\big[|\vec{\nabla}(\vec{u} + \vec{\delta})| > \tau\big] \tag{8.11}$$

Haar-ähnliche (*Haar-like*) Merkmale Der Betrag des Gradienten stellt sich in der Regel durch dünne Kantenverläufe dar. Diese sind schwierig mit pixelbasierten Verfahren zu klassifizieren. Um dieses Problem zu umgehen, wird der Ansatz von Papageorgiou u. a. (Papageorgiou u. a. 1998) und Viola und Jones (Viola und Jones 2001) verfolgt und die Summe des Gradientbetrags ganzer Bildregionen verglichen. Diese Art von Merkmalen wurde bereits erfolgreich bei der Gesichtserkennung sowie bei der Erkennung von mimischen Ausdrücken eingesetzt (Mayer 2012). Um die Anzahl der Regionen klein zu halten, kommen als Regionen nur Linien mit Länge $l \in \{2{,}4{,}8{,}16{,}32{,}64\}$ und Orientierungswinkel $o \in \{0°, 45°, 90°, 135°\}$ zum Einsatz. Sei $\mathrm{f}(\vec{u}, l, o)$ die Summe des Gradientenbetrags entlang der Linie mit Länge l, Orientierung o und Mittelpunkt $\vec{u}$. Die Entscheidungsfunktion ist definiert als

$$\mathrm{h}_{\mathrm{haa}}(\vec{u}, \vec{\rho} = (\vec{\delta}, o, l, \tau)) := \mathbb{I}\big[\,\mathrm{f}(\vec{u} + \vec{\delta}, l, o) > \tau\big] \tag{8.12}$$

HOG Merkmale Die HOG-Merkmale (*Histogram of Oriented Gradients*, siehe (Dalal und Triggs 2005) wurden bereits in Abschnitt 3.7.4 vorgestellt. Die Entscheidungsfunktion, welche diese Merkmale nutzt, betrachtet einzelne Histogrammklassen in der Umgebung des Zielpixels. Da sich die Blöcke überlappen, kann nicht jedem Pixel eine Zelle eindeutig zugeordnet werden. Deswegen wählt $\mathrm{hog}(\vec{u}, o)$ die Zelle aus dem Block mit dem geringsten Abstand des Zellenmittelpunkts zum Zielpixel an der Stelle $\vec{u}$ aus. Die Orientierungsklasse des Histogramms wird durch den Parameter o bestimmt und die Entscheidungsfunktion ist wie folgt definiert:

$$\mathrm{h}_{\mathrm{hog}}(\vec{u}, \vec{\rho} = (\vec{\delta}, o, \tau)) := \mathbb{I}[\,\mathrm{hog}(\vec{u} + \vec{\delta}, o) > \tau] \tag{8.13}$$

Wie gehabt ist dabei $\vec{\delta}$ der Abstandsvektor, τ der Schwellwert und $o \in \{1 \ldots 9\}$ die Orientierung der Histogrammklasse.

Orientierung Ist die Orientierung einer Person bekannt, so kann diese Information dabei helfen, Körperteile robuster zu erkennen (insbesondere bei der Unterscheidung von linken und rechten Körperteilen). Die bisher erläuterten Merkmale werden daher nicht nur zur Klassifikation der Körperteile benutzt, sondern auch zur Klassifikation der Orientierung einer Person mit Hilfe eines *Random Forests*. Dazu wird die Orientierung bezüglich der Aufsicht von oben in acht gleichmäßige Segmente eingeteilt (siehe Abbildung 8.3).

Abbildung 8.3: Beispiele für Spieler und ihre Orientierung. (Bigontina 2014)

Die resultierenden Wahrscheinlichkeiten sind die Eingabewerte für die Klassifikation der Körperteile und die entsprechende Entscheidungsfunktion ist definiert als

$$\mathrm{h}_{\mathrm{ori}}(\vec{u}, \vec{\rho} = (o, \tau)) := \mathbb{I}[p(o) > \tau] \tag{8.14}$$

$p(o|I)$ ist die Wahrscheinlichkeit der Orientierung o im Bild I. Dieser Wert gilt für das ganze Bild und ist unabhängig von den Bildkoordinaten $\vec{u}$. Daher wird diese Funktion nur über $o \in \{1, \ldots, 8\}$ und τ parametrisiert.

Position Die Position eines Pixels innerhalb der Bounding-Box $B(x, y, w, h)$ einer Person hat teilweise eine signifikante Korrelation mit der Körperteilklasse. So ist der Kopf meistens im oberen Drittel lokalisiert und zusammen mit der Orientierung können linke und rechte Körperteile unterschieden werden. Die Position der Koordinaten wird in Bezug zu den Ecken und des Schwerpunkts der Bounding-Box gesetzt:

$$\mathrm{pos}_{\mathit{left}}(\vec{u}) = u_x \tag{8.15}$$
$$\mathrm{pos}_{\mathit{right}}(\vec{u}) = w - u_x \tag{8.16}$$
$$\mathrm{pos}_{\mathit{top}}(\vec{u}) = u_y \tag{8.17}$$
$$\mathrm{pos}_{\mathit{bottom}}(\vec{u}) = h - u_y \tag{8.18}$$
$$\mathrm{pos}_{x}(\vec{u}) = u_x - w/2 \tag{8.19}$$
$$\mathrm{pos}_{y}(\vec{u}) = u_y - h/2 \tag{8.20}$$

Dabei repräsentieren (w,h) die Größe der Bounding-Box und $\vec{u} := (u_x, u_y)$ die Pixelkoordinate in Bezug auf die linke obere Ecke der Bounding-Box. Der Index $i \in \{\mathit{left}, \mathit{right}, \mathit{top}, \mathit{bottom}, x, y\}$ gibt an, welche Art der Berechnung durchgeführt wird. Die Entscheidungsfunktion ist dann wie folgt definiert:

$$\mathrm{h}_{\mathrm{pos}}(\vec{u}, \vec{\rho} = (i, \tau)) := \mathbb{I}\big[\,\mathrm{pos}_i(\vec{u}) > \tau\big] \tag{8.21}$$

8.4.2.5 Klassifikation der Orientierung

Die in Abschnitt 8.4.2.4 vorgestellten Merkmale (außer die Orientierung selbst) werden genutzt, um die Orientierung einer Person zu schätzen. Wie schon erwähnt, werden dabei acht Orientierungsklassen gebildet. Jede Klasse repräsentiert, von oben betrachtet, einen von acht gleichmäßig verteilten Winkelbereichen (siehe Abbildung 8.3). Die Orientierung wird in Bezug auf eine Person (Bounding-Box) bestimmt. Da es sich bei den hier genutzten *Random Forests* allerdings um pixelweise Klassifikatoren handelt, wird immer das zentrale Pixel der Bounding-Box als Zielpixel gewählt. Die Abstandvektoren $\vec{\delta}$ der Entscheidungsfunktionen erstrecken sich vom Zielpixel im gesamten Bereich der Bounding-Box.

8.4.2.6 Klassifikation der Körperteile

Die Klassifikation der Körperteile ist das Kernstück des hier vorgestellten Ansatzes. Auf der einen Seite sollte die Einteilung des Körpers so gestaltet sein, dass die modellierten Körperteile möglichst in sich unveränderlich sind (keine Gelenke u.ä.). Somit können starke Veränderungen in der Erscheinung eines Körperteiles vermieden werden. Auf der anderen Seite erlaubt die teilweise niedrige Auflösung der Spieler in Fußballaufzeichnungen nur eine grobe Aufteilung des Körpers. Das hier vorgestellte Modell teilt den Körper daher in Anlehnung an Hernández-Vela u. a. (Hernández-Vela u. a. 2012) in folgende 14 Klassen ein:

- Kopf
- Torso
- linker und rechter Oberarm / Unterarm
- linker und rechter Oberschenkel / Unterschenkel
- linke und rechte Hand
- linker und rechter Fuß

Für Pixel, die nicht zur Person gehören, gibt es eine zusätzliche Klasse für den Hintergrund.

Wie auch in (Shotton u. a. 2011) vorgeschlagen, wird für jedes einzelne Pixel die Körperteilklasse bestimmt. Für Entscheidungsfunktionen mit einem Abstandsvektor $\vec{\delta}$ wird dieser auf die nahe Umgebung des zu klassifizierenden Pixels beschränkt. Als Ergebnis wird für jedes Pixel und jede Klasse ermittelt, mit welcher Wahrscheinlichkeit das Pixel zu dieser Klasse gehört. Solche Wahrscheinlichkeitswerte sind in Abbildung 8.5 für ein Beispiel dargestellt.

Um die Robustheit der Klassifikation zu steigern, wird jedes Bild zweimal klassifiziert: als Original und als Bild, das an der vertikalen Mittelachse gespiegelt wurde. Die Ergebnisse beider Klassifikationen werden zusammengeführt, indem das Ergebnis im gespiegelten Bild zurücktransformiert wird und dabei linke und rechte Körperteilklassen vertauscht werden. Da sich aufgrund der Zufallsentscheidungen beim Training die Ergebnisse für das Originalbild und das gespiegelte Bild unterscheiden, hat dieses Vorgehen einen deutlichen Einfluss auf die Qualität der Klassifikation.

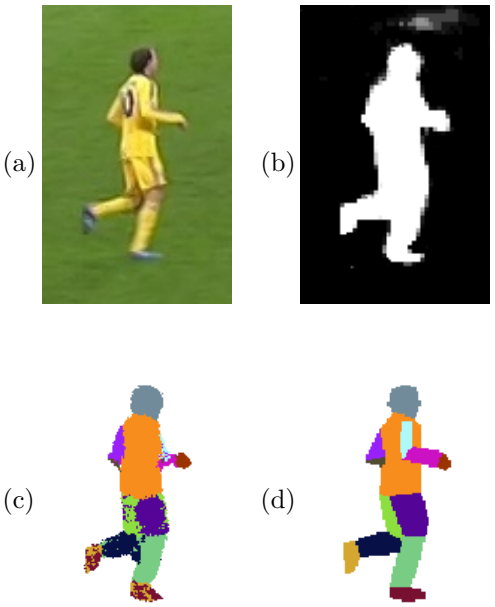


Abbildung 8.4: Klassifikation der Körperteile (Bigontina u. a. 2015): (a) Originalbild, (b) Silhouette von (Hoernig u. a. 2015), (c) geschätzte Körperteile, (d) annotierte Körperteile

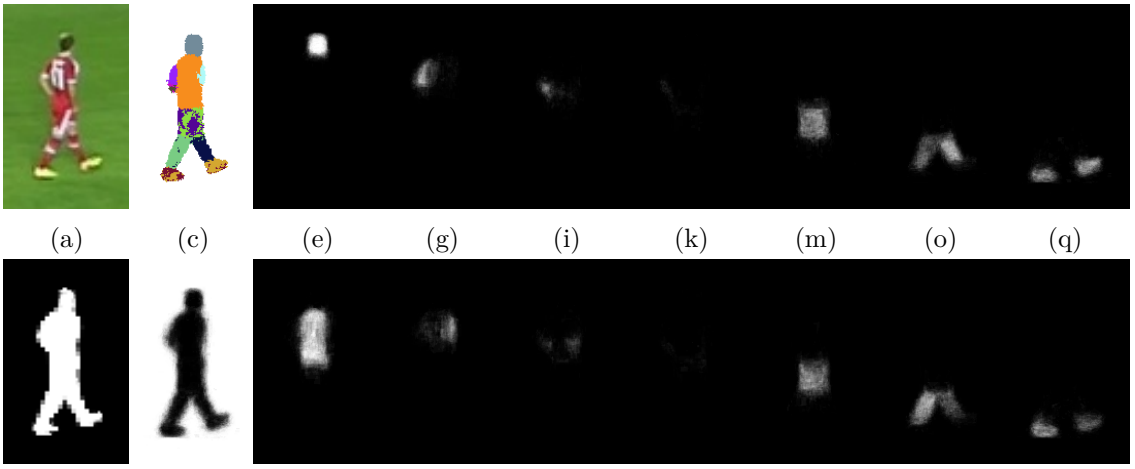


Abbildung 8.5: Wahrscheinlichkeiten der Körperteile: (a) Originalbild, (b) Silhouette, (c) Klassifikation und Wahrscheinlichkeiten für (d) Hintergrund, (e) Kopf, (f) Torso, (g, h) linker / rechter Oberarm, (i, j) linker / rechter Unterarm, (k, l) linke / rechte Hand, (m, n) linker / rechter Oberschenkel, (o, p) linker / rechter Unterschenkel, (q, r) linker / rechter Fuß

8.4.3 Anpassung der Pose

Die Klassifikation resultiert in Wahrscheinlichkeiten für jede Körperteilklasse, die jedem Pixel im Bild zugeordnet sind sowie in Wahrscheinlichkeit für die Orientierungsklassen, die sich auf das komplette Bild beziehen. Im ersten Schritt für die Schätzung der Pose

werden die Mittelpunkte der einzelnen Körperteile ermittelt. Danach wird das Modell eines 3D-Skeletts so angepasst, dass die Rückprojektion ins Bild möglichst gut den Beobachtungen entspricht.

8.4.3.1 Die Mittelpunkte der Körperteile

Ein Hauptproblem bei der Klassifikation ist es, dass häufig linke und rechte Körperteile verwechselt werden. Um diesen Effekt abzumildern, wird nach der Klassifikation das *k-means*-Verfahren (Lloyd 1982) mit $k = 2$ für Körperteile angewendet, die es als linkes und rechtes Körperteil gibt.

Die Besonderheit des hier vorgestellten Ansatzes ist es, dass die einzelnen Pixel bei der Berechnung der Cluster-Schwerpunkte gemäß ihrer Klassenwahrscheinlichkeit gewichtet werden. Seien $a_i \in \mathbb{R}^2$ und $b_i \in \mathbb{R}^2$ die ermittelten Schwerpunkte der Körperteilklassen c_a und c_b in Iteration i. Dann werden die neuen Schwerpunkte in Iteration $i+1$ durch einen gewichteten Mittelwert über alle Pixel $x \in \mathbb{R}^2$ berechnet:

$$a_{i+1} := \frac{\sum_x x \cdot p(c_a|x) \cdot m_i(x)}{\sum_x p(c_a|x) \cdot m_i(x)} \tag{8.22}$$

$$b_{i+1} = \frac{\sum_x x \cdot p(c_b|x) \cdot (1 - m_i(x))}{\sum_x p(c_b|x) \cdot (1 - m_i(x))} \tag{8.23}$$

mit

$$m_i(x) := \mathbb{I}\big[\|x - a_i\|_2 < \|x - b_i\|_2\big] \tag{8.24}$$

Dabei ist $p(c_a|x)$ die im vorhergehenden Schritt bestimmte Wahrscheinlichkeit der Körperteilklasse c_a des Pixels x. Da das Ergebnis sehr stark von der (randomisierten) Initialisierung abhängt, wird das Verfahren mehrmals durchgeführt und das beste Ergebnis (mit minimaler Summe der quadrierten Distanzen) behalten.

Für Körperteile ohne Gegenstück (wie der Torso oder der Kopf) wird der mit der Körperteilwahrscheinlichkeit gewichtete Schwerpunkt berechnet. Das gilt genauso für Körperteile, deren Gegenstück nicht erkannt wurde (das heißt eine Klassen-Wahrscheinlichkeit von Null im ganzen Bild aufweist).

Zuletzt wird sichergestellt, dass die Teile von kompletten Beinen und Armen konsistent (in Bezug auf links / rechts) zugeordnet sind, indem Mittelpunkte in unplausiblen Konstellationen vertauscht werden.

8.4.3.2 Skelettanpassung

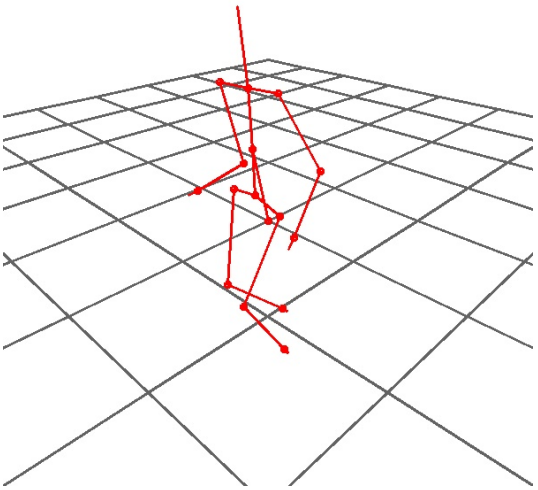


Abbildung 8.6: 3D-Skelettmodell (Bigontina 2014)

Im zweiten Schritt wird das 3D-Skelettmodell (siehe Abbildung 8.6) einer Person entsprechend dem Klassifikationsergebnis und der berechneten Mittelpunkte der Körperteile angepasst. Das Skelettmodell stellt inhärent wichtige Randbedingungen zur Verfügung: es legt fest, welche Körperteile miteinander verbunden sind und welchen Abstand verschiedene Gelenke zueinander haben.

Zunächst wird das Skelett entsprechend der klassifizierten Orientierung der Person gedreht. Danach wird das Skelett iterativ durch ein Verfahren der inversen Kinematik (siehe (Parent 2012)) angepasst, bis die Projektionen der Mittelpunkte der Körperteile des Skeletts möglichst gut mit den geschätzten Mittelpunkten der Klassifikation übereinstimmen. Das Vorgehen nutzt die Jacobi-Matrix für eine lineare Näherung der Änderung der Pose und versucht iterativ den quadrierten Abstand zwischen den projizierten und den geschätzten Mittelpunkten zu minimieren.

Das Modell besteht aus Gelenken (*Joints*) und Gliedern (*Links*), die eine kinematische Hierarchie bilden. Während die Glieder den Abstand zwischen zwei Gelenken vorgeben, beeinflussen Gelenke die Position aller nachfolgenden Glieder und Gelenke. Alle Skelettgelenke sind Rotationsgelenke mit drei Freiheitsgraden (den Rotationswinkeln). Zudem gibt es drei Basisgelenke für die Skalierung, die Translation und die Rotation des kompletten Skeletts.

Die aktuelle Pose des Skeletts in Iteration i ist gegeben durch die Gelenkparameter $\vec{\theta_i} \in \mathbb{R}^{46}$, die sich aus 42 Winkeln ($3 \cdot 14$), drei Translationsparametern und einen Skalierungsfaktor zusammen setzen. Veränderungen des Skeletts werden durch eine Funktion

$T\colon \mathbb{R}^{46} \to \mathbb{R}^{14}$ modelliert, die die Positionen der Glieder $\vec{p}_i\mathbb{R}14$ berechnet, wenn die Gelenkparameter $\vec{\theta}_i$ gegeben sind. Das Skelett wird iterativ angepasst indem die Gelenkparameter $\vec{\theta}_i$ so verändert werden, dass sich die Glieder in die Richtung der geschätzten Mittelpunkte $\vec{p}_G$ aus Abschnitt 8.4.3.1 bewegen. Die z-Koordinate (orthogonal zur Bildebene) dieser Mittelpunkte wird auf 0 gesetzt, um eine Verzerrung des Skelettes in die Tiefe zu vermeiden.

Änderungen in der Pose werden als lineare Näherung über die Jacobi-Matrix $J_T(\vec{\theta}_i) \in \mathbb{R}^{14\times 46}$ geschätzt, die analytisch berechnet werden kann. Ist die Änderung der Gelenkparameter $\Delta\vec{\theta}_i$ gegeben, so kann die Änderung der Pose $\Delta\vec{p}_i$ wie folgt angenähert werden:

$$\Delta\vec{p}_i := J_T(\vec{\theta}_i)\Delta\vec{\theta}_i \tag{8.25}$$

Durch Umformulieren von Gleichung 8.25 kann bei vorgegebener Änderung der Position die notwendige Anpassung der Gelenkparameter geschätzt werden:

$$\Delta\vec{\theta}_i := (J_T(\vec{\theta}_i))^{-1}\Delta\vec{p}_i \tag{8.26}$$

Die gewünschte Änderung der Position ist die Differenz $(\vec{p}_G - \vec{p}_i)$ zwischen der aktuellen Skelettposition $\vec{p}_i$ und der Zielposition $\vec{p}_G$. Durch Einsetzen in Gleichung 8.26 ergibt sich:

$$\Delta\Delta\vec{\theta}_i := \lambda((J_T(\vec{\theta}_i))^{-1}(\vec{p}_G - \vec{p}_i)) \tag{8.27}$$

Dabei ist $\lambda \in \mathbb{R}$ ein Parameter, der die Schrittweite vorgibt. Da $J_T(\vec{\theta}_i)$ nur sehr lokale Änderungen repräsentiert, kann ein zu groß gewähltes λ dazu führen, dass sich der zu minimierende Abstand vergrößert. Daher wird in jeder Iteration eine Liniensuche (*Line Search* (Nocedal und Wright 2006b, S. 30 ff.)) durchgeführt, um einen geeigneten Wert für λ zu finden.

Die Gelenkparameter können danach für die nächste Iteration aktualisiert werden:

$$\vec{\theta}_{i+1} := \vec{\theta}_i + \Delta\vec{\theta}_i \tag{8.28}$$

Als initiale Pose $\vec{\theta}_0$ wird eine realistische Pose mit leicht gebeugten Armen und Beinen gewählt. Da $J_T(\vec{\theta}_i)$ nicht quadratisch ist (und damit nicht invertierbar), kann die Umkehrfunktion $(J_T(\vec{\theta}_i))^{-1}$ lediglich durch die Pseudoinverse von $J_T(\vec{\theta}_i)$ angenähert werden.

Merkmale	Genauigkeit
Silhouette	51,4%
Silhouette + Haut	49,3%
Silhouette + HOG	40,7%
Silhouette + Gradienten	38,5%
Silhouette + RGB	35,0%
RGB	29,3%
Silhouette + Haar-ähnlich	28,6%

TABELLE 8.1: Genauigkeit der Klassifikation der Orientierung (korrekt klassifizierte Bilder)

In diesem Ansatz wird allerdings die effizient zu berechnende Transponierte $J_T(\vec{\theta}_i)^T$ bestimmt, da sich beweisen lässt, dass sich damit der Abstand auch verkleinern lässt (siehe beispielsweise (Aristidou und Lasenby 2009)).

8.4.4 Evaluierung

8.4.4.1 Datensatz

Die vorgestellte Methode wird anhand eines Datensatzes evaluiert, der 200 Bilder von Fußballspielern in unterschiedlichsten Posen umfasst. Dabei sind die Bilder nicht auf einzelne Spieler oder Mannschaften beschränkt. Sie decken eine große Diversität ab, wie zum Beispiel unterschiedliche Trikots, Aufnahmebedingungen und Geschlechter. Da hier nur die eigentliche Posenerkennung untersucht werden soll, sind die Positionen der Spieler (in Form von Bounding-Boxen) manuell vorgegeben und es wird auf ausgeschnittenen Bildern gearbeitet. Die Positionen der Spieler können durch das 2D-Spielertracking (siehe Abschnitt 4.4) erzeugt werden. Die Bilder stammen aus Fernsehaufnahmen und stellen das Verfahre vor typische Herausforderungen, wie beispielsweise schlechte Auflösung und Qualität oder Bewegungsunschärfe. Die Durchschnittsgröße der Bilder liegt bei 78×120 Pixeln. Jedes Bild wird allerdings auf eine Höhe von 200 Pixeln skaliert (bei gleichbleibendem Seitenverhältnis), um Abhängigkeiten von der Bildgröße zu vermeiden.

Die Körperteilklassen in den Bildern wurden pixelweise manuell annotiert und für jedes Bild wurde die Orientierung und die 2D-Position der Gelenke manuell vorgegeben. Der Datensatz wurde in 130 Trainings- und 70 Testbilder aufgeteilt. Zudem wurde für jedes Bild das horizontal gespiegelte Pendant hinzugefügt.

8.4.4.2 Klassifikation der Orientierung

Bei der Klassifikation der Orientierung der Person erzielte der *Random Forest* die besten Ergebnisse, der ausschließlich die Silhouette als Merkmal benutzt. Aus Tabelle 8.1 geht

Merkmale	Genauigkeit
Sil. + Ori. + Position + RGB	90,32%
Sil. + Ori. + Position + Gradienten	89,54%
Sil. + Ori. + Position + Haut	89,46%
Sil. + Ori. + Position + HOG	89,43%
Sil. + Ori. + Position + Haar-ähnlich	89,37%
Sil. + Ori. + Position	89,34%
Silhouette	87,13%
Silhouette + Orientierung	87,03%
Color	85.23%

TABELLE 8.2: Genauigkeit der Klassifikation der Körperteile (korrekt klassifizierte Pixel)

hervor, dass die Klassifikation der Orientierung eine große Herausforderung darstellt, da nur bei knapp über der Hälfte der Bilder (51,4%) die Orientierung richtig erkannt wurde. Dennoch ist diese Information ein hilfreiches Merkmal bei der Klassifikation der Körperteile und der Posenbestimmung. Zudem wird ein ratendes Verfahren, welches bei circa 12,5% liegen würde, deutlich übertroffen. Eine Erklärung für die durchwachsenen Ergebnisse könnte die Anzahl der Trainingsbeispiele sowie die eher willkürliche Wahl der diskreten Klassen sein. Während die 130 Bilder Millionen von Trainingspixel für die Körperteilklassifikation liefern, so sind es für das Einlernen der Orientierungsklassifikatoren nur 130 Beispiele.

8.4.4.3 Klassifikation der Körperteile

Die besten Ergebnisse bei der Klassifikation der Körperteile werden mit einer Kombination der Merkmale Silhouette, RGB-Daten, Position und Orientierung erzielt (siehe das Beispiel in Abbildung 8.4). Damit konnten 90,32% der Pixel im Testdatensatz richtig klassifiziert werden. Dabei gilt ein Pixel als richtig klassifiziert, wenn die Klasse mit der höchsten Wahrscheinlichkeit der annotierten Klasse entspricht. Wie aus Tabelle 8.2 entnommen werden kann, sind die Ergebnisse mit anderen Merkmalen und Merkmalskombinationen nicht signifikant schlechter. Es sei allerdings erwähnt, dass ein großer Teil der Bilder aus Hintergrund besteht und durch das Silhouetten-Merkmal schon sehr gut vorbestimmt ist. Mit einer einfachen Schwellwertoperation auf dem Silhouetten-Bild zur Erkennung der Hintergrundpixel und der Klassifizierung der restlichen Pixel als Torso werden schon 79,95% der Pixel korrekt klassifiziert. Dieses Basisresultat sollte von einem guten Verfahren deutlich übertroffen werden.

Der *Random Forest* mit dem besten Ergebnis enthielt 29 Entscheidungsbäume, die bis zu einer maximalen Tiefe von 20 beziehungsweise bis zu einer minimalen Anzahl von 92 Datenpunkten pro Blatt trainiert wurden. Jeder Entscheidungsbaum wurde auf einer

Methode	Kopf	Torso	Arme	Oben	Unten	Beine	Oben	Unten	Total
IUKS	85%	94%		38%	15%		59%	67%	54%
Yang / Ramanan (Parse-Datensatz)	91%	94%		43%	15%		39%	43%	47%
Yang / Ramanan (Fußballdatensatz)	99%	99%		67%	33%		64%	57%	64%

Tabelle 8.3: PCP

Methode	Kopf	Schulter	Ellbg.	Handgk.	Hüfte	Knie	Knöchel	Total
IUKS	91%	63%	56%	41%	71%	74%	66%	66%
Yang / Ramanan (Parse-Datensatz)	92%	65%	52%	35%	56%	46%	42%	56%
Yang / Ramanan (Fußballdatensatz)	100%	84%	75%	56%	86%	66%	59%	75%

Tabelle 8.4: PCK

zufällig ausgewählten Teilmenge der Trainingspixel trainiert. Wie allgemein üblich beim Einlernen von *Random Forests* wurde bei jedem trainierten Knoten nur eine zufällig ausgewählte Teilmenge der theoretisch möglichen Schwellwerte und anderen Parameter betrachtet. Die Umgebung eines Pixels, die betrachtet wird, ist auf 50 Pixel beschränkt (bei einer Bildhöhe von 200 Pixel). Bei genauerem Betrachten der Struktur dieses *Random Forests* fällt auf, dass die meisten Knoten das Silhouetten-Merkmal abfragen, während Orientierung, RGB und Position das Ergebnis nur geringfügig beeinflussen.

8.4.4.4 Anpassung der Pose

Das Endergebnis der hier vorgestellten Methode ist eine 3D-Pose (bzw. ihre Projektion ins 2D-Bild), welche aus Gelenken und Gliedern besteht. Für eine vergleichende Evaluierung wird der Ansatz von Y. Yang und Ramanan (Y. Yang und Ramanan 2013) mit dem von den Autoren zur Verfügung gestellten Quellcode (Y. Yang und Ramanan 2012) anhand des Fußballdatensatzes ausgewertet. Ihre Veröffentlichung beschäftigt sich auch mit Genauigkeitsmaßen bei der Posenbestimmung. Von den vorgestellten Maßen wurden für diese Evaluierung zwei ausgewählt: PCP und PCK.

PCP wurde von Ferrari u. a. (Ferrari u. a. 2008) eingeführt. Ein Körperteil gilt dabei als richtig platziert, wenn die beiden verbundenen Gelenke innerhalb eines Radius von 50% der Gliedlänge von der jeweiligen Referenzposition liegen. PCK wurde von Y. Yang und Ramanan (Y. Yang und Ramanan 2013) vorgestellt. Dabei gilt ein Gelenk als richtig platziert, wenn es innerhalb eines Radius von $\alpha \cdot \min(w,h)$ von der Referenzposition liegt, wobei w und h die Breite und Höhe des minimal umschließenden Rechtecks des Referenzskeletts sind. Wie von Y. Yang und Ramanan für Ganzkörperposen vorgeschlagen, wird $\alpha := 0{,}1$ gesetzt.

Methode	Kopf	Torso	Arme	Oben	Unten	Beine	Oben	Unten	Total
IUKS	85%	94%		53%	24%		72%	87%	65%
Yang / Ramanan (Parse-Datensatz)	91%	94%		63%	26%		59%	64%	61%
Yang / Ramanan (Fußballdatensatz)	99%	99%		69%	36%		70%	63%	67%

Tabelle 8.5: PCP (bei Billigung von Links-/Rechts-Vertauschungen)

Methode	Kopf	Schulter	Ellbg.	Handgk.	Hüfte	Knie	Knöchel	Total
IUKS	91%	85%	84%	63%	77%	89%	87%	82%
Yang / Ramanan (Parse-Datensatz)	92%	90%	72%	45%	70%	65%	61%	71%
Yang / Ramanan (Fußballdatensatz)	100%	85%	76%	60%	90%	72%	64%	78%

Tabelle 8.6: PCK (bei Billigung von Links-/Rechts-Vertauschungen)

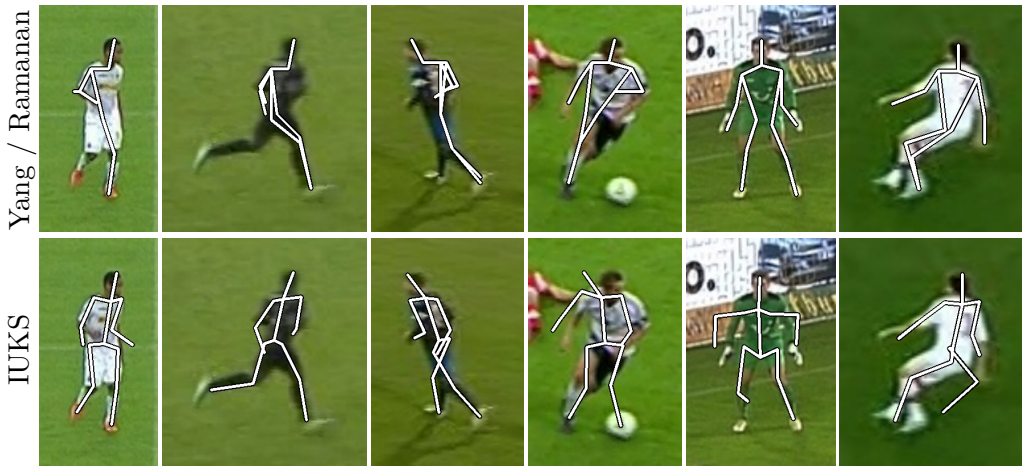


Abbildung 8.7: Der Ansatz von Y. Yang und Ramanan (Y. Yang und Ramanan 2013) tendiert dazu, einzelne Körperteile (insbesondere die Beine) doppelt zu zählen, während die vorgestellte Methode durch den Clusterschritt solche Probleme vermeidet. Bei gebückten Posen und bei zusätzlichen Objekten im Hintergrund (z.B. Feldlinien) schneidet hingegen das Verfahren von Y. Yang und Ramanan meist besser ab. Es sei zu beachten, dass im Modell von Y. Yang und Ramanan die Hüfte direkt mit den Schultern verbunden ist.

Im Folgenden wird die hier vorgestellte Methode mit IUKS bezeichnet. Die Ergebnisse bezüglich PCP sind aus Tabelle 8.3 zu entnehmen und bezüglich PCK aus Tabelle 8.4. Das Verfahren von Y. Yang und Ramanan wurde zum einen mit dem hier vorgestellten Fußballdatensatz sowie mit dem Parse Datensatz (Ramanan 2006) trainiert und evaluiert. Eine vergleichende Gegenüberstellung von Beispielergebnissen ist in Abbildung 8.7 dargestellt. Eine der Hauptschwächen der hier vorgestellten Methode ist die Unterscheidung von linken und rechten Körperteilen. So wurde beispielsweise bei dem Beispiel ganz links in Abbildung 8.7 nur ein PCK von 14,3% erreicht, da sowohl linker und rechter Arm als auch linkes und rechtes Bein verwechselt wurden. Das Verfahren von Y. Yang und Ramanan erzielte 57,1%, obwohl das linke Bein und der rechte Arm doppelt erkannt

wurden und das rechte Bein und der linke Arm gar nicht. Daher wurde eine zweite Evaluierung durchgeführt, bei der die Vertauschung von linken und rechten Körperteilen nicht als Fehler gewertet wurde. Die Ergebnisse sind den Tabellen 8.5 und 8.6 zu entnehmen.

Im Kontext des Fußballs ist es wichtig, die richtige Position der Körperteile (insbesondere der Beine) zu erkennen. Vertauschungen von linken und rechten Körperteilen können durch die zeitliche Information einer Videosequenz aufgelöst werden. In diesem Fall bietet die hier vorgestellte Methode eine Alternative mit einer Genauigkeit, die vergleichbar beziehungsweise besser als der Stand der Forschung ist und zudem durch die pixelweise Klassifikation auch noch hoch parallelisierbar ist.

8.4.5 Fazit und Ausblick

In diesem Abschnitt wurde gezeigt, dass eine akkurate pixelweise Körperteilklassifizierung mit monokularem Bildmaterial möglich ist, falls die Silhouette der Person gegeben ist und es wurden die aussagekräftigsten Merkmale für diese Aufgabe identifiziert. Die klassifizierten Körperteile werden genutzt, um die Pose der Person durch die Anpassung eines Skelettmodells zu schätzen. Die Genauigkeit der Ergebnisse kann sich dabei mit dem Stand der Technik messen.

Wie Sharp u. a. (Sharp u. a. 2008) gezeigt haben, können *Decision Forests* für eine GPU implementiert werden und pixelweise Klassifikation in Echtzeit durchführen (Shotton u. a. 2011). Der hier vorgestellte Ansatz hat somit Vorteile bezüglich der Berechnungszeit und kann mit einem effizienten Verfahren zur Skelettanpassung ein Verfahren zur Posenbestimmung in Echtzeit ermöglichen. Zukünftige Untersuchungen sollten das Potential der zeitlichen Information einer Videosequenz nutzen, um die bestehenden Probleme mit Vertauschungen von linken und rechten Körperteilen, Eigenüberdeckungen und Hintergrundobjekten abzumildern.

Kapitel 9

Diskussion und Ausblick

In den vorangegangenen Kapiteln der vorliegenden Arbeit wurden Verfahren vorgestellt und evaluiert, welche robust und echtzeitfähig die relevanten Objekte in monokularen Aufnahmen von Schwenk-Neige-Zoom-Kameras automatisch erkennen sowie ihre Positionen (2D/3D) bestimmen und nachführen. In Bezugnahme auf die Problemstellung aus Abschnitt 1.3 kann folgendes Fazit für die jeweiligen Teilprobleme gestellt werden:

- Eine gewichtete Kombination von aussagekräftigen Bildmerkmalen bildet eine Konfidenzkarte (siehe Abschnitt 3.7), welche die Grundlage für eine robuste Erkennung von Objekten im Bild ist, sowohl bei der Initialisierung (siehe Abschnitt 4.4.2) als auch während der Objektverfolgung (siehe Abschnitt 4.4.1).

- Die Objektverfolgung in 2D basiert auf einer stochastischen Filterung mit einem linearen Bewegungsmodell und normalverteiltem Prozess- und Messrauschen (siehe Abschnitt 4.3). Die somit bewusst gewählte niedrige Anzahl an Freiheitsgraden dient der Robustheit des Verfahren und kann dennoch durch eine balancierte Wahl der Modellparameter (siehe Abschnitt 5.2.1) nicht-lineare Elemente, wie schnelle Bewegungen von Objekten oder die Veränderung durch Schwenken, Neigen oder Zoomen der Kamera abdecken.

- Die vorgestellten Verfahren verfügen über die notwendige Robustheit, um mit Toleranzen und unterschiedlichsten Randbedingungen (wie etwas schlechte Wetterbedingungen oder ungewöhnliche Aufnahmebedingungen) umzugehen. Neben sorgfältig ausgewählten Bildmerkmalen, spielt dabei die Bestimmung geeigneter Parameterwerte eine entscheidende Rolle. Um die hohe Dimensionalität des Parameterraums zu beherrschen, wird auf Methoden zur automatischen Bestimmung mit Hilfe von repräsentativen Ground-Truth-Datensätzen zurückgegriffen (siehe

Kapitel 5). Ein entscheidender Punkt dabei ist, die Parameter für einzelne Komponenten der Verfahren separat zu bestimmen, da dadurch die Anzahl der möglichen Parameterkombinationen eingeschränkt werden kann.

- Die potentielle Echtzeitfähigkeit wird zum einem erreicht durch eine Konfidenzkarte aus Merkmalen, die parallelisierbar berechnet werden können und durch eine effiziente Nachbarschaftsuche von optimalen Positionen innerhalb der Konfidenzkarte (siehe Abschnitt 3.8). Zudem basiert die Objektverfolgung in 2D auf einem Online-Verfahren und bietet dadurch die Möglichkeit, Objektpositionen in Echtzeit auszugeben.

- Die vorgestellten Verfahren laufen autonom und bedürfen, insbesondere auf Grund der vollautomatischen, initialen Objekterkennung inklusive der Generierung von Farbtemplates (siehe Abschnitt 4.4.2), keiner weiteren manuellen Eingaben.

- In einem Nachbearbeitungsschritt werden, durch einfache Plausibilitätsannahmen der realen Welt (wie etwas physikalische Limits), die Verfolgungstrajektorien von der Bildebene in eine Ebene des Weltkoordinatensystems abgebildet und effizient bereinigt (siehe Abschnitt 4.5). Voraussetzung dafür ist eine geeignete Abbildungsvorschrift (vom Bild ins Weltkoordinatensystem), deren Bestimmung nicht Bestandteil der vorliegenden Arbeit ist.

- Die Robustheit und Praxistauglichkeit der Verfahren wurden mit Hilfe einer Evaluierung auf Basis von mehr als drei Stunden Videomaterial ausführlichst dokumentiert (siehe Kapitel 6). Durch einen Vergleich mit anderen Verfolgungsalgorithmen wurde die Verbesserung des Stands der Technik unter Beweis gestellt. Aufgrund der umfangreichen Verfügbarkeit von Video- und Evaluierungmaterial, hat sich dabei die Wahl der Anwendungsdomäne als vorteilhaft erwiesen.

In der Arbeit von Shaikh u. a. (Shaikh u. a. 2014, S. 8 ff.) werden diverse Herausforderungen bei der Objektverfolgung aufgeführt, die mit den Verfahren der vorliegenden Arbeit unter anderem wie folgt angegangen werden:

Änderung der Ausleuchtung / Störungen im Hintergrund

Durch ein dynamisches Hintergrundmodell mit integriertem Kontextwissen (z.B. Grasfarbe), welches von Einzelbild zu Einzelbild neu ermittelt wird, kann Änderungen der Ausleuchtung im Verlaufe einer Videosequenz effektiv begegnet werden. Probleme durch unterschiedliche Belichtungsbedingungen innerhalb eines Einzelbilds sowie durch Störungartefakte im Hintergrund, wie Sonne und Schatten, werden zusätzlich durch eine lokale Schwellwertbestimmung in der Hintergrundkonfidenzmaske abgeschwächt (siehe

Abschnitt 3.5). Des Weiteren werden bei der Objekterkennung und -verfolgung Merkmale eingesetzt, die eine domänen-unabhängige Allgemeingültigkeit aufweisen (Personendetektor, siehe Abschnitt 3.7.4) oder über mehrere Objekte einer Klasse (und ggf. über mehrere Einzelbilder) gemittelt werden (Farbtemplates, siehe Abschnitt 3.6) und somit der optischen Variabilität der Objekte entgegenwirken.

Dynamischer Hintergrund / Schnelle und abrupte Bewegungen

Wie oben erwähnt, können die vorgestellten Verfahren durch die bildweise Bestimmung des Hintergrunds mit Aufnahmen von dynamischen Kameras umgehen. Abrupte oder schnelle Bewegungen von einzelnen Objekten oder Kameraschwenks können den Bewegungsmustern der verfolgten Objekte einen starken Anteil an nicht-linearen Komponenten verleihen. Diese werden durch eine ausgewogene Modellierung des Prozessrauschens im linearen Bewegungsmodell abgefangen (siehe Abschnitt 5.2.1.1). In extremen Situationen kann es vorkommen, dass die Verfolgung von Objekten durch die starke Bewegungsunschärfe abgebrochen werden muss. Durch eine automatische Wiedererkennung wird diese bei einer Stabilisierung der Kameraposition oder der Objektbewegung jedoch fortgesetzt (siehe Abschnitt 4.4.1).

Überdeckungen

Die Behandlung von Überdeckungen ist eine der größten Herausforderung bei der Verfolgung von Objekten, insbesondere wenn sich mehrere Objekte in der Szene bewegen. Das Anwendungsszenario der vorliegenden Arbeit, die Auswertung von Aufnahmen von Sportbegegnungen, sorgt durch Ereignisse wie Zweikämpfe oder Manndeckung bei Standardsituationen für eine erhöhte Überdeckungswahrscheinlichkeit von zwei oder mehreren Objekten. Die in dieser Arbeit vorgestellten Verfahren behandeln Überdeckungen in erster Linie mit zwei Mechanismen:

- Objekte, die zu einer zusammenhängenden Region in der Vordergrundmaske verschmelzen, werden weiterhin als getrennte Objekte behandelt und bei der Berechnung und Optimierung der Konfidenz berücksichtigt (siehe Abschnitt 3.7). Da die meisten Überdeckungen in der Regel nicht vollständig sind (siehe Abbildung 3.8), sorgt die überlappungsbasierte Konfidenz bei teilweisen Überdeckungen dafür, dass weiterhin beide Objekte verfolgt werden und nicht das dominantere Objekt beide Verfolgungsvorgänge „anzieht“ (siehe Abschnitt 3.7.2).
- Falls durch eine vollständige Überdeckung die Verfolgung eines Objekts verloren geht, wird es nach deren Beendigung automatisch vom System wiedererkannt. Dabei wird durch eine Logik, die auf der Distanz und äußerlichen Ähnlichkeit zu verlorenen Objekten basiert, das Objekt reidentifiziert (siehe Abschnitt 4.4.4).

Initialisierung

Sowohl die Initialisierung des Hintergrundmodells als auch die initiale Erkennung der Objekte und Erstellung der Erscheinungsmodelle erfolgt automatisch. Bei letzterem sorgt ein iteratives Vorgehen mit einer Abschätzung der Objektgrößen für die notwendige Robustheit (siehe Abschnitte 4.4.2 und 3.4). Zusätzlich werden die Objekte automatisch nach ihrer Erscheinung eingeordnet und klassen-spezifische Modelle erstellt (z.B. für Mannschaften), ohne dass die Anzahl der verschiedenen Klassen bekannt ist (siehe Abschnitt 3.6).

In den vorangehenden Kapiteln wurde mehrfach die Eignung der vorgestellten Verfahren für andere Anwendungsdomänen angedeutet. Hier steht vor allem die geeignete Anpassung der Hintergrunderkennung im Vordergrund. Obwohl dies insbesondere bei anderen Sportspielen problemlos möglich ist (siehe das Beispiel Feldhockey in Abbildung 4.3), sind für eine Untermauerung der Aussage weitere empirische Auswertungen notwendig.

Ein kritisches Element der Objektverfolgung ist die Erkennung der Objekte. Nach den Erkenntnissen von Ben Shitrit u. a. (Ben Shitrit u. a. 2013), kann der zusätzliche Einsatz von domänen-spezifischen Objektdetektoren signifikante Vorteile bringen. Der Fortschritt der verfügbaren Rechenkapazitäten und die damit verbundene Möglichkeit, tiefe neuronale Netzwerke zu trainieren, haben in den letzten Jahren die Objekterkennung (und die Bildauswertung im Allgemeinen) revolutioniert (siehe beispielsweise He u. a. (He u. a. 2015)). Ein Problem dabei ist die Gefahr der Überanpassung solcher komplexen Modelle und der notwendige Umfang der Trainingsdaten, welche aufwendig manuell annotiert werden müssen.

In Bezug auf den Bereich der Sportspiele ist die Erstellung der Erscheinungsmodelle der Mannschaften ein kritischer Punkt. Dieser Modellbildung liegt das Problem einer Clusteranalyse zugrunde, ohne die Anzahl der Cluster zu kennen. Hier könnte sich der Einsatz von komplexeren Methoden der Merkmalsraumanalyse und des maschinellen Lernens vorteilhaft auswirken. Dies wäre mit einer detaillierten empirischen Auswertung zu belegen.

Die vorgestellten Verfahren nutzen vorliegende Abbildungen vom Bild- ins Weltkoordinatensystem aus, um die Trajektorien der Objekte zu bereinigen. Solche Abbildungen könne noch weiter ausgenutzt werden, beispielsweise um nach der Auflösung von Überdeckungen, Lücken in den Trajektorien zu schließen und die Reidentifizierung von wiedererkannten Objekten zu verbessern. Ein solches Vorgehen stellt ein kombinatorisches Problem dar, bei dem die beste Lösung in vielen Fällen nicht offensichtlich ist und leicht zu einer objektiven Verschlechterung der Ergebnisse sorgen kann. Zudem erhöht sich der potentielle Berechnungsaufwand und die Erstellung einer echtzeitfähigen Anwendung wird deutlich schwieriger.

Trotz automatisierten Verfahren wird es immer Situationen geben, in denen das System versagt und nicht das gewünschte Ergebnis liefert. Die Implementierung der vorgestellten Verfahren und die dazugehörige Webanwendung (siehe Abschnitt 8.2) bieten für solche Fälle rudimentäre Möglichkeiten für ein manuelles Eingreifen, beispielsweise durch das Verändern von essentiellen Parametern oder die Anpassung von Objektpositionen in einzelnen Frames, die vom Verfahren berücksichtigt werden. Die semi-automatische Analyse von Videomaterial ist kein triviales Problem, da möglichst wenige Eingriffe des Benutzers wünschenswert sind. Dafür sind Mechanismen notwendig, die ein Versagen des Systems automatisch erkennen und den Benutzer um Eingabe bitte und im Idealfall aus den Eingaben des Benutzers lernen.

Anhang A

Relevante Publikationen

A.1 Relevante Publikationen des Autors

Bestimmung der Grasmaske, Kapitel 3

- Hoernig, M., Herrmann, M. und Radig, B. (2013). „Real Time Soccer Field Analysis from Monocular TV Video Data“. In: *11th International Conference on Pattern Recognition and Image Analysis: New Information Technologies (PRIA-11-2013)*. Bd. II. Samara, S. 567–570.
- Hoernig, M., Herrmann, M. und Radig, B. (2015). „Real-Time Segmentation Methods for Monocular Soccer Videos“. In: *Pattern Recogntion and Image Analysis* 25 (2), S. 327–337.

Erkennung und Verfolgung von Spielern, Kapitel 3, 4 und 6

- Herrmann, M., Hoernig, M. und Radig, B. (2014). „Online Multi-player Tracking in Monocular Soccer Videos“. In: *AASRI Procedia* 8, S. 30–37.

Automatische Bestimmung von Parametern, Kapitel 5

- Herrmann, M., Mayer, C. und Radig, B. (2014). „Automatic Generation of Image Analysis Programs“. In: *Pattern Recognition and Image Analysis* 24 (3), S. 400–408.

Bestimmung des individuellen Ballbesitzes, Kapitel 8

- Hoernig, M., Link, D., Herrmann, M., Radig, B. und Lames, M. (2016). „Detection of Individual Ball Possession in Soccer“. In: *10th International Symposium on Computer Science in Sports (ISCSS)*. Hrsg. von Chung, P., Soltoggio, A., Dawson, C. W., Meng, Q. und Pain, M. Bd. 392. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, S. 103–107.

Bestimmung der Spielerposen, Kapitel 8

- Bigontina, A., Herrmann, M., Hoernig, M. und Radig, B. (2015). „Human Body Part Classification in Monocular Soccer Images“. In: *9th Open German-Russian Workshop on Pattern Recognition and Image Understanding (OGRW 2014)*. Hrsg. von Paulus, D., Fuchs, C. und Droege, D. Koblenz: University of Koblenz-Landau, S. 128–131.

A.2 Vom Autor betreute Arbeiten

- Brück, B. (2013). „Performante Körperposenschätzung anhand von 3D Body-Tracking“. Masterarbeit. München: Technische Universität München
- Bigontina, A. (2014). „Vision-based articulated body pose estimation of soccer players in monocular broadcast videos“. Masterarbeit. München: Technische Universität München
- Loipfinger, M. (2014). „Erkennung von Logos und Einblendungen in Fernsehübertragungen von Fußballspielen“. Bachelorarbeit. München: Technische Universität München
- Hopper, M. (2015). „Bildbasierte Erkennung des Balls in monokularen Aufnahmen von Fußballspielen“. Bachelorarbeit. München: Technische Universität München
- Dao, T. H. (2015). „Optical Ball Tracking in Monocular Image Sequences of Sport Games“. Masterarbeit. München: Teschnische Universität München

Anhang B

Wahl der Parameterwerte

Im Folgenden sind die gewählten Parameterwerte der Verfahren aus den Kapiteln 3 und 4 aufgeführt.

Schätzung der Spielergrößen

Die gewählten Werte der Parameter für Algorithmus 3.1 sind aus Tabelle B.1 zu entnehmen.

Parameter	**Wert**	**Bestimmung**
Untergrenze l_0	0,05	empirisch
Untergrenze l_1	-100	empirisch
Obergrenze u_0	0,6	empirisch
Obergrenze u_1	150	empirisch
Iterationslimit d	1e$-$5	empirisch
Konvergenzlimit ϵ	100	empirisch
Tuningkonstante a	4,685	empirisch

Tabelle B.1: Parameter für die Größenbestimmung

Extraktion der Spielersilhouetten

Die gewählten Werte der Parameter für Algorithmus 3.2 sind aus Tabelle B.2 zu entnehmen.

Bestimmung der dominanten Farben

Die gewählten Werte der Parameter für Algorithmus 3.3 sind aus Tabelle B.3 zu entnehmen.

Parameter	**Wert**	**Bestimmung**
Schwellwert τ_{Grass}	97	automatisch
Größe des Strukturelements r	20	automatisch
Toleranzfaktor s_x^l	0,333	automatisch
Toleranzfaktor s_x^u	2,478	automatisch
Toleranzfaktor s_y^l	0,144	automatisch
Toleranzfaktor s_y^u	2,400	automatisch
Dilatationsfaktor s_{dil}	0,117	automatisch

Tabelle B.2: Parameter für die Extraktion der Spielersilhouetten

Parameter	**Wert**	**Bestimmung**
Anzahl Vordergrundfarben k_{FG}	48	automatisch
Anzahl Iterationen Vordergrund i_{FG}	5	empirisch
Anzahl Grasfarben k_G	8	empirisch
Anzahl Iterationen Grass i_G	3	empirisch

Tabelle B.3: Parameter für die Bestimmung der dominanten Farben

Bestimmung der verschiedenen Outfit-Templates

Die gewählten Werte der Parameter für Algorithmus 3.4 sind aus Tabelle B.4 zu entnehmen.

Parameter	**Wert**	**Bestimmung**
Anzahl Iterationen n_i	100	empirisch
Maximale Anzahl Outfits k_{max}	7	empirisch
Abbruchkriterium t	0,4	automatisch

Tabelle B.4: Parameter für die Bestimmung der verschiedenen Outfit-Templates

Berechnung der Konfidenzkarte

Die gewählten Werte der Parameter aus Abschnitt 3.7 sind aus Tabelle B.5 zu entnehmen.

Parameter	**Wert**	**Bestimmung**
Texturbasierte Konfidenz Normalisierung u_D	$-2{,}449$	automatisch
Texturbasierte Konfidenz Normalisierung v_D	0,759	automatisch
Silhouettenbasierte Konfidenz Gewicht w_1	0,578	automatisch
Überlappungsbasierte Konfidenz Gewicht w_2	0,136	automatisch
Farbbasierte Konfidenz Gewicht w_3	0,854	automatisch
Texturbasierte Konfidenz Gewicht w_4	0,154	automatisch
Vorhersagebasierte Konfidenz Gewicht w_5	0,005	empirisch

Tabelle B.5: Parameter für die Berechnung der Konfidenzkarte

Messvorgang

Die gewählten Werte der Parameter aus Algorithmus 4.1 sind aus Tabelle B.6 zu entnehmen.

Parameter	**Wert**	**Bestimmung**
Faktor für die Residualfläche s_a	0,787	automatisch
Größentoleranz für Neu-Detektionen s_w	2,187	automatisch
Größentoleranz für Neu-Detektionen s_h	0,510	automatisch
Skalenwert für die Größenoptimierung s_{min}	0,9	empirisch
Skalenwert für die Größenoptimierung s_{max}	1,5	empirisch
Skalenwert für die Größenoptimierung s_{step}	0,1	empirisch

TABELLE B.6: Parameter für die Berechnung der Konfidenzkarte

Initiale Spielererkennung

Die gewählten Werte der Parameter aus Algorithmus 4.2 sind aus Tabelle B.7 zu entnehmen. Die Parameter für die Vordergrunderkennung sind in Tabelle B.2 aufgeführt.

Parameter	**Wert**	**Bestimmung**
Anzahl an Iterationen n	4	empirisch
Konfidenzschwellwert τ_{conf}	0,380	automatisch
Regionenschwellwerte für die Suche von Spielerkandidaten τ_{hl}	20	empirisch
Regionenschwellwerte für die Suche von Spielerkandidaten τ_{hu}	$0{,}333 \cdot H_I$	empirisch
Regionenschwellwerte für die Suche von Spielerkandidaten τ_{ar}	5,657	automatisch
Regionenschwellwerte für die Suche von Spielerkandidaten τ_{ϕ}	26,449	automatisch
Skalierungsparameter für die Personenerkennung s_{margin}	0,800	automatisch
Skalierungsparameter für die Personenerkennung s_{scale}	1,17292	automatisch

TABELLE B.7: Parameter für die Berechnung der Konfidenzkarte, wobei H_I die Höhe des Bilds darstellt.

2D-Spielerverfolgung

Im Folgenden sind die Parameterwerte für die Verfahren aus Abschnitt 4.4.4 aufgeführt. Die Begründung für die Wahl ist in Abschnitt 5.2.1 zu finden.

$$\overline{h} := 61 \tag{B.1}$$

$$q_p := 100 \cdot \frac{h}{\overline{h}} \tag{B.2}$$

$$q_s := 7{,}5 \cdot \frac{h}{\overline{h}}. \tag{B.3}$$

$$r_p := 3 \cdot \frac{1}{q} \cdot \frac{h}{\overline{h}} \tag{B.4}$$

$$r_s := 20 \cdot \frac{1}{q} \cdot \frac{h}{\overline{h}}. \tag{B.5}$$

$$\tau_{accept} := 0{,}75; \quad \tau_{reject} := 0{,}35; \quad N_{reject} := 2 \tag{B.6}$$

$$N_{label} := 10 \tag{B.7}$$

$$\tau_{ovlp} := 0{,}7; \quad N_{ovlp} := 5; \quad \tau_{meas} := 0{,}15. \tag{B.8}$$

Anhang C

Unterschiede bei Implementierungen von CLEAR-MOT

Zur Verdeutlichung, welchen Einfluss die Implementierung der Evaluierungsmetrik auf das Ergebnis haben kann, sind im Folgenden die Resultate verschiedener Implementierungen für eine Reihe von sehr einfachen Trackingszenarien aufgeführt. Es wurden dabei die folgenden Softwarepakete verwendet:

- PyMOT (Roth u. a. 2014) (Bernardin und Stiefelhagen 2008)
- motutils (Milan 2015) (Milan u. a. 2013)
- Multiple Object Tracking Benchmark Development Kit (Milan, Leal-Taixé, Schindler u. a. 2015a; Milan, Leal-Taixé, Schindler u. a. 2015b; Leal-Taixé u. a. 2015)
- CLEAR-MOT (Masi und Lisanti 2014) (Bagdanov u. a. 2012)
- Multi-target tracking evaluation tool (B. Yang u. a. 2008) (B. Yang und Nevatia 2012) (Li, Huang u. a. 2009)

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	5	20	10	-	-	-	-	0.22	0.56
Motutils	5	20	8	3	1	2	0	0.27	1.0
Devkit	5	20	8	3	1	2	0	0.27	1.0
Masi	5	20	8	-	-	-	-	0.27	1.0
Yang	5	20	0	0	1	2	0	-	-

Tabelle C.1: Beispiel 1a

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	30	120	10	-	-	-	-	0.41	0.93
Motutils	30	120	8	3	1	2	0	0.42	1.0
Devkit	30	120	8	3	1	2	0	0.42	1.0
Masi	30	120	8	-	-	-	-	0.42	1.0
Yang	30	120	10	9	1	2	0	-	-

Tabelle C.2: Beispiel 1b

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	0	15	2	-	-	-	-	0.43	0.87
Motutils	0	15	0	1	0	2	0	0.5	1.0
Devkit	0	15	0	1	0	2	0	0.5	1.0
Masi	0	15	0	-	-	-	-	0.5	1.0
Yang	0	15	1	0	0	2	0	-	-

Tabelle C.3: Beispiel 2a

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	0	90	2	-	-	-	-	0.49	0.98
Motutils	0	90	0	1	0	2	0	0.5	1.0
Devkit	0	90	0	1	0	2	0	0.5	1.0
Masi	0	90	0	-	-	-	-	0.5	1.0
Yang	0	90	2	1	0	2	0	-	-

Tabelle C.4: Beispiel 2b

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	15	0	2	-	-	-	-	-0.13	0.87
Motutils	15	0	2	0	1	0	0	-0.13	1.0
Devkit	15	0	2	0	1	0	0	-0.13	1.0
Masi	15	0	2	-	-	-	-	-0.13	1.0
Yang	15	0	0	1	1	0	0	-	-

Tabelle C.5: Beispiel 3a

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	90	0	2	-	-	-	-	-0.02	0.98
Motutils	90	0	2	0	1	0	0	-0.02	1.0
Devkit	90	0	2	0	1	0	0	-0.02	1.0
Masi	90	0	2	-	-	-	-	-0.02	1.0
Yang	90	0	0	2	1	0	0	-	-

Tabelle C.6: Beispiel 3b

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	17	2	2	-	-	-	-	-0.4	0.71
Motutils	18	3	3	2	0	1	0	-0.6	1.0
Devkit	17	2	5	2	1	0	0	-0.6	0.92
Masi	17	2	5	-	-	-	-	-0.6	1.0
Yang	17	2	0	0	1	0	0	-	-

Tabelle C.7: Beispiel 4a

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	102	12	2	-	-	-	-	-0.29	0.84
Motutils	108	18	3	2	0	1	0	-0.41	1.0
Devkit	102	12	5	2	1	0	0	-0.32	0.92
Masi	102	12	5	-	-	-	-	-0.32	0.98
Yang	102	12	0	5	1	0	0	-	-

Tabelle C.8: Beispiel 4b

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	0	0	0	-	-	-	-	1.0	0.63
Motutils	15	15	0	13	0	2	0	0.0	1.0
Devkit	0	0	0	0	2	0	0	1.0	0.23
Masi	8	8	14	-	-	-	-	0.0	1.0
Yang	8	8	0	0	1	1	0	-	-

Tabelle C.9: Beispiel 5a

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	0	0	0	-	-	-	-	1.0	0.64
Motutils	90	90	0	13	0	2	0	0.0	1.0
Devkit	0	0	0	0	2	0	0	1.0	0.23
Masi	48	48	14	-	-	-	-	0.39	0.91
Yang	48	48	14	20	1	1	0	-	-

Tabelle C.10: Beispiel 5b

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	12	5	3	-	-	-	-	0.29	0.91
Motutils	12	5	2	4	1	1	0	0.38	1.0
Devkit	12	5	2	4	1	1	0	0.32	1.0
Masi	12	5	2	-	-	-	-	0.32	1.0
Yang	12	5	0	0	1	1	0	-	-

Tabelle C.11: Beispiel 6a

Skript	FP	FN	IDS	FM	MT	PT	ML	MOTA	MOTP
PyMOT	72	30	3	-	-	-	-	0.38	0.98
Motutils	72	30	2	4	1	1	0	0.38	1.0
Devkit	72	30	2	4	1	1	0	0.38	1.0
Masi	72	30	2	-	-	-	-	0.38	1.0
Yang	72	30	0	6	1	1	0	-	-

Tabelle C.12: Beispiel 6b

Literatur

Andreakis, A., Hoyningen-Huene, N. v. und Beetz, M. (2009). „Incremental Unsupervised Time Series Analysis Using Merge Growing Neural Gas“. In: *Advances in Self-Organizing Maps*. Hrsg. von Príncipe, J. C. und Miikkulainen, R. Bearb. von Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F. u. a. Bd. 5629. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 10–18.

Andriluka, M., Roth, S. und Schiele, B. (2008). „People-tracking-by-detection and people-detection-by-tracking“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 1–8.

Andriluka, M., Roth, S. und Schiele, B. (2009). „Pictorial structures revisited: People detection and articulated pose estimation“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Hrsg. von Huttenlocher, D., Medioni, G. und Rehg, J. Miami, FL: IEEE, S. 1014–1021.

Andriluka, M., Roth, S. und Schiele, B. (2010). „Monocular 3D pose estimation and tracking by detection“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 623–630.

Andriluka, M., Roth, S. und Schiele, B. (2012). „Discriminative Appearance Models for Pictorial Structures“. In: *International Journal of Computer Vision* 99 (3), S. 259–280.

Aristidou, A. und Lasenby, J. (2009). *Inverse kinematics: a review of existing techniques and introduction of a new fast iterative solver*. Technical Report 632. University of Cambridge, Department of Engineering.

Arthur, D. und Vassilvitskii, S. (2007). „k-means++: the Advantages of Careful Seeding“. In: *18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SODA '07. Philadelphia, PA, USA: Society for Industrial und Applied Mathematics, S. 1027–1035.

Arulampalam, M., Maskell, S., Gordon, N. und Clapp, T. (2002). „A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking“. In: *IEEE Transactions on Signal Processing* 50 (2), S. 174–188.

Bagdanov, A. D., Bimbo, A. D., Dini, F., Lisanti, G. und Masi, I. (2012). „Posterity Logging of Face Imagery for Video Surveillance“. In: *IEEE MultiMedia* 19 (4), S. 48–59.

Beetz, M., Bandouch, J., Gedikli, S., Hoyningen-Huene, N. v., Kirchlechner, B. u. a. (2006). „Camera-based Observation of Football Games for Analyzing Multi-agent Activities“. In: *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, S. 42–49.

Beetz, M., Gedikli, S., Bandouch, J., Kirchlechner, B., Hoyningen-Huene, N. v. u. a. (2007). „Visually Tracking Football Games Based on TV Broadcasts“. In: *Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, S. 2066–2071.

Beetz, M., Hoyningen-Huene, N. v., Kirchlechner, B., Gedikli, S., Siles, F. u. a. (2009). „Aspogamo: Automated sports game analysis models“. In: *International Journal of Computer Science in Sport*.

Bellman, R. E. (1961). *Adaptive control processes: A Guided Tour*. Princeton University Pres.

Belongie, S., Malik, J. und Puzicha, J. (2001). „Shape context: A new descriptor for shape matching and object recognition“. In: *Advances in Neural Information Processing Systems 13 (NIPS 2000)*. Hrsg. von Leen, T. K., Dietterich, T. G. und Tresp, V. MIT Press, S. 831–837.

Ben Shitrit, H., Berclaz, J., Fleuret, F. und Fua, P. (2011). „Tracking multiple people under global appearance constraints“. In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, S. 137–144.

Ben Shitrit, H., Raca, M., Fleuret, F. und Fua, P. (2013). „Tracking Multiple Players using a Single Camera“. In: *Submitted to: Machnie Vision and Applications*.

Benfold, B. und Reid, I. (2011). „Stable Multi-Target Tracking in Real-Time Surveillance Video“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 3457–3464.

Berclaz, J., Fleuret, F., Turetken, E. und Fua, P. (2011). „Multiple Object Tracking Using K-Shortest Paths Optimization“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (9), S. 1806–1819.

Bergstra, J. und Bengio, Y. (2012). „Random Search for Hyper-Parameter Optimization“. In: *Journal of Machine Learning Research* 13 (1), S. 281–305.

Bernardin, K. und Stiefelhagen, R. (2008). „Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics“. In: *EURASIP Journal on Image and Video Processing* 2008, S. 1–10.

Bigontina, A. (2014). „Vision-based articulated body pose estimation of soccer players in monocular broadcast videos“. Masterarbeit. München: Technische Universität München.

Bigontina, A., Herrmann, M., Hoernig, M. und Radig, B. (2015). „Human Body Part Classification in Monocular Soccer Images“. In: *9th Open German-Russian Workshop on Pattern Recognition and Image Understanding (OGRW 2014)*. Hrsg. von Paulus, D., Fuchs, C. und Droege, D. Koblenz: University of Koblenz-Landau, S. 128–131.

Birbach, O. und Frese, U. (2009). „A Multiple Hypothesis Approach for a Ball Tracking System“. In: *Computer Vision Systems*. Hrsg. von Fritz, M., Schiele, B. und Piater, J. H. Bearb. von Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F. u. a. Bd. 5815. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 435–444.

Birchfield, S. T. und Rangarajan, S. (2005). „Spatiograms versus Histograms for Region-Based Tracking“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Bd. 2. IEEE, S. 1158–1163.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Information science and statistics. New York: Springer. 738 S.

Biurrun, D. und Barbato, L. (2015). *libav*. Version 0.8.17. URL: https://libav.org (besucht am 24. 11. 2016).

Blackman, S. S. und Popoli, R. (1999). *Design and analysis of modern tracking systems*. Artech House radar library. Boston: Artech House. 1230 S.

Bochkanov, S. (2014). *ALGLIB*. Version 3.9.0. URL: http://www.alglib.net (besucht am 23. 02. 2016).

Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E. und Van Gool, L. (2011). „Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (9), S. 1820–1833.

Brück, B. (2013). „Performante Körperposenschätzung anhand von 3D Body-Tracking“. Masterarbeit. München: Technische Universität München.

Burghouts, G. J. und Geusebroek, J.-M. (2009). „Performance evaluation of local colour invariants“. In: *Computer Vision and Image Understanding* 113 (1), S. 48–62.

Butt, A. A. und Collins, R. T. (2013). „Multi-target Tracking by Lagrangian Relaxation to Min-cost Network Flow“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 1846–1853.

Canny, J. (1986). „A Computational Approach to Edge Detection“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8 (6), S. 679–698.

Canon Deutschland GmbH (2016). *Canon XF300*. Canon XF300. URL: http://www.canon.de/for_home/product_finder/camcorders/professional/xf300/ (besucht am 23. 02. 2016).

Carr, P., Sheikh, Y. und Matthews, I. (2012). „Monocular Object Detection Using 3D Geometric Primitives“. In: *Computer Vision – ECCV 2012*. Hrsg. von Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y. und Schmid, C. Bearb. von Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F. u. a. Bd. 7572. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 864–878.

Cha, S.-H. (2007). „Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions“. In: *International Journal of Mathematical Models and Methods in Applied Sciences* 1 (4), S. 300–307.

Chen, X. und Yuille, A. L. (2014). „Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations“. In: *Advances in Neural Information Processing Systems 27 (NIPS 2014)*. Hrsg. von Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D. und Weinberger, K. Q. Curran Associates, Inc., S. 1736–1744.

Choi, K. und Seo, Y. (2005). „Tracking Soccer Ball in TV Broadcast Video“. In: *Image Analysis and Processing – ICIAP 2005*. Hrsg. von Roli, F. und Vitulano, S. Bearb. von Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F. u. a. Bd. 3617. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 661–668.

Choi, W., Pantofaru, C. und Savarese, S. (2013). „A General Framework for Tracking Multiple People from a Moving Camera“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (7), S. 1577–1591.

ChyronHego Corporation (2016). *TRACAB Optical Tracking*. ChyronHego. URL: http://chyronhego.com/sports-data/tracab (besucht am 21. 05. 2016).

Collins, R. T. (2012). „Multitarget data association with higher-order motion models“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 1744–1751.

Comaniciu, D. und Meer, P. (2002). „Mean Shift : A Robust Approach Toward Feature Space Analysis“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (5), S. 603–619.

Comaniciu, D., Ramesh, V. und Meer, P. (2003). „Kernel-based Object Tracking“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (5), S. 564–577.

Cooper, G. F. und Herskovits, E. (1992). „A Bayesian method for the induction of probabilistic networks from data“. In: *Machine Learning* 9 (4), S. 309–347.

Criminisi, A., Shotton, J. und Konukoglu, E. (2011). „Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning“. In: *Foundations and Trends in Computer Graphics and Vision* 7 (2), S. 81–227.

Dalal, N. (2006). „Finding People in Images and Videos“. Dissertation. Grenoble: Institut National Polytechnique de Grenoble.

Dalal, N. und Triggs, B. (2005). „Histograms of Oriented Gradients for Human Detection“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Bd. 1. IEEE, S. 886–893.

Dao, T. H. (2015). „Optical Ball Tracking in Monocular Image Sequences of Sport Games“. Masterarbeit. München: Teschnische Universität München.

Das Erste (2012). *DFB-Länderspiel: Niederlande - Deutschland*. Ausgestrahlt am 14.11.2012 um 20:30 Uhr.

Das Erste (2014). *Fußball-Weltmeisterschaft 2014: Deutschland - Argentinien*. Ausgestrahlt am 13.07.2012 um 21:00 Uhr.

Dawes, B., Niebler, E., Rivera, R., James, D., Prus, V. u. a. (2012). *Boost C++ Libraries*. Version 1.51.0. URL: http://www.boost.org/ (besucht am 24. 11. 2016).

Deza, E. und Deza, M.-M. (2006). *Dictionary of distances*. Amsterdam: Elsevier. 391 S.

DFL (2014a). *Scoutingfeed FC Augsburg - SC Paderborn 07*. Aufgezeichnet am 08.11.2014 um 15:30 Uhr.

DFL (2014b). *Scoutingfeed FC Bayern München - Bayer 04 Leverkusen*. Aufgezeichnet am 01.03.2014 um 18:30 Uhr.

DFL (2014c). *Scoutingfeed FC Bayern München - FC Schalke 04*. Aufgezeichnet am 01.03.2014 um 18:30 Uhr.

DFL (2014d). *Scoutingfeed Hertha BSC - Borussia Dortmund*. Aufgezeichnet am 10.05.2014 um 15:30 Uhr.

Divvala, S., Hoiem, D., Hays, J., Efros, A. und Hebert, M. (2009). „An empirical study of context in object detection“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 1271–1278.

Dollár, P. (2014). *Piotr's Computer Vision Matlab Toolbox (PMT)*. Version 3.40. URL: http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html (besucht am 13. 07. 2015).

Dollár, P., Appel, R., Belongie, S. und Perona, P. (2014). „Fast Feature Pyramids for Object Detection“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (8), S. 1532–1545.

Dollár, P., Appel, R. und Kienzle, W. (2012). „Crosstalk Cascades for Frame-Rate Pedestrian Detection“. In: *Computer Vision – ECCV 2012*. Hrsg. von Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y. und Schmid, C. Bearb. von Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F. u. a. Bd. 7573. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, S. 645–659.

Dollár, P., Tu, Z., Perona, P. und Belongie, S. (2009). „Integral Channel Features“. In: *British Machine Vision Conference (BMVC)*. BMVA Press, S. 91.1–91.11.

Dollár, P., Wojek, C., Schiele, B. und Perona, P. (2012). „Pedestrian Detection: An Evaluation of the State of the Art“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (4), S. 743–761.

D'Orazio, T., Guaragnella, C., Leo, M. und Distante, A. (2004). „A new algorithm for ball recognition using circle Hough transform and neural classifier“. In: *Pattern Recognition* 37 (3), S. 393–408.

D'Orazio, T. und Leo, M. (2010). „A review of vision-based systems for soccer video analysis“. In: *Pattern Recognition* 43 (8), S. 2911–2926.

D'Orazio, T., Leo, M., Mosca, N., Spagnolo, P. und Mazzeo, P. L. (2009a). „A Semi-automatic System for Ground Truth Generation of Soccer Video Sequences“. In: *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, S. 559–564.

D'Orazio, T., Leo, M., Mosca, N., Spagnolo, P. und Mazzeo, P. L. (2009b). *ISSIA-CNR Soccer Dataset*. URL: http://www.issia.cnr.it/wp/?portfolio=operation-agreement-cnr-figc-2 (besucht am 23. 02. 2016).

Doucet, A. und Johansen, A. M. (2011). „A Tutorial on Particle Filtering and Smoothing: Fifteen years Later“. In: *The Oxford Handbook of Nonlinear Filtering*. New York: Oxford University Press, S. 656–704.

Durus, M. (2014). „Ball Tracking and Action Recognition of Soccer Players in TV Broadcast Videos“. Dissertation. München: Technische Universität München.

Ekin, A., Tekalp, A. und Mehrotra, R. (2003). „Automatic soccer video analysis and summarization“. In: *IEEE Transactions on Image Processing* 12 (7), S. 796–807.

Ellis, A. und Ferryman, J. (2010). „PETS2010 and PETS2009 Evaluation of Results Using Individual Ground Truthed Single Views“. In: *Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, S. 135–142.

Eurosport (2013a). *Fußball-Afrikameisterschaft 2013: Burkina Faso - Ghana*. Ausgestrahlt am 06.02.2013 um 20:30 Uhr.

Eurosport (2013b). *U-17-Fußball-Europameisterschaften der Frauen 2013: Polen - Schweden*. Ausgestrahlt am 28.06.2013 um 14:30 Uhr.

Eurosport (2013c). *U-19-Fußball-Europameisterschaft 2013: Spanien - Frankreich*. Ausgestrahlt am 29.07.2013 um 20:00 Uhr.

Eurosport (2013d). *WM-Qualifikation Asien 2011/2013: Australien - Irak*. Ausgestrahlt am 18.06.2013 um 11:30 Uhr.

Everingham, M., Gool, L. v., Williams, C. K. I., Winn, J. und Zisserman, A. (2010). „The PASCAL Visual Object Classes (VOC) Challenge“. In: *International Journal of Computer Vision* 88 (2), S. 303–338.

Everingham, M., Eslami, S. M. A., Gool, L. V., Williams, C. K. I., Winn, J. u. a. (2015). „The Pascal Visual Object Classes Challenge: A Retrospective“. In: *International Journal of Computer Vision* 111 (1), S. 98–136.

Everingham, M., Gool, L. v., Williams, C., Winn, J. und Zisserman, A. (2007). *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Annotation Guidelines*. URL: http://host.robots.ox.ac.uk:8080/pascal/VOC/voc2007/guidelines.html (besucht am 23. 02. 2016).

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. und Lin, C.-J. (2008). „LIBLINEAR: A Library for Large Linear Classification“. In: *The Journal of Machine Learning Research* 9 (Aug), S. 1871–1874.

Fédération Internationale de Football Association (FIFA) (2015a). *FIFA Fussball-WM 2014: 3,2 Milliarden Zuschauer, 1 Milliarde beim Finale*. URL: http://de.fifa.com/worldfootball/bigcount/ (besucht am 23. 02. 2016).

Fédération Internationale de Football Association (FIFA) (2015b). *Spielregeln 2015/2016*. URL: http://de.fifa.com/development/education-and-technical/referees/laws-of-the-game.html (besucht am 23. 02. 2016).

Fédération Internationale de Hockey (FIH) (2014). *Feldhockey Champions Trophy der Herren 2014: Deutschland - Pakistan*. URL: https://www.youtube.com/watch?v=Aym9v1Vz5Fk (besucht am 28. 01. 2017).

Fei, Y., Christmas, W. und Kittler, J. (2008). „Layered Data Association Using Graph-Theoretic Formulation with Application to Tennis Ball Tracking in Monocular Sequences". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (10), S. 1814–1830.

Felzenszwalb, P., McAllester, D. und Ramanan, D. (2008). „A discriminatively trained, multiscale, deformable part model". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. und Ramanan, D. (2010). „Object Detection with Discriminatively Trained Part-Based Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (9), S. 1627–1645.

Felzenszwalb, P. F. und Huttenlocher, D. P. (2005). „Pictorial Structures for Object Recognition". In: *International Journal of Computer Vision* 61 (1), S. 55–79.

Feng, Y. und Hamerly, G. (2007). „PG-means: learning the number of clusters in data". In: *Advances in Neural Information Processing Systems 19 (NIPS 2006)*. Hrsg. von Schölkopf, B., Platt, J. C. und Hoffman, T. MIT Press, S. 393–400.

Ferrari, V., Marin-Jimenez, M. und Zisserman, A. (2008). „Progressive search space reduction for human pose estimation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 1–8.

Figueroa, P. J., Leite, N. J. und Barros, R. M. (2006a). „Background recovering in outdoor image sequences: An example of soccer players segmentation". In: *Image and Vision Computing* 24 (4), S. 363–374.

Figueroa, P. J., Leite, N. J. und Barros, R. M. (2006b). „Tracking soccer players aiming their kinematical motion analysis". In: *Computer Vision and Image Understanding* 101 (2), S. 122–135.

Fischler, M. A. und Elschlager, R. A. (1973). „The Representation and Matching of Pictorial Structures". In: *IEEE Transactions on Computers* C-22 (1), S. 67–92.

Fleuret, F., Berclaz, J., Lengagne, R. und Fua, P. (2008). „Multicamera People Tracking with a Probabilistic Occupancy Map". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2), S. 267–282.

Forney, G. D. (1973). „The viterbi algorithm". In: *Proceedings of the IEEE* 61 (3), S. 268–278.

Freund, Y. und Schapire, R. E. (1996). „Experiments with a New Boosting Algorithm“. In: *Thirteenth International Conference on Machine Learning (ICML)*. Bari, Italy, S. 148–156.

Freund, Y. und Schapire, R. E. (1997). „A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting“. In: *Journal of Computer and System Sciences* 55 (1), S. 119–139.

Friedman, J., Hastie, T. und Tibshirani, R. (2000). „Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)“. In: *The annals of statistics* 28 (2), S. 337–407.

Fritzke, B. (1995). „A Growing Neural Gas Network Learns Topologies“. In: *Advances in Neural Information Processing Systems 7 (NIPS 1994)*. Hrsg. von Tesauro, G., Touretzky, D. S. und Leen, T. K. MIT Press, S. 625–632.

Gedikli, S. (2009). „Continual and Robust Estimation of Camera Parameters in Broadcasted Sports Games“. Dissertation. München: Technische Universität München.

Gedikli, S., Bandouch, J., Hoyningen-Huene, N. v., Kirchlechner, B. und Beetz, M. (2007). „An Adaptive Vision System for Tracking Soccer Players from Variable Camera Settings“. In: *5th International Conference on Computer Vision Systems (ICVS)*. Bielefeld.

Gerke, S., Singh, S., Linnemann, A. und Ndjiki-Nya, P. (2013). „Unsupervised Color Classifier Training for Soccer Player Detection“. In: *Visual Communications and Image Processing (VCIP)*. IEEE, S. 1–5.

Gervautz, M. und Purgathofer, W. (1988). „A Simple Method for Color Quantization: Octree Quantization“. In: *New Trends in Computer Graphics*. Hrsg. von Magnenat-Thalmann, N. und Thalmann, D. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 219–231.

Girshick, R. B., Felzenszwalb, P. F. und McAllester, D. (2009). *Discriminatively Trained Deformable Part Models, Release 3.1*. Version 3.1. URL: http://cs.brown.edu/~pff/latent-release3/ (besucht am 23. 02. 2016).

Gomez, G. und Morales, E. F. (2002). „Automatic feature construction and a simple rule induction algorithm for skin detection“. In: *Nineteenth Internation Conference on Machine Learning (ICML) Workshop on Machine Learning in Computer Vision*, S. 5.1–5.8.

Hamerly, G. und Elkan, C. (2004). „Learning the k in k-means“. In: *Advances in Neural Information Processing Systems 16 (NIPS 2003)*. Hrsg. von Thrun, S., Saul, L. K. und Schölkopf, B. MIT Press, S. 281–288.

Hartley, R. und Zisserman, A. (2003). *Multiple view geometry in computer vision*. 2. Aufl. Cambridge: Cambridge University Press. 655 S.

He, K., Zhang, X., Ren, S. und Sun, J. (2015). „Deep Residual Learning for Image Recognition“. In: *CoRR* abs/1512.03385.

Heckbert, P. (1982). „Color image quantization for frame buffer display“. In: *9th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. ACM Press, S. 297–307.

Hernández-Vela, A., Reyes, M., Ponce, V. und Escalera, S. (2012). „GrabCut-based human segmentation in video sequences.“ In: *Sensors* 12 (11), S. 15376–15393.

Herrmann, M., Hoernig, M. und Radig, B. (2014). „Online Multi-player Tracking in Monocular Soccer Videos“. In: *AASRI Procedia* 8, S. 30–37.

Herrmann, M., Mayer, C. und Radig, B. (2014). „Automatic Generation of Image Analysis Programs“. In: *Pattern Recognition and Image Analysis* 24 (3), S. 400–408.

Hoernig, M. (2016). „Kameraparameter-Nachführung durch natürliche Landmarken in Sequenzen monokularer Bilder am Beispiel von Fußballübertragungen mit Anwendungen zu automatischer Ballbesitz- und Spielereigniserkennung“. Dissertation. München: Technische Unversität München.

Hoernig, M., Herrmann, M. und Radig, B. (2013). „Real Time Soccer Field Analysis from Monocular TV Video Data“. In: *11th International Conference on Pattern Recognition and Image Analysis: New Information Technologies (PRIA-11-2013)*. Bd. II. Samara, S. 567–570.

Hoernig, M., Herrmann, M. und Radig, B. (2015). „Real-Time Segmentation Methods for Monocular Soccer Videos“. In: *Pattern Recogntion and Image Analysis* 25 (2), S. 327–337.

Hoernig, M., Link, D., Herrmann, M., Radig, B. und Lames, M. (2016). „Detection of Individual Ball Possession in Soccer“. In: *10th International Symposium on Computer Science in Sports (ISCSS)*. Hrsg. von Chung, P., Soltoggio, A., Dawson, C. W., Meng, Q. und Pain, M. Bd. 392. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, S. 103–107.

Hopper, M. (2015). „Bildbasierte Erkennung des Balls in monokularen Aufnahmen von Fußballspielen“. Bachelorarbeit. München: Technische Universität München.

Hoyningen-Huene, N. v. (2011). „Real-time Tracking of Players Identities in Team Sports“. Dissertation. München: Technische Unversität München.

Hsu, C.-W., Chang, C.-C. und Lin, C.-J. (2003). *A Practical Guide to Support Vector Classification*. Technical Report. Department of Computer Science, National Taiwan University.

Huber, P. J. und Ronchetti, E. (2009). *Robust statistics*. 2. Aufl. Wiley series in probability and statistics. Hoboken, N.J: John Wiley & Sons, Inc. 354 S.

IEEE (1998). „1062-1998 - IEEE Standard for a Software Quality Metrics Methodology“. In: *IEEE Std 1061-1998*.

Itoh, H., Takiguchi, T. und Ariki, Y. (2012). „3D tracking of soccer players using time-situation graph in monocular image sequence“. In: *21st International Conference on Pattern Recognition (ICPR)*. IEEE, S. 2532–2536.

itseez (2011). *OpenCV*. Version 2.3.1. URL: http://opencv.org/ (besucht am 23. 02. 2016).

itseez (2015). *OpenCV*. Version 2.4.11. URL: http://opencv.org/ (besucht am 23. 02. 2016).

Izadinia, H., Saleemi, I., Li, W. und Shah, M. (2012). „(MP)2T: Multiple People Multiple Parts Tracker“. In: *Computer Vision – ECCV 2012*. Bd. 7577. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, S. 100–114.

Jaccard, P. (1912). „The Distribution of the Flora in the Alpine Zone“. In: *New Phytologist* 11 (2), S. 37–50.

Jacob, B. und Guennebaud, G. (2013). *Eigen*. Version 3.1.3. URL: http://eigen.tuxfamily.org (besucht am 24. 11. 2016).

Jalal, A. S. und Singh, V. (2012). „The State-of-the-Art in Visual Object Tracking“. In: *Informatica: an International Journal of Computing and Informatics* 36 (3), S. 227–248.

Joo, S.-W. und Chellappa, R. (2007). „A Multiple-Hypothesis Approach for Multiobject Visual Tracking“. In: *IEEE Transactions on Image Processing* 16 (11), S. 2849–2854.

Julier, S. J. und Uhlmann, J. K. (1997). „New extension of the Kalman filter to nonlinear systems“. In: *Signal Processing, Sensor Fusion, and Target Recognition VI*. Hrsg. von Kadar, I. Bd. 3068. SPIE, S. 182–193.

Kálmán, R. E. (1960). „A New Approach to Linear Filtering and Prediction Problems“. In: *Journal of Basic Engineering* 82 (1), S. 35–45.

Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J. u. a. (2009). „Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (2), S. 319–336.

Kazemi, V., Burenius, M., Azizpour, H. und Sullivan, J. (2013). „Multi-view Body Part Recognition with Random Forests“. In: *British Machine Vision Conference (BMVC)*. Hrsg. von Burghardt, T., Damen, D., Mayol-Cuevas, W. und Mirmehdi, M. BMVA Press, S. 48.1–48.11.

Khan, R., Hanbury, A. und Stoettinger, J. (2010). „Skin detection: A random forest approach“. In: *17th IEEE International Conference on Image Processing (ICIP)*. IEEE, S. 4613–4616.

Kim, H., Nam, S. und Kim, J. (2003). „Player Segmentation Evaluation for Trajectory Estimation in Soccer Games“. In: *Image and Vision Computing New Zealand 2003*. Institute of Information Sciences und Technology, Massey University.

Klein, L. A. (2004). *Sensor and data fusion: a tool for information assessment and decision making*. Bellingham, Washington: SPIE. 362 S.

Kristan, M., Pers, J., Perse, M., Kovacic, S. und Bon, M. (2005). „Multiple interacting targets tracking with application to team sports“. In: *4th International Symposium on Image and Signal Processing and Analysis (ISPA)*. IEEE, S. 322–327.

Kuhn, H. W. (1955). „The Hungarian method for the assignment problem“. In: *Naval Research Logistics Quarterly* 2 (1), S. 83–97.

Kviatkovsky, I., Adam, A. und Rivlin, E. (2013). „Color Invariants for Person Reidentification“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (7), S. 1622–1634.

Laplante, P. A., Hrsg. (2001). *Dictionary of computer science, engineering, and technology*. Boca Raton, Florida: CRC Press. 543 S.

Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B. und Savarese, S. (2014). „Learning an Image-Based Motion Context for Multiple People Tracking“. In: IEEE Confernece on Computer Vision and Pattern Recognition (CVPR). IEEE, S. 3542–3549.

Leal-Taixé, L., Milan, A., Reid, I. D., Roth, S. und Schindler, K. (2015). „MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking“. In: *ArXiv e-prints* 1504.01942.

Leo, M., Mosca, N., Spagnolo, P., Mazzeo, P. und Distante, A. (2008). „Real-time multiview event detection in soccer games“. In: *2nd ACM/IEEE International Conference on Distributed Smart Cameras*. IEEE, S. 1–10.

Leo, M., Mazzeo, P. L., Nitti, M. und Spagnolo, P. (2013). „Accurate ball detection in soccer images using probabilistic analysis of salient regions“. In: *Machine Vision and Applications* 24 (8), S. 1561–1574.

Li, Y., Dore, A. und Orwell, J. (2005). „Evaluating the performance of systems for tracking football players and ball“. In: *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, S. 632–637.

Li, Y., Liu, G. und Qian, X. (2009). „Ball and Field Line Detection for Placed Kick Refinement“. In: *WRI Global Congress on Intelligent Systems (GCIS)*. Bd. 4. IEEE, S. 404–407.

Li, Y., Huang, C. und Nevatia, R. (2009). „Learning to associate: HybridBoosted multi-target tracker for crowded scene“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 2953–2960.

Liao, S., Zhu, X., Lei, Z., Zhang, L. und Li, S. Z. (2007). „Learning Multi-scale Block Local Binary Patterns for Face Recognition“. In: *Advances in Biometrics*. Hrsg. von Lee, S.-W. und Li, S. Z. Bd. 4642. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 828–837.

Lienhart, R. und Maydt, J. (2002). „An extended set of Haar-like features for rapid object detection“. In: *International Conference on Image Processing (ICIP)*. Bd. 1. IEEE, S. I.900–I.903.

Liu, J., Tong, X., Li, W., Wang, T., Zhang, Y. u. a. (2009). „Automatic player detection, labeling and tracking in broadcast soccer video“. In: *Pattern Recognition Letters* 30 (2), S. 103–113.

Liu, J., Collins, R. und Liu, Y. (2011). „Surveillance Camera Autocalibration based on Pedestrian Height Distributions“. In: *British Machine Vision Conference (BMVC)*. British Machine Vision Association, S. 117.1–117.11.

Liu, Y., Liang, D., Huang, Q. und Gao, W. (2006). „Extracting 3D information from broadcast soccer video“. In: *Image and Vision Computing* 24 (10), S. 1146–1162.

LiveLike VR (2016). *LiveLike VR*. URL: http://www.livelikevr.com/ (besucht am 26. 04. 2016).

Lloyd, S. P. (1982). „Least squares quantization in PCM“. In: *IEEE Transactions on Information Theory* 28 (2), S. 129–137.

Loipfinger, M. (2014). „Erkennung von Logos und Einblendungen in Fernsehübertragungen von Fußballspielen“. Bachelorarbeit. München: Technische Universität München.

Lu, W.-L., Okuma, K. und Little, J. J. (2009). „Tracking and recognizing actions of multiple hockey players using the boosted particle filter“. In: *Image and Vision Computing* 27 (1), S. 189–205.

Lu, W.-L., Ting, J.-A., Murphy, K. P. und Little, J. J. (2011). „Identifying players in broadcast sports videos using conditional random fields“. In: IEEE, S. 3249–3256.

Lu, W.-L., Ting, J.-A., Little, J. J. und Murphy, K. P. (2013). „Learning to Track and Identify Players from Broadcast Sports Videos“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (7), S. 1704–1716.

Mackenzie, R. und Cushion, C. (2013). „Performance analysis in football: A critical review and implications for future research“. In: *Journal of Sports Sciences* 31 (6), S. 639–676.

Maggio, E. und Cavallaro, A. (2011). *Video tracking: theory and practice*. West Sussex, UK: Wiley. 266 S.

Marco, T. D., Leo, M. und Distante, C. (2013). „Soccer Ball Detection with Isophotes Curvature Analysis“. In: *Image Analysis and Processing*. Hrsg. von Petrosino, A. Bearb. von Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F. u. a. Bd. 8156. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 793–802.

Masi, I. und Lisanti, G. (2014). *CLEAR-MOT*. Version Feb 11, 2014. URL: https://github.com/glisanti/CLEAR-MOT (besucht am 23. 02. 2016).

Mayer, C. (2012). „Facial Expression Recognition With A Three-Dimensional Face Model“. Dissertation. München: Technische Universität München.

Mazzeo, P. L., Spagnolo, P., Leo, M. und D'Orazio, T. (2008). „Visual Players Detection and Tracking in Soccer Matches“. In: *5th International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, S. 326–333.

MDR (2013). *3. Liga 2013/2014: Chemnitzer FC - RB Leipzig*. Ausgestrahlt am 26.10.2013 um 14:00 Uhr.

Milan, A. (2015). *motutils*. Version 2015-01-25. URL: https://bitbucket.org/amilan/motutils (besucht am 23. 02. 2016).

Milan, A., Leal-Taixé, L., Reid, I. D., Roth, S. und Schindler, K. (2016). „MOT16: A Benchmark for Multi-Object Tracking“. In: *ArXiv e-prints* 1603.0083.

Milan, A., Leal-Taixé, L., Schindler, K., Roth, S. und Reid, I. (2015a). *MOT Challenge Development Kit*. Version 1.0 - Jan. 23, 2015. URL: http://motchallenge.net/data/devkit.zip (besucht am 23. 02. 2016).

Milan, A., Leal-Taixé, L., Schindler, K., Roth, S. und Reid, I. (2015b). *Mutliple Object Tracking Benchmark*. URL: http://www.motchallenge.net/ (besucht am 23. 02. 2016).

Milan, A., Schindler, K. und Roth, S. (2013). „Challenges of Ground Truth Evaluation of Multi-target Tracking“. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, S. 735–742.

Milan, A., Schindler, K. und Roth, S. (2015). „Multi-Target Tracking by Discrete-Continuous Energy Minimization“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (99), S. 1–17.

Misu, T., Matsui, A., Naemura, M., Fujii, M. und Yagi, N. (2007). „Distributed Particle Filtering for Multiocular Soccer-Ball Tracking“. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Bd. 3. IEEE, S. III/937–III/940.

Miura, J. und Kubo, H. (2008). „Tracking players in highly complex scenes in broadcast soccer video using a constraint satisfaction approach“. In: *International Conference on Content-based Image and Video Retrieval (CIVR)*. ACM Press, S. 505–514.

Moeslund, T. B., Hilton, A. und Krüger, V. (2006). „A survey of advances in vision-based human motion capture and analysis“. In: *Computer Vision and Image Understanding* 104 (2), S. 90–126.

Mori, G. und Malik, J. (2002). „Estimating human body configurations using shape context matching“. In: *Computer Vision - ECCV 2002*. Hrsg. von Heyden, A., Sparr, G., Nielsen, M. und Johansen, P. Lecture Notes in Computer Science 2352.

Mori, G. und Malik, J. (2006). „Recovering 3D human body configurations using shape contexts“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (7), S. 1052–1062.

Munkres, J. (1957). „Algorithms for the Assignment and Transportation Problems“. In: *Journal of the Society for Industrial and Applied Mathematics* 5 (1), S. 32–38.

MVTec Software GmbH (2012). *HALCON*. Version 11. URL: http://www.mvtec.com/products/halcon/ (besucht am 24. 11. 2016).

Nam, W., Dollár, P. und Han, J. H. (2014). „Local Decorrelation For Improved Pedestrian Detection“. In: *Advances in Neural Information Processing Systems 27 (NIPS 2014)*. Hrsg. von Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D. und Weinberger, K. Q. Curran Associates, Inc., S. 424–432.

Nghiem, A. T., Bremond, F., Thonnat, M. und Valentin, V. (2007). „ETISEO, performance evaluation for video surveillance systems“. In: *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, S. 476–481.

Nillius, P., Sullivan, J. und Carlsson, S. (2006). „Multi-Target Tracking - Linking Identities using Bayesian Network Inference“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Bd. 2. IEEE, S. 2187–2194.

Nocedal, J. und Wright, S. J. (2006a). *Numerical optimization*. 2. Aufl. New York: Springer.

Nocedal, J. und Wright, S. J. (2006b). *Numerical Optimization*. Springer.

Ojala, T., Pietikäinen, M. und Maenpaa, T. (2002). „Multiresolution gray-scale and rotation invariant texture classification with local binary patterns“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (7), S. 971–987.

Ojala, T., Pietikäinen, M. und Harwood, D. (1996). „A comparative study of texture measures with classification based on featured distributions“. In: *Pattern Recognition* 29 (1), S. 51–59.

Okuma, K., Taleghani, A., Freitas, N. d., Little, J. J. und Lowe, D. G. (2004). „A Boosted Particle Filter: Multitarget Detection and Tracking“. In: *Computer Vision - ECCV 2004*. Hrsg. von Pajdla, T. und Matas, J. Bearb. von Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C. u. a. Bd. 3021. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 28–39.

Otsu, N. (1979). „A Threshold Selection Method from Gray-Level Histograms“. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9 (1), S. 62–66.

Pallavi, V., Mukherjee, J., Majumdar, A. K. und Sural, S. (2008). „Graph-Based Multiplayer Detection and Tracking in Broadcast Soccer Videos“. In: *IEEE Transactions on Multimedia* 10 (5), S. 794–805.

Papageorgiou, C., Oren, M. und Poggio, T. (1998). „A general framework for object detection“. In: *Sixth International Conference on Computer Vision (ICCV)*, S. 555–562.

Parent, R. (2012). „Inverse kinematics“. In: *Computer Animation - Algortihms & Techniques*. 3. Aufl. Waltham, Massachusetts: Morgan Kaufmann, S. 161–186.

Pelleg, D. und Moore, A. W. (2000). „X-means: Extending K-means with Efficient Estimation of the Number of Clusters“. In: *International Conference on Machine Learning (ICML 2000)*. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., S. 727–734.

PERFORM Media Deutschland GmbH (2016). *Die Datenerfassung*. Opta | we live sport. URL: http://www.optasports.de/%C3%BCber-uns/so-arbeiten-wir/the-data-collection-process.aspx (besucht am 26. 05. 2016).

Pingali, G., Opalach, A. und Jean, Y. (2000). „Ball tracking and virtual replays for innovative tennis broadcasts“. In: *15th International Conference on Pattern Recognition (ICPR)*. Bd. 4. IEEE, S. 152–156.

Pirsiavash, H., Ramanan, D. und Fowlkes, C. C. (2011). „Globally-optimal greedy algorithms for tracking a variable number of objects“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 1201–1208.

Plataniotis, K. N. und Venetsanopoulos, A. N. (2000). *Color Image Processing and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Poiesi, F., Daniyal, F. und Cavallaro, A. (2010). „Detector-less ball localization using context and motion flow analysis“. In: *17th International Conference on Image Processing (ICIP)*. IEEE, S. 3913–3916.

Poiesi, F., Mazzon, R. und Cavallaro, A. (2013). „Multi-target tracking on confidence maps: An application to people tracking“. In: *Computer Vision and Image Understanding* 117 (10), S. 1257–1272.

Poppe, R. (2007). „Vision-based human motion analysis: An overview“. In: *Computer Vision and Image Understanding* 108 (1), S. 4–18.

Powers, D. M. W. (2011). „Evaluation: From precision, recall and f-measure to ROC, informedness, markedness & correlation“. In: *Journal of Machine Learning Technologies* 2 (1), S. 37–63.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. und Flannery, B. P., Hrsg. (2007). *Numerical recipes: the art of scientific computing*. 3. Aufl. Cambridge, UK ; New York: Cambridge University Press. 1235 S.

Prozone Sports (2016). *Sports Performance Analysis Software*. Prozone Sports. URL: http://prozonesports.stats.com/performance-analysis/ (besucht am 21. 05. 2016).

Puhalla, J., Krans, J. und Goatley, M. (2010). *Sports fields: design, construction, and maintenance*. 2nd ed. Hoboken, N.J. : [Lawrence, Kan.]: Wiley ; SportsTurf Managers Association. 516 S.

Ramanan, D. (2006). „Learning to parse images of articulated bodies“. In: *Advances in Neural Information Processing Systems 19 (NIPS 2006)*. Hrsg. von Schölkopf, B., Platt, J. und Hoffman, T. MIT Press, S. 1129–1136.

Rauch, H. E., Striebel, C. T. und Tung, F. (1965). „Maximum likelihood estimates of linear dynamic systems“. In: *AIAA Journal* 3 (8), S. 1445–1450.

Reid, D. (1979). „An algorithm for tracking multiple targets“. In: *IEEE Transactions on Automatic Control* 24 (6), S. 843–854.

Ren, J., Orwell, J., Jones, G. und Xu, M. (2008). „Real-Time Modeling of 3-D Soccer Ball Trajectories From Multiple Fixed Cameras“. In: *IEEE Transactions on Circuits and Systems for Video Technology* 18 (3), S. 350–362.

Rodriguez, M., Laptev, I., Sivic, J. und Audibert, J.-Y. (2011). „Density-aware person detection and tracking in crowds“. In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, S. 2423–2430.

Roth, M., Bäuml, M., Fischer, M. und Bernardin, K. (2014). *PyMOT*. Version 12 Aug 2014. URL: https://github.com/Videmo/pymot (besucht am 23. 02. 2016).

Rujikietgumjorn, S. und Collins, R. T. (2013). „Optimized Pedestrian Detection for Multiple and Occluded People“. In: IEEE, S. 3690–3697.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S. u. a. (2015). „ImageNet Large Scale Visual Recognition Challenge“. In: *International Journal of Computer Vision* 115 (3), S. 211–252.

Russell, S. J. und Norvig, P. (2003). *Artificial intelligence: a modern approach*. 2. Aufl. Prentice Hall series in artificial intelligence. Upper Saddle River, NJ: Prentice Hall. 1081 S.

Sabirin, H., Sankoh, H. und Naito, S. (2015). „Automatic Soccer Player Tracking in Single Camera with Robust Occlusion Handling Using Attribute Matching“. In: *IEICE TRANSACTIONS on Information and Systems* E98-D (8), S. 1580–1588.

Särkkä, S. (2013). *Bayesian filtering and smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge, UK: Cambridge University Press. 232 S.

Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N. u. a. (2014). „High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth“. In: *Pattern Recognition*. Hrsg. von Jiang, X., Hornegger, J. und Koch, R. Bd. 8753. Lecture Notes in Computer Science. Cham: Springer International Publishing, S. 31–42.

Schlipsing, M. (2014). „Videobasierte Leistungserfassung im Fußball“. Dissertation. Ruhr-Universität Bochum.

Schlipsing, M., Salmen, J. und Igel, C. (2013). „Echtzeit-Videoanalyse im Fußball: Ein Live-System zum Spieler-Tracking“. In: *KI - Künstliche Intelligenz* 27 (3), S. 235–240.

Sebastian, P., Voon, Y. V. und Comley, R. (2008). „The effect of colour space on tracking robustness“. In: *3rd IEEE Conference on Industrail Electronics and Applications (ICIEA)*. IEEE, S. 2512–2516.

Seo, Y., Choi, S., Kim, H. und Hong, K.-S. (1997). „Where are the ball and players? Soccer game analysis with color-based tracking and image mosaick“. In: *Image Analysis and Processing*. Hrsg. von Del Bimbo, A. Bearb. von Goos, G., Hartmanis, J. und Leeuwen, J. van. Bd. 1311. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 196–203.

Shaikh, S. H., Saeed, K. und Chaki, N. (2014). *Moving Object Detection Using Background Subtraction*. SpringerBriefs in Computer Science. Cham: Springer International Publishing.

Shannon, C. E. (2001). „A mathematical theory of communication“. In: *ACM SIGMOBILE Mobile Computing and Communications Review* 5 (1), S. 3.

Sharp, T., Forsyth, D., Torr, P. und Zisserman, A. (2008). „Implementing Decision Trees and Forests on a GPU“. In: *Computer Vision - ECCV 2008*. Bd. 5305. Lecture Notes in Computer Science, S. 595–608.

Shimawaki, T., Sakiyama, T., Miura, J. und Shirai, Y. (2006). „Estimation of Ball Route under Overlapping with Players and Lines in Soccer Video Image Sequence“. In: *18th International Conference on Pattern Recognition (ICPR)*. IEEE, S. 359–362.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M. u. a. (2011). „Real-time human pose recognition in parts from single depth images“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 1297–1304.

Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M. u. a. (2013). „Efficient human pose estimation from single depth images.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (12), S. 2821–2840.

Siles Canales, F. (2014). „Automated Semantic Annotation of Football Games from TV Broadcast“. Dissertation. München: Technische Universität München.

Smith, K., Gatica-Perez, D., Odobez, J.-M. und Ba, S. (2005). „Evaluating Multi-Object Tracking“. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Bd. 3. IEEE, S. 36–36.

Sonka, M., Hlavac, V. und Boyle, R. (2008). *Image processing, analysis, and machine vision*. 3. Aufl. Toronto: Thompson Learning. 829 S.

Stark, P. B. und Parker, R. L. (1995). „Bounded-variable least-squares: an algorithm and applications“. In: *Computational Statistics* 10, S. 129–141.

Stauffer, C. und Grimson, W. (1999). „Adaptive background mixture models for real-time tracking“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Bd. 2. IEEE, S. 246–252.

Steger, C., Ulrich, M. und Weidemann, C. (2008). *Machine vision algorithms and applications*. Weinheim: Wiley-VCH. 360 S.

Stiefelhagen, R., Bernardin, K., Bowers, R., Garofolo, J., Mostefa, D. u. a. (2007). „The CLEAR 2006 Evaluation“. In: *Multimodal Technologies for Perception of Humans*. Hrsg. von Stiefelhagen, R. und Garofolo, J. Bd. 4122. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 1–44.

Stroustrup, B. (2013). *The C++ programming language*. 4. Aufl. Upper Saddle River, NJ: Addison-Wesley. 1346 S.

Sullivan, J. und Carlsson, S. (2006). „Tracking and Labelling of Interacting Multiple Targets“. In: *Computer Vision – ECCV 2006*. Hrsg. von Leonardis, A., Bischof, H. und Pinz, A. Bearb. von Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F. u. a. Bd. 3953. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 619–632.

Sutherland, I. E. und Hodgman, G. W. (1974). „Reentrant polygon clipping“. In: *Communications of the ACM* 17 (1), S. 32–42.

Suzuki, S. und Abe, K. (1985). „Topological structural analysis of digitized binary images by border following“. In: *Computer Vision, Graphics, and Image Processing* 30 (1), S. 32–46.

Theobalt, C., Albrecht, I., Haber, J., Magnor, M. und Seidel, H.-P. (2004). „Pitching a baseball: tracking high-speed motion with multi-exposure images“. In: *ACM Transactions on Graphics* 23 (3), S. 540–547.

Tompson, J., Jain, A., LeCun, Y. und Bregler, C. (2014). „Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation“. In: *Advances in Neural Information Processing Systems 27 (NIPS 2014)*. Advances in Neural Information Processing Systems (NIPS). Montreal, Canada: Curran Associates, Inc., S. 1799–1807.

Tong, X.-F., Lu, H.-Q. und Liu, Q.-S. (2004). „An effective and fast soccer ball detection and tracking method“. In: *17th International Conference on Pattern Recognition (ICPR)*. Bd. 4. IEEE, S. 795–798.

TRACAB (2015). *TRACAB image tracking system (TM)*. URL: http://tracab.hegogroup.com/ (besucht am 23. 02. 2016).

University of Reading (2002). *VS-PETS Football Dataset*. Third IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. URL: http://www.cvg.reading.ac.uk/VSPETS/vspets-db.html (besucht am 23. 02. 2016).

Urtasun, R., Fleet, D. und Fua, P. (2005). „Monocular 3-D Tracking of the Golf Swing“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Bd. 2. IEEE, S. 932–938.

Utsumi, O., Miura, K., Ide, I., Sakai, S. und Tanaka, H. (2002). „An object detection method for describing soccer games from video“. In: *IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, S. 45–48.

Viola, P. und Jones, M. (2001). „Rapid Object Detection using a Boosted Cascade of Simple Features“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Bd. 1. IEEE, S. I.511–I.518.

Virtually Live US, Inc. (2016). *Virtually Live*. URL: http://virtuallylive.com/ (besucht am 26. 04. 2016).

Viterbi, A. (1967). „Error bounds for convolutional codes and an asymptotically optimum decoding algorithm“. In: *IEEE Transactions on Information Theory* 13 (2), S. 260–269.

Vondrick, C., Patterson, D. und Ramanan, D. (2012). „Efficiently Scaling up Crowdsourced Video Annotation: A Set of Best Practices for High Quality, Economical Video Labeling“. In: *International Journal of Computer Vision* 101 (1), S. 184–204.

Wang, X., Ablavsky, V., Shitrit, H. B. und Fua, P. (2014). „Take your eyes off the ball: Improving ball-tracking by focusing on team play“. In: *Computer Vision and Image Understanding* 119, S. 102–115.

Wei, X., Zhang, P. und Chai, J. (2012). „Accurate realtime full-body motion capture using a single depth camera“. In: *ACM Transactions on Graphics* 31 (6), 188:1–188:12.

Weijer, J. v. d., Gevers, T. und Bagdanov, A. (2006). „Boosting color saliency in image feature detection“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (1), S. 150–156.

Wu, B. und Nevatia, R. (2006). „Tracking of Multiple, Partially Occluded Humans based on Static Body Part Detection“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 951–958.

Xu, R. und Wunsch II, D. (2005). „Survey of Clustering Algorithms“. In: *IEEE Transactions on Neural Networks* 16 (3), S. 645–678.

Yang, B., Li, Y., Huang, C. und Nevatia, R. (2008). *Multi-target tracking evaluation tool*. Version 11/13/2008. URL: http://iris.usc.edu/people/yangbo/data/EvaluationTool.zip (besucht am 23. 02. 2016).

Yang, B. und Nevatia, R. (2012). „An online learned CRF model for multi-target tracking“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 2034–2041.

Yang, H., Shao, L., Zheng, F., Wang, L. und Song, Z. (2011). „Recent advances and trends in visual tracking: A review“. In: *Neurocomputing* 74 (18), S. 3823–3831.

Yang, Y. und Ramanan, D. (2011). „Articulated pose estimation with flexible mixtures of parts“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 1385–1392.

Yang, Y. und Ramanan, D. (2012). *Articulated Pose Estimation with Flexible Mixtures of Parts*. Version 1.3. URL: http://www.ics.uci.edu/~yyang8/research/pose/ (besucht am 01. 05. 2016).

Yang, Y. und Ramanan, D. (2013). „Articulated human detection with flexible mixtures of parts“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (12), S. 2878–2890.

Yilmaz, A., Javed, O. und Shah, M. (2006). „Object tracking: A survey“. In: *ACM Computing Surveys* 38 (4).

Yu, X., Hay, T. S., Yan, X. und Chng, E. (2005). „A Player-Possession Acquisition System for Broadcast Soccer Video“. In: IEEE International Conference on Multimedia and Expo (ICME). IEEE, S. 522–525.

Yu, X., Leong, H. W., Xu, C. und Tian, Q. (2006). „Trajectory-Based Ball Detection and Tracking in Broadcast Soccer Video“. In: *IEEE Transactions on Multimedia* 8 (6), S. 1164–1178.

Yu, X., Tian, Q. und Wan, K. W. (2003). „A novel ball detection framework for real soccer video“. In: *IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, S. II/265–268.

Yu, X., Tu, X. und Ang, E. L. (2007). „Trajectory-Based Ball Detection and Tracking in Broadcast Soccer Video with the Aid of Camera Motion Recovery“. In: *IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, S. 1543–1546.

ZDF (2012). *UEFA Championsleague 2012/13: FC Bayern München - OSC Lille*. Ausgestrahlt am 07.11.2012 um 20:45 Uhr.

ZDF (2013a). *UEFA Championsleague 2013/14: Borussia Dortmund - FC Arsenal*. Ausgestrahlt am 06.11.2013 um 20:45 Uhr.

ZDF (2013b). *WM-Qualifikation Europa: Schweden - Deutschland*. Ausgestrahlt am 15.10.2013 um 20:45 Uhr.

ZDF (2016). *ZDF App: Champions League im 2nd Screen*. ZDF.de. URL: http://www.zdf.de/zdf/champions-league-second-screen-in-der-zdf-app-37329870.html (besucht am 26. 05. 2016).

Zhang, J., Presti, L. L. und Sclaroff, S. (2012). „Online Multi-person Tracking by Tracker Hierarchy“. In: *IEEE 9th International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*. IEEE, S. 379–385.

Zhang, L., Li, Y. und Nevatia, R. (2008). „Global data association for multi-object tracking using network flows“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, S. 1–8.

Zhang, T., Ghanem, B. und Ahuja, N. (2012). „Robust multi-object tracking via cross-domain contextual information for sports video analysis“. In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, S. 985–988.

Zhang, Z. (1997). „Parameter estimation techniques: a tutorial with application to conic fitting“. In: *Image and Vision Computing* 15 (1), S. 59–76.