

Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik

**Entwicklung und Evaluation eines
Lokalisierungsverfahrens für mobile
fußballspielende Roboter auf Basis des Iterative
Closest Points Algorithmus**

Masterarbeit

Leipzig, 1. März 2016

vorgelegt von

Mende, Jonas
Studiengang Informatik

Betreuender Hochschullehrer: Prof. Dr. Mario Hlawitschka

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Ort, Datum

Unterschrift

Inhaltsverzeichnis

1	Einleitung	1
2	Hintergrund: Der Robocup	3
2.1	Wettbewerbe des RoboCup	4
2.2	Die Standard Platform League	5
2.2.1	Hardwarebasis: Nao	5
2.2.2	Spielfeldabmessungen	8
3	Grundlagen: Lokalisierung autonomer Systeme	9
3.1	Position	10
3.2	Unterteilung von Lokalisierungsproblemen	10
3.2.1	Lokale und globale Lokalisierung	11
3.2.2	Änderung der Umgebung	11
3.2.3	Aktive und passive Verfahren	12
3.2.4	Lokalisierung in der Gruppe	12
3.2.5	Intrinsische und extrinsische Verfahren	13
3.3	Verbreitete Lokalisierungsverfahren	14
3.3.1	Kalman-Filter	14
3.3.1.1	Zustandsmodellierung	15
3.3.1.2	Parameterschätzung	15

3.3.1.3	Optimalität und Erweiterungen	16
3.3.1.4	Multihypothesen-Kalman-Filter	17
3.3.2	Partikelfilter	17
3.4	Iterative Closest Points Algorithmus	18
3.4.1	Ablauf	19
4	Exakte Positionsbestimmung mittels Deckenkameras	21
4.1	Versuchsaufbau	21
4.2	Grundlagen: Markerbasierte Erfassung bewegter Objekte	22
4.2.1	Intrinsische Marker	23
4.2.2	Extrinsische Marker	23
4.2.3	Aktive Marker	23
4.2.4	Passive Marker	24
4.3	Verfahren 1: Aktive LED Marker	25
4.3.1	Vorgehen	25
4.3.2	Evaluation	26
4.4	Verfahren 2: Passive Marker mit vergrößelter Oberfläche	30
4.4.1	Vergrößerung der Markeroberfläche	30
4.4.2	Kanalzerlegung	31
4.4.3	Positions- und Rotationsbestimmung mittels Hauptkomponenten- analyse	32
4.4.4	Spielerkodierung	37
4.4.5	Algorithmus	37
4.4.6	Auswertung	40
5	Entwicklung eines Lokalisierungsalgorithmus für mobile fußballspielende Roboter	41
5.1	Voraussetzungen	41

5.1.1	Vorverarbeitete Kamerabilder	41
5.1.2	Odometrische Daten	43
5.2	Ziel	44
5.3	Anwendung des Iterative Closest Points Algorithmus	44
5.3.1	Synthese der Modellpunktmenge	44
5.3.2	Auswahl der Startpunkte	46
5.3.3	Winkelstabilisierung durch Randlinienerkennung	48
5.3.4	Behandlung von Ausreißern	51
5.4	Rekursive Filterung und Zuordnung korrespondierender Funde	53
6	Auswertung	57
6.1	Versuchsaufbau	57
6.1.1	Ergebnisse	58
7	Zusammenfassung und Ausblick	63

Kapitel 1

Einleitung

Die seit 1997 abgehaltenen Wettbewerbe des RoboCup erforschen aktiv die Entwicklung und den Einsatz moderner Verfahren der Robotik sowie der künstlichen Intelligenz. Durch die stetige Verfeinerung der Regeln dieser Wettbewerbe bis zum Erreichen des durch die RoboCup Initiative gesetzten Ziels, bis spätestens zum Jahre 2050 das amtierende menschliche Fußballweltmeisterteam zu besiegen, stellt auch die Lokalisierung mobiler autonomer Systeme ein prominentes Problem dieser Disziplinen dar. Diese Arbeit beschäftigt sich in zweierlei Hinsicht mit diesem Thema.

Zuerst soll ein Verfahren beschrieben werden welches es ermöglicht, sowohl bereits existierende als auch zukünftige Algorithmen für die autonome Lokalisierung fußballspielender Roboter der Standard Platform League des RoboCups anhand eines Goldstandards evaluieren zu können. Dazu wurde ein Verfahren entwickelt, welches die korrekte Position eben jener Roboter anhand hochauflösender Deckenkameras mit Weitwinkelobjektiven unter Laborbedingungen bestimmt.

An zweiter Stelle wurde ein konkreter Lokisierungsalgorithmus entwickelt welcher Positionsschätzungen durch die Angleichung von Punktwolken mittels einer modifizierten Variante des Iterative Closest Points Verfahren ermittelt.

Es soll ferner gezeigt werden, dass sich besagtes Verfahren insbesondere für initiale globale Positionsschätzungen im beschriebenen Szenario als auch unter den Bedingungen des sogenannten Kidnapped Robot Problems eignet. Dazu wird die Eignung des beschriebenen Algorithmus abschließend unter Zuhilfenahme des im ersten Abschnitt vorgestellten Validierungsverfahrens untersucht.

Kapitel 2

Hintergrund: Der Robocup

Die 1995 vorgestellte und seit 1997 aktive RoboCup Initiative hat sich der Definition und Lösung neuer Probleme der künstlichen Intelligenz sowie der aktiven Erforschung damit verbundener Verfahren und Techniken verschrieben [1]. Aus dem Bedarf heraus, neue Referenzprobleme der künstlichen Intelligenz und der Robotik zu definieren, stellte die Initiative den RoboCup als mögliche Plattform für diese vor. Im Gegensatz zu den kurz zuvor gelösten bisherigen Standardproblemen dieser Disziplin, wie dem Sieg eines Computers über den zu diesem Zeitpunkt amtierenden Schachweltmeister Garry Kasparov im Jahre 1997 [2], sollten neue Referenzprobleme die verteilte Lösung dynamischer Echtzeitprobleme ohne perfekte Information in den Mittelpunkt stellen. Dies bedeutete, völlig neue Herangehensweisen für die Entwicklung autonomer Systeme, deren Signalverarbeitung, Entscheidungsfindung, Kooperation und Simulation zu finden. Der rasante technologische Fortschritt Ende des 20. Jahrhunderts begünstigte die Inangriffnahme neuartiger Probleme zusehends. Neben dem rein akademischen Fortschritt bieten die Wettbewerbe und Konferenzen des RoboCup außerdem die Möglichkeit, die Entwicklung von in der Industrie nutzbaren Verfahren und Technologien durch sportlichen Wettstreit und Austausch zu beschleunigen. In der Tat sind viele Probleme des Roboterfußballs direkt oder indirekt auf Probleme der Logistik, der Raumfahrt, der Fertigungstechnik, dem Verkehrswesen oder sogar der Medizin anwendbar. Prominente Beispiele hierfür sind die intelligenten Lokalisierungs- und Wegfindungsverfahren selbstfahrender Autos und logistischer Flugkörper.

2.1 Wettbewerbe des RoboCup

Das erklärte Ziel der RoboCup Initiative ist der Sieg eines ausschließlich aus menschenähnlichen Robotern bestehenden Fußballteams gegen den amtierenden menschlichen Fußballweltmeister bis spätestens 2050. Zur Umsetzung dieses ambitionierten Plans, werden seit 1997 jährlich Wettkämpfe zwischen verschiedenen Forschungseinrichtungen ausgetragen, um die Erforschung der dafür notwendigen Komponenten zu beschleunigen. Um einen schrittweisen Fortschritt zu gewährleisten, werden die Teilprobleme dieser Aufgabe in verschiedenen Ligen des Wettbewerbes unterschiedlich gewichtet gelöst. So erforscht die Humanoid League vornehmlich die verschiedenen Konstruktionsmöglichkeiten humanoider Roboter sowie die richtige Auswahl, Kombination und algorithmische Auswertung unterschiedlicher Komponenten. Im Gegensatz dazu bieten die 2D und 3D Simulationsligen Umgebungen, in welchen abstraktere Probleme wie Teamentscheidungen und Strategievergleiche ohne Rücksicht auf etwaige Fragestellungen bezüglich der Hardware der Roboter untersucht werden können. Zusätzlich ermöglicht die Standard Platform League (SPL) den reinen Vergleich unterschiedlicher algorithmischer Lösungsmöglichkeiten bei gleicher Hardwarekonfiguration aller teilnehmender Roboter.

Um die gestellten Probleme dem aktuellen technischen Fortschritt entsprechend schrittweise lösen zu können, werden die Regeln der Wettbewerbe außerdem jährlich verändert und auf lange Sicht denen des menschlichen Fußballspiels angeglichen. So können die Roboter der meisten Ligen nach aktuellem Regelwerk untereinander kommunizieren und erhalten außerdem die Zustandsübergänge des aktuellen Spiels übermittelt. Zukünftige Versionen der Regelwerke werden diese wenig menschlichen Aspekte voraussichtlich nicht mehr zulassen und die Kommunikation von Spielern und Schiedsrichtern mutmaßlich nur noch auf akustischem und visuellem Wege erlauben. Ein weiteres Beispiel stellt die zunehmende optische Symmetrie der Spielfelder dar: die noch vor drei Jahren unterschiedlich gefärbten Tore innerhalb des Spielfeldes wurden einheitlich weiß gefärbt, um die algorithmische Orientierung der Roboter innerhalb dieses Spielfeldes zu erschweren.

Zusätzlich wurde der RoboCup um weitere Ligen erweitert, welche den Einsatz von Robotern im Arbeitsumfeld, im Heimeinsatz oder in Rettungsszenarien erforschen. Der gesamte Wettbewerb setzt sich im Jahre 2016 aus den folgenden Ligen zusammen [3]:

- RoboCup Soccer
 - Humanoid League
 - Standard Platform League
 - Small Size League
 - Middle Size League
 - Simulation League (2D, 3D)
- Rescue League
- Robocup@Home
- Robocup@Work
- RoboCup Logistics League
- RoboCup Junior

2.2 Die Standard Platform League

Der Standard Platform League soll an dieser Stelle besondere Bedeutung zukommen, da die hier vorgestellten Verfahren für den Einsatz in den Wettbewerben dieser Ligen konzipiert und entwickelt wurden. Wie vorhergehend beschrieben, konzentriert sich diese Liga weniger auf den Vergleich unterschiedlicher Hardwarelösungen als vielmehr auf die Entwicklung der leistungsfähigsten Algorithmen bei gleicher Hardwarebasis. In diesem Abschnitt sollen deshalb die sensorischen Eingabemöglichkeiten dieser Plattform sowie die für diese Arbeit wichtigsten Regeln der Standard Platform League skizziert und erläutert werden.

2.2.1 Hardwarebasis: Nao

Das französische Unternehmen Aldebaran Robotics produziert die aktuelle Hardwarebasis der Standard Platform League: den bipedischen humanoiden Roboter Nao. Abbildung

2.1 zeigt ein Exemplar dieses Fabrikats. Zusätzlich wurden die für den RoboCup wichtigsten Sensoren wie Kameras, Mikrofone und Ultraschallsensoren markiert.

Kameras und Odometrische Daten stellen dabei die Grundlage einer effektiven Lokalisierung der Roboter innerhalb des vorgegebenen Spielfeldes dar. Es gilt zu beachten, dass die Regeln der Standard Platform League jegliche Modifikation der Ausstattung der Roboter strikt untersagen. Dies bedeutet im Speziellen, dass nicht auf genauere Messverfahren wie Laserscans oder richtungsauflösende Komponenten wie Kompassse zugegriffen werden kann. Für alle anfallenden Aufgaben steht jedem Roboter die in Tabelle 2.1 aufgelistete Systemkonfiguration zur Verfügung.

Eine spezifischere Auflistung der verbauten Komponenten sowie deren Leistungsmerkmale können der offiziellen Aldebaran Dokumentation entnommen werden [4].

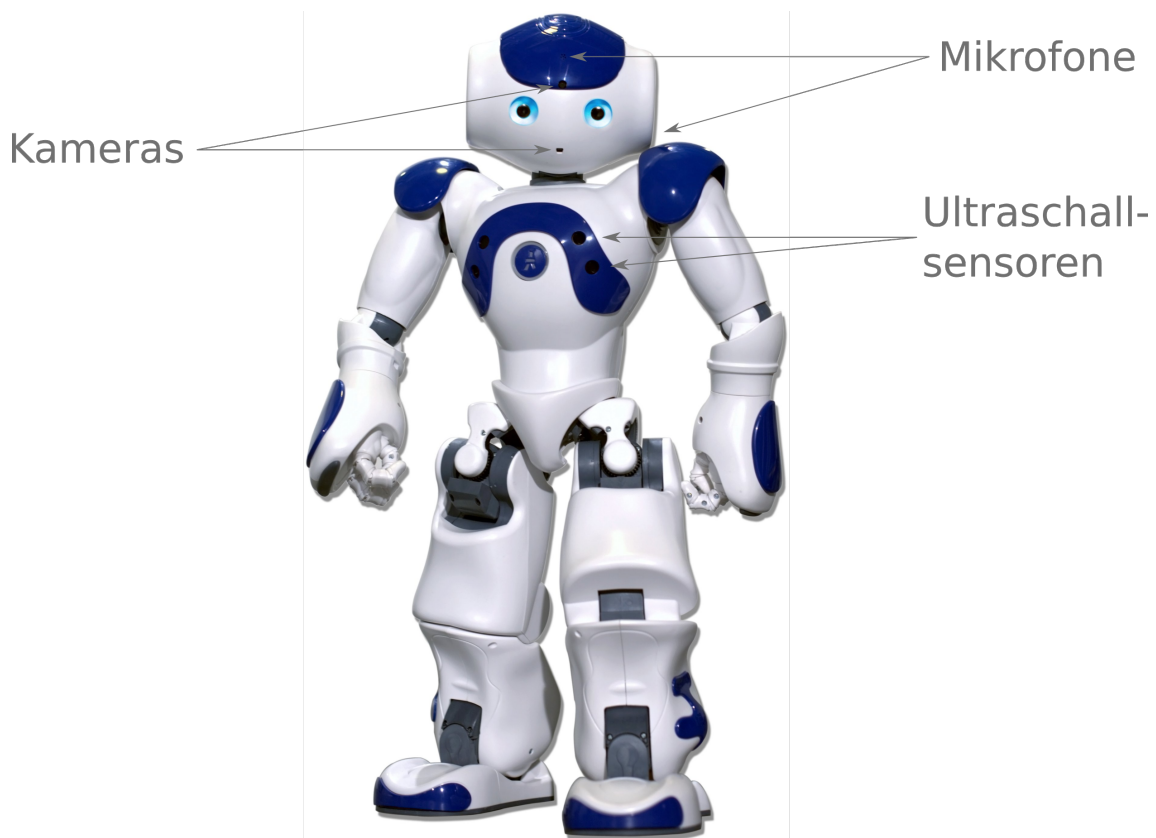


Abbildung 2.1: Nao Roboter mit für den RoboCup relevanten Sensoren (Auswahl)¹

¹Original: <http://vignette3.wikia.nocookie.net/robotics/images/6/60/NAO.jpg>, abgerufen am 24.01.2016

Tabelle 2.1: Leistungsmerkmale der Nao Plattform

Komponente	Spezifikation
Prozessor	Intel ATOM Z530 @1.6 GHz
Arbeitsspeicher	1 GB
Kameras	2 Stück Auflösung 1280x960 Aufnahmefrequenz 30 fps
Mikrofone	4 Stück Frequenzumfang 150 Hz-12k Hz
Innertiale Messeinheit	3 Achsen Gyrosensor 3 Achsen Beschleunigungssensor
Gelenkpositionssensoren	36 magnetische Drehimpulsgeber
Infrarotemitter und -sensoren	2 Stück Abstrahlwinkeln $\pm 60^\circ$
Ultraschallemitter und -sensoren	2 Stück Abstrahlwinkel $\pm 60^\circ$ Auflösung 1-4 cm effektive Aufnahmeentfernung 20-80 cm

2.2.2 Spielfeldabmessungen

Das Spielfeld der Standard Platform League wurde dem Spielfeld menschlicher Fußballspiele maßstäblich nachempfunden. Abbildung 2.2 stellt die Maße in cm der jeweiligen Spielfeldmarkierungen dar. Die Torräume werden durch 80 cm hohe Tore in weißer Farbe begrenzt. Neben den dargestellten Markierungen und den Toren dürfen keinerlei externe Markierungen zur Lokalisierung der Roboter genutzt werden. Dies bedeutet, dass eine effektive Auswertung von Kameradaten und Odometrie essentiell für die Bestimmung der Position eines Roboters sind.

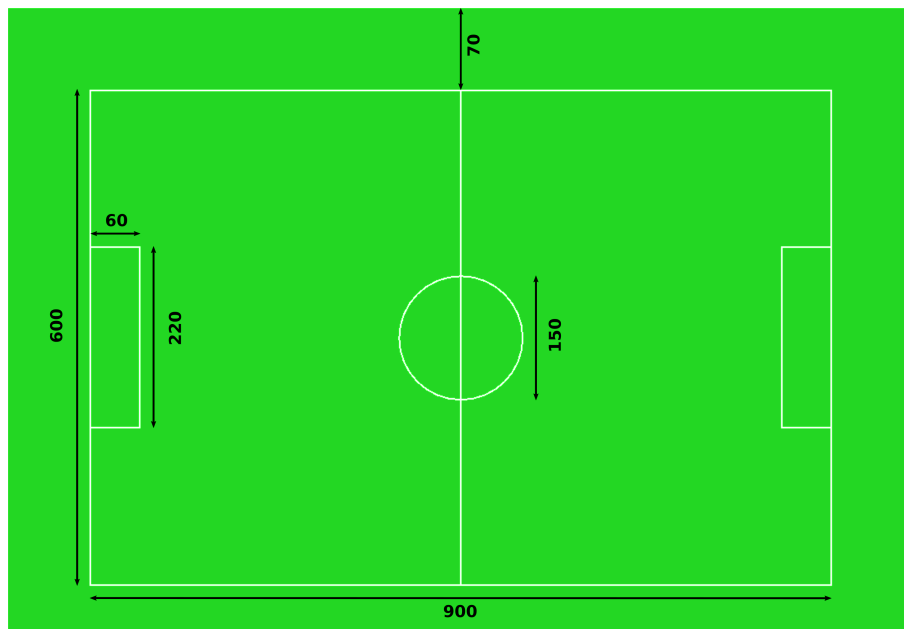


Abbildung 2.2: Spielfeldmaße

Kapitel 3

Grundlagen: Lokalisierung autonomer Systeme

Als Lokalisierung wird im allgemeinen die Schätzung der Position eines autonomen Systems in einer bekannten oder unbekanntem Umwelt bezeichnet. Thrun, Burgard und Fox stellen das Problem der Lokalisierung außerdem als die Überführung des zu dem Roboter relativen Koordinatensystems in das globale Koordinatensystem der Umwelt dar (vgl. [5], S. 190).

Die Bestimmung der globalen Position weiterer relevanter Objekte ist demnach ebenfalls direkt mit der Lokalisierung des Roboters verbunden. Übertragen auf das eingangs beschriebene Szenario des Roboterfußballs bedeutet dies, dass die Bestimmung und die Verfolgung der Positionen des Balls, der Tore und anderer Mitspieler nur dann korrekt erfolgen kann, wenn die aktuelle Position des Roboters möglichst genau geschätzt wurde.

In den meisten Anwendungsfällen ist die globale Position des Roboters nicht direkt bestimmbar. Stattdessen müssen die Messdaten verschiedener Sensoren über einen größeren Zeitraum hinaus verarbeitet und aggregiert werden um so eine genaue und stabile Schätzung der eigenen Position zu erlauben. Für den Roboterfußball bedeutet dies die Auswertung von Kamerabildern sowie Beschleunigungs-, und Lagesensoren als auch deren zeitliche Zusammenführung.

3.1 Position

Die Position eines autonomen Systems ist ein Bestandteil des Zustandes desselbigen. Im beschriebenen Anwendungsfall besteht diese aus der Spielfeldposition $(x, y)^T$ und der Orientierung α .

$$\vec{\chi} = \begin{pmatrix} x \\ y \\ \alpha \end{pmatrix}, x \in [0, 9], y \in [0, 6], \alpha \in [0, 2\pi] \quad (3.1)$$

Abbildung 3.1 zeigt beispielhaft eine solche Positionsinformation innerhalb der gegebenen Wertebereiche.

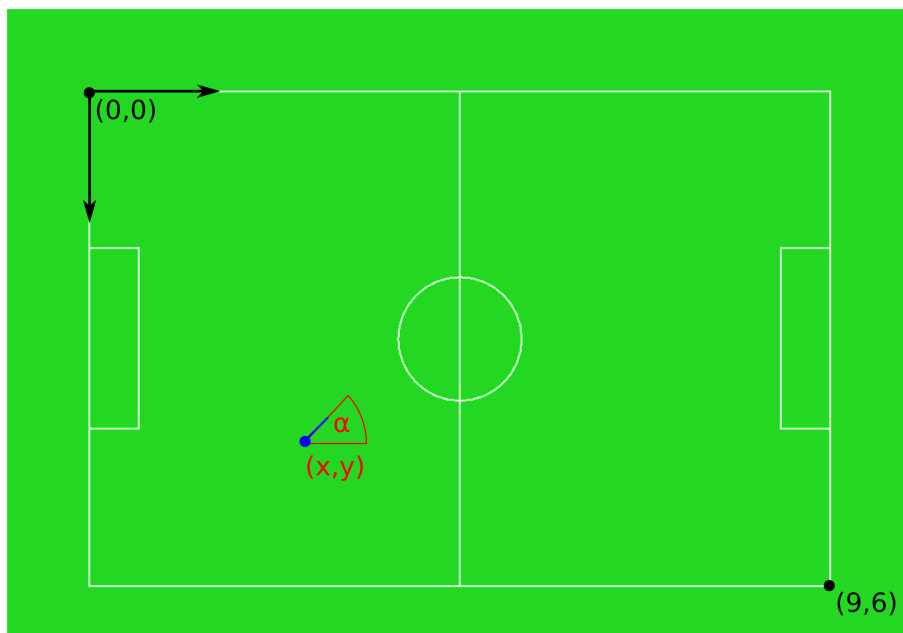


Abbildung 3.1: Roboterposition

3.2 Unterteilung von Lokalisierungsproblemen

Autonome Systeme finden in einer Vielzahl möglicher Szenarien Anwendung. Beispiele lassen sich in der Medizin, der Logistik, der Raumfahrt, dem Transportwesen und vielen

weiteren Zweigen der Industrie und des öffentlichen Leben finden. Als ebenso vielfältig zeigen sich auch die damit auftretenden Probleme welche angewandte Lokalisierungsverfahren zu lösen suchen. An dieser Stelle soll Beispielfhaft eine mögliche Unterteilung dieser Lokalisierungsprobleme nach [5] S. 193ff skizziert und im Kontext der Standard Platform League des RoboCups erläutert werden.

3.2.1 Lokale und globale Lokalisierung

Die lokale Lokalisierung beschränkt sich auf die Positionsschätzung in einer relativ kleinen Umgebung bei bekannter Startposition. In der Regel ist die Verfolgung und Verbesserung einer unimodalen Positionsschätzung ausreichend.

Globale Lokalisierungsverfahren ermitteln die Position des Roboters innerhalb seiner Umwelt ohne spezifischem Vorwissen. Üblicherweise erfordert dies die zusätzliche zeitliche Verfolgung und Schätzung der Position des Roboters unter Zuhilfenahme multimodaler Modelle. Ist die globale Position des Roboters bekannt, kann die weitere Verfolgung dessen unter Umständen lokal betrachtet werden. Eine Ausnahme dafür stellt das Kidnapped Robot Problem dar. Dieses in der Robotik grundlegende Problem bezeichnet den vollständigen Verlust der aktuellen Positionsinformation des Roboters bedingt durch eine externe Neupositionierung desselbigen.

Bei der Lokalisierung fußballspielender Roboter handelt es sich um ein globales Problem mit hoher Eintrittswahrscheinlichkeit des Kidnapped Robot Problems, beispielsweise durch die regelgebundene Entfernung und Wiedereinsetzung von Robotern. Dieser Sachverhalt muss für die Entwicklung eines leistungsfähigen Lokalisierungsalgorithmus unbedingt berücksichtigt werden.

3.2.2 Änderung der Umgebung

Lokalisierungsverfahren hängen sehr stark von der Veränderlichkeit der betrachteten Umgebung ab. Die Lokalisierung in dynamischen Umgebungen wird dabei als deutlich schwereres Problem als die Orientierung in statischen Umgebungen betrachtet. Im ersten Fall muss der Roboter neben der eigentlichen Lokalisierung auch ein dynamisches Modell seiner Umgebung erstellen um sich in dieser zu orientieren. Es gilt zu beachten,

dass eine Umgebung nur dann als statisch angesehen werden kann, wenn der Roboter das einzige Veränderliche Objekt der Umgebung darstellt. Die Lokalisierung eines Roboters auf einem Fußballspielfeld muss aufgrund sich verändernder Positionen des Balls und anderer Roboter demnach als dynamisches Problem betrachtet werden. Lediglich die Begrenzung des Spielfeldes sowie die Position und Form der Tore sind a priori bekannt.

3.2.3 Aktive und passive Verfahren

Nimmt das Lokalisierungsverfahren Einfluss auf die Position des Roboters - beispielsweise durch die Manövrierung in ein Terrain höherer Landmarkendichte oder durch die Vermeidung des Betretens gefährlicher Bereiche - so wird dieses Verfahren als aktiv bezeichnet. Im Gegensatz dazu beschränken sich passive Verfahren nur auf die Ermittlung der Position ohne etwaige Einflussnahme auf diese.

Fußballspielende Roboter besitzen die Möglichkeit, bei Verlust ihrer aktuellen Positionsschätzung, verstärkt Bereiche mit höherer Lokalisierungswahrscheinlichkeit anzusteuern und zu beobachten. Ob diese Möglichkeit der aktiven Einflussnahme jedoch genutzt wird, obliegt der jeweiligen Lokalisierungsstrategie.

3.2.4 Lokalisierung in der Gruppe

Zuletzt unterteilen die Autoren Lokalisierungsverfahren in die Anzahl der beteiligten Systeme. Denkbar ist eine mehrheitliche Lokalisierung multipler Roboter unter Austausch ihrer Informationen.

Im Roboterfußball ist auch hier keine eindeutige Aussage zu treffen. Generell besitzen alle Roboter die Möglichkeit, Informationen über ihre aktuelle Position sowie den Standort anderer relevanter Objekte mit ihren Teammitgliedern über WLAN, akustische Signale oder Infrarot auszutauschen. Deren Einsatz ist jedoch optional und eine zukünftige regelbedingte Unterbindung dieser ist nicht auszuschließen. Konkret stellt die Symmetrie des Spielfeldes ein Problem dar, welches nur schwer von einem einzelnen Roboter gelöst werden kann.

3.2.5 Intrinsische und extrinsische Verfahren

Die Autoren beziehen sich für die Klassifikation von Lokalisierungsverfahren lediglich auf jene Verfahren welche die den zu lokalisierenden Robotern eigene Sensorik nutzen. Diese intrinsischen Verfahren erfassen häufig nur einen Teil der Umgebung der zu beobachtenden Objekte. Extrinsische Verfahren bieten den inhärenten Vorteil der weiträumigeren Erfassung der Umgebung. In vielen Anwendungsfällen ist die extrinsische Erfassung jedoch entweder gar nicht oder nur unter starken Einschränkungen möglich.

In der Standard Platform League des RoboCup ist nur die Auswertung der Sensoren aktiver Roboter gestattet. Die Nutzung externer Erfassungssysteme unter Spielbedingungen ist somit ausgeschlossen. Im Gegensatz dazu werden in der Small Size League die Positionen aller agierender Roboter global erfasst und an die partizipierenden Teams übermittelt.

3.3 Verbreitete Lokalisierungsverfahren

Die an den Wettbewerben des RoboCup teilnehmenden Institute und Forschungseinrichtungen beteiligen sich seit dessen Gründung im Jahre 1997 aktiv an der Erforschung und Erprobung sowohl bereits bestehender als auch neuartiger Lokalisierungsverfahren. An dieser Stelle sollen einige dieser Verfahren vorgestellt sowie deren Vor- und Nachteile skizziert werden. Die genannten Verfahren unterscheiden sich teilweise erheblich in der Genauigkeit ihrer zugrundeliegenden Modelle als auch ihres geforderten Berechnungsaufwandes. Ein leistungsfähiger Algorithmus bedarf einer ausführlichen Abwägung dieser Dimensionen um eine ausreichende Genauigkeit bei Einhaltung der durch die jeweilige Plattform gegebenen Bedingungen in puncto Speicher- und Rechenbedarf zu ermöglichen. Den ersten beiden vorgestellten Verfahren gemein ist die Anwendung probabilistischer Modelle zur Bestimmung der jeweils besten Schätzung der aktuellen Position des Roboters sowie zur zeitlichen Aggregation, Filterung und Stabilisierung derselbigen. Das im Abschluss erläuterte Verfahren bietet ein Beispiel zur direkten Bestimmung von Positionshypothesen aus Punktwolken durch Angleichung dieser Punktmenge an ein gegebenes Modell.

3.3.1 Kalman-Filter

Das Kalman-Filter ist ein bereits 1960 von Rudolph Kalman vorgestelltes Verfahren der Klasse Gauß'scher Filter [6]. Es findet seit seiner Einführung weite Verbreitung zur Filterung und Vorhersage linearer kontinuierlicher Systeme.

Als rekursives Verfahren unter Annahme der Markov-Vermutung schätzt es den aktuellen Systemzustand ausschließlich anhand des vorhergehenden Zustandes sowie aktueller Beobachtungen. Dabei werden zwei grundlegende Annahmen über das zugrundeliegende System getroffen ([6], [5] S. 31)

- Der zeitliche Zustandsübergang des zugrundeliegenden Systems sowie die Abbildung von Messwerten auf Systemzustände lassen sich als lineare Abbildungen beschreiben.
- Fehler sowohl in der Beobachtung des Systems als auch in der Schätzung des Zustandsüberganges sind multivariat normalverteilt.

3.3.1.1 Zustandsmodellierung

Ausgehend von diesen Annahmen lässt sich der Zustandsübergang des Systems wie folgt modellieren:

$$\vec{\chi}_t = F_{t-1}\vec{\chi}_{t-1} + B_{t-1}\vec{u}_{t-1} + \vec{\epsilon}_{t-1} \quad (3.2)$$

F beschreibt hierbei den intrinsischen Übergang des Systems zwischen zwei Zeitpunkten. B ist der bestimmbare externe Einfluss auf das System. Er wird des öfteren auch als Steuerungseingabe interpretiert. Der unbestimmbare äußere Einfluss, bzw. der zufällige Fehler der Steuerungseingabe $\vec{\epsilon}_{t-1}$ ist mittelwertfrei normalverteilt mit Kovarianz R_{t-1} :

$$\vec{\epsilon}_{t-1} \sim \mathcal{N}(0, R_{t-1}) \quad (3.3)$$

Auch die Beobachtungen der Systemzustände unterliegen in der Regel einem statistischen Rauschen. Die Abbildung von Systemzuständen auf einen Beobachtungsvektor wird deshalb wie folgt modelliert:

$$\vec{z}_t = H\vec{\chi}_t + \vec{\delta}_t \quad (3.4)$$

H_t beschreibt die Abbildung von Systemzuständen auf Beobachtungsgrößen. Der dabei auftretende additive Fehler

$$\vec{\delta}_t \sim \mathcal{N}(0, Q_t) \quad (3.5)$$

ist wiederum mittelwertfrei normalverteilt mit Kovarianz Q_t

3.3.1.2 Parameterschätzung

Die zeitlich diskrete Schätzung des kontinuierlichen Zustandes erfolgt durch Schätzung der Parameter der zugrundeliegenden Normalverteilung mit Mittelwert $\vec{\mu}_t$ und Kovarianz P_t :

$$\hat{\vec{\chi}}_t \sim \mathcal{N}(\hat{\vec{\mu}}_t, \hat{P}_t) \quad (3.6)$$

Die eigentliche Schätzung erfolgt in 2 unabhängigen Schritten: der Prädiktion

$$\begin{aligned}\hat{\chi}_{t|t-1} &= F_{t-1}\hat{\chi}_{t-1} + B_{t-1}\vec{u}_{t-1} \\ \hat{P}_{t|t-1} &= F_{t-1}\hat{P}_{t-1}F_{t-1}^T + R_{t-1}\end{aligned}\quad (3.7)$$

des folgenden Systemzustandes sowie der anschließenden Korrektur

$$\begin{aligned}\hat{\chi}_t &= \hat{\chi}_{t|t-1} + \hat{K}_t\hat{y}_t \\ \hat{P}_t &= \hat{P}_{t|t-1} - \hat{K}_tS_t\hat{K}_t^T\end{aligned}\quad (3.8)$$

dieser Vorhersage mittels gegebener Beobachtung \vec{z}_t . Eine besonderes Augenmerk ist auf die Interpretation der Hilfsgrößen Innovation

$$\vec{y}_t = \vec{z}_t - H_t\hat{\chi}_{t|t-1}\quad (3.9)$$

und Kalman-Verstärkung

$$\hat{K}_t = \hat{P}_{t|t-1}H_t^T S_t^{-1}\quad (3.10)$$

mit

$$S_t = H_t\hat{P}_{t|t-1}H_t^T + Q_t\quad (3.11)$$

zu richten. Diese beschreiben sowohl die Kompatibilität des vorhergesagten Systemzustandes und der beobachteten Messgröße als auch die Gewichtung dieser Werte anhand der angenommen Fehler dieser.

3.3.1.3 Optimalität und Erweiterungen

Das Kalman-Filter ist unter der genannten Voraussetzungen der Linearität und der Kenntnis der Kovarianzen von System- und Messrauschen optimal. Praktisch können jedoch beide Voraussetzungen nur selten erfüllt werden. Dennoch liefert dieses Filter bei leichten Verletzungen dieser Forderungen bereits sehr gute Ergebnisse und findet deshalb in einer Vielzahl technischer Anwendungsgebiete Verwendung. Zur Filterung nichtlinearer Systeme existieren des Weiteren analytische und statistische Erweiterungen dieses Filters.

Das Extended Kalman-Filter ersetzt die Nichtlinearität durch eine linearisierte Abbildung derselbigen mittels Taylorreihenentwicklung ([5] S. 48). Das Unscented Kalman-Filter [7] liefert noch stabilere Ergebnisse für hoch nichtlineare Problemstellungen durch statistische Stützpunkte um den Mittelwert der Abbildungen.

3.3.1.4 Multihypothesen-Kalman-Filter

Es gilt zu beachten, dass das Kalman-Filter eine unimodale Normalverteilung des Systemzustandes annimmt. Die Filterung und Vorhersage einer multimodalen Verteilung der Zustände ist also nicht oder nur bedingt möglich. Nach der vorgenommenen Einteilung von Lokalisierungsalgorithmen in Abschnitt 3.2 eignet sich das Kalman-Filter besonders für lokale Lokalisierungsprobleme, ohne Erweiterung jedoch nicht für die globale Lokalisierung eines Roboters. Thrun et al. ([5] S. 179) sowie Bazzani [8] beschreiben den Einsatz mehrerer Filterinstanzen als sogenanntes Multihypothesen-Kalman-Filter. Durch die intelligente Zuordnung von Beobachtungen zu einer oder mehreren Filterinstanzen - der Lösung des sogenannten Korrespondenzproblems - lässt sich so eine multimodale Verteilung als Mischung mehrerer Normalverteilungen annähern. Die Anzahl benötigter Instanzen wird durch die geschickte Initialisierung, Zusammenführung und Aufteilung je Iterationsschritt optimiert. Jochmann et al. [9] beschreiben den Einsatz eines solchen Multihypothesen-Kalman-Filters für den Einsatz im RoboCup.

3.3.2 Partikelfilter

Ein weit verbreitetes Filter für die globale Positionsbestimmung mobiler Roboter stellt das zu den parameterlosen Verfahren gehörende Partikelfilter dar. Im Gegensatz zum zuvor beschriebenen Kalman-Filter stützt sich dieses Verfahren also nicht auf ein durch Parameter beschreibbares Modell (im Falle des Kalman-Filters eine Normalverteilung) der Wahrscheinlichkeitsdichte. Stattdessen approximiert dieses Monte-Carlo Verfahren eine unbekannte Dichtefunktion durch zufällig aus dieser Verteilung gezogene Punkte. Jeder dieser so generierten Punkte oder Partikel $\chi_t^{[m]}$ repräsentiert eine mögliche Realisierung des Zustandes des zu beobachtenden Systems ([5] S. 77).

$$X_t := \chi_t^{[1]}, \chi_t^{[2]}, \chi_t^{[3]}, \dots, \chi_t^{[M]} \quad (3.12)$$

Ausgehend von einer intialen informierten oder uninformierten Verteilung der Partikel aktualisiert der Algorithmus die anzunähernde Verteilung rekursiv in mehreren Schritten. Zuerst werden alle aktuell existierenden Partikel anhand der bekannten Steuerungseingabe des Systems aktualisiert und nach der Wahrscheinlichkeit der Sichtung gemessener Daten bei Annahme des durch den Partikel repräsentierten Systemzustandes gewichtet:

$$w_t^{[m]} = p(\vec{z}_t | \vec{\chi}_t^{[m]}) \quad (3.13)$$

Anschließend werden aus diesen aktualisierten Punkten M Partikel mit Zurücklegen gezogen. Die Wahrscheinlichkeit dass der Partikel $\vec{\chi}_t^{[m]}$ gezogen wird soll dabei genau $w_t^{[m]}$ entsprechen.

Durch eine große Zahl von Partikeln lässt sich somit eine Wahrscheinlichkeitsdichte beliebig genau approximieren und über die Zeit verfolgen. Als nachteilig zeigt sich jedoch, dass - je nach Art der zu beobachtenden Wahrscheinlichkeitsverteilung - unter Umständen eine sehr hohe Anzahl von Partikeln benötigt wird um diese statistisch zu erfassen. Damit einher geht demzufolge ein steigender Berechnungsaufwand. Burchardt et al. [10] zeigen den erfolgreichen Einsatz dieser Art Filter in den Wettbewerben des Robocup.

3.4 Iterative Closest Points Algorithmus

Der Iterative Closest Points Algorithmus dient der Angleichung von Punktmengen aneinander. Einsatz findet dieser Algorithmus zum Beispiel in der Registrierung von Laserscans.

Als Eingabe des Algorithmus dienen 2 Mengen von Punkten: Model M und Source S . Dabei wird die Transformation T gesucht unter deren Anwendung der Fehler zwischen der transformierten Sourcemenge und der Modellmenge möglichst gering wird.

Anderson et. al. [11] demonstrieren den erfolgreichen Einsatz des Iterative Closest Points Verfahrens für zuvor klassifizierte Merkmale des Spielfeldes der Standard Platform League.

3.4.1 Ablauf

Ausgehend von einer initialen Transformation arbeitet der Algorithmus in folgenden Schritten:

- Bestimmung des jeweils nächsten Punktes m_i der Modellmenge zu jeden Punkt s_i der Sourcemenge: $\arg \min_{m \in M} \|m - s_i\|$
- (Ausschließung etwaiger Ausreißer)
- (Gewichtung der Korrespondenzen)
- Bestimmung der Transformation T welche den Fehler zwischen korrespondierenden Punkten minimiert
- Anwendung der Transformation auf die Sourcemenge

Diese Schritte werden so lange wiederholt bis der summierte Fehler über alle korrespondierenden Punkte einen Schwellwert unterschreitet.

Übliche Fehlermaße sind der mittlere quadratische Abstand zwischen Modell- und Sourcepunkten m_i und s_i :

$$E = \frac{1}{N} \sum_{i=1}^N \|m_i - T * s_i\|^2 \quad (3.14)$$

sowie der mittlere quadratische Abstand zwischen Sourcepunkt s_i und der durch Normale n_i definierten Oberfläche an Modellpunkt m_i [12].

$$E = \frac{1}{N} \sum_{i=1}^N \|(m_i - T * s_i) \cdot n_i\|^2 \quad (3.15)$$

Algorithmus 1 Iterativer Ablauf des Iterative Closest Points Algorithmus

Berechnung der optimalen Transformation T sowie des mittels des Punkt-zu-Punkt Fehlermaßes bestimmten Fehlers $error$ aus Modellpunktmenge M und Sourcepunktmenge S

```
1: function ICP( $S, M$ )
2:   while not converged do
3:     for all  $s_i \in S$  do
4:        $m_i = \arg \min_{m \in M} \|m - s_i\|$ 
5:        $T = \arg \min_T \frac{1}{N} \sum_{i=1}^N \|m_i - T * s_i\|^2$ 
6:     for all  $s_i \in S$  do
7:        $s_i = T * s_i$ 
8:        $error = \frac{1}{N} \sum_{i=1}^N \|m_i - s_i\|^2$ 
9:   return  $T, error$ 
```

Kapitel 4

Exakte Positionsbestimmung mittels Deckenkameras

Zum Vergleich konkurrierender Lokalisierungsalgorithmen muss ein geeignetes Validierungsverfahren für generierte Lokalisierungshypothesen gefunden werden. Da eine manuelle Auswertung von Kamerabildern und deren assoziierten Lokalisierungspositionen aus praktischen Gründen nicht möglich ist, muss ein Verfahren entwickelt werden, um eine Lokalisierungshypothese mit einem Goldstandard vergleichen zu können. In diesem Kapitel soll ein Verfahren beschrieben werden, um Roboter auf dem Spielfeld zu erkennen und deren Position möglichst genau zu bestimmen. Von Vorteil ist hierbei, dass die Ermittlung des Goldstandards auf Methoden und Hardware zurückgreifen kann, welche für die eigentliche Lokalisierung nicht infrage kommen, da sie durch die Regeln des RoboCups ausgeschlossen wären. Es wurden deshalb verschiedene Versuche unternommen, um Roboter anhand von Markern mittels an der Decke befestigter hochauflösender Kameras mit Weitwinkelobjektiven zu erkennen und zu verfolgen.

4.1 Versuchsaufbau

Abbildung 4.1 illustriert beispielhaft die Ausgabe der technischen Laborausstattung, bestehend aus 2 deckenmontierten Kameras mit Weitwinkelobjektiven und einem softwaregestützten Verfahren zur Entzerrung aufgenommener Bilder. Diese stellen die Eingabe

für die in dieser Arbeit untersuchten Verfahren zur exakten Erkennung und Verfolgung sowohl unbewegter als auch bewegter Roboter innerhalb des Spielfeldes dar.

Abbildung 4.1 verdeutlicht außerdem einen weiteren durch die Entzerrung besagter Kamerabilder bedingten Sachverhalt: der tatsächliche aufgenommene Bildbereich (rotes Trapez) umfasst einen wesentlich größeren Teil des Raumes als erwünscht. Das entzerrte Bild weist somit ein deutliches Auflösungsgefälle in Richtung der Basis des durch die Entzerrung entstandenen Trapezes auf. Zu untersuchende Verfahren müssen diesen Sachverhalt explizit berücksichtigen um eine robuste Erkennung der Roboter in den Außenbereichen des Bildes im Allgemeinen und nahe der Basis des entstandenen Trapezes im Besonderen zu ermöglichen.



Abbildung 4.1: Entzerrte Beispielaufnahme einer Deckenkamera
rot: gesamter Aufnahmebereich

4.2 Grundlagen: Markerbasierte Erfassung bewegter Objekte

In diesem Abschnitt sollen gängige Verfahren zur Erfassung bewegter Objekte mittels vorrangig extrinsischer Marker vorgestellt werden. Um die Einteilung der verschiedenen Verfahren zu erleichtern, soll zuvor eine Einteilung der unterschiedlichen Markierungsarten vorgenommen werden.

4.2.1 Intrinsische Marker

Die Verwendung intrinsischer Marker beschränkt die Möglichkeiten zur Erkennung von Objekten auf die von den Objekten a priori bekannten Merkmale ohne die Anbringung zusätzlicher Markierungen. Als Beispiel sei die Erfassung und Verfolgung von Gesichtern anhand der Position von Mund, Nase, Ohren und anderer denkbarer Gesichtsmerkmale genannt. Ein weiteres Beispiel stellt die automatisierte Erfassung und Verfolgung vaskulärer Gefäße anhand charakteristischer Verzweigungspunkte dar.

4.2.2 Extrinsische Marker

Durch die zusätzliche Anbringung von Markern können Objekte deutlich stärker von ihrer Umgebung abgegrenzt und somit algorithmisch erfasst werden. Beispielsweise stellen Fahrbahnbegrenzungen optische extrinsische Marker sowohl für maschinelle Fahrspurerkennungssysteme, als auch für menschliche Fahrer dar.

Anwendung finden extrinsische Marker vorrangig in Szenarien in welchen der Bedarf einer genaueren Positionsbestimmung den erhöhten Aufwand der Anschaffung und Anbringung solcher Markierungen rechtfertigt. Beispiele sind industrielle Fertigungsstraßen oder die computerassistierte Chirurgie.

4.2.3 Aktive Marker

Aktive Marker senden selbsttätig Signale in für den Beobachter wahrnehmbaren Spektren aus und bieten somit eine inhärent bessere Erkennungsrate als vergleichbare passive Verfahren. Durch die Erweiterung der durch die Versuchsdomäne gegebenen Wahrnehmungsmöglichkeiten werden somit Ergebnisse verbessert oder eine Erkennung erst ermöglicht.

Als Beispiel für aktive Erkennungsverfahren sei die Positionsbestimmung von Operationsbesteck und Kleinstgeräten in der computerassistierten Chirurgie genannt ([13]). Hierbei wird ein elektromagnetisches Feld erzeugt und die zu verfolgenden Objekte mit Spulen versehen. Bewegen sich diese nun in dem erzeugten Feld, so kann durch Messen der Veränderung der Feldstärke Rückschlüsse auf Lage und Orientierung der zu beobachtenden Objekte gezogen werden.

Die seit kurzen populären Innenraumlokalisierungsverfahren auf Basis niederfrequenter Bluetooth-Signale stellen ein weiteres Beispiel für aktive Marker dar ([14]). Bei diesem Verfahren werden Signalgeber an vorher bekannten Positionen innerhalb von Gebäuden platziert. Mobile Geräte erfassen die Präsenz dieser Signale und ermitteln den Abstand zu diesen Quellen um somit ihre Position zu triangulieren. Die Positionsbestimmung von Mobiltelefonen mittels Triangulation über die Signalstärke sich in der Nähe befindender Funkzellen ist eng mit diesem Verfahren verwandt.

Leuchtdioden bieten eine kostengünstige Möglichkeit zur Verbesserung von bildbasierten Erkennungsverfahren. Sichtbares Licht emittierende Dioden können dabei als heller Punkt in ihrer Umgebung wahrgenommen werden - eine Verwirrung technischer Aufnahmegeräte oder menschlicher Beobachter kann dabei allerdings nicht ausgeschlossen werden. Infrarot-LEDs verlassen das sichtbare Spektrum des Lichts und beeinträchtigen somit weniger Menschen und Fotoaufnahmen (entsprechende Technik vorausgesetzt). Als nachteilig zeigen sich jedoch erhöhte Anschaffungskosten für Marker und Aufnahmegeräte.

Zuletzt sei die Möglichkeit akustischer Verfahren genannt. Vorstellbar wären Geräusche erzeugende Lautsprecher welche von im Raum verteilten Mikrofonen aufgenommen werden. Mittels Triangulation der aufgenommenen Signale ließe sich die Position von Objekten effektiv bestimmen. Durch Nutzung von für den Menschen nicht hörbaren Frequenzen im Ultra- oder Infraschallbereich könnte die Beeinträchtigung von Personen minimiert werden. Der erhöhte Installationsaufwand sowie die zusätzliche Erschwernis bei der Erkennung verschiedener gleichzeitiger Signale sowie der Erkennung von Objektrotationen schließen dieses Verfahren jedoch für das beschriebene Versuchsszenario aus.

4.2.4 Passive Marker

Die Emission erfassbarer Signale vereinfacht die Erkennung aktiver Marker deutlich. Damit einher gehen erhöhte Anschaffungskosten sowie ein deutlich gesteigerter Installationsaufwand. Durch die geschickte Wahl und Installation passiver Marker können inhärente Nachteile jedoch je nach Einsatzzweck oft kompensiert werden. Passive Marker werden je nach Aufnahmemodalität so gewählt, dass sie sich besonders stark von ihrer Umgebung abgrenzen. Dies kann zum Beispiel durch den Einsatz stark reflektierender

Materialien im den verschiedenen Lichtspektrern geschehen. Möglich ist auch die Verwendung eindeutiger Muster zur Erhöhung der Erkennbarkeit.

4.3 Verfahren 1: Aktive LED Marker

In einem ersten Versuch wurden Leuchtdioden der Marke Kingbright in den in Tabelle 4.1 aufgeführten Farben und Intensitäten als aktive Marker verwendet. Der Abstrahlwinkel betrug jeweils 20 Grad.

Tabelle 4.1: Verwendete LED-Farben und -Intensitäten

Leuchtfarbe	Lichtstärke
rot	8000 mcd
grün	10000 mcd
weiß	18000 mcd
blau	8000 mcd

Die Leuchtdioden wurden zusammen mit einer Knopfzelle provisorisch sowohl seitlich als auch mittig auf den Köpfen der Roboter platziert.

4.3.1 Vorgehen

Algorithmus 2 beschreibt den Ablauf zur Erkennung der zuvor beschriebenen aktiven Marker basierend auf einfachen Schwellwert- und Fensterungsverfahren.

Da die im Versuchsaufbau beschriebenen Deckenkameras fest montiert sind, kann in einem vorverarbeitenden Schritt auf die durch ein Differenzbild gewonnene Vorauswahl relevanter Pixel zurückgegriffen werden.

Zuerst wird für jedes Pixel (x, y) des übergebenen Bildes der Dimension $width \times height$ eine Differenzierung zwischen relevanten und irrelevanten Pixeln anhand deren Luminanz mittels der Funktion *processPixel* durchgeführt. Anschließend wird für alle relevanten Pixel der Durchschnitt aller Farbwerte innerhalb eines Fensters um besagtes Pixel gebildet (Funktion *mean*). Zu diesem Wert wird in einem letzten Schritt mittels der Funktion

closestColor die Farbe gesucht, deren euklidischer Abstand zu zuvor berechnetem Wert im RGB Farbraum minimal ist.

Die im Algorithmus genutzten Konstanten wurden wie folgt initialisiert:

Tabelle 4.2: Verwendete Konstanten

Konstante	Wert
<i>FRAMESIZE</i>	5
<i>T_{DIFF}</i>	0.2
<i>T_{LIT}</i>	0.8

Abbildung 4.2 visualisiert den Ablauf des Algorithmus anhand der zuvor beschriebenen Zwischenschritte.

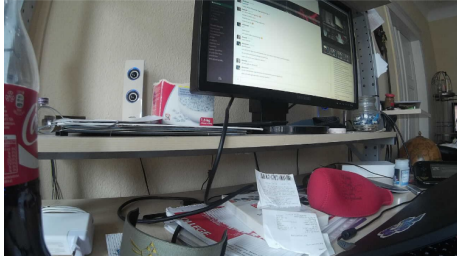
4.3.2 Evaluation

Obwohl das hier vorgestellte Verfahren sehr gute Ergebnisse bei Bildern hoher Auflösung lieferte, versagte es leider im beschriebenen Einsatzfeld. Zwar waren die Marker in Bildern nahe der Kamera sehr gut zu erkennen. Bedingt durch den zu geringen Abstrahlwinkel der Leuchtdioden und der damit einhergehenden großen Änderung der Intensität des einfallenden Lichtes bereits bei leichter Veränderung des Einfallswinkels (Abbildung 4.3), stellte sich die Erkennungsrate am Rand des Spielfeldes durch das zuvor beschriebene Auflösungsgefälle jedoch als zu gering heraus. In Anbetracht der schwierigen Anbringung der Marker sowie einer relativ hohen Anfälligkeit gegenüber Stürzen musste ein anderes Verfahren gefunden werden.

Algorithmus 2 Farbklassifikation

Berechnung des farbklassifizierten Bildes *processed* aus Eingabebild *img* und Basisbild *base* mit Bilddimensionen $width \times height$

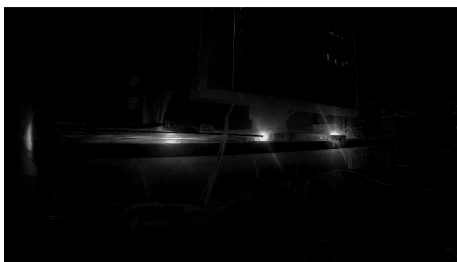
```
1: function PROCESSIMAGE(img, base, width, height)
2:   processed  $\leftarrow$  CREATE
3:   for all (x, y)  $\in$  img do
4:     if PROCESSPIXEL(img(x, y), base(x, y))  $>$  0 then
5:        $m \leftarrow \frac{\sum \text{SUB}(\text{img}, x, y, \text{FRAMESIZE})}{\text{FRAMESIZE}^2}$ 
6:       processed(x, y)  $\leftarrow$  CLOSESTCOLOR(m)
7:   return processed
8:
9: function PROCESSPIXEL(pixel, base)
10:  l  $\leftarrow$  luminance(pixel)
11:  b  $\leftarrow$  luminance(base)
12:  diff  $\leftarrow$  l - b
13:  if diff  $<$   $T_{DIFF}$  then diff  $\leftarrow$  0
14:  if diff  $\leq$  0 then l  $\leftarrow$  0
15:  if l  $<$   $T_{LIT}$  then l  $\leftarrow$  0
16:  return l
17:
18: function CLOSESTCOLOR(pixel)
19:  return color with minimal euclidian distance to pixel in RGB space
20:
21: function CREATE(width, height)
22:  return new image of size  $width \times height$ 
23:
24: function LUMINANCE(pixel)
25:  return  $0.2126 * R + 0.7152 * G + 0.0722 * B$ 
26:
27: function SUB(img, x, y, width)
28:  return subimage with bounds (x-width, x+width, y-width, y+width)
```



(a) Basisbild



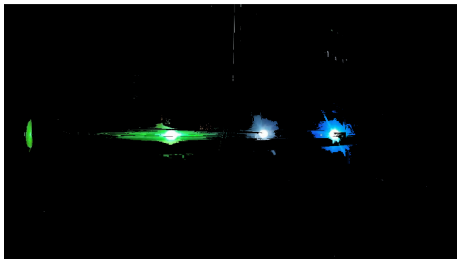
(b) Dioden erleuchtet



(c) Differenzbild



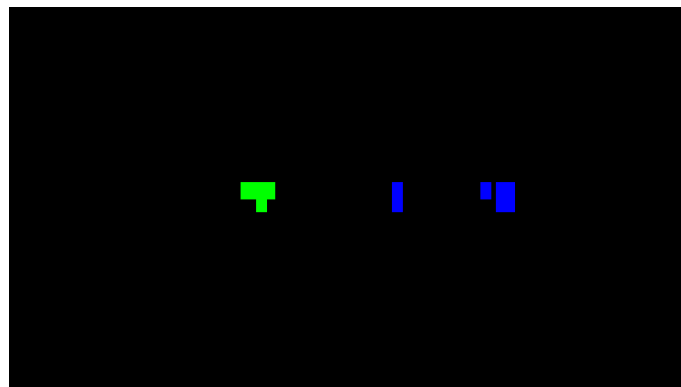
(d) Differenzbild mit Schwellwert



(e) Gefiltertes RGB-Bild



(f) Schwellwert auf RGB-Bild



(g) diskretisierte Farbzuoordnung

Abbildung 4.2: Zwischenschritte der LED-Lichtpunkterkennung



(a) Direkter Einfall



(b) Geneigter Einfall

Abbildung 4.3: Beispielaufnahmen eines Roboters mit angebrachten Leuchtdioden bei unterschiedlichen Einfallswinkeln

4.4 Verfahren 2: Passive Marker mit vergrößerter Oberfläche

Aktive Marker können so gewählt werden, dass sie in Aufnahmen besonders hervorstechen - im elektromagnetischen Spektrum zum Beispiel durch Einsatz von Infrarot-LEDs oder wie zuvor beschrieben als helle Punkte in der Luminanz-Transformation in vorangegangenem Versuchsaufbau. Damit einher gehen allerdings erhöhte Anschaffungskosten sowie eine erschwerte Anbringung aufgrund der benötigten Energieversorgung. Des Weiteren zeigten sich in besagtem Versuchsaufbau Probleme bei der Kippung der angebrachten Marker aufgrund des eingeschränkten Abstrahlwinkels.

Deshalb soll nun ein Verfahren beschrieben werden welches durch Einsatz passiver Marker die beschriebenen Probleme zu kompensieren versucht. Dies bedeutet, dass sich eingesetzte Marker als günstig in der Anschaffung, stabil bei Stürzen und unanfällig gegenüber dem zu Beginn dieses Kapitels beschriebenen Auflösungsgefälle der Aufnahmen beweisen müssen.

4.4.1 Vergrößerung der Markeroberfläche

In einem ersten Schritt sollte die insgesamt zu erkennende Fläche der verwendeten Marker vergrößert werden, um sowohl im Zentrum als auch an den Rändern des Aufnahmebereiches annähernd gleich hohe Erkennungsraten zu erzielen. Für die weitere Verwendung von LEDs hätte dies entweder den Einsatz größerer Strahler oder eine Vergrößerung des Abstrahlwinkels bedeutet. Dies hätte allerdings entweder den Einsatz einer stärkeren Energiequelle oder die Anschaffung deutlich teurer LED Modelle nach sich gezogen. Stattdessen zeigten Experimente mit unterschiedlich gefärbten Folien- und Papiermarkern deutlich bessere Erkennungsraten bei drastisch niedrigerem Installationsaufwand (Abbildung 4.4).

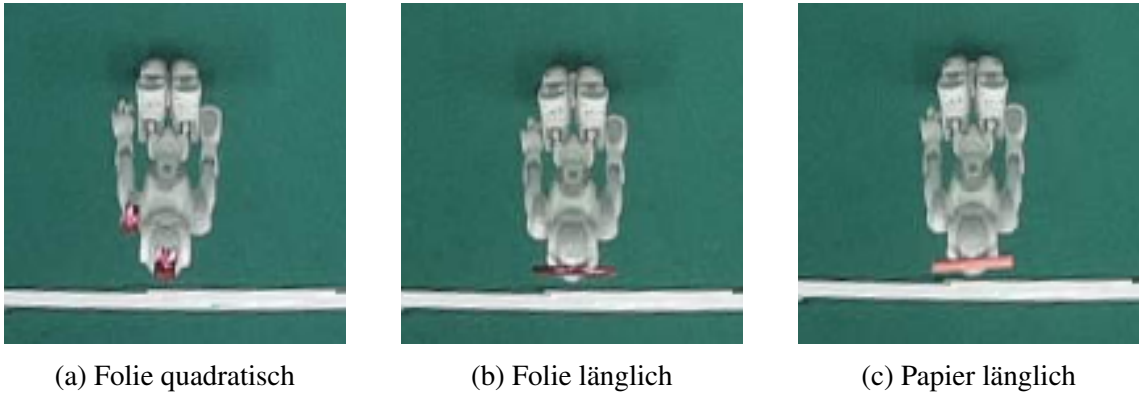


Abbildung 4.4: Aufnahmen eines Roboters mit unterschiedlichen angebrachten Markern aus reflektierender Folie und Papier

4.4.2 Kanalzerlegung

Zur Erkennung der Marker werden die Eingabebilder in einem ersten Schritt zu Kanälen entsprechend der zu untersuchenden Farben zerlegt. Die Kanalzerlegung erfolgt nach folgender Formel:

$$p^t = c - p \quad (4.1)$$

wobei p^t den transformierten Pixelwert, p den Originalwert und c den Median der Farbwerte der Zielfarbe darstellen.

$$p = \begin{pmatrix} r \\ g \\ b \end{pmatrix}, p^t = \begin{pmatrix} r^t \\ g^t \\ b^t \end{pmatrix}, c = \begin{pmatrix} \text{median}(R^{\text{col}}) \\ \text{median}(G^{\text{col}}) \\ \text{median}(B^{\text{col}}) \end{pmatrix} \quad (4.2)$$

Anschließend werden die so entstandenen Farbwerte wie folgt binär klassifiziert:

$$f(p^t) = \begin{cases} 1, & \text{wenn } r < \lambda_r \wedge g < \lambda_g \wedge b < \lambda_b \\ 0, & \text{sonst} \end{cases} \quad (4.3)$$

Wobei $\lambda_r \sim \text{MAD}(R^{\text{col}})$, $\lambda_g \sim \text{MAD}(G^{\text{col}})$, $\lambda_b \sim \text{MAD}(B^{\text{col}})$ gilt. Die statistischen Schätzgrößen Median und MAD (median absolute deviation) werden zuvor aus einer aus-

reichend großen Menge manuell positiv klassifizierter Beispielpixel bestimmt (R^{col} , G^{col} und B^{col}). Praktische Versuche zeigten dabei dass der Einsatz der robusten Median- und MAD-Schätzer deutlich bessere Klassifikationsergebnisse lieferte als die gegenüber Ausreißern anfälligen Mittelwert- und Varianzschätzer. Abbildung 4.6 zeigt die Histogramme der roten, grünen und blauen Farbkanäle für jede der vier Zielfarben. Zusätzlich wurden die angenommenen Normalverteilungen mit den unterschiedlich geschätzten Parametern (grün: $\mathcal{N} \sim (\mu, \sigma^2)$, blau: $\mathcal{N} \sim (\tilde{x}, mad)$) dargestellt.

Abbildung 4.5 zeigt beispielhaft die Kanaltransformation und anschließende Klassifizierung für die Zielfarben rot und gelb.

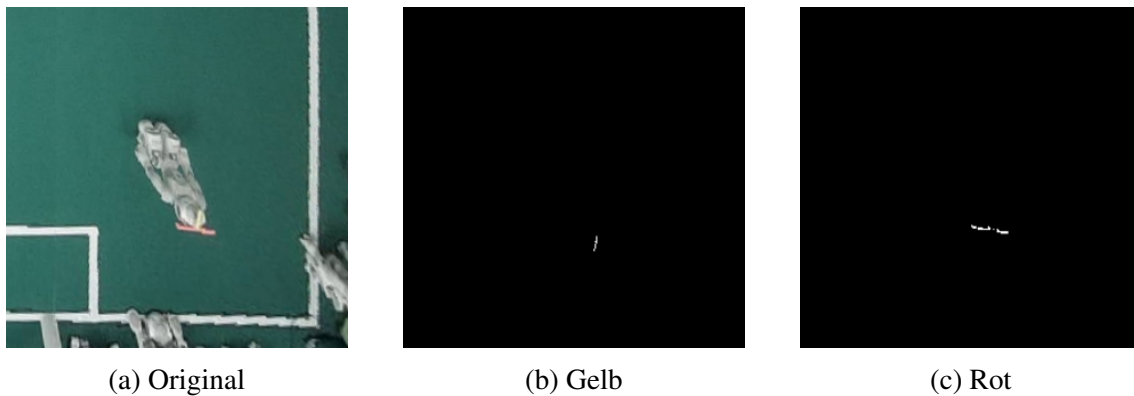


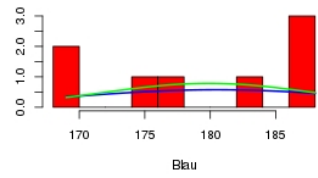
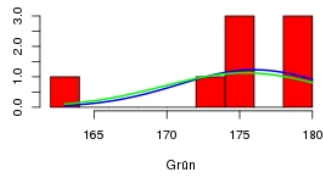
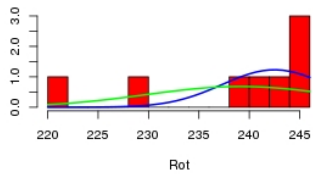
Abbildung 4.5: Kanaltransformation und Klassifizierung

4.4.3 Positions- und Rotationsbestimmung mittels Hauptkomponentenanalyse

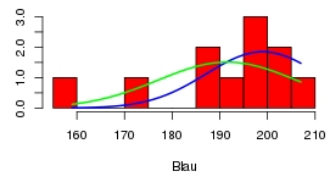
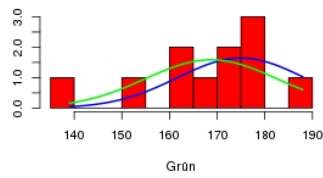
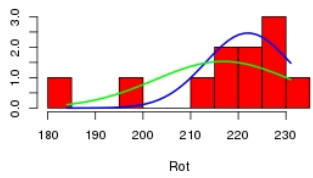
Die Positionsinformation

$$\vec{\chi} = \begin{pmatrix} x \\ y \\ \alpha \end{pmatrix} \quad (4.4)$$

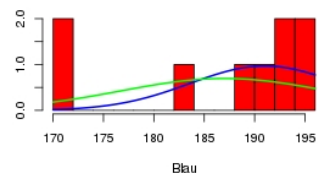
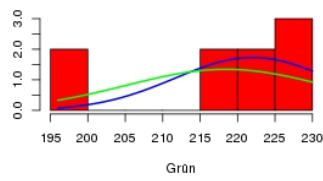
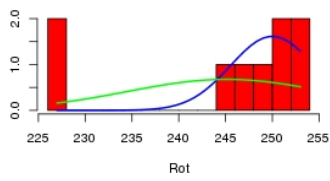
jedes Roboters auf dem Spielfeld, bestehend aus den translativen Komponenten x und y sowie dem Rotationswinkel α sollen aus eingegebenen Bilddaten extrahiert werden. Einfache kreisförmige oder quadratische Marker sind dementsprechend nicht mehr ausreichend und müssen durch Komponenten ergänzt werden die einen Rückschluss auf die



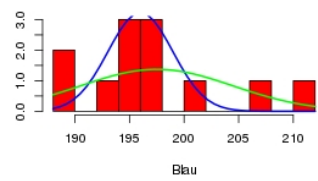
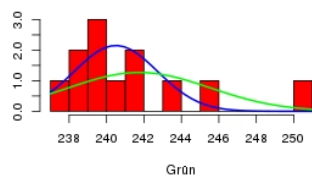
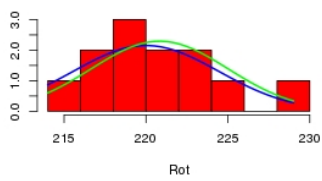
(a) Rot



(b) Magenta



(c) Orange



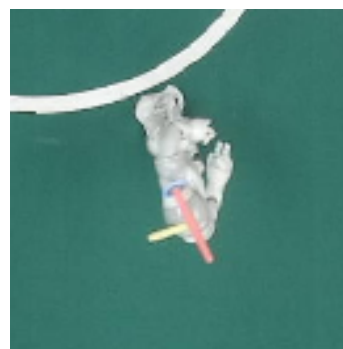
(d) Grün

Abbildung 4.6: Kanalspezifische Farberthistogramme
 Grün: Mittelwert-Schätzer
 Blau: Median-Schätzer

Rotation des Markers und somit des Roboters ermöglichen. Die in Abbildung 4.7 dargestellten länglichen Marker erlauben bereits eine richtungsabhängige Bestimmung der Drehung. Da aber keine weiteren Rückschlüsse auf die Orientierung des Roboters gezogen werden können, wird ein zusätzlicher Marker für die Auflösung dieser Symmetrie benötigt. Abbildung 4.7 verdeutlicht dies beispielhaft. Zur einfacheren Bestimmung der zwei unabhängigen Richtungen wurde in einem weiteren Schritt ein andersfarbiger Marker zur Bestimmung der Richtung angebracht.



(a) einfarbig



(b) zweifarbig

Abbildung 4.7: Aufnahmen eines Roboters mit angebrachten rotationsauflösenden Markern

Ausgehend von zuvor binarisierten Bildern wie sie das im vorhergehenden Abschnitt beschriebene Verfahren liefert, kann nun nach länglichen Objekten innerhalb zuvor bestimmter Bereiche gesucht werden. Ein geeignetes Verfahren dafür stellt die Hauptkomponentenanalyse dar.

Die Hauptkomponentenanalyse dient in der Multivariaten Statistik der Dimensionsreduktion hochdimensionaler Daten bei gleichzeitiger Erhaltung der dem Datensatz inhärenten Information. Konkret werden für einen p -dimensionalen Datensatz $q \leq p$ minimal korrelierte Variablen gesucht die besagten Datensatz möglichst genau beschreiben, wobei die Varianz dieser Hauptkomponenten von erster zu letzter Hauptkomponente abnimmt [15]. Die daraus resultierende Transformation ist orthogonal. Dies bedeutet, dass alle Hauptkomponenten zueinander senkrecht stehen und gemeinsam eine Orthonormalbasis bilden.

Im beschriebenen Anwendungsfall sind die Dimensionalitäten sowohl des Ursprungsraumes als auch des Zielraumes zwei. Dies bedeutet, dass anhand womöglich stark korrelierter Ausgangsvariablen zwei minimal korrelierte Variablen gesucht werden deren Varianz

möglichst groß ist. Das Ergebnis des Verfahrens liefert die normierten Hauptkomponenten \vec{C}_i , deren Längen λ_i sowie das Zentrum \vec{P} des Datensatzes.

Die im vorhergehenden Schritt erzeugten Fenster der Binärbilder können somit auf Hauptkomponenten untersucht werden. Dabei ist zu erwarten, dass die erste Hauptkomponente in Längsrichtung des Markers auftritt. Über die Länge der dazu senkrechten zweiten Hauptkomponente kann des Weiteren auf die Form des Markers geschlossen werden. Überschreitet diese einen definierten Schwellwert, kann davon ausgegangen werden, dass das so erkannte Objekt nicht der länglichen Markerform entspricht und es sich hier um einen falsch positiven Fund der zuvor vorgenommenen Klassifizierung handeln muss. Abbildung 4.8 zeigt beispielhaft die Hauptkomponenten \vec{C}_1, \vec{C}_2 eines Kanalfensters.

Wendet man dieses Verfahren auf die Binärbilder jedes Farbkanals an, so kann damit eine Menge von Markern innerhalb dieser Kanalbilder bestimmt werden. Die relative räumliche Nähe zweier Markerteilstücke in unterschiedlichen Kanälen lassen auf die Zusammengehörigkeit zu einem mehrfarbigen Marker schließen. Das Markerteilstück mit der längsten bestimmten ersten Hauptkomponente bildet dabei die Basis des Markers. Die Bestimmung der eigentlichen Position des Markers kann nun aber nicht nur aus den Schwerpunkten der Datensätze gewonnen werden, da diese einer hohen Varianz entlang der ersten Hauptachse der Markerteilstücke unterworfen sind. Abbildung 4.9 verdeutlicht diesen Sachverhalt übertrieben anhand eines Ausreißers des Datensatzes. Stattdessen kann der Schnittpunkt der Geraden $\vec{P}_1 + r * \vec{C}_1$ und $\vec{P}_2 + s * \vec{C}_2$ als stabilisierter Basispunkt des Markers ermittelt werden:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \vec{P}_1 + r * \vec{C}_1 \quad (4.5)$$

wobei r durch die Lösung des Gleichungssystems

$$\vec{P}_1 + r * \vec{C}_1 = \vec{P}_2 + s * \vec{C}_2 \quad (4.6)$$

definiert ist.

Die Rotation des Markers kann nun aus den Richtungen der Markerteilstücke gewonnen werden. Da die Basis des Markers als größerer der beiden Datensätze im allgemeinen statistisch stabiler ist, wurde die Rotation wie folgt bestimmt:

$$\theta = \begin{cases} \phi + \pi/2 & \text{falls } \angle \overrightarrow{P_2 P_1} - \phi > \frac{\pi}{2} \\ \phi - \frac{\pi}{2} & \text{sonst} \end{cases} \quad (4.7)$$

wobei ϕ den Winkel der ersten Hauptkomponente zur Abszissenachse angibt. Die dargestellte Fallunterscheidung wurde vorgenommen, da die Orientierung der ermittelten Hauptkomponenten variieren kann und bestimmt werden muss, ob die erste Hauptkomponente der Markerbasis gleich oder entgegengesetzt dem durch die Mittelpunkte der Markerteilstücke bestimmten Vektor $\overrightarrow{P_2 P_1}$ orientiert ist.

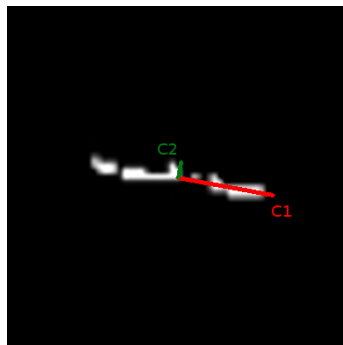


Abbildung 4.8: Hauptkomponenten \vec{C}_1, \vec{C}_2 eines zweidimensionalen Datensatzes



Abbildung 4.9: Ausreißerbedingte Varianz des Mittelpunktes P

4.4.4 Spielerkodierung

Um die Position mehrerer unabhängiger Roboter erfassen und verfolgen zu können, müssen die erfassten Beobachtungen den unterschiedlichen Spielern zugewiesen werden können. Mögliche Verfahren könnten die Roboter zum Beispiel räumlich durch versetzte Anbringung des zweiten Richtungsmarkers oder, im Falle der Verwendung aktiver Marker, durch die Verwendung unterschiedlicher Aussendefrequenzen kodieren. Im dargestellten Szenario hat sich jedoch die Verwendung unterschiedlicher Farbkombinationen als überlegen erwiesen. Als besonders gut durch den verwendeten Algorithmus erkennbar (siehe Abschnitt 4.4.2) stellten sich dabei die Farben rot, magenta, grün und orange heraus. Die Kombination dieser Farben an den unterschiedlichen Markerpositionen (lange oder kurze Seite) - ergibt eine maximal mögliche Spieleranzahl von $4 * 3 = 12$ Spielern, was der maximal erlaubten Spieleranzahl von 5 Robotern je Team genügt.

4.4.5 Algorithmus

Ausgehend von den zuvor beschriebenen Erkenntnissen wurde ein Algorithmus entwickelt der den am Anfang dieses Kapitels genannten Ansprüchen genügen sollte.

Der beschriebene Ablauf dieses Verfahrens sowie dessen Teilschritten, namentlich der Kanalzerlegung, Schwellwertfilterung, Extraktion von Markerhypothesen je Kanal sowie der anschließenden Bestimmung der Roboterposition wurde in den Algorithmen 3 und 4 noch einmal detailliert dargestellt.

Algorithmus 3 Kanaltransformation, Schwellwertfilterung und Markerbestimmung

Bestimmung aller Markerpositionen *hypotheses* aus Eingabebild *img* mittels Fensterungsverfahren nach vorhergehender Kanaltransformation und Schwellwertfilterung

```
1: function THRESHOLDED(img, color, width, height)
2:   thresholded  $\leftarrow$  CREATE(width, height)
3:   for all (x, y)  $\in$  img do
4:     r  $\leftarrow$   $|p_r - color_r|$ 
5:     g  $\leftarrow$   $|p_g - color_g|$ 
6:     b  $\leftarrow$   $|p_b - color_b|$ 
7:     if r  $<$   $\lambda_r$  & g  $<$   $\lambda_g$  & b  $<$   $\lambda_b$  then
8:       thresholded(x, y)  $\leftarrow$  1
9:     else
10:      thresholded(x, y)  $\leftarrow$  0
11:   return thresholded
12:
13: function EXTRACTMARKERS(img, color)
14:   hypotheses  $\leftarrow$   $\emptyset$ 
15:   t  $\leftarrow$  THRESHOLDED(img, color)
16:   for all (x, y)  $\in$  t every STEP do
17:     inner  $\leftarrow$  sub(t, x, y, innerWindow)
18:     outer  $\leftarrow$  sub(t, x, y, outerWindow)
19:     surrounding  $\leftarrow$   $\sum$  outer -  $\sum$  inner
20:     if surrounding = 0 &  $\sum$  inner  $>$  0 then
21:       (center, dir)  $\leftarrow$  PCA(inner)
22:       if (center, dir)  $\notin$  hypotheses then
23:         hypotheses  $\leftarrow$  hypotheses  $\cup$  (center, dir)
24:   return hypotheses
25:
26: function SUB(img, x, y, width)
27:   return subimage with bounds (x-width, x+width, y-width, y+width)
28:
29: function PCA(img)
30:   return (center, direction) of pointcloud as defined by binary image img
31:
32: function CREATE(width, height)
33:   return new image of size width  $\times$  height
```

Algorithmus 4 Positionsbestimmung

Bestimmung aller Markerpositionen je Farbkanal sowie anschließende Ermittlung aller Roboterpositionen $poses$ innerhalb des Eingabebildes img

```
1: function EXTRACTPOSITIONS( $img, width, height$ )
2:    $hypotheses \leftarrow \emptyset$ 
3:    $poses \leftarrow \emptyset$ 
4:   for all  $color \in colors$  do
5:      $hypotheses[color] \leftarrow \text{EXTRACTMARKERS}(img, color)$ 
6:   for all  $(col_1, col_2) \in colors \times colors$  do
7:     for all  $((\vec{c}_1, \vec{d}_1), (\vec{c}_2, \vec{d}_2)) \in hypotheses[col_1] \times hypotheses[col_2]$  do
8:        $\vec{c} \leftarrow \text{solve}(\vec{c}_1 + r \cdot \vec{d}_1 = \vec{c}_2 + s \cdot \vec{d}_2)$ 
9:       if  $d_1^1 > d_2^1$  then
10:          $\vec{d} \leftarrow \begin{pmatrix} d_1^2 \\ d_1^1 \end{pmatrix}$ 
11:       else
12:          $\vec{d} \leftarrow \begin{pmatrix} d_2^2 \\ d_2^1 \end{pmatrix}$ 
13:       if  $\angle \vec{c}_1 - \angle \vec{c}_2 < \frac{\pi}{2}$  then
14:          $\vec{d} \leftarrow -\vec{d}$ 
15:        $poses \leftarrow poses \cup (\vec{c}, \vec{d})$ 
16:   return  $poses$ 
```

4.4.6 Auswertung

Die Qualität des entstandenen Algorithmus wurde abschließend anhand mehrerer handvermessener Testaufnahmen ermittelt. Ausschlaggebend für die Qualität des Verfahrens sollten einerseits der mittlere absolute Fehler zwischen handvermessenen und mithilfe des Verfahrens ermittelten Positionen und Orientierungen, als auch die maximale Streuung der Positionsinformationen sein. Die Streuung wurde als die Differenz aus Minimum und Maximum einer Testaufnahme an festen Punkten des Spielfeldes ermittelt. Die so definierten Metriken ergaben die in Tabelle 4.3 aufgelisteten Messwerte.

Tabelle 4.3: Positions- und Rotationsfehler

Metrik	Wert
Positionsfehler	$\mu = 2.2cm, \sigma^2 = 0.02$
Orientierungsfehler	$\mu = 0.92, \sigma^2 = 0.03$
Streuung Position	$0.6cm$
Streuung Orientierung	1.55

Kapitel 5

Entwicklung eines Lokalisierungsalgorithmus für mobile fußballspielende Roboter

In diesem Kapitel soll die Entwicklung eines Lokalisierungsalgorithmus auf Basis des Iterative Closest Points Verfahrens für das in Kapitel 1 beschriebene Szenario mobiler fußballspielender Roboter in der Standard Platform League im Rahmen des RoboCup Wettbewerbes beschrieben werden.

5.1 Voraussetzungen

Die Roboter der Standard Platform League besitzen unterschiedliche Sensoren zur Erfassung ihrer Umgebung. Diese dienen als Eingabe zur Schätzung ihrer Position, sowie dem Zustand anderer Spielteilnehmer. Die wichtigsten dieser Eingabedaten stellen für das vorgestellte Verfahren vorverarbeitete Kamerabilder sowie odometrische Messdaten dar.

5.1.1 Vorverarbeitete Kamerabilder

Wie in Abschnitt 2.2.1 beschrieben besitzt jeder Roboter zwei am Kopf angebrachte Kameras. Für diese Arbeit wird angenommen, dass diese Bilder in einem vorangestellten

Schritt bereits verarbeitet wurden und direkt nutzbar zur Verfügung stehen.

Die mathematischen Grundlagen dieser Vorverarbeitung können der Masterarbeit Thomas Reinhardts ([16]) entnommen werden.

Die zur Verfügung gestellten Daten sind dabei die Menge aller im Kamerabild C_t zum Zeitpunkt t erkannten Übergänge von Feldlinien zur Spielfeldfläche sowie deren Gradienten, beziehungsweise die angenommenen Normalen zur Spielfeldmarkierung:

$$c = \begin{pmatrix} x \\ y \\ dx \\ dy \end{pmatrix} c \in C_t \quad (5.1)$$

Abbildung 5.1 zeigt beispielhaft eine Visualisierung dieser Daten. Mögliche Fehler sind hierbei vor allem die Erkennung zusätzlicher, nicht vorhandener Spielfeldübergänge sowie Fehler in der Bestimmung der Positionen und Gradienten.

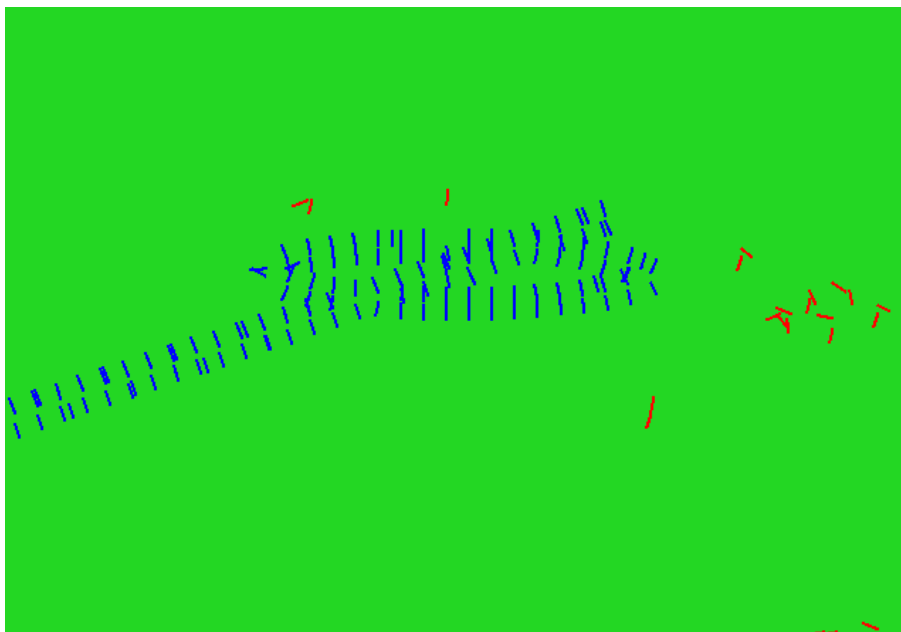


Abbildung 5.1: Vorverarbeitetes Kamerabild
Blick vom Rand des Spielfeldes auf den Mittelkreis
blau: korrekt angenommene Spielfeldübergänge
rot: fehlerhaft angenommene Spielfeldübergänge

Des Weiteren werden für diese Arbeit die Kenntnis aller für die Projektion der Kamerabilder auf das Spielfeld benötigten Körperwinkel des Roboters als gegeben erachtet um somit die Umrechnung von Kamerabildpunkten in Spielfeldkoordinaten bei gegebener Position $\vec{\chi}$ des Roboters zu ermöglichen.

Abbildung 5.2 illustriert die beschriebene Projektion bei angenommener perfekter Schätzung der aktuellen Position des Roboters am Rand des Spielfeldes.

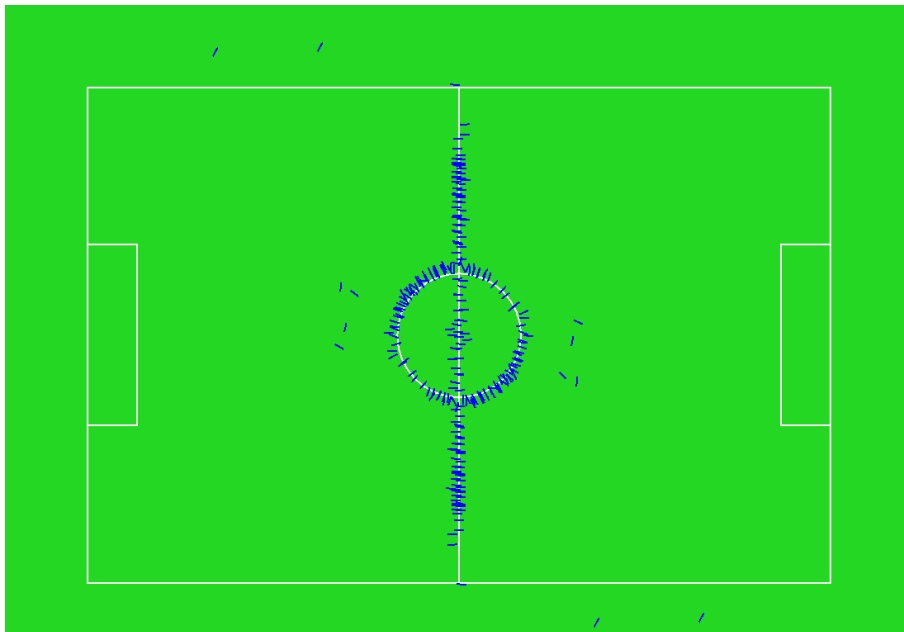


Abbildung 5.2: Projiziertes Kamerabild

5.1.2 Odometrische Daten

Als Odometrie wird die Schätzung der Positionsveränderung eines Systems anhand von Messungen dessen Antriebssystems bezeichnet. Im Falle fußballspielender Roboter bedeutet dies die Schätzung der Positionsveränderung \vec{u}_t anhand der Auswertung der Beschleunigungssensoren des Systems.

$$\vec{u} = \begin{pmatrix} x' \\ y' \\ \alpha' \end{pmatrix} \quad (5.2)$$

Bei perfekter Messung des Systemausgaben sowie angenommener Schätzung $\vec{\chi}_{t-1}$ der vorherigen Position würde somit die Folgeposition $\vec{\chi}_t$ wie folgt bestimmbar sein:

$$\vec{\chi}_t = \vec{\chi}_{t-1} + \vec{u}_t \quad (5.3)$$

Da durch Messsysteme gewonnene Parameterschätzungen nicht als korrekt angenommen werden können, muss eine Schätzung der Positionsveränderung als Summe der tatsächlichen Positionsveränderung \vec{u}_t sowie eines Fehlers $\vec{\varepsilon}_t$ angenommen werden:

$$\hat{\vec{u}}_t = \vec{u}_t + \vec{\varepsilon}_t \quad (5.4)$$

5.2 Ziel

Der finale Algorithmus soll eine bestmögliche Schätzung der aktuellen Position $\hat{\vec{\chi}}_t$ des Roboters zum Zeitpunkt t ausgehend von den vorliegenden Kameradaten, den odometrischen Messdaten der Beschleunigungssensoren sowie der aktuellen Schätzung der Position ermitteln.

5.3 Anwendung des Iterative Closest Points Algorithmus

Der in Abschnitt 3.4 vorgestellte Iterative Closest Points Algorithmus wird im vorgestellten Verfahren für die Suche der plausibelsten Spielfeldposition anhand gegebener Kameradaten eingesetzt. Dazu werden im Kamerabild gefundene Kantenübergänge auf das Spielfeld projiziert und an die durch die Spielfeldbegrenzung gegebene Modellpunktmenge angeglichen.

5.3.1 Synthese der Modellpunktmenge

Da der Iterative Closest Points Algorithmus zwei Punktmenge als Eingaben erwartet, muss das in Abschnitt 2.2.2 dargestellte Spielfeld als solche repräsentierbar sein. Eine

Möglich dafür wäre die Generierung einer hochaufgelösten Punktwolke entlang besagter Markierungen. Dies würde allerdings einen enormen Berechnungsaufwand je nach gewählter Auflösung bedeuten. Eine auflösungsunabhängige Lösung bei nur von der Sourcemenge abhängigem Berechnungsaufwand stellt die Repräsentation des Spielfeldes durch Geometrische Primitive dar. Die zuvor beschriebene Suche nach dem nächsten Modellpunkt für einen Sourcepunkt besteht nun aus zwei Teilschritten:

- Berechnung der Schnittpunkte zwischen der durch den Sourcepunkt und dessen Gradientenrichtung definierten Linie und aller geometrischen Primitive.
- Bestimmung des nächstgelegenen Schnittpunktes aus zuvor bestimmter Menge

Da das Spielfeld ausschließlich durch Strecken und Kreise repräsentierbar ist, beschränkt sich die Suche dieser Schnittpunkte auf die Lösung der jeweiligen Schnittpunktgleichungen.

Die in den Kamerabildern gefundenen Gradienten und Punkte können wie in Abschnitt 5.1.1 beschrieben statistischen Fehlern unterworfen sein. Deshalb müssen außerdem eventuell nicht vorhandene Schnittpunkte durch Punkte auf besagten Primitiven ergänzt werden. Diese werden dabei jeweils auf den Endpunkten dieser platziert.

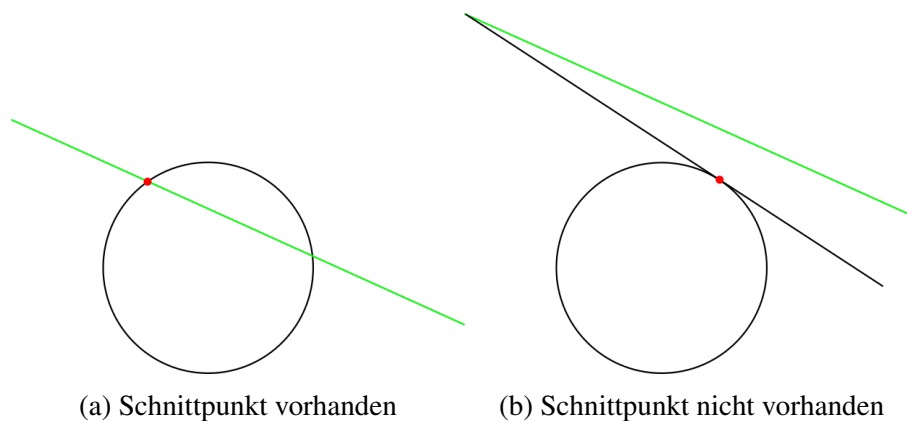


Abbildung 5.3: Synthese der Modellpunkte für kreisförmige Spielfeldabschnitte

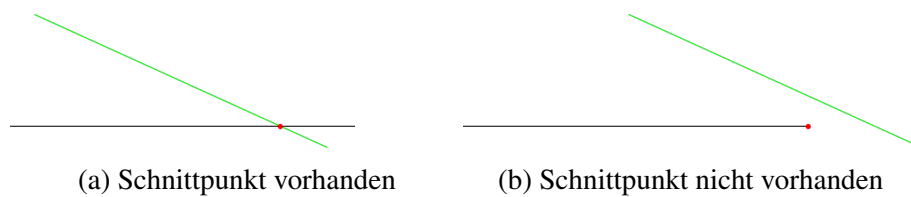


Abbildung 5.4: Synthese der Modellpunkte für Strecken

5.3.2 Auswahl der Startpunkte

Die Konvergenz des Iterative Closest Points Algorithmus ist für lokale Minima garantiert. Nur jedoch bei geeigneter Wahl des Startpunktes gelingt es, die global beste Anpassung der Punktwolke an das Spielfeld zu erreichen. Zwei mögliche Szenarien sind hierbei zu unterscheiden:

- Der Roboter hat sich zuvor lokalisiert und dessen Position soll nun anhand neuerer Daten aktualisiert werden.
- Der Roboter ist gänzlich unlokalisiert (Spielstart) oder die Qualität seiner Lokalisierung ist stark gefallen (Kidnapped Robot Problem).

Im ersten Fall kann die aktuelle Position als Ausgangspunkt des Iterative Closest Points Algorithmus genutzt werden. Da auch die bisherige Position ungenau oder falsch sein kann und außerdem eine Konvergenz gegen ein naheliegendes lokales Minimum nicht ausgeschlossen werden kann, müssen jedoch weitere Startpunkte in Betracht gezogen werden. Abbildung 5.5 zeigt, wie stark die Chance der Konvergenz gegen das korrekte lokale Minimum von der Wahl des Startpunktes abhängt. Es wurde eine künstlich erzeugte Punktmenge in der Form des Torraumes an 10.000 zufälligen Positionen im Umkreis des Torraums mittels des Iterative Closest Points Algorithmus an das Spielfeld angeglichen. Wie deutlich zu sehen ist, hat vor allem ein vertikaler Fehler in der Wahl des Startpunktes eine große Auswirkung auf die Auswahl des korrekten lokalen Minimums. Weiterhin zeigt Abbildung 5.5 dass dieses Verhalten auch bei einer Rotation der Punktmenge um 45 Grad zu beobachten ist.

Im zweiten Fall muss der Algorithmus an verschiedenen Stellen auf dem Spielfeld initialisiert werden, um im optimalen Fall alle möglichen lokalen Minima zu berücksichtigen. Die Auswahl dieser Positionen wurden dabei wie folgt unternommen:

- Zufällige Verteilung der Roboterpositionen
- Zufällige Verteilung der projizierten Datenschwerpunkte
- Ermittlung von Senken und Initialisierung der projizierten Datenschwerpunkte an diesen

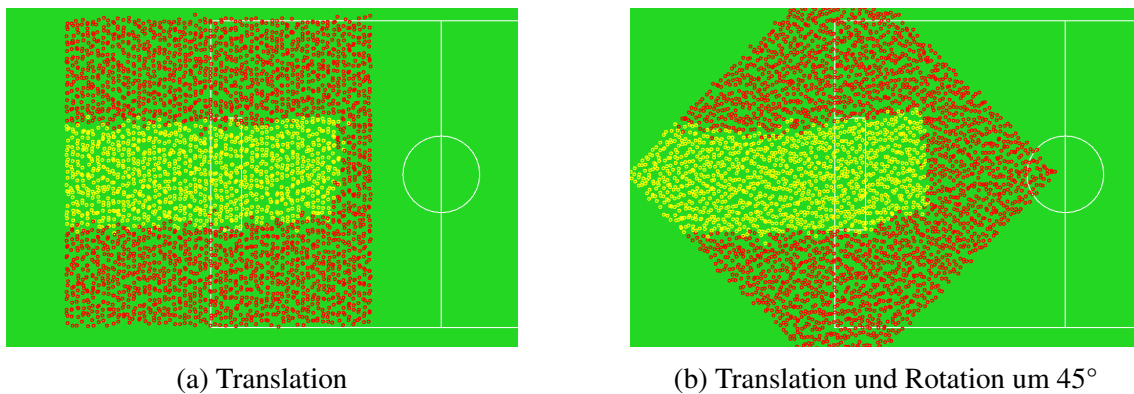


Abbildung 5.5: Abhängigkeit der Konvergenz des Iterative Closest Points Verfahrens von gewählten Starttranslationen und -rotationen

Die Initialisierung der Roboterpositionen an zufälligen Punkten innerhalb des Spielfeldes stellt die einfachste dieser Möglichkeiten dar. Allerdings müssen, um genügend viele lokale Minima miteinzubeziehen und eventuell redundante Initialisierungspunkte zu berücksichtigen, eine Vielzahl von Positionen initialisiert werden. Weiterhin besteht die Möglichkeit der falschen Projektion von Kameradaten außerhalb des Spielfeldes.

Die zweite Methode platziert anstatt der Roboter die Schwerpunkte der auf das Spielfeld projizierten Daten an zufällig gewählten Positionen innerhalb des Spielfeldes. Die Redundanz der generierten Positionen ist dabei ein ähnlich schwerwiegendes Problem wie im erstgenannten Verfahren. In beiden Fällen mussten 20-50 Startpositionen initialisiert werden um eine ausreichende Abdeckung des Spielfeldes und damit aller möglichen Konvergenzpunkte zu gewährleisten.

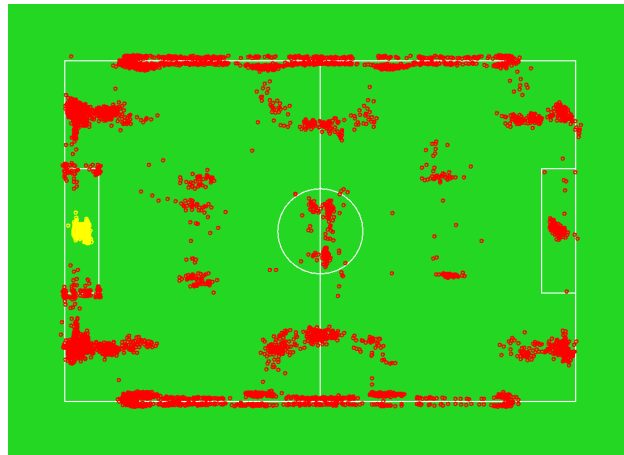
Für die dritte Möglichkeit mussten zunächst alle potenziellen Senken innerhalb des Spielfeldes ermittelt werden. Anschließend wurden die Roboter so initialisiert, dass die projizierten Schwerpunkte der Kameradaten auf diesen liegen. Abbildung 5.6 illustriert die ermittelten Senken für generierte Punktwolken in Form von Torraum, Mittelkreis und T-Stück. Es wurden jeweils 10.000 zufällige Startpunkte und Drehwinkel gewählt. Jeder

Punkt stellt dabei den Punkt dar, an welchem der Algorithmus bei Start an jenem generierten Punkt konvergierte. Gelbe Punkte markieren korrekte Konvergenzen des Verfahrens gegen das ausgewählte Spielfeldsegment und zeigen noch einmal die starke lokale Begrenztheit des Verfahrens. Es wurden die in 5.8 dargestellten Punkte als Senken des Algorithmus identifiziert. Zusätzlich musste untersucht werden, wie viele Startpunkte um eine Senke initialisiert werden müssen, um eine rotationsunabhängige Effektivität des Verfahrens zu ermöglichen. Abbildung 5.9 stellt eine Initialisierung des Algorithmus mit einer Punktwolke in Form des Torraums mit der korrekten Position dessen als Startpunkt dar. Die Punktwolke wurde in $\frac{\pi}{8}rad$ Schritten gedreht und auf korrekte Endrotation untersucht. Wie zu sehen ist, konvergiert der Algorithmus nur bei Wahl der Startrotation mit $-\pi < \alpha < \pi$ korrekt. Um jede Senke müssen also mindestens zwei Positionen initialisiert werden.

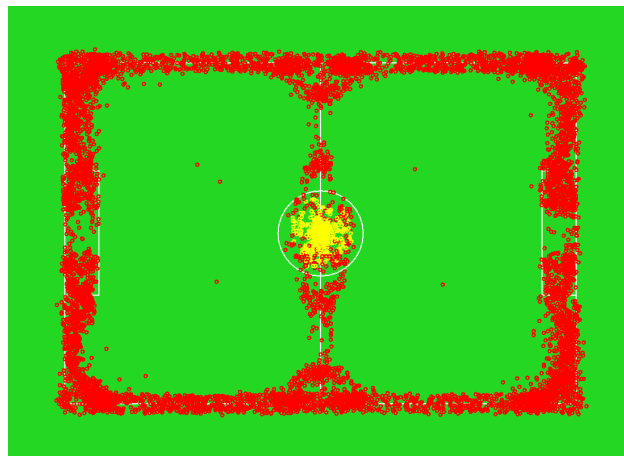
Die Auswahl der dargestellten Senken mit je zwei möglichen Startrotationen bedeutet somit eine Menge von 16 Startpunkten, die zu sehr großer Wahrscheinlichkeit alle Möglichen lokalen Minima abdecken. Die jeweils fehlenden Position auf der gegenüberliegenden Seite des Spielfeldes können durch Spiegelung der gefundenen Endpositionen des Algorithmus mit in die folgenden Schritte einbezogen werden ohne den Iterative Closest Points Algorithmus zusätzlich für diese ausführen zu müssen.

5.3.3 Winkelstabilisierung durch Randlinienerkennung

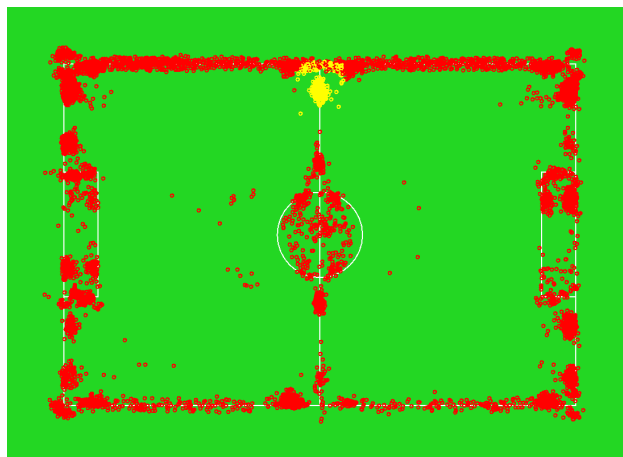
Der zuvor beschriebene grundlegende Iterative Closest Points Algorithmus konvergiert gegen beliebige Liniensegmente sofern die Sourcepunktmenge nur aus Beobachtung einer einzigen Linie besteht. Zur Stabilisierung einer Positionshypothese kann ein somit gewonnener Punkt also nicht genutzt werden, da der Algorithmus grundsätzlich entlang des gesamten Liniensegmentes konvergieren kann - nur bei perfekten zuvor bestehenden Positionshypothesen konvergiert das Verfahren direkt gegen den bereits bekannten Punkt. Stattdessen kann aber der in diesem Schritt bestimmte Winkel des Roboters zu einer Feldlinie genutzt werden, um den Winkel der ursprünglichen Hypothese zu stabilisieren. Zur Erkennung einer reinen Linienform im projizierten Kamerabild wurde die Hauptkomponentenanalyse analog Abschnitt 4.4.3 genutzt.



(a) Torraum



(b) Mittelkreis



(c) T-Stück

Abbildung 5.6: Identifizierte Senken für generierte Punktwolken.

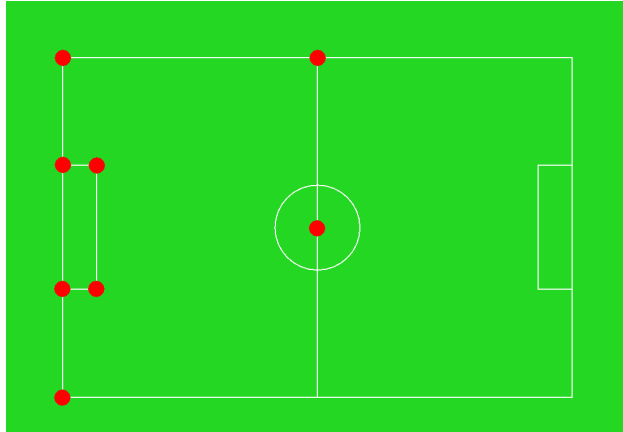


Abbildung 5.8: Ermittelte Senken des dargestellten Spielfeldes

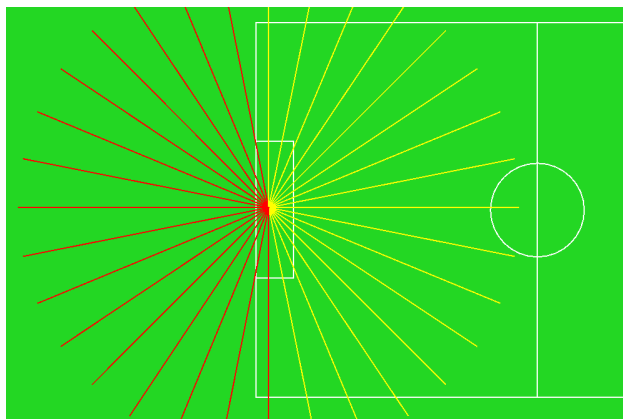


Abbildung 5.9: Einfluss der Rotation auf die Konvergenz des ICP-Verfahrens

5.3.4 Behandlung von Ausreißern

Während der Arbeit an dem hier vorgestellten Verfahren zeigte sich eine hohe Anfälligkeit des Iterative Closest Points Algorithmus gegenüber im Bild befindlichen Ausreißern. Diese Ausreißer lassen sich vornehmlich wie folgt unterscheiden:

- Zufällige Ausreißer. Diese sind in keinem Maße mit den Daten des Kamerabildes korreliert.
- Ausreißer durch Verzerrungseffekte.

Abbildungen 5.1 und 5.2 zeigten bereits beispielhaft die erste dieser beiden Arten entstehender Eingabefehler: die Ausreißer nahe dem Mittelkreis des Spielfeldes stehen im keinem Zusammenhang mit einer Spielfeldlinie. Derartige Erscheinungen wurden in beliebigen Bildern an beliebigen Positionen des Spielfeldes beobachtet. Besonders häufig traten diese an den Rändern des Kamerabildes auf.

Unkorrelierte Ausreißer wurden mit einem einfachen in Algorithmus 5 beschriebenen Verfahren entfernt.

Algorithmus 5 Ausreißerbeseitigung

Bereinigung der Menge aller Spielfeldübergänge in Kamerabild C von zufälligen Ausreißern

```
1: function REMOVEOUTLIERS( $C$ )
2:    $remove \leftarrow \emptyset$ 
3:   for all  $c_1 \in C$  do
4:      $close \leftarrow 0$ 
5:     for all  $c_2 \in C$  do
6:       if  $c_1 \neq c_2$  &  $distance(c_1, c_2) < distanceThreshold$  &
          $angleBetween(c_1, c_2) < angleThreshold$  then
7:          $close \leftarrow close + 1$ 
8:       if  $close > closeThreshold$  then
9:          $remove \leftarrow removed \cup c_1$ 
10:   $C^* \leftarrow C \setminus remove$ 
11:  return  $C^*$ 
```

Wie ersichtlich ist, werden somit alle Gradienten aus der Menge des Kamerabildes entfernt, in deren unmittelbarer Umgebung sich keine ausreichend große Anzahl ähnlich

gerichteter Gradienten befindet.

Da diese Bereinigung des Datensatzes für jedes Kamerabild ausgeführt werden muss, wurde auf den Ausschluss von Punkten entsprechend der Häufigkeitsverteilung der Parameter Entfernung, Winkel und Anzahl je Bild verzichtet. Stattdessen wurden diese a priori aus einer Vielzahl von Kamerabildern durch Analyse der entsprechenden Histogramme der Parameter bestimmt.

Die zweitgenannte Art von Ausreißern zeigte sich sich als systematisch für alle Kameraaufnahmen. Sowohl Qualität als auch Quantität der Ausreißer stellten sich als proportional zur Entfernung der auf das Spielfeld projizierten Kamerapunkte zum Roboter heraus. Beispielhaft stellt Abbildung 5.10 leicht verzerrte Spielfeldlinien im Torbereich des Spielfeldes dar.

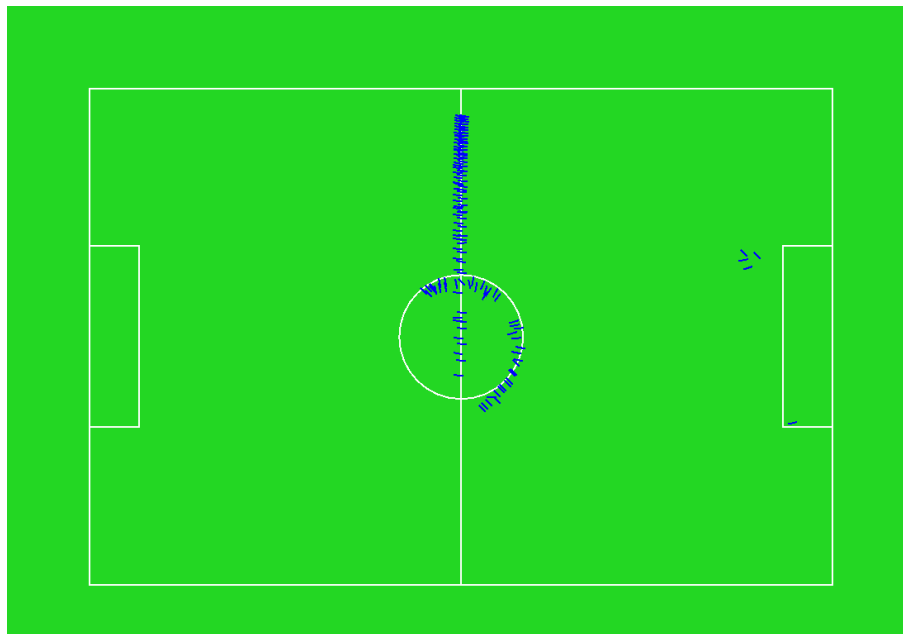


Abbildung 5.10: Verzerrungseffekte in großer Entfernung zum Aufnahmeort des Bildes

Diesem Sachverhalt musste in den Teilschritten des Iterative Closest Points Algorithmus Beachtung geschenkt werden. Verschiedene Arbeiten benennen grundsätzlich zwei Möglichkeiten zur Behandlung unerwünschter Werte: Ausschluss und Gewichtung. Im Rahmen dieser Arbeit wurden beide Verfahren geprüft. Der Ausschluss zu weit vom Roboter entfernt liegender Punkte hat sich dabei als zuverlässiger erwiesen. Außerdem verringert

sich durch Ausschluss etwaiger Ausreißer die effektive Mächtigkeit der Sourcepunktmenge und verringert somit den benötigten Berechnungsaufwand.

Des Weiteren wurden die Auswirkungen des von Shaoyi et al [17] vorgestellten Scaled ICP Verfahrens auf die Qualität der erzielten Punktwolkenangleichungen untersucht. Besagtes Verfahren erweitert die Klasse erlaubter Transformationen in den Iterationen des Iterative Closest Points Algorithmus von Translation und Rotation um eine zusätzliche Skalierung der Punktwolken. Leider musste festgestellt werden dass die bestmögliche Transformation der Punktwolken bei gegebener Synthese der Modellpunktmenge nahezu ausschließlich diejenigen Skalierungen bevorzugt, welche die gesamte Quellenpunktmenge auf einzelne Geraden abbilden.

5.4 Rekursive Filterung und Zuordnung korrespondierender Funde

Das bisher beschriebene Verfahren liefert für jedes Kamerabild die Menge der Spielfeldpunkte an welchen der Iterative Closest Points Algorithmus konvergierte. Wie zuvor beschrieben, kann sich in dieser Menge eine gegebenenfalls große Anzahl falsch positiver Funde befinden. Aufgrund statistischer Abweichungen kann nicht immer der Fund mit dem geringsten Fehlermaß als bester Fund angenommen werden. Vielmehr müssen aus aufeinanderfolgenden Kamerabildern diejenigen Spielfeldpositionen ermittelt werden, an welchen der Algorithmus ähnlich gut konvergiert. Dazu müssen jedoch auch Funde aufeinanderfolgender Kamerabilder einander zugeordnet werden können.

Zur rekursiven Filterung aufeinanderfolgender Funde wurden für diese Arbeit multiple Instanzen des Kalman-Filters wie in Abschnitt 3.3.1.4 beschrieben verwendet. Nach Lösen des zuvor beschriebenen Korrespondenzproblems können durch den Einsatz dieser Filter mehrere Hypothesen der aktuellen Roboterposition ermittelt und verfolgt werden. Die Zuordnung der Funde erfolgt nach in Algorithmus 6 beschriebener Regel.

Ausgehend vom von Ausreißern bereinigten Kamerabild C_t^* , den odometrischen Messdaten \vec{x}_t sowie den aktuell besten Positionshypothesen $robots$ ermittelt der Algorithmus die besten Hypothesen zum Zeitpunkt t . Dazu wird in Zeile 4 zuerst die in Abschnitt 3.3.1.2 beschriebene Vorhersage der Filterinstanzen ausgeführt. Anschließend werden diese vor-

hergesagten Positionshypothesen um die in Abschnitt 5.3.2 beschriebenen Positionen um Senken des Algorithmus erweitert, um nicht abgedeckte Senken ebenfalls zu berücksichtigen. Diese erweiterte Menge von Hypothesen stellt nun die Grundlage für die in Zeile 7 stattfindenden Ausführungen des zuvor beschriebenen Iterative Closest Points Algorithmus dar. Die nach Konvergenz der Instanzen ermittelten Positionen werden anschließend für die Korrektur der bestehenden Positionshypothesen genutzt. In den Zeilen 13-16 wird immer dann eine neue Hypothese erzeugt, wenn sich einer der Konvergenzpunkte in relativer Nähe zu einer bestehenden Hypothese befindet. Die relative Nähe bestimmt sich dabei aus der Qualität besagter Hypothese $quality(r)$ sowie den Gewichtungen λ_1, λ_2 als Koeffizienten besagter Qualität zum Vergleich mit der Distanz des Punktes und der Hypothese, respektive dem Winkel zwischen den Orientierungen derselbigen. Sollte sich kein Konvergenzpunkt in der Nähe einer Positionshypothese r befinden, so wird diese Hypothese unverändert übernommen. Sollte diese Hypothese auch weiterhin nicht in der Nähe von Konvergenzpunkten liegen, wird diese nach einigen Iterationen des Algorithmus im letzten Schritt desselbigen wie weiter unten beschrieben entfernt. Die neu erzeugte Hypothese wird durch den in Abschnitt 3.3.1.2 beschriebenen Korrekturschritt der entsprechenden Filterinstanzen bestimmt. Es gilt zu beachten, dass hierbei unter Umständen mehr Hypothesen entstehen als zuvor bestanden. Entstehen für eine zuvor bestehende Hypothese mehrere neue Hypothesen mit ausreichendem Unterschied zueinander, so wird dies als Track-Splitting bezeichnet. Im Gegensatz zu diesem durchaus erwünschten Verhalten, können sich nun aber auch mehrere zuvor unterschiedliche Hypothesen in unmittelbarer Nähe zueinander befinden. Um die Anzahl von Hypothesen zu verringern werden diese im sogenannten Track-Merging in Zeile 21 auf ausreichend unterschiedliche Hypothesen reduziert. In einem letzten Schritt werden die n besten Hypothesen in die nächste Iteration des Algorithmus übernommen. Somit wird sichergestellt, dass Hypothesen schlechter Qualität keinen weiteren unnötigen Berechnungsaufwand erfordern.

Algorithmus 6 Multihypothesen-Kalman-Filter

Prädiktion und Aktualisierung mehrerer Filterinstanzen sowie Vergrößerung und Verkleinerung dieser Menge mittels Track-Merging und -Splitting

```
1: function FILTER( $C_t^*, \vec{u}_t, robots$ )
2:    $predicted \leftarrow \emptyset$ 
3:   for all  $r \in robots$  do
4:      $predicted \leftarrow predicted \cup predict(robot, \vec{u}_t)$ 
5:    $hyptheses \leftarrow \emptyset$ 
6:   for all  $r \in predicted$  do
7:      $(h, error) \leftarrow ICP(r, C_t^*, FIELD)$ 
8:     if  $error < errorThreshold$  then
9:        $hyptheses \leftarrow hypotheses \cup h$ 
10:     $hyptheses \leftarrow merge(hyptheses)$ 
11:   $corrected \leftarrow \emptyset$ 
12:   $newTracks \leftarrow \emptyset$ 
13:  for all  $r \in robots$  do
14:    for all  $h \in hypotheses$  do
15:      if  $distance(h, r) < \lambda_1 \cdot quality(r) \ \& \ angle(h, r) < \lambda_2 \cdot quality(r)$  then
16:         $newTracks \leftarrow corrected \cup correct(r, h)$ 
17:       $corrected \leftarrow corrected \cup r$ 
18:      if  $newTracks = \emptyset$  then
19:        else
20:           $corrected \leftarrow corrected \cup newTracks$ 
21:      select  $n$  best from  $merge(corrected)$ 
22:
23:  return  $n$  best robot hypotheses
24:
25: function ICP( $r, S, M$ )
26:  return transformation  $T$  between  $S$  and  $M$  with initial position  $r$ 
```

Kapitel 6

Auswertung

Der in Kapitel 5 vorgestellte Algorithmus soll nun anhand des in Kapitel 4 beschriebenen Verfahrens evaluiert werden. Besonderes Augenmerk soll hierbei auf die Ermittlung initialer Positionsschätzungen ohne Vorwissen entweder durch den vollkommenen Kontrollverlust des Lokalisierungssystems, dem Kidnapped Robot Szenario oder zum Start einer Spielpartie gelegt werden. Wie bereits beschrieben, wird die durch Deckenkameras gewonnene Position als beste ermittelbare Position angenommen und deshalb als Goldstandard zur Bewertung aller anderen Verfahren gewertet. Das hier vorgestellte Verfahren soll sich aber nicht nur an diesem, sondern auch am bisher bestehenden Lokisierungsalgorithmus messen.

6.1 Versuchsaufbau

Abbildung 6.1 stellt eine Auswahl vordefinierter Positionen und Blickrichtungen dar von welchen markante Ausschnitte des Spielfeldes ohne vorherige Information über Position und Rotation des Roboters beobachtet werden sollen. Die Testfälle 1 bis 3 dienen der Untersuchung des Algorithmus im Hinblick auf dessen Potenzial zur Gewinnung der aktuellen Spielfeldposition aus der Beobachtung einzelner Spielfeldabschnitte. Diese Abschnitte sind: der Mittelkreis (Testfall 1), ein T-Stück an der Mittellinie (Testfall 2) sowie dem Torraum (Testfall 3). Testfall 4 erweitert diese Testfälle um die Untersuchung der Aggregation und Filterung zeitlich aufeinanderfolgender Beobachtungen multipler Spiel-

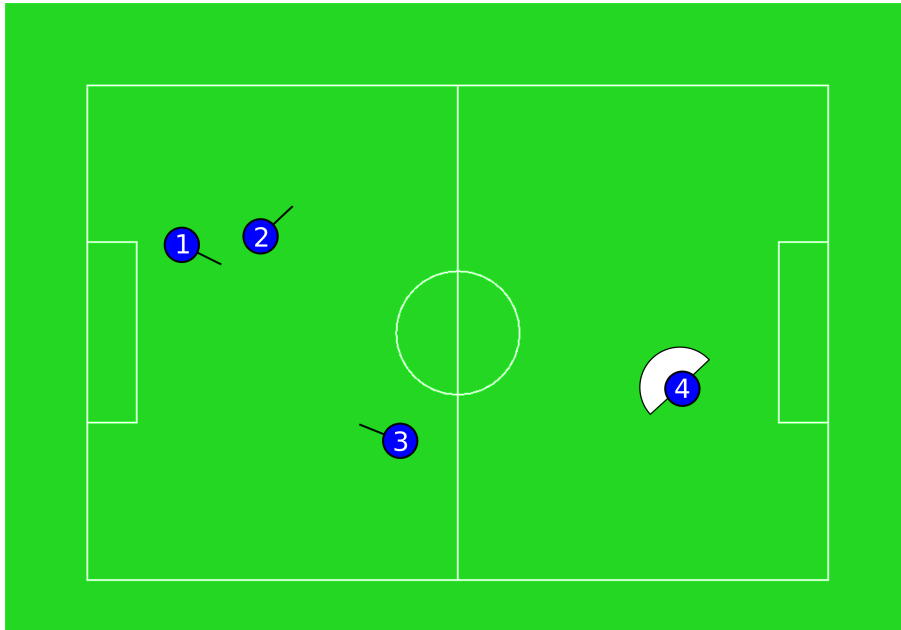


Abbildung 6.1: Roboterpositionen mit Blickrichtung und ggf. Öffnungswinkel

feldabschnitte durch Schwenken des Roboterkopfes. Der weiße Halbkreis dient der Visualisierung des ungefähren Winkel in welchem die Blickrichtung des Roboters dabei schwankt.

6.1.1 Ergebnisse

Abbildung 6.5 zeigt die Ergebnisse zuvor beschriebener Versuche. In jedem Fall wurden die Positionsfehler sowohl für das Standardverfahren als auch für das in dieser Arbeit vorgestellte, auf dem Iterative Closest Points Algorithmus basierende Verfahren zusammen mit den Mittelwerten und Standardabweichungen der jeweiligen Messreihen dargestellt.

Im ersten Testfall zeigt sich ein ähnliches Ergebnis für beide Algorithmen mit einem Positionsfehler von etwa 30 cm mit vergleichsweise geringer Schwankung.

Im zweiten Testfall konnte das Standardverfahren keine Positionsinformation schätzen und initialisierte diese deshalb auf Position $(4.5, 3, 0)^T$ in der Mitte des Spielfeldes. Es gilt zu bemerken, dass das neuere Verfahren eine Positionsschätzung liefert, diese jedoch mit einem Durchschnittlichen Fehler von 50 cm relativ ungenau ist.

In Testfall 3 wiederum schätzten beide Verfahren die Position des Roboters nahezu korrekt.

In Testfall 4 schätzten beide Algorithmen ähnliche Ergebnisse - das ICP-basierte Verfahren jedoch stabiler. Zwar weist das Standardverfahren einen geringeren minimalen Fehler auf - der maximale Fehler sowie die Schwankung der Position sind jedoch erheblich höher. Der Mittelwert der durch dieses Verfahren gewonnenen Positionen schwanken in etwa um den des ICP-basierten Verfahrens.

Diese Ergebnisse lassen bisher zwei Interpretationen zu: beide Verfahren weisen eine systematische Verschiebung der Positionsschätzung abhängig von der tatsächlichen Position des Roboters auf. Außerdem liefert das ICP-basierte Verfahren stabilere Positionsschätzungen bei gleichzeitig höherem minimalem Fehler dieser Schätzungen.

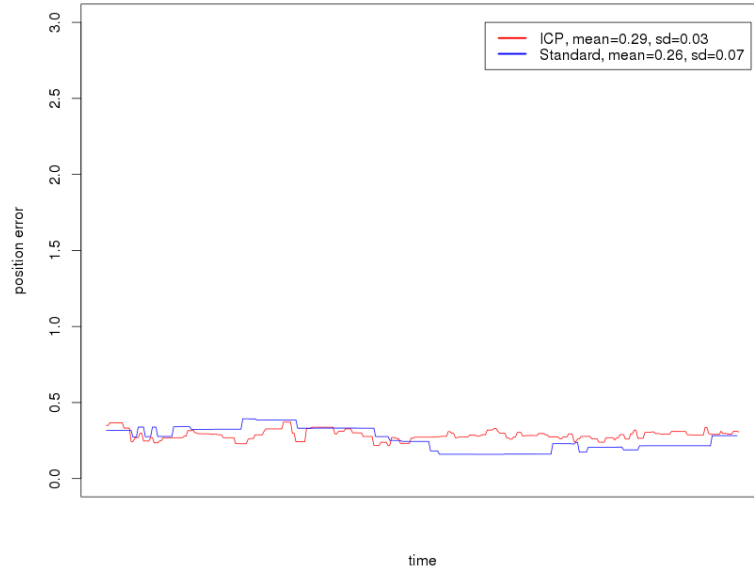


Abbildung 6.2: Positionsevaluation Testfall 1: Blick auf Mittelkreis

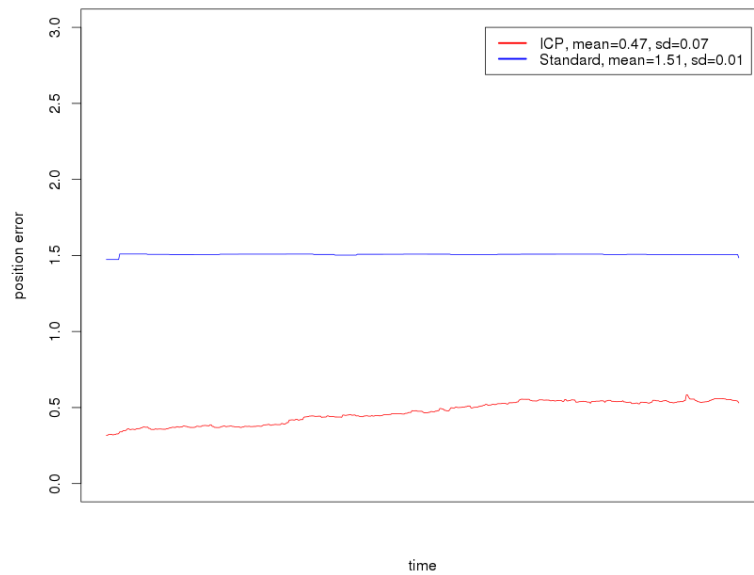


Abbildung 6.3: Positionsevaluation Testfall 2: Blick auf Torraum

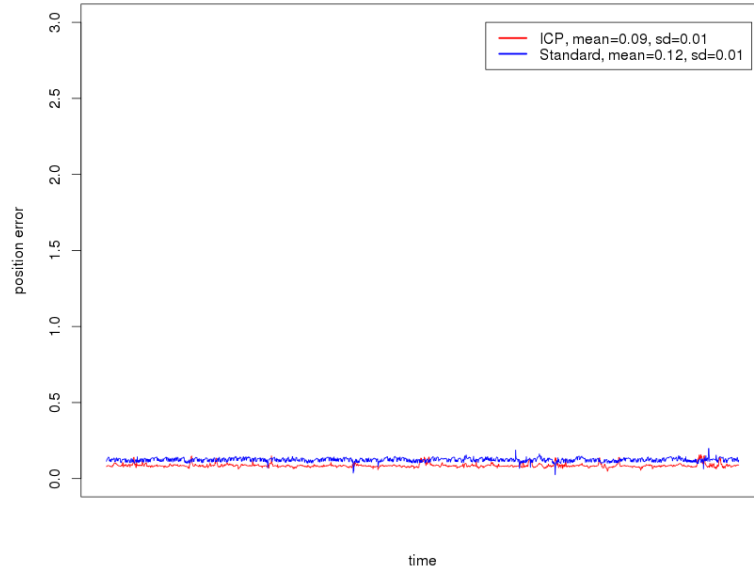


Abbildung 6.4: Positionsevaluation Testfall 3: Blick auf T-Stück

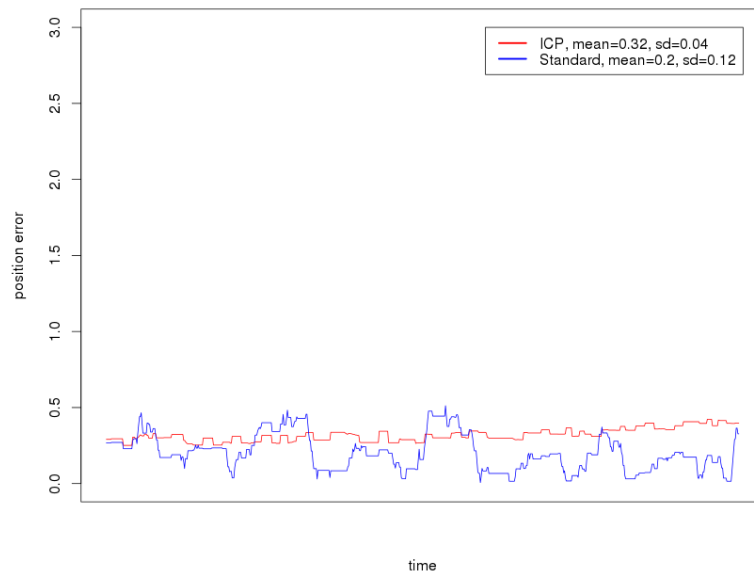


Abbildung 6.5: Positionsevaluation Testfall 4: Blick auf Mittelkreis, Kopf bewegt

Kapitel 7

Zusammenfassung und Ausblick

In dieser Arbeit wurden zwei gleichsam verwandte und dennoch in ihren Voraussetzungen grundverschiedene Problemstellungen der Lokalisierung mobiler fußballspielender Roboter der Standard Platform League des RoboCups untersucht. Ihnen gemein war die Anforderung, die Position mobiler Roboter anhand gegebener Eingabedaten zu bestimmen. Der Unterschied lag dabei in den Anforderungen welche die Verfügbarkeit und Qualität der vorliegenden Sensordaten an jene Lokalisierungsverfahren stellte. Das erste entstandene Verfahren konnte auf hochauflösende externe Kameras zurückgreifen, um somit eine auf wenige Zentimeter genaue Positionsauflösung der zu beobachtenden Roboter zu ermöglichen. Gleichzeitig lieferte es die Grundlage zur Evaluation der Ergebnisse des zweiten Verfahrens. Dieses sollte die Positionen von Robotern ausschließlich anhand der von ihnen selbst gelieferten Kamera- und Odometriedaten mithilfe einer modifizierten Variante des Iterative Closest Points Verfahrens ermitteln. Die beiden Hauptaspekte dieser Arbeit, nämlich der Definition eines Goldstandards zur Bewertung aktueller und zukünftiger Lokalisierungsverfahren unter Laborbedingungen sowie die Untersuchung der Eignung des Iterative Closest Points Verfahrens zur Bestimmung von Roboterpositionen in besonderer Hinsicht auf dessen Möglichkeiten zur Ermittlung initialer Positionsschätzungen, wurden somit ausgiebig untersucht.

Die in im Rahmen dieser Arbeit gewonnen Erkenntnisse zeigen, dass mithilfe des Iterative Closest Points Algorithmus sehr verlässlich Positionsschätzungen innerhalb des Spielfeldes der Standard Platform League bereits anhand kleiner gesichteter Ausschnitte desselbigen ermitteln werden können. Es wurde außerdem dargestellt, welche Schwierig-

keiten und Besonderheiten der Einsatz jenes Verfahrens in dieser Versuchsdomäne nach sich zog und welche Schritte unternommen werden können, um diese zu lösen. Weiterhin wurde beispielhaft gezeigt, wie stabilisierte Positionsschätzungen mithilfe rudimentärer Filterungstechniken unter Einsatz eines Multihypothesen-Kalmanfilters zu ermitteln sind.

Der Einsatz dieses Verfahrens kann grundsätzliche Probleme bei gegebener Sensorik, wie der systematischen Verzerrung von Eingabedaten, nicht allein bewältigen. Allerdings bietet es die Grundlage für weitere Untersuchungen dieses Sachverhalts.

Diese Aspekte können nun die Grundlage für den weiteren Einsatz dieses Verfahrens für das genannte Szenario darstellen.

Abbildungsverzeichnis

2.1	Nao Roboter mit für den RoboCup relevanten Sensoren (Auswahl)	6
2.2	Spielfeldmaße	8
3.1	Roboterposition	10
4.1	Entzerrte Beispielaufnahme einer Deckenkamera	22
4.2	Zwischenschritte der LED-Lichtpunkterkennung	28
4.3	Beispielaufnahmen eines Roboters mit angebrachten Leuchtdioden bei unterschiedlichen Einfallswinkeln	29
4.4	Aufnahmen eines Roboters mit unterschiedlichen angebrachten Markern aus reflektierender Folie und Papier	31
4.5	Kanaltransformation und Klassifizierung	32
4.6	Kanalspezifische Farbwertistogramme	33
4.7	Aufnahmen eines Roboters mit angebrachten rotationsauflösenden Markern	34
4.8	Hauptkomponenten \vec{C}_1, \vec{C}_2 eines zweidimensionalen Datensatzes	36
4.9	Ausreißerbedingte Varianz des Mittelpunktes P	36
5.1	Vorverarbeitetes Kamerabild	42
5.2	Projiziertes Kamerabild	43
5.3	Synthese der Modellpunkte für kreisförmige Spielfeldabschnitte	45
5.4	Synthese der Modellpunkte für Strecken	46

5.5	Abhängigkeit der Konvergenz des Iterative Closest Points Verfahrens von gewählten Starttranslationen und -rotationen	47
5.6	Identifizierte Senken für generierte Punktwolken.	49
5.8	Ermittelte Senken des dargestellten Spielfeldes	50
5.9	Einfluss der Rotation auf die Konvergenz des ICP-Verfahrens	50
5.10	Verzerrungseffekte in großer Entfernung zum Aufnahmepunkt des Bildes	52
6.1	Roboterpositionen mit Blickrichtung und ggf. Öffnungswinkel	58
6.2	Positionsevaluation Testfall 1: Blick auf Mittelkreis	60
6.3	Positionsevaluation Testfall 2: Blick auf Torraum	60
6.4	Positionsevaluation Testfall 3: Blick auf T-Stück	61
6.5	Positionsevaluation Testfall 4: Blick auf Mittelkreis, Kopf bewegt	61

Algorithmenverzeichnis

1	Iterativer Ablauf des Iterative Closest Points Algorithmus	20
2	Farbklassifikation	27
3	Kanaltransformation, Schwellwertfilterung und Markerbestimmung	38
4	Positionsbestimmung	39
5	Ausreißerbeseitigung	51
6	Multihypothesen-Kalman-Filter	55

Literatur

- [1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda und E. Osawa, *RoboCup: The Robot World Cup Initiative*, 1995.
- [2] IBM Research, *Deep Blue*. Adresse: <http://www.research.ibm.com/deepblue/> (besucht am 03.02.2016).
- [3] Leipziger Messe GmbH, *RoboCup 2016 Leagues*, 2016. Adresse: <http://www.robocup2016.org/en/leagues> (besucht am 24.01.2016).
- [4] Aldebaran Robotics, *Nao h25 Technical Specifications*. Adresse: http://doc.aldebaran.com/2-1/family/nao_h25/index_h25.html (besucht am 23.12.2015).
- [5] S. Thrun, W. Burgard und D. Fox, *Probabilistic Robotics*. The MIT Press, 2006.
- [6] R. Kalman, “A new approach to linear filtering and prediction problems”, *Journal of basic Engineering*, Bd. 82, Nr. 1, S. 35–45, 1960.
- [7] E. a. Wan und R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation”, *Technology*, Bd. v, S. 153–158, 2000.
- [8] L. Bazzani, D. Bloisi und V. Murino, “A comparison of multi hypothesis kalman filter and particle filter for multi-target tracking”, *Performance Evaluation of Tracking and Surveillance workshop at CVPR*, 2009.
- [9] G. Jochmann, S. Kerner, S. Tasse und O. Urbann, “Efficient multi-hypotheses unscented kalman filtering for robust localization”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Bd. 7416 LNCS, S. 222–233, 2012.

- [10] A. Burchardt, T. Laue und T. Röfer, “Optimizing particle filter parameters for self-localization”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Bd. 6556 LNAI, S. 145–156, 2011.
- [11] P. Anderson, Y. Hunter und B. Hengst, “An icp inspired inverse sensor model with unknown data association”, *2013 IEEE International Conference on Robotics and Automation*, S. 2713–2718, 2013.
- [12] K. Low, “Linear least-squares optimization for point-to-plane icp surface registration”, *Chapel Hill, University of North Carolina*, Nr. February, S. 2–4, 2004.
- [13] X. Guo, C. Wang und R. Yan, “An electromagnetic localization method for medical micro-devices based on adaptive particle swarm optimization with neighborhood search”, *Measurement*, Bd. 44, Nr. 5, S. 852–858, 2011.
- [14] MiBeacons, “Beacons : the technical overview”, 2014.
- [15] I. T. Jolliffe, “Principal component analysis, second edition”, *Encyclopedia of Statistics in Behavioral Science*, Bd. 30, Nr. 3, S. 487, 2002.
- [16] T. Reinhardt, *Kalibrierungsfreie Bildverarbeitungsalgorithmen zur echtzeitfähigen Objekterkennung im Roboterfußball*, 2011.
- [17] D. Shaoyi, Z. Nanning, Y. Shihui, Y. Qubo und W. Yang, “An extension of the icp algorithm considering scale factor”, *Proceedings - International Conference on Image Processing, ICIP*, Bd. 5, S. 193–196, 2006.

Anlagen auf DVD

- Beispielaufnahmen: Deckenkameras
- Beispielaufnahmen: Roboterkameras und -odometrie
- Ausführbarer Programmcode: deckenkamerabasierte Lokalisierung
- Ausführbarer Programmcode: ICP-basierte Lokalisierung
- Ausführbarer Programmcode: Lokalisierungsevaluation