

Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät II
Institut für Informatik
Lehrstuhl für Adaptive Systeme
Prof. Dr. Verena V. Hafner



Studienarbeit

**Methoden und Werkzeuge zur Analyse von
Teamverhalten und -strategie im Roboterfußball**

Autor: Philipp Strobel
philippstrobel@posteo.de

Betreuer: Prof. Dr. Verena V. Hafner

Berlin, den 6. Januar 2019

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 1 |
| 1.1 | RoboCup | 2 |
| 1.2 | Das Nao-Team Humboldt | 3 |
| 1.3 | Zielstellung | 4 |
| 1.4 | Struktur der Arbeit | 5 |
| 2 | Infrastruktur der SPL im RoboCup | 6 |
| 2.1 | GameController | 7 |
| 2.2 | Teamkommunikation | 8 |
| 2.3 | Logdateien | 9 |
| 3 | Werkzeuge zur Auswertung und Analyse | 12 |
| 3.1 | RobotControl | 12 |
| 3.2 | VideoLogLabeling | 15 |
| 4 | System zur synchronen Videoaufnahme | 18 |
| 4.1 | Ähnliche Systeme anderer Teams und Ligen | 18 |
| 4.2 | Evaluation und Anforderungen | 20 |
| 4.3 | Umsetzung und Implementierung | 22 |
| 4.4 | Empirische Erfahrungen im realen Einsatz | 25 |
| 5 | Zusammenfassung und Diskussion | 31 |
| | Literaturverzeichnis | 33 |
| | Abbildungsverzeichnis | 35 |

1 Einleitung

Die Entwicklung komplexer autonomer Systeme ist aufwendig und fehleranfällig. Zusätzlich kommen eine Vielzahl von verschiedenen Parametern zur Konfiguration eines Systems hinzu. Die damit steigende Komplexität führt dazu, dass die erwartete Funktionsweise beeinträchtigt wird und unerwartete Rückkopplungseffekte auftreten können. Solchen Problemen kann mit umfangreichen Analyse- und Auswertungsmethoden begegnet werden.

Im Rahmen des RoboCups stellen autonom agierende, fußballspielende Roboter ein solches komplexes System dar. Wobei die Konfiguration der Roboter und Interaktion der eingesetzten Algorithmen die Analyse und Auswertung erschweren.

Die Analyse bezieht sich dabei vor allem auf bestimmte Probleme während der Entwicklung und soll die Ursachen nachvollziehbar bzw. reproduzierbar machen. Auswertungen beziehen sich auf die Leistung der implementierten Algorithmen und deren Zusammenspiel. Gute Auswertungsmethoden decken Fehler, Probleme und Defizite im Gesamtkonzept bzw. einzelnen Teilen davon auf.

Im RoboCup sind die Auswertungsmethoden an die verwendeten Roboter angepasst und bieten entsprechend spezialisierte Analysemöglichkeiten. Neben den internen Analysewerkzeugen (wie die Aufnahme von Logdateien), werden auch „externe“ Werkzeuge zur Analyse der Systeme eingesetzt (z.B. Tracking- und Videoaufnahmen). Diese betrachten das System von außen und versuchen ohne interne Kenntnisse des Roboters Auswertungen auf einer höheren, abstrakten Ebene zu ermöglichen.

Für die Auswertung jener internen und externen Daten ist manueller Aufwand erforderlich. Insbesondere die Synchronisierung der aufgezeichneten Logdateien und Videoaufnahmen auf einen bestimmten Zeitpunkt ist aufwendig.

In der SPL-Liga des RoboCups gibt es bisher kein einheitliches Aufnahmesystem und nur die wenigsten Teams nehmen systematisch ihre Wettkampfspiele per Video auf. Das Nao-Team Humboldt ist eines der Teams, die seit einigen Jahren ihre Spiele manuell per Video aufnehmen. Auf Grund des oben beschriebenen Synchronisierungsaufwands, wurden Auswertungen von Logdateien und Videos nur vereinzelt durchgeführt. Für eine systematische Analyse des Teamverhaltens ist dieser Aufwand allerdings notwendig, da das Zusammenspiel der Roboter sonst nur schwer einzuschätzen ist.

Um diesen Aufwand zu vereinfachen, wird im Rahmen dieser Arbeit ein Aufnahmesystem entwickelt, das synchronisierte Videos aufzeichnet und damit die Auswertung und

Synchronisation der Daten automatisiert und beschleunigt. Zu Beginn wird dazu ein Überblick über die bestehenden Werkzeuge und Methoden zur Analyse und Auswertung von fußballspielenden Robotern gegeben. Anschließend werden die in dieser Arbeit weiterentwickelten Werkzeuge vorgestellt und schließlich die Entwicklung des Aufnahmesystems beschrieben.

1.1 RoboCup

Der RoboCup¹ ist eine jährliche Veranstaltung und eine Gemeinschaft, die sich die Entwicklung und Erforschung von künstlicher Intelligenz und autonomen Robotern auf die Fahnen geschrieben hat. Dabei soll der RoboCup auch als Plattform zur gesellschaftlichen Kommunikation und Aufklärung über künstliche Intelligenz und der Robotikforschung dienen. Das „große Ziel“ dabei ist, bis Mitte des 21. Jahrhunderts mit vollautonomen Robotern gegen den aktuellen Fußball-Weltmeister zu gewinnen².

Erstmals erwähnt wurde eine solche Initiative 1992 von Alan Mackworth. In den darauffolgenden Jahren wurden weltweit Überlegungen angestellt, wie die Forschung an künstlicher Intelligenz und Robotik allgemein der Öffentlichkeit besser präsentiert werden und gleichzeitig die Entwicklung und den Austausch von Forschung vorantreiben kann. Vor allem in Japan fanden verschiedene Workshops und Diskussionsrunden statt, die zur Entwicklung der RoboCup-Idee beitrugen. Auf Grund des internationalen Interesses wurde schließlich die bis heute bestehende RoboCup-Initiative 1993 gegründet und führte 1997 zur ersten RoboCup-Weltmeisterschaft³.

Die Idee, Fußball als Test- und Wettkampfumgebung zur Erforschung und Entwicklung autonomer Systeme zu verwenden, war von Beginn an der Schwerpunkt der Initiative. Im Vergleich zu anderen (spielerischen) Herausforderungen (wie z.B. Schach), stellt ein Fußballspiel eine weit größere Herausforderung im Umgang mit der komplexen, dynamischen Welt dar. Beispielsweise ist Schach ein statisches bzw. deterministisches System, indem jeder Schritt nur eine Aktion erfordert und sich das System nur in diesem ändert. Außerdem hat in einem solchen statischen System der Agent sehr umfangreiche oder auch vollständige Informationen über den Zustand der Umgebung und unter Umständen viel Zeit zur Berechnung des nächsten Schritts bzw. zur Ermittlung einer optimalen Strategie. Im Gegensatz dazu hat ein Agent im Fußball nur unvollständige und teilweise fehlerhafte Informationen über seine lokale und globale Umgebung zur Verfügung. Gleichzeitig muss in Echtzeit eine „sinnvolle“ Entscheidung anhand dieser Informationen getroffen werden. Zusätzlich kommen weitere unabhängige Agenten hinzu, mit denen zusammen kooperiert

¹<http://www.robocup.org> (30.07.2018)

²<http://robocup.org/objective> (06.08.2018)

³http://robocup.org/a_brief_history_of_robocup (06.08.2018)

1 Einleitung

oder auch gegeneinander gespielt werden muss. Für ein gutes Teamspiel sollte sich die Mannschaft auch auf eine gemeinsame Strategie einigen können.

Zu Beginn des RoboCups wurden Wettkämpfe in drei Ligen ausgetragen: Simulation, Small-Size und Middle-Size. Im Laufe der Zeit wurden die Spielregeln je Liga weiterentwickelt. Zusätzlich kamen weitere Ligen und weitere Einsatzgebiete für autonome Roboter hinzu: SPL, Humanoids, Rescue, @Home, Industrial, sowie einige Unterligen.

Jede Liga legt ihren Fokus jeweils auf einen anderen Themenschwerpunkt, entsprechend hat jede Liga unterschiedliche Herausforderungen. Die Simulationsliga hat ihren Schwerpunkt auf Algorithmen zur Teamstrategie und kooperativem Verhalten. Die Small- und Middle-Size-Liga baut ihre Roboter gemäß den Regeln selbst und hat ebenfalls ihren Fokus auf Teamstrategie und kooperativen Verhalten. Die @Home-Liga legt ihren Fokus auf Service-, Haushalts- und Assistenzroboter, daher sollten diese Roboter sich gut in unterschiedlichen häuslichen Umgebungen zurecht finden. Bei der Humanoids-Liga steht das Design und die Entwicklung autonomer humanoider Roboter im Vordergrund. Die Rescue-Liga entwickelt autonome Roboter, die in Katastrophengebieten zum Einsatz kommen können. Und in der SPL, kurz für „Standard Plattform Liga“, werden standardisierte Roboter der Firma SoftBank Robotics verwendet und es steht die Entwicklung und der Wettkampf der eingesetzten Algorithmen im Vordergrund.

Trotz der unterschiedlichen Themenschwerpunkte ist die Auswerte- und Analysemöglichkeit über alle Ligen hinweg zentraler Bestandteil für eine erfolgreiche Weiterentwicklung der Roboter und Algorithmen. Auf Grund der Rahmenbedingungen ist es in einigen Ligen leichter Auswertungen und Analysen zu ermöglichen. So ist es in der Simulationsliga möglich das gesamte Weltmodell und die Entscheidungen der einzelnen Agenten aufzuzeichnen und Fehler, sowie bestimmtes Verhalten, genau zu untersuchen. Andererseits hat bspw. die Small-Size-Liga ein einheitliches Trackingsystem, das allen Teams die gleichen Informationen bereitstellt (siehe Abschnitt 4.1). Im Gegensatz dazu fehlt in anderen Ligen ein einheitliches System zur Aufzeichnung eines Wettkampfspiels oder es ist nur in definierter Umgebung (z.B. Labor) möglich.

1.2 Das Nao-Team Humboldt

Das Nao-Team Humboldt (kurz: NaoTH) nimmt jährlich am RoboCup teil und hat in der Vergangenheit sowohl in der SPL-Liga, als auch in der 3D-Simulationsliga Erfolge feiern können [6].

Die dabei verwendeten Auswerte- und Analysewerkzeuge wurden schon frühzeitig entwickelt und verschiedene Konzepte zur Fehleranalyse, Konfiguration und Auswertung eines Roboters eingesetzt (siehe dazu [2, 4, 11]).

Die Hauptwerkzeuge sind zum einen die Aufnahme von Logdateien auf dem Roboter, dabei wird der interne Zustand des Roboters in jedem Ausführungsschritt in eine Logdatei geschrieben. Zum anderen wird das Programm „RobotControl“ für die Konfiguration und Auswertung des Roboters im laufenden, als auch im „offline“ Betrieb verwendet (siehe Abschnitte 2.3 und 3.1). Im Rahmen einer Forschungsarbeit wurde ein Analyse-Tool für die Auswertung der aufgenommenen Log- und Videodateien entwickelt (Abschnitt 3.2).

1.3 Zielstellung

Dem Nao-Team Humboldt steht ein über mehrere Jahre gesammeltes Archiv von Logdateien und manuellen Videoaufnahmen zur Verfügung. Auf diesen Logdateien wurden und werden einzelne Analysen und Auswertungen für spezielle Probleme durchgeführt.

Um die Logdateien gemeinsam mit den Videoaufzeichnungen auswerten zu können, müssen diese manuell synchronisiert werden, da sonst nur schwer Schlussfolgerungen gezogen werden können, wenn ein Ereignis nicht zur gleichen Zeit in der Logdatei und im Video dargestellt wird. Beispielsweise sollte der An- bzw. Abpfiff des Spiels bei der Auswertung zum gleichen Zeitpunkt erfolgen. Je mehr Logdateien auf einmal verwendet werden, umso größer wird der Aufwand zur Synchronisierung.

Ein weiterer Punkt, der die Auswertung eines Teams erschwert, ist die Kenntnis von der tatsächlichen Position des Balls und der Spieler auf dem Feld, die s.g. „Ground Truth“. Jeder einzelne Roboter nimmt die Welt aus seiner Sicht wahr. Entsprechend gibt es mehrere Ansichten, wo der Ball gerade liegen könnte. Soll die Korrektheit der Informationen geprüft werden, sollten die genauen Informationen „extern“ ermittelt werden (z.B. durch ausmessen oder ein externes Tracking-System). Das ist nur unter Laborbedingungen möglich und während eines Wettkampfspiels nicht zu leisten.

Das Nao-Team Humboldt hat zum Tracking ein fest installiertes Kamerasystem (OptiTrack Flex 13⁴) in ihrem Labor, das die Position von markierten Objekten im Raum erfassen kann. Dazu müssen die zu erfassenden Objekte mit Reflektionskugeln markiert werden. Falls notwendig, ist das gesamte System zuvor zu kalibrieren. Wenn diese Vorarbeit abgeschlossen ist, kann das Experiment durchgeführt und die Position(en) aufgezeichnet werden.

Ein solches System könnte unter Wettkampfbedingungen und während des RoboCups nur mit großem Aufwand betrieben werden. Außerdem müssen die Verhältnisse vor Ort den Aufbau eines solchen Systems ermöglichen.

Neben der einfacheren Informationserfassung in einem, bzw. nach einem Spiel, ist eine häufige Anforderung, das Aufnahmesystem schnell aufbauen zu können und portabel zu

⁴<http://www.optitrack.com/products/flex-13> (27.08.2018)

sein. Das würde es auch ermöglichen, auf verschiedene Anforderungen an den Wettkampforten flexibel zu reagieren und sich an die gegebenen Bedingungen anzupassen.

In dieser Studienarbeit wurde sich mit der bestehenden Infrastruktur und Analysewerkzeuge des Nao-Team Humboldts im Rahmen des RoboCups vertraut gemacht und weitere Varianten zur Auswertung von Robotern in einem Fußballteam entwickelt. Dabei wurde auf bestehende Arbeiten aufgebaut und um zusätzliche Komponenten erweitert. Im Besonderen wird das entwickelte System zur externen Videoaufnahme von Spielen vorgestellt und die Erfahrungen in der Nutzung des Systems während zweier Wettkämpfe erläutern.

1.4 Struktur der Arbeit

Kapitel 2 befasst sich mit dem Feldaufbau eines SPL-Spiels und der Infrastruktur der SPL-Liga und des Nao-Team Humboldts. Dabei wird auf die Bestandteile eingegangen, die insbesondere für die Aufnahme von Logdateien und dessen spätere Auswertung nützlich bzw. notwendig sind. In Kapitel 3 werden die im Nao-Team Humboldt verwendeten und entwickelten Werkzeuge zur Analyse und Auswertung von einzelnen Robotern und die Aufnahme von Logdateien eingegangen. Anschließend wird in Kapitel 4 das entwickelte Aufnahmesystem beschrieben. Dabei wird zu Beginn kurz auf ähnliche bzw. vergleichbare Systeme anderer Ligen und Teams eingegangen (Abschnitt 4.1). Im Hauptteil werden die Anforderungen für ein synchronisiertes Aufnahmesystem (Abschnitt 4.2), die Evaluation der eingesetzten Hardware und die Implementierung des gesamten Systems beschrieben (Abschnitt 4.3). Abgeschlossen wird das Kapitel mit der empirischen Auswertung zum aktiven Einsatz des Aufnahmesystems während der GermanOpen 2018 in Magdeburg und des RoboCups 2018 in Montreal, Kanada (Abschnitt 4.4). Zum Schluss wird in Kapitel 5 die Arbeit zusammengefasst und zukünftige Entwicklungen diskutiert, sowie die Integration bereits bestehender Arbeiten angeregt.

2 Infrastruktur der SPL im RoboCup

Der regelkonforme (technische) Aufbau für ein SPL-Wettkampfspiel im RoboCup ist in Abbildung 2.1 schematisch dargestellt. Er besteht aus dem Spielfeld, einem GameController, einem WLAN-Netzwerk und den eingesetzten Robotern. Die Roboter kommunizieren zum einen über das Netzwerk untereinander und tauschen Informationen über ihre Wahrnehmung und Strategie aus. Zum anderen kommunizieren die Roboter mit dem GameController und erhalten von diesem Informationen bzgl. des Spielstandes, der Spielzeit und Strafzeiten der einzelnen Spieler.

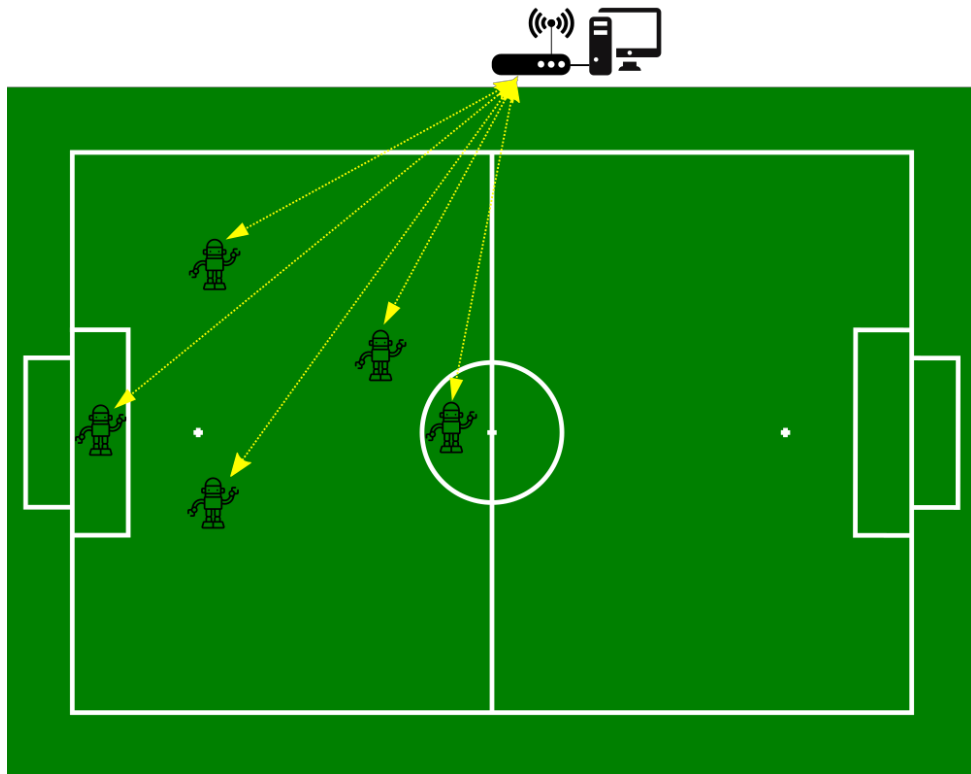


Abbildung 2.1: Schematische Darstellung eines SPL-Feldaufbaus

Im Folgenden werden die wichtigsten Komponenten des Aufbaus beschrieben.

2.1 GameController



Abbildung 2.2: Ansicht des GameController nach dem Starten und vor einem Spiel

Der GameController¹ ist die Schnittstelle zwischen dem menschlichen Schiedsrichter und den Robotern. Er wird vom Assistenten des Schiedsrichters bedient und kommuniziert dessen Entscheidungen an die Roboter. Entwickelt wird die Java-basierte Anwendung vom Team „B-Human“² aus Bremen und wird jährlich gemeinsam mit den SPL-Regeln veröffentlicht [9].

Der GameController kommuniziert mit den Robotern mehrmals pro Sekunde über das WLAN-Netzwerk mittels UDP-Broadcasts. Eine Nachricht („RoboCupGameControlData“) enthält Informationen zum Spielzustand und die einzelnen Strafzeiten der eingesetzten Roboter.

Die Roboter müssen mit einer definierten Nachricht antworten („RoboCupGameControlReturnData“), um ihre Funktionstüchtigkeit zu signalisieren. Bleibt eine solche Antwort aus, wird der Spieler als „inaktiv“ angezeigt.

Der kommunizierte Spielzustand enthält neben den spielenden Teams, der aktuellen Spielzeit und dem Punktestand insbesondere auch den „Zustand“ des Spiel. Folgende Zustände sind in den Regeln definiert:

- **Initial:** vor Spielbeginn müssen alle Roboter gestartet und am Seitenrand aufgestellt werden
- **Ready:** die eingesetzten Roboter laufen auf ihre Startposition; dazu haben sie maximal 45 Sekunden Zeit

¹<https://github.com/bhuman/GameController> (01.08.2018)

²<http://www.b-human.de> (13.09.2018)

- **Set:** die Roboter dürfen sich nicht mehr bewegen, es werden Korrekturen oder Strafen vom Schiedsrichter vorgenommen
- **Playing:** das Spiel wurde angepfiffen und die Spielzeit läuft
- **Penalized:** gilt nur je Roboter und wird bei einem Regelverstoß bzw. Foul verhängt
- **Finished:** die Spielzeit ist abgelaufen bzw. das Spiel wurde vom Schiedsrichter beendet, die Roboter sollten aufhören sich zu bewegen

Jeder eingesetzte Spieler sollte auf diese Zustände reagieren, ansonsten darf er nicht am Spiel teilnehmen.

Neben der eigentlichen Spielkontrolle zeichnet der GameController alle gesendeten und empfangenen Daten in einer Logdatei auf, insbesondere die Kommunikation der Spieler untereinander.

2.2 Teamkommunikation

Während eines Spiels ist den Robotern nur erlaubt untereinander und mit dem GameController zu kommunizieren. Die Kommunikation läuft über ein WLAN-Netzwerk, das für jedes Spiel bereitgestellt wird. Dabei erfolgt die Datenübertragung mittels des UDP-Protokolls und eine Nachricht darf nur drei Mal pro Sekunde je Roboter gesendet werden.

Die SPL-Regeln legen außerdem das Datenformat für die Nachrichten fest. Neben den 8 Standardfeldern, dienen die Nachrichtfelder 9 und 10 für die teamspezifischen Informationen (Vergleich Abbildung 2.3). Jedes Team kann dort nach eigenem Ermessen zusätzliche Daten übertragen.

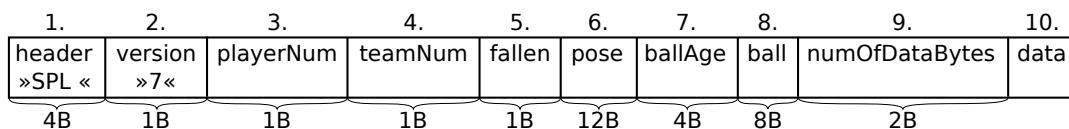


Abbildung 2.3: Format des standardisierten Nachrichtenteils

Das Nao-Team Humboldt verwendet den variablen Datenbereich (10. Feld) zur Übertragung von Informationen für die Visualisierung bzw. den Zustand des Roboters. Zum Beispiel werden Batteriezustand, max. Gelenk- und CPU-Temperatur, Pfeifenerkennung und Teamball rein zur Beurteilung des Roboterzustands bzw. der Funktionsweise der eingesetzten Algorithmen verwendet. Zum anderen werden im variablen Datenbereich Informationen übertragen, die für die Rollenabstimmung und Strategieentscheidung der

Roboter untereinander notwendig sind. Dazu zählen insbesondere Zeitsynchronisierungsinformationen und Stürmer-Entscheidungen des jeweiligen Roboters. Auf Grundlage dieser Informationen stimmen sich die Roboter über die Ballposition ab und wer an den Ball gehen sollte.

Für das Format im variablen Datenbereich wird Googles Protobuf³ verwendet (siehe dazu folgenden Abschnitt 2.3). Die zu kommunizierenden Daten werden dafür mittels Protobuf in eine binäre Zeichenkette konvertiert und in den variablen Bereich der Nachricht übertragen.

Standardmäßig werden alle von einem Roboter empfangenen und gesendete Nachrichten während eines Wettkampfspiels protokolliert (siehe dazu auch folgenden Abschnitt 2.3). Damit ist es möglich die Kommunikation im Nachhinein nachzuvollziehen und darauf aufbauende Entscheidungen des Roboters zu bewerten.

2.3 Logdateien

Die Infrastruktur und das Framework des Nao-Team Humboldts basiert auf einem Modul- und Repräsentationskonzept. Die Idee hierbei ist, dass einzelne Funktionseinheiten in Module gekapselt werden. Diese Module können bestimmte Repräsentationen verlangen und wiederum eigene Repräsentationen für andere Module zur Verfügung stellen. Repräsentation sind dabei einfache Datenobjekte die zur „Kommunikation“ zwischen Modulen dienen und nur einfache Funktionen implementieren (siehe Abbildung 2.4)[6].

Damit Probleme nach einem Spiel oder einem Test analysiert werden können, existiert eine Log-Infrastruktur. Sie ermöglicht es, den Verhaltensgraphen des Roboters und die verwendeten Repräsentationen aufzuzeichnen. Dazu werden ausgewählte Repräsentation serialisiert und in eine Logdatei geschrieben. Die erstellten Logdateien lassen sich anschließend dahingehend auswerten, welche Daten als Eingabe für ein Modul dienten und welche Ausgaben daraus erzeugt wurden.

Als Datenformat wird dafür Googles Protobuf verwendet. Damit ist es besonders einfach das Datenformat zu spezifizieren. Außerdem unterstützt Protobuf eine Vielzahl von Programmiersprachen in denen die Logdateien gelesen werden können. Zusätzlich ermöglicht die Formatspezifikation erweiterbare und versionierbare Formate zu entwickeln.

Die Aufzeichnung einer Logdatei erfolgt frameweise und enthält alle ausgewählten, serialisierbaren Repräsentationen. Ein Frame stellt dabei einen Ausführungszyklus des Roboters dar. Die serialisierbaren Repräsentationen implementieren eine Serialisierungsfunktion, welche die Daten der Repräsentation in das Protobufformat überträgt. Gespeichert

³<https://developers.google.com/protocol-buffers/> (27.08.2018)

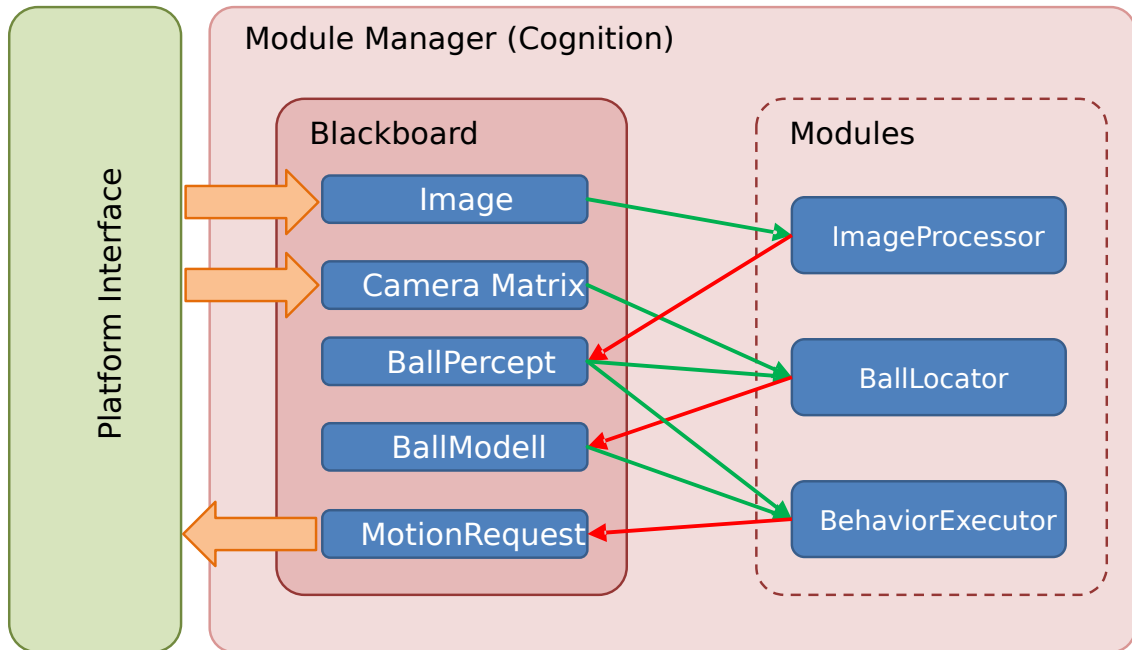


Abbildung 2.4: Übersicht der verschiedenen Komponenten des Modul-Frameworks; das **Blackboard** dient dabei als Ablageort für die Repräsentationen; der **Module Manager** ist für die Ausführung der Module zuständig [6]

wird die von Protobuf serialisierte binäre Repräsentation in einer Logdatei, wobei die Daten nach folgender Struktur gespeichert werden:

```
<frame number><representation name><size><data>
```

Die „frame number“ gibt die fortlaufende Nummer des Frames an, „representation name“ den Name der Repräsentation, „size“ die Größe der folgenden Daten und „data“ die mit Protobuf serialisierten Daten.

Der Roboter zeichnet standardmäßig eine bestimmte Auswahl an Repräsentationen während eines Spiels auf, die nach dem Spiel vom Roboter mit einem USB-Stick eingesammelt werden kann. Außerdem lassen sich für Testzwecke, mittels des „LogfileRecorder“-Dialogs von RobotControl, eine bestimmte Auswahl an verfügbaren Repräsentationen auswählen, in eine Logdatei aufnehmen und anschließend per USB-Stick oder über eine Netzwerkverbindung kopieren.

2 Infrastruktur der SPL im RoboCup

Die gesammelten Logdateien können anschließend einzeln ausgewertet werden, wobei folgende Varianten möglich sind:

- Logsimulator
- RobotControl (LogfileInspector/LogfilePlayer/VideoAnalyzer)
- VideoLogLabeling-Tools
- Python oder eine andere Programmiersprache, für die Protobuf einen (De-)Serialisierer generieren kann

Ein weiterer Nutzen dieser Logdateien liegt in der Auswertung und im Vergleich relevanter Kennzahlen, sowohl zwischen verschiedenen Algorithmen als auch zwischen einzelnen Robotern und (Wettkampf-)Spielen. Zum Beispiel lassen sich verschiedene Stabilisierungsvarianten anhand der Häufigkeit vergleichen, wie oft die Roboter während eines Spiels hingefallen sind. Oder es lassen sich die Balldetektoren in verschiedenen Spielen bewerten – je nachdem wie oft und lange ein Ball gesehen wurde.

3 Werkzeuge zur Auswertung und Analyse

Im Nao-Team Humboldt ist „RobotControl“ das Hauptwerkzeug zur Konfiguration, zum Debugging und zur Auswertung von Roboterdaten – sowohl „online“, während der Roboter gerade läuft, als auch „offline“, wenn Logdateien ausgewertet werden.

Eines der neusten Auswerte- und Analyse-Tools ist das „VideoLogLabeling“-Tools, das im Rahmen einer wissenschaftlichen Arbeit im Nao-Team Humboldt entstanden ist [5] und im Rahmen dieser Arbeit weiterentwickelt wurde. Mit diesem Werkzeug lassen sich bestimmte Situationen annotieren und anschließend genauer auswerten.

Beide Analysewerkzeuge werden im Weiteren ausführlicher vorgestellt.

3.1 RobotControl

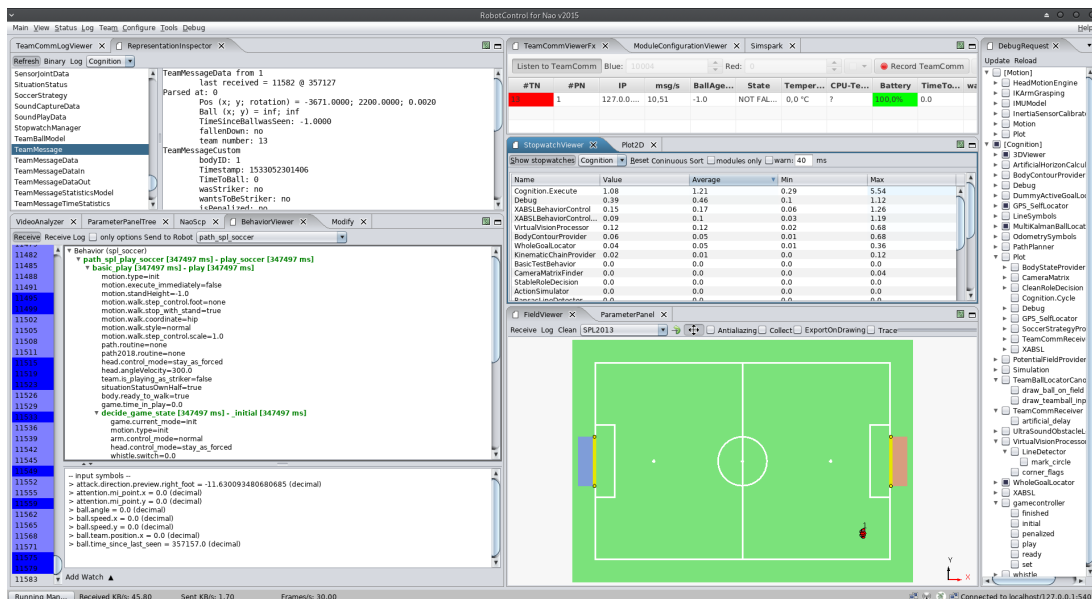


Abbildung 3.1: RobotControl 2018, es werden verschiedene Dialoge gezeigt: RepresentationInspector, BehaviorViewer, TeamCommViewer, StopwatchViewer, FieldViewer, DebugRequest

3 Werkzeuge zur Auswertung und Analyse

RobotControl ist das plattformübergreifende, Java-basierte Hauptwerkzeug bei der Arbeit mit den Robotern im Nao-Team Humboldt. Es ist seit vielen Jahren im Einsatz und wird stetig weiterentwickelt und an neue Entwicklungen angepasst. Dabei ist das Grundkonzept, einen Rahmen für verschiedene Dialoge zu bieten, gleich geblieben. Dieser Rahmen wird durch ein Plugin-System realisiert und ermöglicht es neue Dialoge zu entwickeln und auf einfache Weise zu integrieren. Die einzelnen Dialoge können auch von einander abhängen, bzw. Daten für andere Dialog bereitstellen, die wiederum über so genannte Manager an die einzelnen Dialoge verteilt werden. Zusätzlich sind externe Java-Anwendungen in RobotControl integriert, um die Arbeit mit den Robotern zu erleichtern und alle notwendigen Werkzeuge an einem Ort zu behalten. Die externen Java-Anwendungen wurden dabei so modifiziert, dass sie weiterhin als eigenständige Programme funktionieren und gleichzeitig als Dialoge in RobotControl eingebunden werden können.

Die einzelnen Dialoge sind in Kategorien zusammengefasst und lassen sich über das Menü aufrufen. Im Weiteren werden nur die für Auswertungen und Analyse verfügbaren Dialoge stichpunktartig beschrieben.

- **BehaviorViewer:** zeigt das aktuell auf dem Roboter laufende Verhalten, den Zustandsbaum, sowie die vom Verhalten verwendete Symbole an; wird eine Logdatei abgespielt, wird das entsprechend in der Logdatei aufgenommene Verhalten angezeigt
- **RepresentationInspector:** zeigt die vom Roboter verwendeten Repräsentationen und deren enthaltenen Werte an; wird eine Logdatei abgespielt, werden die in der Logdatei aufgenommenen Daten angezeigt
- **RobotHealth:** zeigt die aktuellen Temperaturen der Robotergelenke und die Werte der Drucksensoren im Fuß des Roboters an
- **StopWatchViewer:** die Ausführungszeit der einzelnen Module wird hier tabellarisch dargestellt
- **3DViewer:** visualisiert den Roboter und dessen (Gelenk-)Haltung im dreidimensionalen Raum
- **FieldViewer:** zeigt das Spielfeld und die (gedachte) Roboterposition, sowie weitere relevante Informationen der einzelnen Roboter; Logdateien und direkt verbundene Roboter können zusätzlich ausgewählte Informationen in das Feld einzeichnen
- **ImageViewer:** zeigt das Kamerabild des Roboters an (obere und untere Kamera), sowie aktivierte Zeichnungen darin
- **Plot2D:** zeichnet Graphen von ausgewählten numerischen Werten

3 Werkzeuge zur Auswertung und Analyse

- **LogFileRecorder:** ermöglicht die Aufnahme und Speicherung ausgewählter Repräsentationen
- **LogFileInspector:** stellt eine schnelle Variante zur Inspektion von Logdateien zur Verfügung und gibt extrahierte Informationen an andere Dialoge weiter
- **LogFilePlayer:** ermöglicht die Steuerung des LogSimulators. Der LogSimulator ist ein Programm, das ähnlich wie das Programm für den Roboter funktioniert, aber im Gegensatz dazu seine Informationen aus Logdateien zieht. Damit lassen sich neue Algorithmen auf Logdateien testen, ohne dafür einen Roboter zu verwenden.
- **TeamCommLogViewer:** die von einer „externen“ Instanz (z.B. GameController) aufgezeichneten Nachrichten können hier abgespielt und die enthaltenen Informationen angezeigt, sowie im *FieldViewer* visualisiert werden

Besonders hervorzuheben ist der „TeamCommViewer“, der zur Analyse des Teamverhaltens und -strategie eingesetzt wird (Abbildung 3.2). Er wurde vollständig neu mit einer JavaFX-Oberfläche entwickelt und zeigt die kommunizierten Informationen der Roboter von beiden Teams tabellarisch an. Zusätzlich werden für jeden Roboter bestimmte Informationen im FieldViewer eingezeichnet, z.B. dessen Position auf dem Feld, die Ballposition, die Position des Teamballs und die Entscheidung ob der Roboter „Stürmer“ ist. Anhand dieser Informationen lässt sich beurteilen, ob die Rollenentscheidung der Spieler der gewünschten entspricht oder ob in der Abstimmung Fehler aufgetreten sind.

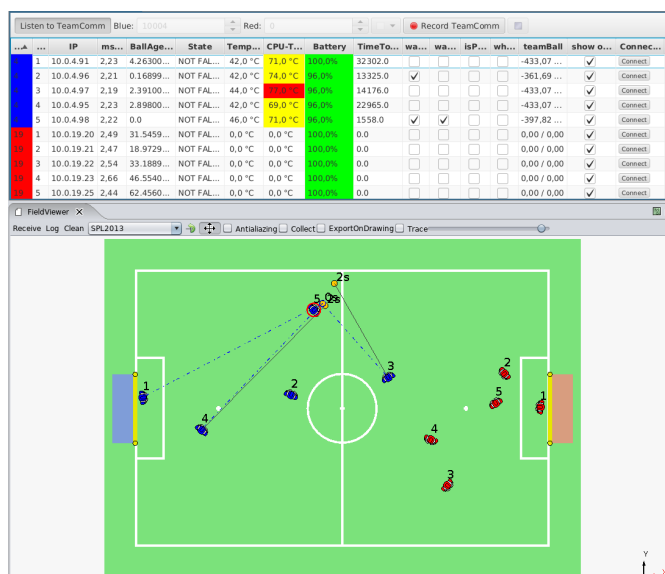


Abbildung 3.2: Ansicht des TeamCommViewers und des FieldViewers während eines Spiels.

3 Werkzeuge zur Auswertung und Analyse



Abbildung 3.3: Synchronisierung eines Videos mit einer Logdatei im VideoAnalyzer-Dialog von RobotControl.

Der „VideoAnalyzer“ ist ein weiterer Dialog, der hervorgehoben werden sollte (Abbildung 3.3). Mit diesem können unsynchronisierte Logdateien und Videos aneinander angeglichen werden. Unsynchronisierte Videos beginnen die Aufnahme mehrere Sekunden bis Minuten vor dem Spiel. Entsprechend können diese nicht gleichzeitig mit einer Logdatei abgespielt werden, das Bild und die aufgezeichnete Informationen aus der Logdatei stimmen nicht überein. Mit dem VideoAnalyzer kann ein Synchronisationspunkt gewählt werden, an dem beide Dateien ausgerichtet werden sollen.

Mit den bisher verfügbaren Dialogen ist es nur möglich jeweils einen Roboter zu konfigurieren bzw. entsprechend nur eine Logdatei auszuwerten.

3.2 VideoLogLabeling

Das „VideoLogLabeling“-Tool¹ entstand im Rahmen der Forschungsarbeit „Simulation Based Selection of Actions for a Humanoid Soccer-Robot“ von Mellmann, Schlotter & Blum und ermöglicht die Annotation von zuvor extrahierten Spielereignissen eines (oder mehrerer) Roboter.

Das Programm ist als PHP-Webapplikation geschrieben und erfordert für den Betrieb eine passende Serverumgebung. Damit ist es möglich kooperativ Wettkampfspiele zu analysieren.

¹<https://www2.informatik.hu-berlin.de/~naoth/videolabeling/> (01.08.2018)

3 Werkzeuge zur Auswertung und Analyse



Abbildung 3.4: Interface zum annotieren von Logdateien

In Abbildung 3.5 ist die Reihenfolge der Datenverarbeitungsschritte konzeptionell dargestellt. Zur Analyse sind neben den Logdateien der Roboter mindestens auch ein entsprechend aufgezeichnetes Video notwendig. Bevor Spiele und bestimmte Situationen mit diesem Programm genauer analysiert werden können, müssen noch einige Vorbereitungen getroffen werden.



Abbildung 3.5: Konzeptionelle Darstellung der Datenverarbeitungsreihenfolge

Zum einen sollten die einzelnen Logdateien jeweils einzeln mit dem Video synchronisiert werden. Dazu wird der „VideoAnalyzer“-Dialog aus RobotControl verwendet und die jeweilige Logdatei, als auch das Video ausgewählt. Es lässt sich ein Ereignisframe auswählen und die Videozeit daran synchronisieren. Der „VideoAnalyzer“ legt daraufhin eine Datei (game.log.videoanalyzer.properties) mit den entsprechenden Informationen an. Anschließend muss ein Python-Skript ausgeführt werden, dass aus den Logdateien bestimmte

3 Werkzeuge zur Auswertung und Analyse

„Ereignisse“ extrahiert und als JSON-Datei speichert (labels.json). Zur Zeit werden nur bestimmte „Kick“-Varianten extrahiert und gespeichert. Das Python-Skript lässt sich aber leicht erweitern, so dass auch andere „Ereignisse“ bzw. interessante Stellen extrahiert werden können.

Die Ordner mit den Logdateien, dem Video und den generierten Dateien müssen anschließend in das Unterverzeichnis „log“ des VideoLogLabeling-Tools abgelegt werden, um die eigentliche Spielanalyse durchzuführen.

Im Zuge der Entwicklung des Aufnahmesystems, wurde auch das VideoLogLabeling-Tool weiterentwickelt. Es wurde um ein Skript zur Konvertierung der Videos, in verschiedene Auflösungen, erweitert. Außerdem wurde das Tool auf eine „Model-View-Controller“-Architektur (MVC) umgestellt. Damit werden Datenobjekte von ihrer Darstellung getrennt und können in verschiedenen Ansichten wiederverwendet werden. Außerdem lässt sich das Tool leichter erweitern und um neue Module ergänzen.

4 System zur synchronen Videoaufnahme

In der SPL-Liga des RoboCups gibt es kein einheitliches, transportables Aufnahmesystem, um Wettkampfspiele außerhalb des Labors aufnehmen zu können. Außerdem müssen die Logdateien der Roboter für die Auswertung mit dem VideoLogLabeling-Tool (Abschnitt 3.2) und dem VideoAnalyzer-Dialog aus RobotControl (Abschnitt 3.1) mit dem Video manuell synchronisiert werden.

In diesem Kapitel wird das entwickelte Aufnahmesystem zur Erstellung synchronisierter Videos vorgestellt. Die Entwicklung wurde durch zwei Projekte der RoboCup-Federation gefördert. Das System ist transportabel, um während verschiedenster Veranstaltungen eingesetzt werden zu können. Es wurde während der GermanOpen 2018 in Magdeburg und während des Robocups 2018 in Montreal eingesetzt und erste Erfahrungen gesammelt.

4.1 Ähnliche Systeme anderer Teams und Ligen

Zeitgleich mit der Entwicklung des Aufnahmesystems haben die Nao Devils aus Dortmund¹ ein Streamingsystem entwickelt, das die Wettkampfspiele auf YouTube live überträgt. Der Schwerpunkt liegt hier auf der Übertragung der Spiele und weniger auf Auswertung und Synchronisation der Aufnahmen mit anderen Logdaten.

Für ihr Streaming-System verwenden die Dortmunder ein Raspberry Pi mit einer „Raspberry Pi Camera“. Der Raspberry verbindet sich mit dem SPL-WLAN-Netzwerk und empfängt von dort die Informationen des GameControllers. Im Rahmen der Team-Kooperation hat das Nao-Team Humboldt die bereits entwickelten GameController-Komponenten den Dortmundern zur Verfügung gestellt. Anhand der empfangenen Informationen wird das übertragene Video mit zusätzlichen Informationen angereichert, bspw. die aktuellen Gegner und der aktuelle Punktestand.

Über die LAN-Verbindung des Raspberry Pis wird die Aufnahme ins Internet (YouTube) gestreamt. Dort sind auch die Aufnahmen des letzten RoboCups abrufbar ².

¹<https://naodevils.de> (07.08.2018)

²https://www.youtube.com/channel/UCp_3raHDiDfTqcIygMrg-0w (07.08.2018)

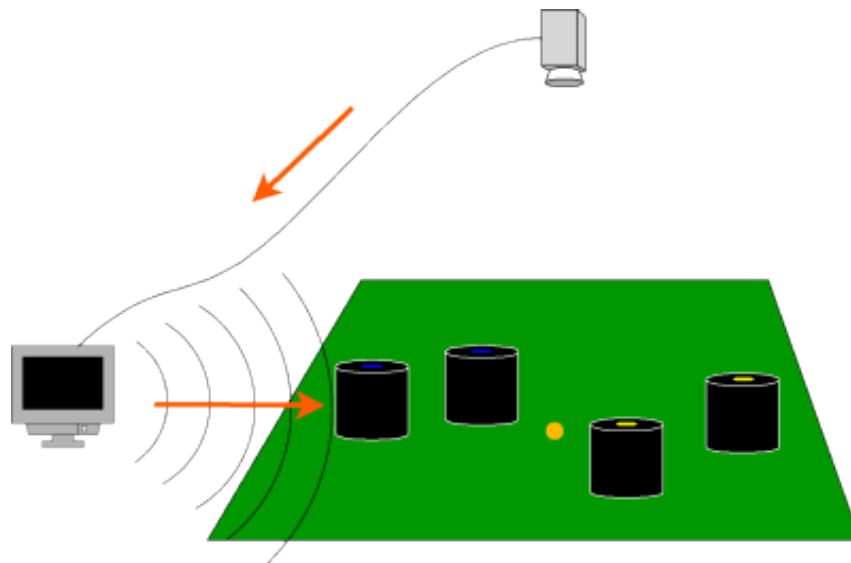


Abbildung 4.1: Der Datenfluss von der Wahrnehmung des Feldes zur Steuerung der Roboter in der Small Size League.

Andere Ligen, neben der SPL-Liga, benötigen ebenfalls visuelle Auswerte- und Analysewerkzeuge. Teilweise sind sie sogar darauf angewiesen, da ohne ein solches System ein Wettkampf nicht stattfinden könnte.

So ist ein Kamerasystem in der Small-Size-Liga obligatorisch und wird seit 2010 einheitlich von einem Unterkomitee bereitgestellt (SSL-Vision³). Bis 2009 musste jedes Small-Size-Team sein eigenes Kamerasystem vor Ort aufsetzen und kalibrieren. Entsprechend konnte nur auf diesem Feld gespielt werden. Mit der steigenden Teamanzahl, bzw. mit der Vergrößerung des Spielfeldes, war der Installations- und Wartungsaufwand der einzelnen Teams nicht mehr praktikabel und man hat sich auf ein einheitliches Kamerasystem geeinigt [12].

Das vereinheitlichte SSL-Vision-System besteht aktuell aus einem Array von Kameras, die 4m über dem Spielfeld fest installiert und von einem (oder mehreren) zentralen Rechner ausgewertet werden. Die extrahierten Informationen werden anschließend an alle Spieler des aktuellen Spiels übertragen. Das gesamte System wird zu Beginn einer Veranstaltung aufgebaut, kalibriert und bei mehreren Auswerterechnern auch synchronisiert. Zur Unterscheidung der einzelnen Spieler sind die Roboter für die Auswertung und Informationsextraktion farbkodiert gekennzeichnet und vereinfachen damit den Aufwand bei der Analyse und Auswertung (siehe dazu Abbildung 4.1 und die aktuellen Regeln der Small-Size-Liga[1]).

³<https://github.com/RoboCup-SSL/ssl-vision/wiki> (07.08.2018)

4.2 Evaluation und Anforderungen

Ein transportables System zur Aufnahme von Wettkämpfen in der SPL-Liga lässt sich mit einer auf einem Stativ befestigten Kamera realisieren. Entsprechende Erfahrung wurde bereits vom Nao-Team Humboldt in den Jahren zuvor gemacht, wobei die Aufnahme immer manuell für ein Spiel gestartet und nach einem Spiel wieder gestoppt wurde. Dieser Aufbau hat sich bewährt und wird für das Aufnahmesystem beibehalten.

Für die Synchronisierung mit den Logdateien sind eindeutige Events oder Zeitangaben notwendig, anhand derer die Synchronisierung durchgeführt werden kann. Das bedeutet, dass die folgenden Systeme die gleichen Zeiten haben sollten:

- Roboter: Zeitstempel für die Logdateien
- GameController: Zeitstempel für die Spielelogs
- Steuerungseinheit für die Kamera: Zeitstempel für die Zuordnung von Video und Mannschaften eines Spiels
- Kamera: Zeitstempel der Videos

Dabei dürfen nicht nur allein die aktiv eingesetzten Komponenten synchronisiert werden, sondern auch Ersatzkomponenten sollten ebenfalls synchronisiert werden, falls eine genutzte Komponente ausfällt.

Eine Möglichkeit zur Synchronisierung ist, die Systemzeit aller beteiligten Komponenten vor Veranstaltungsbeginn auf den gleichen Stand zu bringen, d.h. die Uhrzeiten müssen synchronisiert werden. Je nach Roboteranzahl und Anzahl der aufzuzeichnenden Spielfelder bedeutet das bereits einen hohen initialen Aufwand.

Eine andere Möglichkeit ist die Komponenten relativ zueinander, an bestimmten Zeitpunkten, zu synchronisieren. Dazu bietet sich an, anhand der vom GameController kommunizierten Spielzustände, die Synchronisierung auszurichten. Wird der GameController gestartet, befindet sich das Spiel im „INITIAL“-Zustand und das Ende eines Spiels wird mit dem „FINISH“-Zustand angezeigt. Das heißt, dass zwischen diesen beiden Zuständen das Spiel und damit der für eine Logaufzeichnung relevante Teil stattfindet.

Die Roboter des Nao-Team Humboldts beginnen ihre Logaufzeichnung sobald der Spielzustand in „READY“ wechselt und stoppen die Aufzeichnung im „FINISH“-Zustand. Folglich sollte die Kamera ebenfalls die Aufzeichnung mit „READY“ beginnen und mit „FINISH“ stoppen. Eine Steuereinheit für die Kamera sollte die spielenden Teams und die jeweilige Videodatei speichern, damit eine anschließende Zuordnung leichter möglich ist. Wird eine Aufzeichnung mit diesen relativen Zeitpunkten durchgeführt, ist kein manueller initialer Konfigurationsaufwand notwendig. Das resultierende Video und die vom Roboter

4 System zur synchronen Videoaufnahme

aufgezeichnete Logdatei sind bereits mit den „READY“ und „FINISH“-Spielzuständen synchronisiert.

Wie zu Beginn erwähnt, gibt es im Nao-Team Humboldt bereits Erfahrungen mit der manuellen Aufnahme von Spielen. Diese wurde mit einer auf einem Stativ befestigten GoPro Hero (2014)⁴ aufgenommen, sowie vereinzelt mit einer GoPro Hero 3+⁵. Die gemachten Erfahrungen mit diesen Kameras bildeten die Grundlage zur Evaluation weiterer Kamerasysteme. Folgende Vorteile wurden bei der Verwendung der GoPro ausgemacht:

- hohe Auflösung von 1920x1080 (Full HD) und weitwinkliger Aufnahmebereich zur Erfassung des gesamten Spielfelds
- einfache Bedienung und Steuerung mit einem PC bzw. Raspberry Pi
- viele Teams haben bereits eine GoPro und insgesamt ist die Verwendung einer GoPro in der RoboCup-Community weit verbreitet

In Tabelle 4.1 sind alle evaluierten Kameras und einige ausgewählten Einstellungen aufgelistet.

| Kamerabezeichnung | Auflösung | FOV | FPS | Steuerung über |
|----------------------|-----------|------|-----|--------------------|
| GoPro Hero 3+ Silver | 1920x1080 | 170° | 60 | WLAN |
| GoPro Hero 4 Session | 1920x1440 | 170° | 30 | WLAN |
| | 1920x1080 | 170° | 48 | WLAN |
| | 1920x1080 | 170° | 60 | WLAN |
| GoPro Hero 5 Black | 3840x2160 | 170° | 24 | WLAN |
| | 1920x1080 | 150° | 120 | WLAN |
| | 1920x1080 | 170° | 80 | WLAN |
| Pi Camera v2.1 | 1920x1080 | 74° | 30 | Raspberry Pi (CSI) |
| | 1280x720 | 74° | 60 | Raspberry Pi (CSI) |
| Rollei Actioncam 425 | 3840x2160 | 170° | 25 | WLAN |
| | 1920x1080 | 170° | 60 | WLAN |
| | 1920x1080 | 170° | 30 | WLAN |
| Genius F100 Widecam | 1920x1080 | 120° | 30 | Raspberry Pi (USB) |
| | 1280x720 | 120° | 30 | Raspberry Pi (USB) |

Tabelle 4.1: Evaluation verschiedener Kameras

Im Ergebnis hat sich die GoPro durchgesetzt. Ausschlaggebend dabei war, dass die GoPro den größten Aufnahmebereich, bei gleichzeitig hoher Auflösung (Full HD) und hoher

⁴<https://gopro.com/update/hero> (08.08.2018)

⁵https://gopro.com/update/hero3_plus (08.08.2018)

Aufnahmegeschwindigkeit hat. Außerdem besitzen mehrere Teams bereits eine (neuere) GoPro. Zusätzlich haben Experimente gezeigt, dass sich die GoPro Kameras ab Version Hero 3, über WLAN steuern lassen. Für die Aufnahme während der GermanOpen 2018 in Magdeburg wurde eine GoPro Hero 4 Session gekauft und eingesetzt. Später wurde im Rahmen des von der RoboCup-Federation geförderten Projekts zur Aufnahme beim RoboCup 2018 in Montreal zwei weitere GoPro Hero 4 Sessions und zwei GoPro Hero 5 Black gekauft.

4.3 Umsetzung und Implementierung

Basierend auf der Entscheidung die GoPro als Kamera zu verwenden, ist gleichzeitig auch festgelegt, wie die Kamera zu steuern ist. In Abbildung 4.2 ist der schematische Aufbau des Aufnahmesystems dargestellt. Dieser besteht aus dem normalen SPL-Spielaufbau: einem GameController und den Spielern, die beide miteinander und untereinander kommunizieren (siehe auch Kapitel 2).

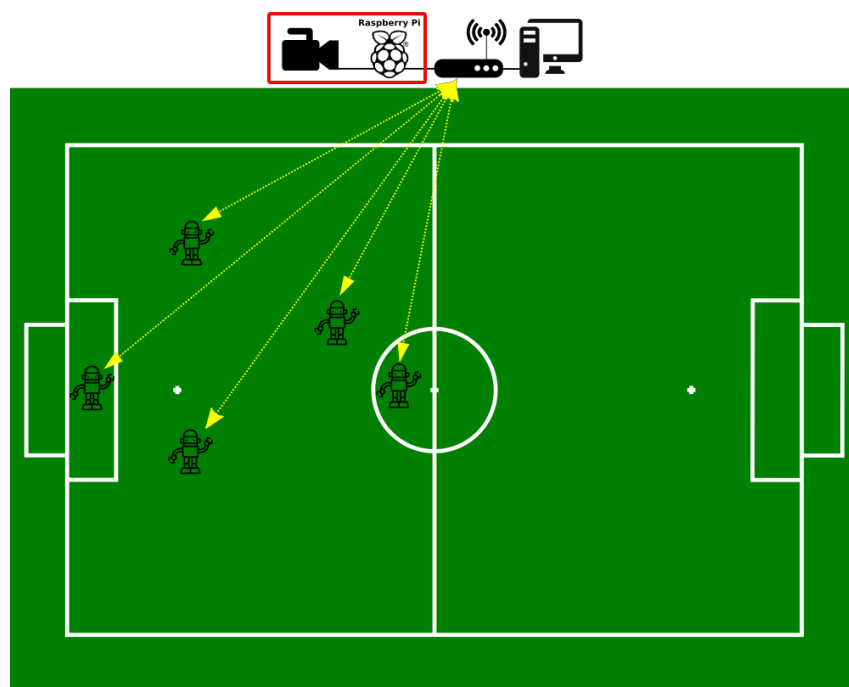


Abbildung 4.2: schematischer Aufbau des Aufnahmesystems

Neu in diesem Aufbau ist das Aufnahmesystem. Es besteht aus einem Raspberry Pi und einer GoPro. Die GoPro und der Raspberry Pi werden dabei auf ein Stativ in ca. 2-3m Höhe installiert, damit das gesamte Spielfeld erfasst und insbesondere Verdeckungen durch Schiedsrichter oder andere Personen vermieden wird. Der Raspberry Pi ist notwendig, da die GoPro ihr eigenes WLAN-Netzwerk aufspannt und nur darüber steuerbar ist. In

4 System zur synchronen Videoaufnahme

das bereits vorhanden SPL-Netzwerk kann die GoPro nicht eingebunden werden, da eine solche Funktion vom Hersteller nicht vorgesehen ist. Der Raspberry Pi sollte in nächster Nähe zur Kamera angebracht werden, um Verbindungsabbrüche durch Frequenzstörungen zu vermeiden. Die Stromversorgung der Kamera wird über einen USB-Anschluss des Raspberry Pis sichergestellt. Dafür wurden die Datenleitungen des USB-Kabels abgeklebt, damit die GoPro den Raspberry Pi nicht als PC erkennt und in den Übertragungsmodus geht. Im Übertragungsmodus kann die Kamera nicht gesteuert werden und keine Videos aufnehmen.

Die „GoPro Hero 4 Session“-Firmware wurde dahingehend untersucht, ob die Kamera über weitere Kanäle (USB-Verbindung, Bluetooth) gesteuert werden kann, bzw. ob es möglich ist, die GoPro in das SPL-Netzwerk einzubinden. Es konnte aber kein anderer Weg gefunden werden, ohne die Firmware der GoPro zu modifizieren. Da dies nicht der Schwerpunkt der Arbeit ist, wurde von weitergehenden Untersuchungen abgesehen.

Zur Steuerung der Kamera wird auf dem Raspberry Pi ein Python-Programm gestartet, das die Verbindung sowohl zum GameController, als auch zur GoPro herstellt. Dazu muss der Raspberry Pi mit dem SPL-Netzwerk (via LAN-Kabel) und mit dem WLAN-Netzwerk der GoPro verbunden werden. Für das SPL-Netzwerk wird eine statische IP festgelegt, um Fehlkonfigurationen zu vermeiden. Bei bestehender Netzwerkverbindung sollte der Raspberry Pi die vom GameController kommunizierten Nachrichten empfangen.

Da jede GoPro ihr eigenes WLAN-Netzwerk bereitstellt, sollte der Raspberry Pi konfigurierbar bleiben, damit er sich mit verschiedenen GoPros verbinden kann. Entsprechend wird in einer Konfigurationsdatei festgelegt, mit welcher GoPro sich der Raspberry Pi verbindet.

Das Python-Programm hat die folgenden Aufgaben:

- Verbindung mit der GoPro herstellen (unter Umständen WLAN aktivieren) und überwachen
- Nachrichten vom GameController empfangen und Spielzustand verfolgen
- anhand des Spielzustands die Videoaufnahme starten bzw. stoppen
- Protokoll über die Spiele und aufgezeichneten Videos führen
- den aktuellen Zustand bzw. Status von sich selbst anzeigen (zur Fehlersuche und -erkennung)

Dabei besteht das Programm aus 5 Softwarekomponenten, die über ein Eventsystem miteinander Daten austauschen.

4 System zur synchronen Videoaufnahme

Die Netzwerkkomponente ist für den Aufbau und Überwachung der Verbindung zur GoPro verantwortlich. Dabei versucht sich die Komponente anhand der gegebenen Konfiguration mit dem GoPro-WLAN-Netzwerk zu verbinden und bei Abbruch der Verbindung diese wiederherzustellen.

Die GameController-Komponente ist dafür verantwortlich den Spielzustand vom GameController zu überwachen und die abhängigen Komponenten über eine Änderungen im Spiel zu informieren.

Die GoPro-Komponente ist für die Steuerung der GoPro verantwortlich. Sie übernimmt das Starten der Aufnahme, sobald der Spielzustand entweder READY, SET oder PLAY ist und stoppt die GoPro-Aufnahme, wenn keiner der genannten Zustände aktiv ist. Insbesondere stoppt sie die Kamera im FINISH-Zustand, ermittelt den Dateinamen der letzten Aufnahme auf der GoPro und übergibt diesen an die Log-Komponente.

Die zwei zusätzlichen Komponenten sind für das aktive Debugging bzw. die anschließende Zuordnung von Dateinamen und Spiel. Die Log-Komponente dient dabei der Zuordnung von Dateinamen zu den Spielen (Wer, wann, wo gegen wen gespielt hat).

Die LED-Komponente dient dazu den Zustand der Software anzuzeigen. Dabei zeigen Fehlercodes, an welcher Stelle im Ablauf es gerade zu Problemen kommt bzw. ob das System gerade aufnimmt oder nicht. Dazu wurde der Raspberry Pi um drei LEDs (rot, grün, blau) erweitert, die von einem Python-LED-Programm angesteuert werden (siehe Abbildung 4.3). Die LED-Komponente ist als Service realisiert und läuft unabhängig von der Steuerung der GoPro bzw. Verbindung zum Netzwerk. Die Kommunikation zwischen LED-Server und GoPro-Steuerungsprogramm findet über Sockets statt.

Während der Entwicklung des Systems fiel ein großer Unterschied zwischen den zwei verwendeten GoPro-Varianten erst im Laufe der Zeit auf. Die GoPro Hero 4 Session hält ihre WLAN-Verbindung solange aktiviert, bis diese entweder ausgeschaltet wird oder der Akku der Kamera leer ist. Im Gegensatz dazu schaltet die GoPro Hero 5 Black ihr WLAN nach einiger Zeit aus Stromspargründen ab und muss erneut aktiviert werden.

Für den aktiven (unbeobachteten) Einsatz kann die Hero 5 nicht ohne zusätzlichen Aufwand verwendet werden. Eine Reaktivierung ist nur manuell oder über eine separate Bluetooth-Verbindung möglich. Folglich wurde das Python-Programm so erweitert, dass eine Bluetooth-Komponente das WLAN der Hero 5 Kamera wieder aktiviert, sobald dieses sich ausschaltet. Im Testbetrieb hat sich dieser Aufbau als erfolgreich bewiesen.

Durch die Verwendung der Bluetoothverbindung, entsteht allerdings für die Erstkonfiguration zusätzlicher Aufwand. Der Raspberry Pi und die GoPro Hero 5 Black müssen vor dem ersten Einsatz miteinander gepairt werden. Dieser Schritt ließ sich nicht automatisieren, da dazu die manuelle Bedienung der Kamera notwendig ist. Die GoPro Hero 5 Black wurde mit dem erweiterten System erstmals beim RoboCup 2018 eingesetzt.



Abbildung 4.3: Modifizierter Raspberry Pi zur Statusanzeige

Das gesamte Aufnahmesystem wurde im Rahmen der geförderten RoboCup-Projekte veröffentlicht und anderen Teams zur Verfügung gestellt⁶.

4.4 Empirische Erfahrungen im realen Einsatz

Das beschriebene Aufnahmesystem wurde bereits erfolgreich während der GermanOpen 2018 in Magdeburg und während des RoboCups 2018 in Montreal eingesetzt. Außerdem haben die HULKs aus Hamburg eine Vorabversion des Systems im Rahmen einer Kooperation auch während den IranOpen 2018 eingesetzt. Die dort gewonnenen Erkenntnisse dienen zur Verbesserung des Systems.

In Magdeburg wurde auf einem Feld gespielt, folglich wurde das System nur einmal mit einer GoPro Hero 4 Session aufgebaut. Die Gegebenheit vor Ort waren insbesondere von Vorteil, da ein Gerüst zur Verfügung stand, auf dem das Aufnahmesystem in ca. 4m Höhe montiert werden konnte. Entsprechend gut war der Aufnahmebereich – das gesamte Spielfeld konnte aufgenommen werden und es gab kaum Verdeckung durch die Schiedsrichter oder andere Personen (siehe Abbildung 4.4). Von Nachteil erwies sich der

⁶<https://github.com/BerlinUnited/RoboCupTools> (09.08.2018)

4 System zur synchronen Videoaufnahme



Abbildung 4.4: Schnappschuss von der GoPro Hero Session 4 während des Spiels „Berlin United - NaoTH“ gegen „Nomadz“ in der 2. Halbzeit. Das gesamte Spielfeld wird erfasst und auf dem Feld befindliche Schiedsrichter verdecken nur einen kleinen Teil des Feldes.

erhöhte Aufbau in soweit, dass es nicht so leicht war die SD-Karte am Ende eines Tages zu tauschen und die Aufnahmen zu sichern.

Insgesamt wurden während der GermanOpen 23 Wettkampfspiele durchgeführt. Von denen konnten alle Spiele vollständig aufgezeichnet und den interessierten Teams anschließend vor Ort und auf YouTube⁷ zur Verfügung gestellt werden.

Beim RoboCup 2018 in Montreal wurde auf 5 Feldern (A-E) gespielt. Damit war der Aufbau und die Betreuung während der Wettkämpfe herausfordernder, insbesondere weil das Team zum großen Teil mit der Vorbereitung der eigenen Wettkampfspiele beschäftigt war.

Für ein Aufnahmesystem, das mehrere Komponenten mit Strom zu versorgen hat, sollte vor allem darauf geachtet werden, dass am Spielfeld ausreichend Steckplätze vorhanden sind, um die erforderlichen Komponenten zu betreiben. Der RoboCup 2018 fand in Montreal/Kanada statt, folglich sollten für alle Komponenten passende Stromadapter vorhanden sein.

Ein Problem ist erst im laufenden Wettkampf aufgefallen: die Teams und Teilnehmer benötigen am Spielfeldrand ebenfalls Strom. Auf Grund der knappen – vom Organisa-

⁷https://www.youtube.com/watch?v=ijoAK3cmqUI&list=PLVoczrk_MzTy6zvTtRyeYprXvTu-zsjit (06.09.2018)

4 System zur synchronen Videoaufnahme

tor nicht ausreichend zur Verfügung gestellten – Steckdosen, wurde in einigen Fällen der Strom vom Aufnahmesystem entfernt. Dementsprechend kam es zu nicht aufgenommenen Spielen. Obwohl die Teamleiter über das Aufnahmesystem und der damit benötigten Strom und LAN-Verbindungen informiert wurden, kam es trotzdem zu weiteren Ausfällen. Erst das abkleben der Stecker mit Klebeband und einem offen angebrachten Hinweisschild hat dieses Problem einigermaßen beheben können.

Unvorhersehbar war, dass die mit einer GoPro Hero 5 Black betriebenen Varianten häufig ausgefallen sind. Dies lag zum einen daran, dass die Kamera ihr WLAN aus Stromspargründen ausschaltet und das die Kamera mehr Strom zur Aufnahme und Aufrechterhaltung der WLAN-Verbindung benötigt, als ihr zur Verfügung gestellt werden konnte.

Obwohl die Reaktivierung des WLAN zuvor im Labor getestet und als einsatzbereit eingeschätzt wurde (siehe Abschnitt 4.3), ereigneten sich trotzdem spontane Verbindungsabbrüche, die bisher nicht nachvollzogen und aufgeklärt werden konnten. Die Vermutung ist, dass es auf Grund der gleichen Sendefrequenzbereiche von WLAN (2,400 - 2,483 GHz) und Bluetooth (2,402 - 2,480 GHz) und der großen Anzahl von WLANs bzw. der sendenden Geräte, es zu spontanen Abbrüchen kam und keine neue Verbindung hergestellt werden konnte. Auch können zum Beispiel auf Messen eingesetzte Schnurlostelefone oder von Teams verwendete Mikrowellenherde, die im selben Frequenzband arbeiten, zu Störungen führen und die Herstellung einer neuen Verbindung behindern.

Das Versorgungsproblem trat tatsächlich erst während der letzten zwei Wettkampftage auf. Auch der Wechsel auf eine externe Stromanbindung der Kamera und Tests verschiedener Adapter hat nicht zum Ergebnis geführt, dass die Kamera für den dauerhaften Betrieb ausreichend Strom bekommen hat.

Auf den insgesamt 5 Feldern wurden die folgenden Kameras eingesetzt:

| Feld | Kamera |
|-------------|----------------------|
| A | GoPro Hero 4 Session |
| B | GoPro Hero 4 Session |
| C | GoPro Hero 5 Black |
| D | GoPro Hero 5 Black |
| E | GoPro Hero 4 Session |

Tabelle 4.2: Feld/Kamera-Zuordnung

4 System zur synchronen Videoaufnahme

Die aufgezeichneten Spiele wurden den interessierten Teams bereits während und nach dem Wettkampf zur Verfügung gestellt. Nach der Aufbereitung und Auswertung der Videos, wurden diese auf YouTube hochgeladen⁸.

Die Auswertung der aufgenommenen Spiele ergab, dass von den insgesamt 76 Spielen:

- 54 Spiele vollständig aufgenommen wurden
- 9 Spiele teilweise aufgenommen wurden, wobei manchmal nur wenige Sekunden fehlen, aber auch ganze Halbzeiten
- 13 Spiele nicht aufgenommen wurden

In Tabelle 4.3 wird detailliert gezeigt, welche Spiele vollständig aufgezeichnet wurden, teilweise fehlen oder ganz fehlen. In der Tabelle ist zu sehen, dass in den ersten Tagen mehr Spiele nicht aufgezeichnet wurden bzw. teilweise nicht aufgezeichnet wurden, als in den letzten zwei Tagen. Das ist vor allem auf die Entfernung der Stromversorgung bzw. Unterbrechung der Netzwerkverbindung zum GameController durch die Teams am Spielfeld zu erklären. Hingegen sind die fehlenden bzw. teilweisen Spiele auf Feld D am dritten Tag auf die unzureichende Stromversorgung der GoPro Hero 5 Black zurückzuführen. Außerdem sind die nur teilweise aufgezeichneten Spiele auf den Feldern C und D mit Verbindungsabbrüche zwischen Raspberry Pi und GoPro Hero 5 Black erklärbar. Im Nachhinein nicht nachvollziehbar ist hingegen, warum am dritten Tag auf Feld B und am letzten Wettkampftag auf Feld A jeweils ein Spiel nicht aufgezeichnet wurde – dort kamen die zuverlässig laufenden GoPro Hero 4 Sessions zum Einsatz.

Zusammengefasst lassen sich die folgenden Punkte für eine (größere) Veranstaltung festhalten:

- für ausreichende Stromversorgung am Feld und evtl. benötigte Adapter sorgen
- die Stecker (LAN und Strom) ausreichend sichern (bspw. Klebeband), um unbeabsichtigtes Herausziehen zu vermeiden
- die teilnehmenden Teams darüber informieren, dass ein Aufnahmesystem installiert wurde und die Stecker nicht entfernt werden dürfen

⁸https://www.youtube.com/watch?v=F6_YM11_H9Y&list=PLVoczrk_MzTyjs3My-P8ro-V-LNLamG0u (06.09.2018),

https://www.youtube.com/watch?v=syA3tKSXZmM&list=PLVoczrk_MzTwtX370A9FreobGbJu46c1Y (06.09.2018),

https://www.youtube.com/watch?v=Pwrbwwe3qv4&list=PLVoczrk_MzTwcrpQ1t60NjBfx54Bp0g2I (06.09.2018),

https://www.youtube.com/watch?v=gWxsrinJQ7k&list=PLVoczrk_MzTxbw5u8jqr6yiCn3sxuFiV4 (06.09.2018)

4 System zur synchronen Videoaufnahme

| Datum | Feld | fehlt | teilweise | vollständig | Summe |
|--------------------|--------------|-----------|-----------|-------------|-----------|
| 18.06.2018 | A | 1 | | 4 | 5 |
| | B | 3 | | 2 | 5 |
| | C | | 3 | 2 | 5 |
| | D | | 1 | 3 | 4 |
| | E | 2 | | 2 | 4 |
| <i>Summe</i> | | <i>6</i> | <i>4</i> | <i>13</i> | <i>23</i> |
| 19.06.2018 | A | 1 | | 5 | 6 |
| | B | | | 5 | 5 |
| | C | 2 | 2 | 3 | 7 |
| | D | | 1 | 2 | 3 |
| | E | | | 4 | 4 |
| <i>Summe</i> | | <i>3</i> | <i>3</i> | <i>19</i> | <i>25</i> |
| 20.06.2018 | A | | | 6 | 6 |
| | B | 1 | | 5 | 6 |
| | C | | | 4 | 4 |
| | D | 2 | 2 | 1 | 5 |
| | <i>Summe</i> | | <i>3</i> | <i>2</i> | <i>16</i> |
| 21.06.2018 | A | 1 | | 4 | 5 |
| | D | | | 2 | 2 |
| | <i>Summe</i> | | <i>1</i> | <i>6</i> | <i>7</i> |
| Gesamtsumme | | 13 | 9 | 54 | 76 |

Tabelle 4.3: Auswertung der Videoaufzeichnungen während des RoboCup 2018 in Montreal

4 System zur synchronen Videoaufnahme

- in regelmäßigen Abständen prüfen, ob ein Fehler aufgetreten ist (Hardware-, Software- oder Benutzerfehler)
- zur Zeit wird die Verwendung der GoPro Hero 5 Black auf Grund der ungelösten Verbindungsabbrüche und mangelhaften Stromversorgung nicht empfohlen

Abschließend lässt sich sagen, dass in nächster Zeit nur GoPro Hero 4 Sessions zum Einsatz kommen sollten und das weitere Tests in außerordentlichen Situationen, die insbesondere die WLAN- bzw. Bluetooth-Verbindung belasten, durchgeführt und aufgezeichnet werden sollten.

5 Zusammenfassung und Diskussion

Mit der Implementierung des Aufnahmesystems ist ein weiterer Schritt getan, die Auswertung von Spielen und Logdateien zu vereinfachen und zu beschleunigen. In dieser Arbeit wurde ein System zur Aufnahme von synchronisierten Videos entworfen, implementiert und erste Erfahrungen im Einsatz beschrieben. Das System ist portabel und flexibel und damit jederzeit an den unterschiedlichsten Wettkampforten einsetzbar.

In Kapitel 2 wurde die vorhandene Infrastruktur der Liga und des Nao-Team Humboldt beschrieben. Anschließend wurde in Kapitel 3 die vorhandenen Werkzeuge zur Analyse und Auswertung von Logdateien und aufgezeichneten Spielen dargestellt. Im Hauptteil (Kapitel 4) wurde dargelegt welche Anforderungen an ein Aufnahmesystem für synchronisierte Videos bestehen und warum sich für die GoPro als Aufnahmekamera entschieden wurde (Abschnitt 4.2). Außerdem wurde die Umsetzung des Systems und dessen Implementierung beschrieben (Abschnitt 4.3), sowie die Erfahrungen, die im realen Einsatz in Wettkämpfen (GermanOpen 2018 und RoboCup 2018) gesammelt wurden (Abschnitt 4.4).

Mit dem beschriebenen Aufnahmesystem ist eine schnellere Auswertung eines gesamten Spiels bzw. bestimmte Spielsituationen möglich. Logdateien müssen nicht mehr aufwendig einzeln mit dem Video synchronisiert werden – es reicht den Zeitpunkt der READY-Spielphase aus den Logdateien zu extrahieren. Dieser Schritt ist mit einem Skript automatisierbar und kann anschließend direkt in dem VideoLogLabeling-Tool verwendet werden.

Die nächsten Entwicklungsschritte gehen hin zur Integration und Aufbau auf bereits bestehenden Arbeiten. So sollte die Arbeit von Lucas Schwaß „Visuelle Verfolgung humanoider Roboter im RoboCup-Kontext“[10] integriert werden, um damit die Position der Roboter und der Kamera bzgl. des Feldes ermitteln zu können. Darauf aufbauend lassen sich alle aus dem Video gewonnenen Information auf das Spielfeld zurückprojizieren und entsprechend „Ground Truth“ für bspw. die Roboter- oder Ballposition ermitteln. Auch lassen sich anhand dieser extrahierten Informationen verschiedene Algorithmen besser beurteilen, da die exakten Werte bekannt sind. Zusätzlich können die extrahierten Informationen an die Roboter zurückgemeldet werden und darauf aufbauend können „Machine Learning“-Algorithmen mit Feedback versorgt werden.

Für die Informationsextraktion lässt sich auf die Arbeit von Dominik Krienelke zum Thema „Visuelle Detektion humanoider Roboter basierend auf Histogrammen orientierter Gradienten“[3] aufbauen. So kann der in Kapitel 4 beschriebene und in Abbildung 3.5

5 Zusammenfassung und Diskussion

dargestellte Ablauf, um die Roboterdetektion im Python-Skript erweitert werden. Damit würde man eine gute Positionsabschätzung aller Roboter erhalten und könnte diese als „Ground Truth“ mit den aufgezeichneten Erkenntnissen der einzelnen Roboter vergleichen.

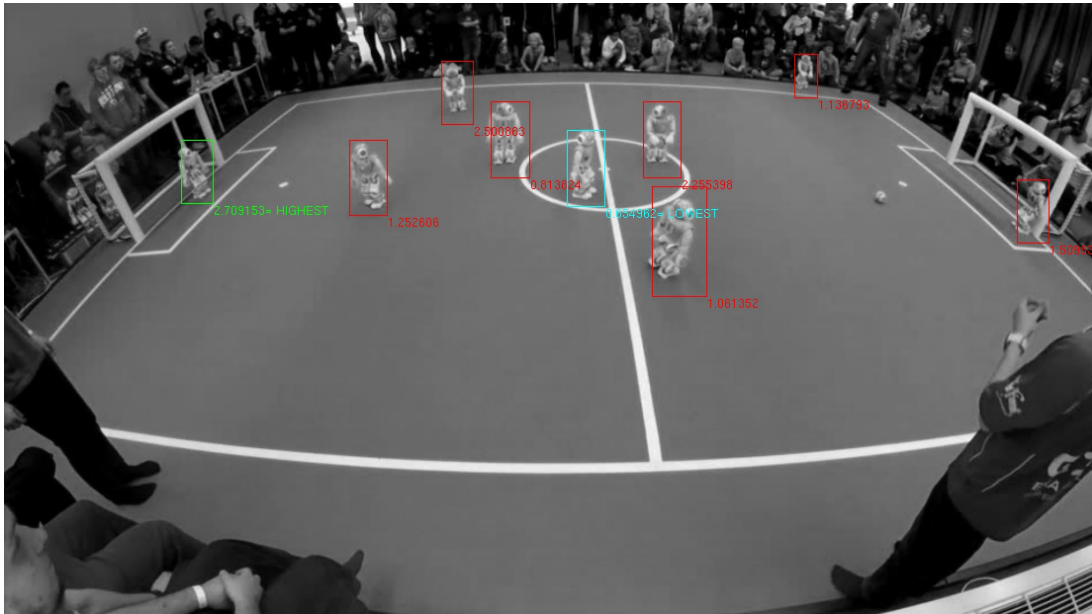


Abbildung 5.1: beispielhaftes Ergebnis einer Informationsextraktion aus einem aufgezeichneten Video[3]

Neben der Integration und Adaption weiterer Arbeiten (wie zum Beispiel die Arbeit von Robin Papke „Deep Learning Based Robot Detection in the Context of RoboCup“[7] oder Denis Ring „Visuelle Echtzeit Detektion humanoider Roboter im RoboCup Kontext“[8]), sollten auch weitere Kamerasysteme evaluiert werden.

Die Steuerung der GoPro und Ermittlung von Informationen über die Aufnahme sind während eines Spiels mit der bisherigen Anbindung nur eingeschränkt durchführbar. Findet man eine Kamera, die die genannten Kriterien erfüllt und sich zusätzlich einfacher ansteuern lässt (z.B. eine verbesserte RaspiCam), wären zukünftig Analysen und Auswertungen bereits während des Spiels möglich. Entsprechend könnte die gewonnene „Ground Truth“ auch direkt an die Roboter geleitet werden und damit die Algorithmen verbessern.

Ein solches ist System universell während verschiedenster Veranstaltungen einsetzbar und automatisiert bzw. beschleunigt den Analyse- und Auswerteprozess. Experimente und Auswertungen, die zur Zeit nur im Labor möglich sind, könnten auch während des RoboCups durchgeführt werden.

Literaturverzeichnis

- [1] Small Size League Technical Committee. *Laws of the RoboCup Soccer Small Size League 2018*.
- [2] Florian Holzhauer. Nao Team Humboldt: Entwicklung und Qualitätssicherung. Diplomarbeit, Humboldt-Universität zu Berlin, 2012.
- [3] Dominik Kriemelke. Visuelle Detektion humanoider Roboter basierend auf Histogrammen orientierter Gradienten. Diplomarbeit, Humboldt-Universität zu Berlin, Februar 2017.
- [4] Martin Löttsch. Debugging Mechanisms in the Software Architecture of the GermanTeam, Oktober 2004.
- [5] Heinrich Mellmann, Benjamin Schlotter, and Christian Blum. Simulation Based Selection of Actions for a Humanoid Soccer-Robot. In *RoboCup 2016: Robot Soccer World Cup XX*, 2016. to appear.
- [6] Heinrich Mellmann, Benjamin Schlotter, Steffen Kaden, Philipp Strobel, Thomas Krause, and Claas-Norman Ritter. Berlin United - Nao Team Humboldt: Team Report 2017. Technical report, Humboldt-Universität zu Berlin, Adaptive Systems Group, 2017.
- [7] Robin Papke. Deep Learning Based Robot Detection in the Context of RoboCup. Bachelorarbeit, Humboldt-Universität zu Berlin, November 2017.
- [8] Denis Ring. Visuelle Echtzeit Detektion humanoider Roboter im RoboCup Kontext. Bachelorarbeit, Humboldt-Universität zu Berlin, Februar 2018.
- [9] Thomas Röfer, Tim Laue, Yannick Bülter, Daniel Krause, Jonas Kuball, Andre Mühlenbrock, Bernd Poppinga, Markus Prinzler, Lukas Post, Enno Roehrig, René Schröder, and Felix Thielke. B-Human Team Report and Code Release 2017, 2017. Only available online: <http://www.b-human.de/downloads/publications/2017/coderelease2017.pdf>.

Literaturverzeichnis

- [10] Lucas Schwaß. Visuelle Verfolgung humanoider Roboter im RoboCup-Kontext. Master's thesis, Humboldt-Universität zu Berlin, Oktober 2017.
- [11] Michael Spranger. An Architecture supporting Research and Education in Robotics, 2005.
- [12] Stefan Zickler, Tim Laue, Oliver Birbach, Mahisorn Wongphati, and Manuela Veloso. SSL-Vision: The Shared Vision System for the RoboCup Small Size League.

Abbildungsverzeichnis

| | | |
|-----|--|----|
| 2.1 | Schematische Darstellung eines SPL-Feldaufbaus | 6 |
| 2.2 | Ansicht des GameControllers | 7 |
| 2.3 | Datenformat der SPLStandardMessage | 8 |
| 2.4 | Komponenten des Modul-Frameworks | 10 |
| 3.1 | verschiedene Dialoge von RobotControl 2018 | 12 |
| 3.2 | TeamCommViewer-Dialog aus RobotControl 2018 | 14 |
| 3.3 | VideoAnalyzer-Dialog aus RobotControl 2018 | 15 |
| 3.4 | VideoLogLabeling – Annotation von Logdateien | 16 |
| 3.5 | Datenfluss zur Analyse im VideoLogLabeling-Tool | 16 |
| 4.1 | Datenfluss des SSL-Vision-Systems in der Small-Size-Liga | 19 |
| 4.2 | schematischer Aufbau des Aufnahmesystems | 22 |
| 4.3 | Darstellung des modifizierten Raspberry Pi | 25 |
| 4.4 | Schnappschuss eines Spiels während der GermanOpen 2018 | 26 |
| 5.1 | Ergebnis einer Videoextraktion | 32 |

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Studienarbeit selbständig und nur unter Zuhilfenahme der angegebenen Literatur verfasst zu haben.

Ich bin damit einverstanden, dass ein Exemplar dieser Arbeit in der Bibliothek des Instituts für Informatik der Humboldt-Universität zu Berlin ausgestellt wird.

6. Januar 2019

Philipp Strobel