

Ballverifikation

Robocup Projektbericht

Lasse Einig
Kay Peikert
Anja Richter

Informatik B.Sc.
WS 2010/11 & SoSe 2011

Universität Hamburg
Betreuer: Janis Schönefeld

10. Februar 2012

Inhaltsverzeichnis

1. Einleitung	4
1.1. RoboCup Soccer (Humanoid, Middle-Size, Simulation, Small-Size, Standard Platform Liga(SPL), Mixed-Reality)	5
1.2. Rescue League (Robot, Simulation)	6
1.3. RoboCup@Home	7
1.4. RoboCup Junior (Soccer, Dance, Rescue)	7
2. Standard Platform League im Projekt	8
2.1. Objekterkennung	8
2.2. Kollisionserkennung	9
2.3. Bewegungsabläufe	9
2.4. Verhalten	9
2.5. Hardware des Roboters ansteuern	10
3. Aufgabenstellung	11
4. Vorüberlegungen	12
4.1. Schwierigkeiten der Objekterkennung	12
4.2. Annahme	13
5. Ausgangssystem	15
6. Realisierung	15
6.1. Ansatz	15
6.2. Histogramm	16
6.3. Farbräume	17
6.3.1. RGB	17
6.3.2. HSV	19
7. Funktionsweise und Implementierung der Ballverifikation	21
8. Bewertung unseres Verfahrens	26

9. Fazit	27
10. Anhang	28

1. Einleitung

Die Teilnahme am RoboCup ist Teil des gleichnamigen Projektes an der Universität Hamburg. *RoboCup ist eine internationale Initiative zur Förderung der Forschung und interdisziplinären Ausbildung in den Bereichen Künstliche Intelligenz und autonome mobile Systeme.*[1]

Im RoboCup werden Wettbewerbe in verschiedenen Disziplinen bzw. Ligen ausgetragen. Da der RoboCup der Forschung und somit dem Fortschritt dient, werden nach dem Wettkampf die Lösungsansätze der Teams veröffentlicht. Dies soll verhindern, dass ein Team „gute“ Software so lange unverändert benutzt, wie die anderen Teams damit zu schlagen sind. Die einzelnen Ligen unterscheiden sich vor allem in der Art der Roboter und in der Art der zu bewältigenden Aufgaben. Diese Unterscheidung ist aufgrund der derzeitigen Problematik in einzelnen Bereichen der Robotik notwendig.



Abbildung 1: SPL-Spielfeld mit Naos

Da die humanoide Fortbewegung der Roboter noch ein eigenes Forschungsgebiet ist und die entsprechenden Roboter sich derzeit noch sehr langsam über das Spielfeld bewegen, eignen sich diese beispielsweise noch nicht für die praktische Erforschung von kollektivem Verhalten bzw. Schwarmintelligenz. Um jedoch auch diesen Bereich voran zu bringen, verzichtet man in anderen Ligen auf eine anspruchsvolle Fortbewegung und spielt mit radgetriebenen Robotern.

Die Anforderungen bzw. Spielregeln der einzelnen Ligen werden zusätzlich jedes Jahr dem Fortschritt angepasst, um den Anspruch und die Anforderungen an die Leistungsfähigkeit der Roboter zu erhöhen.

Folgende Ligen werden derzeit angeboten

- RoboCup Soccer
- Rescue League
- RoboCup@Home
- RoboCup Junior

1.1. RoboCup Soccer (Humanoid, Middle-Size, Simulation, Small-Size, Standard Platform Liga(SPL), Mixed-Reality)

Die humanoiden Ligen benutzen alle (wie im Namen schon angedeutet) menschenähnliche Roboter. Dabei kommt es auf die jeweilige Liga an, wie streng die Vorschriften bezüglich der Hardware sind. In der SPL gibt es ein Reglement, dass jegliche Modifikation der Hardware verbietet. Die Hardware ist vorgefertigt, also sind die Roboter nicht von den Teams selbst gebaut. Die Schwierigkeit der Humanoiden ist hauptsächlich die Fortbewegung. Das Laufen auf zwei Beinen ist nach wie vor schwierig. Aber auch das Aufstehen nach einem Sturz ist nicht trivial. Generell ist das Ausführen der Aktionen hier vordergründig, nicht so sehr das Fußballspielen an sich. Taktiken spielen lediglich eine Nebenrolle.



Abbildung 2: Middle-Size Roboter auf ihrem Spielfeld

Die Middle-Sized Liga, die sich auch mit Fußball spielenden Robotern beschäftigt, hat andere Schwerpunkte. Da hier keine menschenähnlichen Roboter verwendet werden, gibt es das Problem der Fortbewegung auf zwei Beinen nicht. Die Roboter sehen ein wenig aus wie eine Boje, die aus dem Wasser schaut. Die Roboter haben einen niedrigen Schwerpunkt und bewegen sich auf Rädern. Somit sind sie deutlich agiler als z. B. unsere Naos [5]. Deshalb geht es in dieser Liga in erster Linie um Teamplay und nicht darum, z. B. die Agilität zu erhöhen.

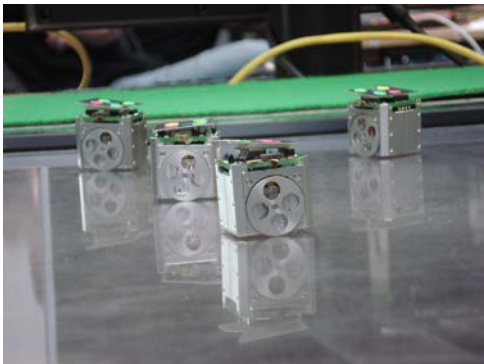


Abbildung 3: Mixed-Reality
Roboter auf
einem Bildschirm

Die omnidirektionale Kamera, die es dem Roboter jederzeit ermöglicht, Objekte zu erkennen, ohne ihnen mit der Vorderseite zugewandt sein zu müssen ist eine weitere Erleichterung. Die Roboter sollen, wie im richtigen Fußball auch, im Team zusammenspielen.

Typische Situationen sind z. B. das Freistellen bei einem Einwurf, um anspielbar zu sein, das Blocken eines ankommenden Gegners durch Rückwärtsfahren oder das Ausspielen des Gegners durch Passspiel.

Eine andere Herangehensweise an das Thema Roboterfußball verfolgt die Mixed-Reality Liga. Hier sind

lediglich die Roboter real, Spielfeld und Ball werden auf einem Monitor dargestellt, auf dem sich die Roboter bewegen. Über Deckenkameras wird das Spielgeschehen beobachtet und mit den gewonnenen Daten gesteuert.

1.2. Rescue League (Robot, Simulation)

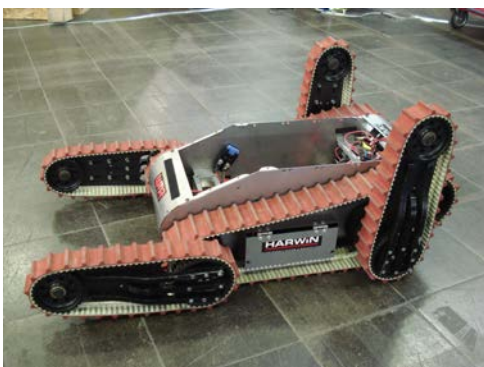


Abbildung 4: Rescue-Roboter

Diese Liga beschäftigt sich mit dem Aufspüren von eventuell verletzten Personen, dem Zurechtfinden in fremden Umgebungen und deren Kartographierung etc. Dafür gab es beim RoboCup einen Hindernisparcours, der durch farbige Bänder an der Bandenoberseite deutlich macht, welcher Klasse die Roboter angehören. Es gibt dabei Kriterien wie *autonom* und *halb autonom* sowie verschiedene Schwierigkeitsgrade bei der Fortbewegung z. B. holprige Abschnitte, Treppen, steile Schrägen und weitere.

1.3. RoboCup@Home

Aufgabe von Teilnehmern dieser Liga ist die Entwicklung von Haushaltshilferobotern. Dabei müssen die Roboter sich selbstständig in der häuslichen Umgebung zurechtfinden. Außerdem sollen sie Personen kennenlernen können, sich also deren Gesichter einprägen und sie somit wiedererkennen, wenn sie sie in der Wohnung treffen. Per Stimmeneingabe empfangen die Roboter Befehle, die sie ausführen sollen. Dabei geht es um typische Alltagsaufgaben wie Ketchup oder ein Getränk aus dem Kühlschrank holen und zu einer bestimmten Person bringen.

1.4. RoboCup Junior (Soccer, Dance, Rescue)

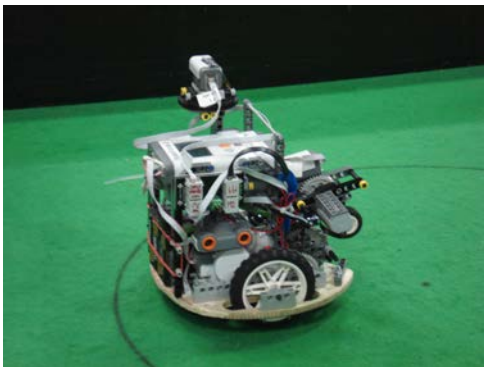


Abbildung 5: Junior-League-Roboter

Die Junior Liga ist für Schüler. Es geht hier mehr um das spielerische Ausprobieren und Arbeiten mit den Robotern, als um die Weiterentwicklung des Fachgebietes. Entsprechend locker ist das Regelwerk. Die Roboter sind meist eine Kombination aus Lego mit verbauten Motoren. Dabei reicht die Bandbreite der Roboter von Fußballrobotern (ähnlich der Middle-Sized Liga) bis zu Themenrobotern, wie z. B. Sponge Bob, Tom und Jerry und andere, die auf vorgefertigten Parcours umherfahren. Der Robotik-Anteil reduziert sich hier auf ein einfaches fest implementiertes Verhalten, wie z. B. dem Folgen einer schwarzen Linie. In erster Linie soll der Nachwuchs Interesse an der Robotik entwickeln und sich spielerisch der Thematik widmen.

2. Standard Platform League im Projekt

Das RoboCup Projekt zielt auf die Teilnahme an der Standard Platform League ab. Wie bereits oben erwähnt, treten in diesem Wettkampf humanoide Roboter zum Fußballspielen an. Die Roboter dieser Liga werden von der französischen Firma Aldebaran Robotics hergestellt. Die Schwierigkeit liegt in der Programmierung der Roboter, da die Hardware unverändert bleiben muss. Ziel ist es, autonom Fußball zu spielen. Sobald ein Spiel gestartet ist, dürfen von außen keinerlei Eingriffe mehr erfolgen. Die Roboter müssen also von sich aus z. B. Ball und Tor erkennen können, um dann auf das Tor schießen zu können. Dazu sind Eigenschaften, wie eine gute Selbstlokalisierung und ein Weltmodell unabdingbar. Die Arbeiten im Projekt orientieren sich an den Richtlinien der Standard Platform League des RoboCups. Daraus ergeben sich unter anderem die folgenden Teilgebiete

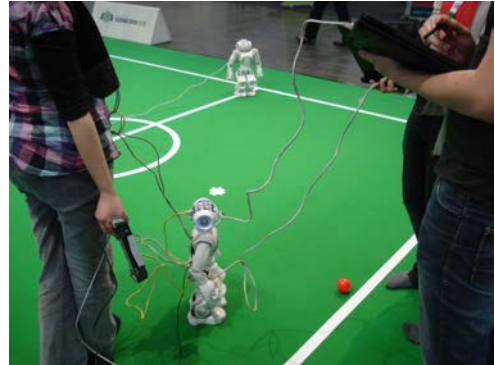


Abbildung 6: Nao-Roboter auf dem Testfeld

- Objekterkennung
- Kollisionserkennung
- Bewegungsabläufe
- Verhalten
- Hardware des Roboters ansteuern

2.1. Objekterkennung

Die Objekterkennung im Kontext des RoboCups beinhaltet das Erkennen von Ball, Tor und Spielfeld. Dabei gibt es verschiedene Anhaltspunkte, um dies zuverlässig zu erreichen. Die Tore haben unterschiedliche Farben, aber gleiche Formen. Zur Unterscheidung der Tore ist also hauptsächlich die Farbe von Bedeutung. Das Spielfeld hat charakteristische Linien, anhand derer man sich orientieren kann. Der Ball ist, auch wenn das mittlerweile

nicht mehr erstrebenswert ist, rot und kann so relativ gut auf dem Spielfeld gesehen werden. Es gab dieses Jahr aber unterschiedliche Rottöne, sodass die Farbe allein nicht mehr ausreichend war, um den Ball zuverlässig zu finden.

2.2. Kollisionserkennung

Durch die vier Ultraschallsensoren in der Brust ist der Nao in der Lage, dicht vor ihm befindliche Gegenstände, z. B. andere Roboter oder einen Torpfosten, zu erkennen und ihnen gegebenenfalls auszuweichen. Dies ist besonders hilfreich, um Strafen wegen *pushing*[7, Seite 20] zu vermeiden.

2.3. Bewegungsabläufe

Jeder Vorgang hat einen eigenen Bewegungsablauf. Damit alles „wie aus einem Guss“ wirkt und auch von der Geschwindigkeit her ausreichend ist, müssen die die Bewegungsabläufe intensiv analysiert werden. So gab es dieses Jahr große Unterschiede beim Schießen des Balles. Viele Teams ließen den Roboter dafür das Gewicht auf einen der beiden Füße verlagern, er kippte also leicht zur Seite. Dann holte er mit dem anderen Fuß aus, um zu schießen. Insgesamt dauerte der Vorgang zu lange, sodass im Spiel mehrfach ein Schienbein des Gegners getroffen wurde, weil der Ball sich nicht mehr vor dem Fuß des Roboters befand, als der eigentliche Schuss ausgeführt wurde. Andere Teams hingegen ließen den Roboter lediglich den Fuß schnell nach vorne bewegen, was sich in der Umsetzung wesentlich komplizierter gestaltet, als zu vermuten wäre. Denn auch hier muss natürlich die Balance gehalten werden, etc. Es war im Vergleich zum klassischen Schuss allerdings sehr effektiv, weil es deutlich schneller ging.

2.4. Verhalten

Dieser Teilbereich ist noch nicht sehr ausgeprägt, wenn man das Teamplay mit einbezieht. Ein Spiel besteht aus mehreren Phasen, und in jeder Phase muss sich der Roboter anders verhalten. Ein grundsätzliches Verhalten der Roboter besteht also aus mindestens einem

Teilverhalten für jede Phase. Wie dieses Verhalten aussieht, ist allerdings von Team zu Team verschieden. Vom simplen Finden und ins Tor schießen des Balles bis zum Ausspielen des Gegners per Seitenschuss oder gar Passspiel ist alles vorhanden. Das Zusammenspiel der Roboter im Team ist noch in der Findungsphase. Größere Ziele zu verfolgen, ist jedoch schwierig, wenn die Roboter noch immer mit der Bewegung kämpfen.

2.5. Hardware des Roboters ansteuern

Sinnvoll ist z. B., die vorhandenen LEDs zu nutzen, um dem Team während eines Spiels Rückmeldung zu geben über das, was der Roboter gerade tut. In dieser Phase ist das die einzige Möglichkeit, etwas von den Robotern zu erfahren, da sie im Spiel autonom agieren müssen und daher keinerlei Verbindung zu ihnen hergestellt werden darf. So könnte man beispielsweise eine Farbkodierung der Augen-LED benutzen, um das Ball-Tracking (das Verfolgen des Balles allein mit Blicken) anzuzeigen. Eine mögliche Kombination wäre *Auge rot* für Ball nicht gefunden, *Auge gelb* für Ball suchen sowie *Auge grün* für Ball gefunden.

Unser Team hat sich im Projekt RoboCup der Objekterkennung – und hier speziell der Ballerkennung – gewidmet.

3. Aufgabenstellung

Die konkrete Aufgabenstellung lag in der Entwicklung einer effizienten Ballerkennung unter Berücksichtigung des Einsatzkontextes und der zugrunde liegenden Hardware. Der Roboter nimmt mit der Kamera fortlaufend Bilder auf, diese werden vorsegmentiert und auf Vorhandensein eines Balls geprüft. Schlußendlich soll die Ballerkennung anhand einer Abweichungsfunktion berechnen, ob es sich um einen Ball handelt.

Folgende Eigenschaften galt es zu berücksichtigen

- Eigenschaften der Roboter
 - x86 AMD GEODE 500 MHz CPU
 - 256 MB SDRAM
 - 2 GB Flash Memory
 - OS (Embedded Linux)
 - Kameraauflösung 640 x 480 px
 - Kameraframerate 30 fps
 - Neigungswinkel des Kopfes -39° - $+30^\circ$
- Eigenschaften der Spielregeln
 - Datenübertragung im Wireless-Lan
 - Keine Deckenkamera
 - Keine Markierungselemente am Feldrand
 - Anzahl der Roboter im Spiel

Als Basis für unsere Ballerkennung dienen so genannte Regions of Interest (ROI). Hierbei handelt es sich um nicht vorsegmentierte Bilder der Größe 12 x 12 Pixel, welche von der bereits existierenden Bildverarbeitungssoftware in einem Datenformat des Frameworks VIGRA [6] bereitgestellt werden. Diese sogenannten ROIs sollen im weiteren Verlauf darauf untersucht werden, ob sie einen Ball enthalten. Die Einbettung der Ballerkennung in den Kontext Fußball spielender Roboter stellt Anforderungen an Echtzeitfähigkeit der

Ballerkennung mit weichen Echtzeitanforderungen. Dies bedeutet in unserem Fall, dass ein Fehler in der Ballerkennung zwar zu einem Gegentor führen kann, aber keinen Schaden an der Hardware zur Folge hat. Das bedeutet, dass die Bilder innerhalb von fest vorgegebenen Zeitgrenzen segmentiert und durch die Ballerkennung geprüft werden, um eine Wahrscheinlichkeit für das Enthaltensein des Balls zurückzugeben. Die Zeitbedingungen werden zum einen durch die maximale Bildrate der Kamera beschränkt. Allerdings lässt die derzeitige Fortbewegungsgeschwindigkeit der Roboter und die daraus resultierende geringe Spielgeschwindigkeit zu, dass eine Verarbeitung von 30 Bildern pro Sekunde für eine effiziente Objekterkennung und Ballverfolgung ausreichend sind. Da bei zeitweiligem Überschreiten der Zeitanforderungen kein Schaden entsteht und die Ergebnisse für den weiteren Spielverlauf verwendet werden können, sind keine harten Echtzeitanforderungen gegeben.

4. Vorüberlegungen

4.1. Schwierigkeiten der Objekterkennung

Die großen Herausforderungen der Objekterkennung sind der Umgang mit wechselnden Lichtverhältnissen, die Verdeckung des gesuchten Gegenstands sowie hardwareseitig die Leistungsfähigkeit des Rechners aufgrund der großen Datenmengen von Bildern und die Gewährleistung von Echtzeitfähigkeit.

Wechselnde Lichtverhältnisse sorgen beispielsweise dafür, dass Objekte je nach Lichteinstrahlung heller oder dunkler erscheinen. Bei der Untersuchung eines Bildes weist ein und dasselbe Objekt somit unterschiedliche RGB-Farbwerte auf, je nachdem wo es aufgenommen wurde.

Das wohl größte Problem, für das bisher keine Lösungsansätze gefunden werden konnten, ist jedoch das Bildverstehen.

Unsere Roboter besitzt kein semantisches Modell von einem Ball. Er hat keine Erfahrungswerte bezüglich der Eigenschaften eines Balles und weiß somit nicht, wie dieser in bestimmten Situationen aussehen kann. Somit können wir nicht erwarten, dass der

Roboter das Erkennen eines Balls verifiziert, sondern lediglich, dass er erkennt, ob das Ball-Prädikat (die für seine Entscheidung relevanten Eigenschaften des Balls) erfüllt ist. Die Physiologie des menschlichen Sehens ermöglicht es dem Menschen, das Gesehene auf verschiedene Weise zu interpretieren. So vervollständigt unser Gehirn beispielsweise Kanten von zwei sich überlagernden Objekten, selbst wenn diese auf einem Bild wegen einer Farbähnlichkeit tatsächlich gar nicht vorhanden sind. Ein Roboter hingegen sieht nur das, was wirklich da ist. Kantenvervollständigung ist bei Robotern zwar möglich, jedoch in der Standardausführung unserer Roboter nicht vorhanden.

4.2. Annahme

Zunächst erfassten wir die Eigenschaften des zu erkennenden Objektes, in unserem Falle die Eigenschaften des Balles. Für die Ballerkennung im Rahmen des RoboCups können folgende Eigenschaften vorausgesetzt werden

Die Ballfarbe wird vom Veranstalter vorgegeben, in diesem Jahr ein kräftiges Rot.

Der Farbverlauf wird von der Beleuchtung beeinflusst. Das Spielfeld wird durch eine Deckenbeleuchtung erhellt. Diese ist meist zentral über der Mitte des Spielfelds angebracht. Bei der Größe des Spielfeldes und der relativen Entfernung der Deckenbeleuchtung sowie aller möglichen Positionen des Balls auf dem Spielfeld kann ein konstanter Farbverlauf in der Mitte des Balls ermittelt werden. Ein konstanter Farbverlauf bedeutet, dass die Änderung des Farbwertes für alle ROIs denselben Verlauf nimmt. Eine Farbwertberechnung, die das ambiente Licht der Halle und die sehr intensive Punktlichtquelle beinhaltet ergibt folgendes Muster:

Werde der Ball vollständig auf einem 12 x 12 Pixel großem 2 dimensionalem Bild abgebildet. Der Mittelpunkt des Balls befindet sich hierbei exemplarisch im Nullpunkt eines kartesischen Koordinatensystems. Dann erkennt man stets eine abnehmende Helligkeit und einen stärker werdenden Rot-Ton entlang der abnehmenden y-Achse, also von $y = 1$ bis $y = 10$, in der Mitte des Balls, also von $x = 5$ bis $x = 6$.

Dieser Farbverlauf entsteht u.a. durch die Form des Balles und dadurch, dass die Deckenbeleuchtung über dem Spielfeld sehr viel intensiver ist als sonstige umgebende Lichtquellen der Halle. So ist eine deutliche Unterscheidung zwischen dem

Ball und anderen gleichfarbigen Objekten wie z. B. dem T-Shirt eines Kindes am Spielfeldrand.

Der Ballschatten resultiert aus der Beleuchtung. Aus der zuvor beschriebenen Beleuchtung und der Form des Balls ergibt sich eine weitere Eigenschaft, die man sich für eine Ballerkennung zu Nutze machen kann: der Ballschatten. Dieser hat während des Spiels einen immer gleichen Verlauf. Er befindet sich zentral unter dem Ball und hat stets den gleichen Helligkeitswert, sofern die Lichtquellen nicht von anderen Robotern oder den Schiedsrichtern verdeckt werden. Tatsächlich treten geringe Abweichungen des Helligkeitswertes auf, wenn sich die Position des Balles von der Spielfeldmitte zum Spielfeldrand verlagert. Allerdings werden diese bedingt durch die Eigenschaften der Roboterkamera auf dem 2D-RGB-Bild nicht dargestellt.

Die Ballgröße ist vorgegeben, deren Abbildung auf der Kamera des Roboters ändert sich jedoch in Abhängigkeit der relativen Entfernung zum Roboter. Die runde Form des Balls kann im Rahmen des Wettkampfes für eine effiziente Objekterkennung genutzt werden, muss allerdings aufwendig implementiert werden und ist zum Zeitpunkt dieser Arbeit noch nicht geplant.

5. Ausgangssystem

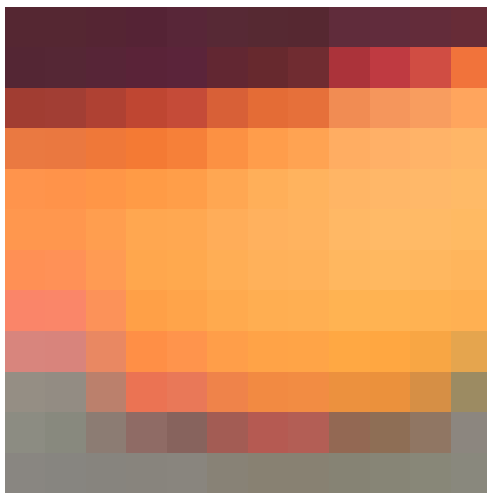
Die Ballerkennung wird in das bereits vorhandene NodeFramework [2] von Steven Köhler integriert und benutzt die Bildverarbeitungsbibliothek VIGRA [6] sowie das Multi-Sensordaten-Fusion-Framework [3] von Janis Schönefeld. Die Vorverarbeitung rechnet das Kamerabild von 640 x 480 Pixeln auf 160 x 120 Pixel herunter, um eine Echtzeitbearbeitung zu gewährleisten. Das neue Bild wird segmentiert und in Regions of Interest (ROI) zerteilt. ROIs enthalten Feldlinien-, Torelement- oder Ballkandidaten. Die Ballkandidaten werden als unsegmentiertes 12 x 12 Pixel Bild übergeben. Der Ball kann komplett, teilweise, teilweise verdeckt oder gar nicht auf dem Bild zu sehen sein.

6. Realisierung

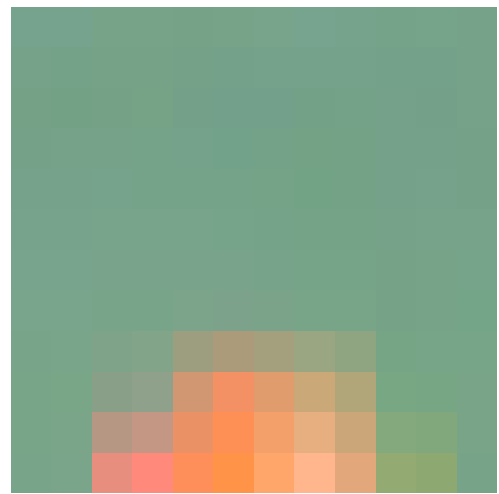
6.1. Ansatz

Die Ballverifikation wird in dieser Arbeit auf der Basis eines Histogrammvergleichs durchgeführt. Ziel dieses Vergleiches ist es, unabhängig von Umgebungslicht und anderen äußeren Einflüssen, den Ball möglichst eindeutig zu identifizieren. Zunächst benötigen wir eine Basis von ROIs. Diese Basis enthält sowohl „gute“ als auch „schlechte“ Bilder. Ein „gutes“ Bild enthält dabei eine von der Vorverarbeitung ausgewählte ROI, in der ein Ball enthalten und auch als solcher leicht zu erkennen ist. Die Bewertung der ROIs erfolgt dabei durch den Menschen mit Hilfe einer grafischen Oberfläche in zwei Kategorien. Durch mehrfache positive oder negative Auswahl können Bilder mit starken positiven oder negativen Eigenschaften höher bewertet werden als Bilder mit schwachen Eigenschaften.

Aus der Basis der „guten“ und „schlechten“ ROIs wird ein durchschnittliches Histogramm in den von uns ausgewählten Farbräumen erstellt, an dem später die Ball-Kandidaten gemessen werden.



(a) „Gute“ ROI



(b) „Schlechte“ ROI

Abbildung 7: Beispiel-ROIs die von der Vorverarbeitung ausgewählt wurden

6.2. Histogramm

Die Histogramme werden auf Basis der 12x12 Pixel ROIs erstellt. Jedes Bild hat somit 144 Pixel, von denen wir lediglich 20 Pixel bewerten, die jeweils drei Werte (Rot, Grün und Blau) enthalten. Jeder Punkt im Histogramm gibt an, wie oft der Farbwert (z. B. Grün) in einer Intensität vorkommt. Die Häufigkeit wird dabei auf der y-Achse aufgetragen, die Intensität auf der x-Achse.

Die durchschnittlichen Histogramme werden durch Aufsummieren der Intensitätshäufigkeiten der Einzelbildhistogramme erstellt und durch die Anzahl der Testbilder dividiert. Ein Wert von 0.1 bei 255 bedeutet also, dass in jeder zehnten Ballkandidaten ROI ein Pixel mit der roten Intensität von 255 vorkommt bzw. in 99 keiner und in einem 10.

6.3. Farbräume

6.3.1. RGB

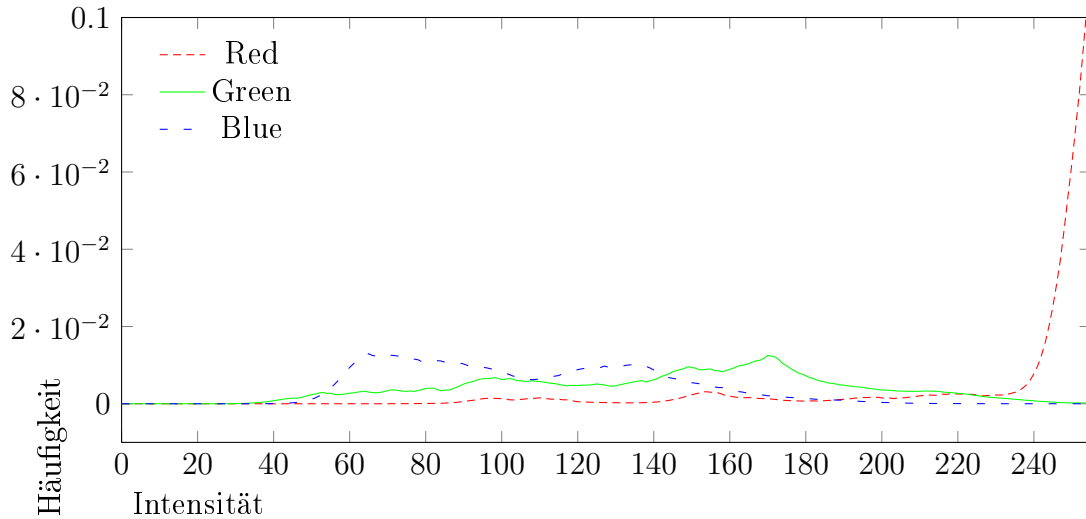


Abbildung 8: Durchschnittshistogramm der positiven Ball-ROIs im RGB-Farbraum

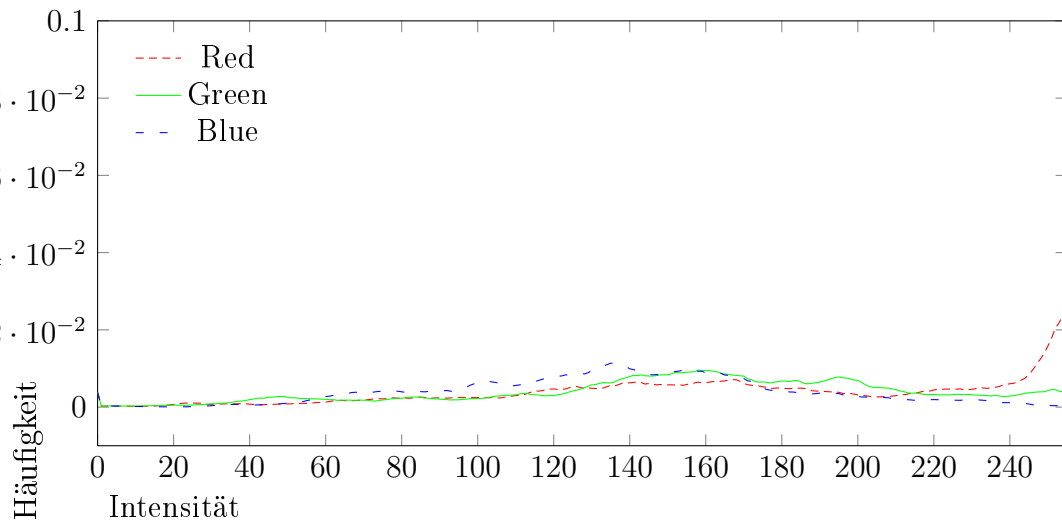


Abbildung 9: Durchschnittshistogramm der negativen Ball-ROIs im RGB-Farbraum

In dieser Arbeit verwenden wir sowohl den RGB Farbraum als auch den HSV-Farbraum. Der RGB-Farbraum bietet sich nicht nur aufgrund der bereits im RGB-Format vorliegenden ROIs an. Der von dem SPL-Reglement vorgeschriebene Ball ist vorwiegend rot,

während das Spielfeld grün und Roboter und Feldlinien weiß sind. Aufgrund dieser Farben erhalten wir im RGB Vergleichshistogramm eine sehr spezifische Kennlinie für den Rotanteil, welcher im Wertebereich von 230 bis 256 eine starke Häufigkeit aufweist, während die Werte für Blau und Grün eine deutlich flachere Kurve aufweisen und im mittleren Wertesegment (Blau: 55 - 150, Grün: 90 - 180) gehäuft auftreten (siehe Abbildung 8). Das durchschnittliche Histogramm der positiv bewerteten ROIs unterscheidet sich vor allem in der Ausprägung der roten Kennlinie. Auf den ersten Blick ähnelt der Verlauf der Kennlinie der negativen ROIs aus Abbildung 9 der Kennlinie der positiven ROIs aus Abbildung 8. Auf der y-Achse ist jedoch zu erkennen, dass die rote Kennlinie der positiven ROIs mehr als vier mal so stark ausgeprägt ist. Die blauen und grünen Kennlinien unterscheiden sich in beiden Histogrammen kaum, verlaufen aber bei den negativ bewerteten ROIs etwas diffuser.

6.3.2. HSV

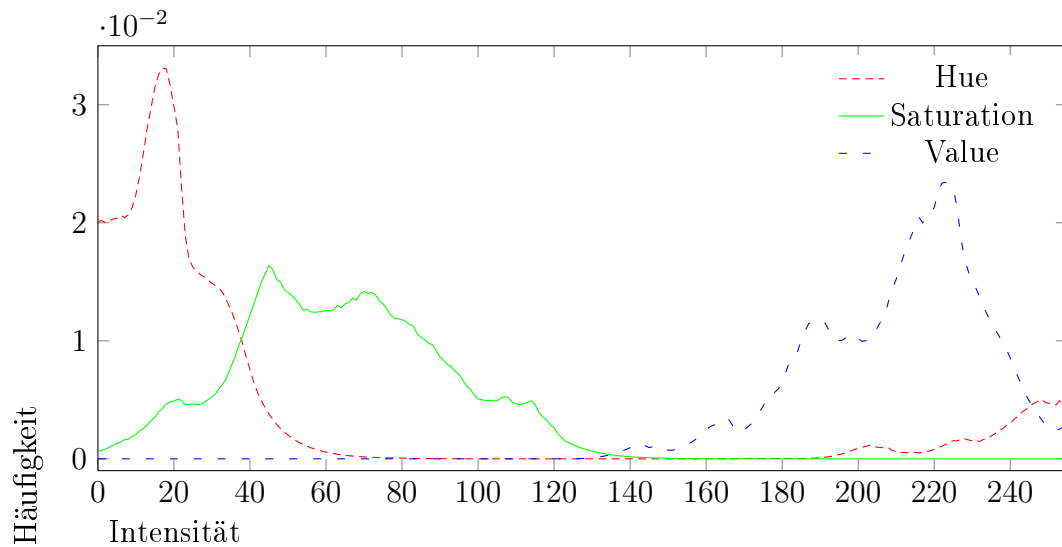


Abbildung 10: Durchschnittshistogramm der positiven Ball-ROIs im HSV-Farbraum

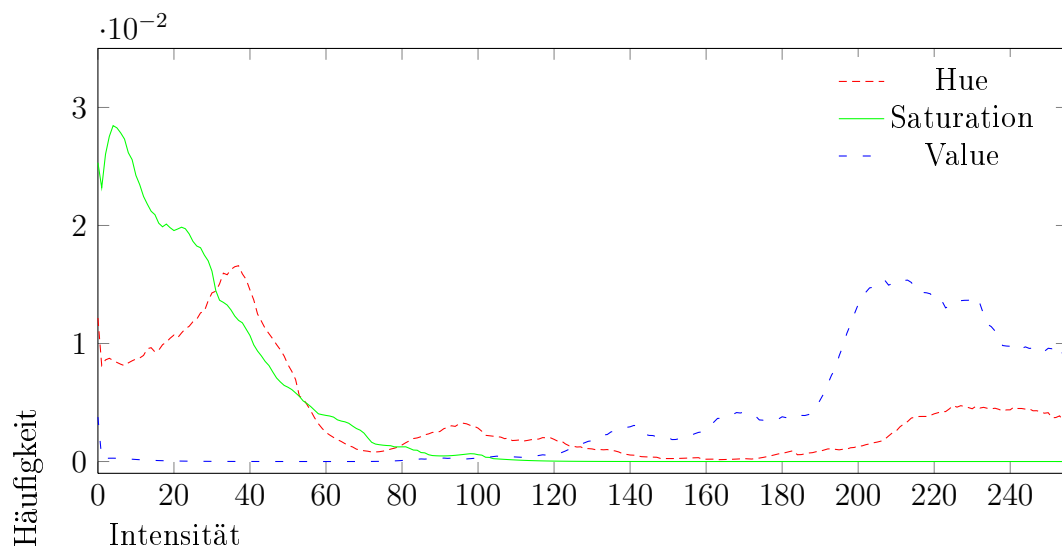


Abbildung 11: Durchschnittshistogramm der negativen Ball-ROIs im HSV-Farbraum

Der HSV-Farbraum wird, anders als der RGB-Farbraum, nicht durch die einzelnen Farbkomponenten beschrieben, sondern repräsentiert vor allem Helligkeit (**V**alue) und Farbsättigung (**S**aturation). Zusätzlich gibt der Farbwinkel (**H**ue) den Farbton an, welcher aus

einem Farbkreis abgelesen wird. Die Ballfarbe liegt im roten Bereich, zwischen 320° und 40° . Überbelichtete bzw. unterbelichtete Bereiche sollten nicht berücksichtigt werden.

Im Vergleich der beiden durchschnittlichen Histogramme im HSV-Farbraum für positive und negative Ballkandidaten sehen wir eine deutliche Differenzierung zwischen den Kennlinien beider Histogramme für Farbton (Hue) und Sättigung (Saturation), während die Werte für Helligkeit (Value) einen ähnlichen Verlauf zeigen. Der Schnitt an der $360^\circ/0^\circ$ -Marke wird durch die Glättung der Kurve verursacht und entsteht aufgrund der Skalierung auf den RGB Farbraum auf 255 Werte an der x-Achse bei 255. Das Histogramm der positiven Beispiele weist eine starke Häufung im roten (im Histogramm 250 bis 15) und orangen (im Histogramm 15 bis 25) Farbton auf. Im negativen Histogramm ist der Spitzenwert erst bei 40 (gelb-grüner Farbton) erreicht und die Kurve verläuft deutlich flacher. Des Weiteren sind im negativen Histogramm deutlich mehr ungesättigte Farben vorhanden (0 bis 40) als im Histogramm mit den positiven Beispielen (40 bis 80).

7. Funktionsweise und Implementierung der Ballverifikation

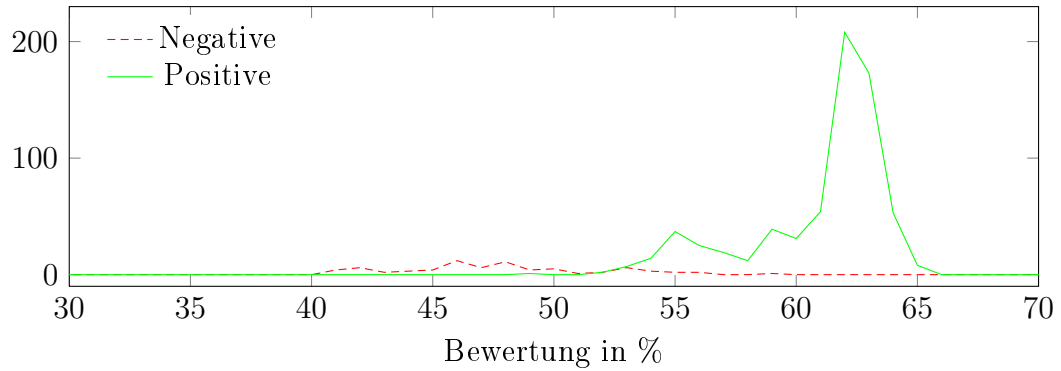


Abbildung 12: Visualisierung der Bewertung von zufälligen Ball-ROIs im RGB-Farbraum

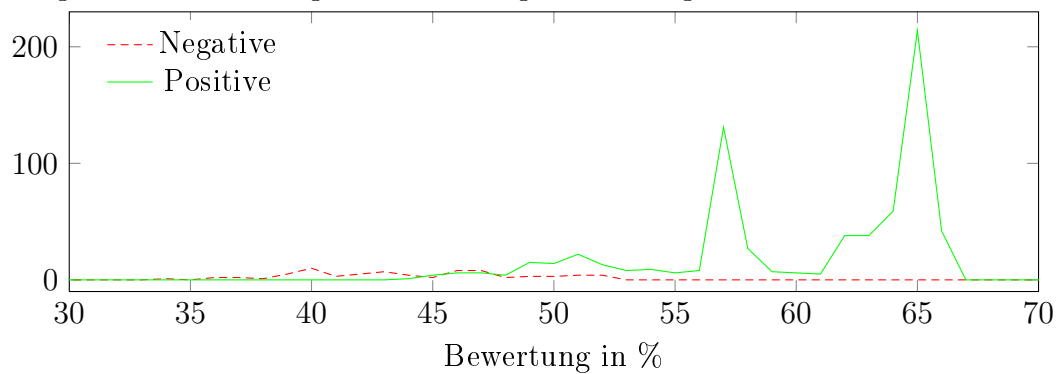


Abbildung 13: Visualisierung der Bewertung von zufälligen Ball-ROIs im HSV-Farbraum mit Helligkeit

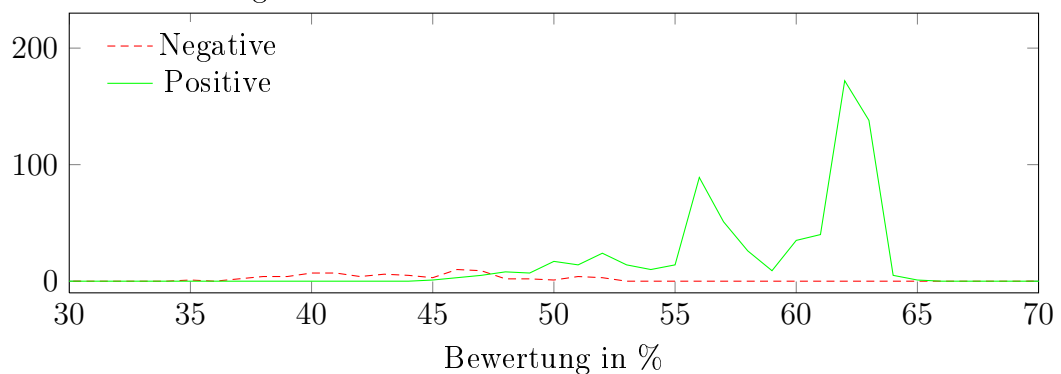


Abbildung 14: Visualisierung der Bewertung von zufälligen Ball-ROIs im HSV-Farbraum ohne Helligkeit

Wie zuvor erwähnt, basiert unsere Ballverifikation auf einer berechneten Abweichung von dem Histogramm des zu bewertenden Bildes und den Durchschnittshistogrammen „gut“ bzw. „schlecht“. Zunächst müssen dafür Histogramme vorliegen, die wir natürlich erst erstellen müssen.

Hierfür sind grundsätzlich drei Schritte nötig

1. Normalisierung auf 250
2. Gauß-Glättung
3. Normalisierung auf die Fläche

Das folgende Vorgehen ist für jede neue Testumgebung durchzuführen.

Wir beschreiben das Vorgehen für die Durchschnittshistogramme, die mit hoher Wahrscheinlichkeit einen Ball darstellen. Dazu benötigen wir Bilder, auf denen der Ball entsprechend gut zu erkennen ist. Die Bilder liegen bereits in einer Größe von 12 x 12 Pixeln (ausgeschnitten aus dem Kamerabild) vor. Dies verringert den späteren Rechenaufwand erheblich. Die nun beschriebenen Verfahren müssen je für alle Farbkanäle gemacht werden. Außerdem einmal im RGB-Farbraum, sowie im HSV-Farbraum. Also im RGB-Farbraum für Rot, Grün und Blau, im HSV-Farbraum für Hue, Saturation und Value. Wir beschränken uns hier auf die Beschreibung im RGB Farbraum (konkret auf den roten Farbkanal), da das Vorgehen für die restlichen Kanäle, sowie für den HSV-Farbraum identisch ist.

Ein Histogramm sei im Folgenden ein Bild mit den Maßen 256 x 1 Pixel.

Von diesen 144 Pixeln betrachten wir im Folgenden nur $5 \leq x \leq 6$ und $1 \leq y \leq 10$, also 20 Pixel pro Bild. Diese 20 Pixel bilden die Basis für die Erstellung der Histogramme. Auf der x-Achse werden die einzelnen Farbwerte repräsentiert, die y-Achse die tatsächliche Häufigkeit des Farbwertes in allen Bildern. Für jedes dieser 20 Pixel wird der Farbwert ermittelt und der entsprechende Wert im Histogramm inkrementiert um Eins. Als Beispiel: Hat jedes unserer 224 Testbilder in den 20 betrachteten Pixeln denselben Farbwert (hier zur Veranschaulichung 144), hätte das Histogramm bei $x = 144$ den Wert $y = 4480$. Dieses Beispiel verdeutlicht die Notwendigkeit von Schritt 1 (dem Normalisieren auf 250).

Zu 1. Wir brauchen einen Normalisierungsfaktor, der sich ergibt aus 250 dividiert durch den maximalen Farbwert im Histogramm (im vorherigen Beispiel 4480). 250 ist gewählt, da die Werte in einem RGB-Bild gespeichert werden, welches maximal Werte bis 256 halten kann. Jetzt füllen wir ein neues 256 x 1 Pixel großes Bild pixelweise mit Farbwert * Normalisierungsfaktor. Nun liegt ein auf 250 normalisiertes Histogramm vor.

Zu 2. Auf das in Schritt 1 erstellte Histogramm wird nun eine Gauß-Glättung angewandt.[4, Seite 79]

Zu 3. Weil nach Ausführung aller Berechnungen eine Aussage getroffen werden soll, mit welcher Wahrscheinlichkeit es sich bei den untersuchten Bildmotiven um einen Ball handelt, müssen wir unser Histogramm noch auf die Fläche normalisieren. Gemeint ist, dass die Fläche von $x = 0$ bis $x = 256$ und $y = 0$ bis $y = 250$ den Wert 1 (also 100%) ergeben soll. Dazu müssen wir nun ähnlich wie in Schritt 1 vorgehen. Da wir nun Werte zwischen 0 und 1 erhalten wollen, berechnen wir zunächst den Normalisierungsfaktor. Dieser ergibt sich aus 1 dividiert durch den maximalen Farbwert in dem aus 2. gewonnenen Histogramm. Entsprechend 1. wird nun der Farbwert jedes Pixels mit dem so ermittelten Normalisierungsfaktor multipliziert.

Diese Drei Schritte führen wir für jedes zu bewertende Bild und die Testbilder durch, sodass wir letztendlich folgende Histogramme erhalten:

- Histogramm vom zu bewertenden Bild
- Histogramm von den Testbildern (RGB gut)
- Histogramm von den Testbildern (RGB schlecht)
- Histogramm von den Testbildern (HSV gut)
- Histogramm von den Testbildern (HSV schlecht)

Diese Histogramme sind nun Grundlage für die eigentliche Bewertung. Auch dieser Vorgang erfolgt in Drei Schritten.

1. Berechnung der Abweichung (Histogramm bewertendes Bild zu Histogramm der Testbilder)
2. Umwandlung der Abweichung in Prozentangaben
3. Bewertung der in 2. errechneten Angaben / „Ballentscheidung“

Zu 1. Wir berechnen pixelweise die Differenz (Testbildhistogramm (gut / schlecht) - Histogramm zu bewertendes Bild). Dies Rechnung für jeden einzelnen Farbkanal in beiden Farbräumen.

Zu 2. Aus den in 1. berechneten Abweichungen müssen nun Prozentwerte generiert werden. Die Absolutwerte der Differenzen in 1. (gilt wieder für beide Farbräume) werden getrennt nach gut und schlecht aufsummiert und dann durch 6 (3 Kanäle für gut und 3 Kanäle für schlecht) dividiert. Anschließend berechnen wir die Differenz $\frac{\text{schlecht}-\text{gut}+1}{2}$ und wandeln diese durch Multiplikation mit 100 in einen Prozentwert um.

Zu 3. Wir teilen die Prozentskala (0 - 100) in drei Bereiche. Dazu wählen wir eine untere Schranke u und eine obere Schranke o . Alle Werte $< u$ werden nicht als Ball anerkannt, Werte zwischen u und o gelten als unsicher, Werte $> o$ sind als Ball anerkannt. In praktischen Tests erwiesen sich die Werte $u = 40\%$ und $o = 60\%$ als sinnvoll, in anderen Testumgebungen könnten andere Werte sinnvoller sein.

	$x < 47$	$47 \leq x \leq 49$	$49 < x$
Anzahl schlechter ROI im Wertebereich	83	19	6
Anzahl guter ROI im Wertebereich	1	15	459

Tabelle 1: Konfusionsmatrix im HSV-Farbraum ohne Helligkeitsbewertung für eine dreistufige Bewertung

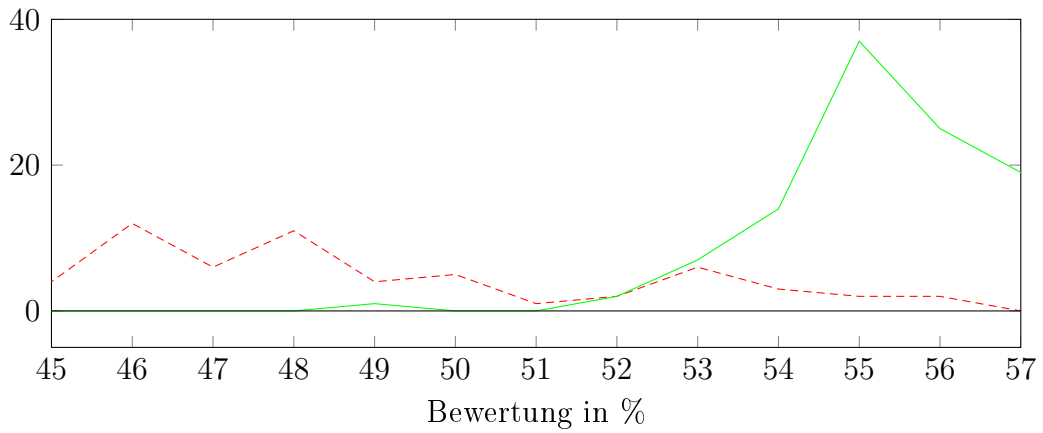


Abbildung 15: Ausschnitt Abbildung 12

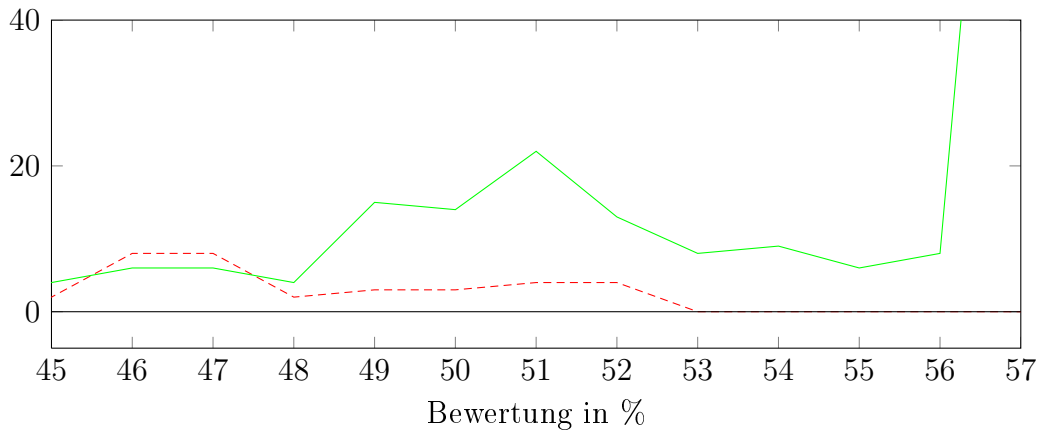


Abbildung 16: Ausschnitt Abbildung 13

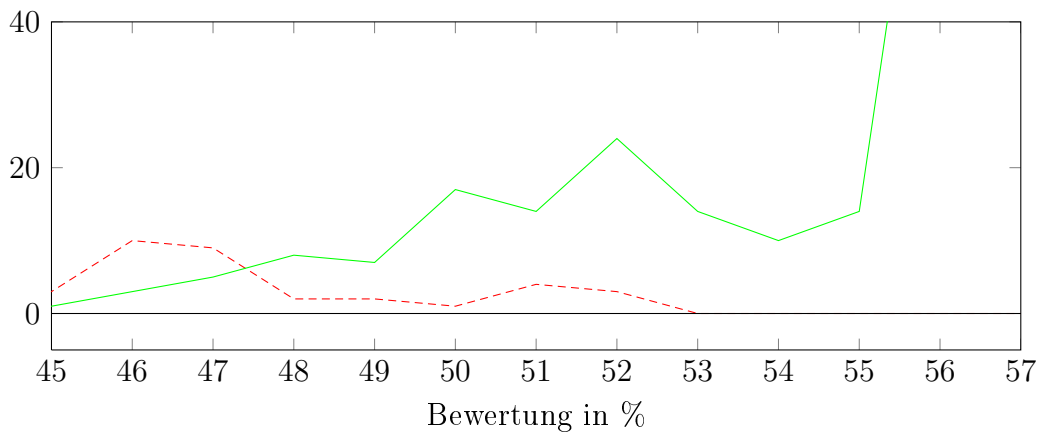


Abbildung 17: Ausschnitt Abbildung 14

8. Bewertung unseres Verfahrens

Die Vorteile des HSV-Farbraum Verfahrens ohne Helligkeitsberücksichtigung würde sich mit entsprechenden Testbildern noch klarer hervorheben lassen.

Anhand der Bewertungsdaten ist zu erkennen, dass sich von den drei ausgewählten Bewertungsverfahren der HSV-Farbraum ohne Helligkeitsbewertung am besten für die eindeutige Identifizierung von Ballkandidaten eignet. Der Anteil an falsch identifizierten ROIs ist minimal, gleichzeitig ist der Schnittpunkt der Kennlinien klar erkennbar. Die Verbesserung zwischen den beiden HSV-Farbraum-Verfahren ist relativ gering, weil alle Testbilder unter denselben Bedingungen entstanden sind. Der Sonnenlicht-Anteil im Testraum war relativ gering und die künstliche Beleuchtung stark und gleichmäßig.

Das Bewertungsergebnis des Verfahrens lässt sich am besten mit einem dreistufigen Maßstab benutzen. Der Grenzbereich zwischen 47 % und 52 % wird als nicht eindeutig eingestuft. Der Bereich 0 % - 46 % wird als eindeutig kein Ball und 53 % - 100 % eindeutig als Ball aufgefasst. Dies bietet sich vor allem an, wenn die Ballbeweisprozedur nicht auf einem Einzelbild sondern auf einer Folge von mindestens zwei Bildern basiert. Bei ersten Live-Tests auf der German Open 2011 in Magdeburg und im Labor konnte das Ergebnis bestätigt werden.

Unsere praktischen Tests waren sehr erfolgreich. Der Roboter findet zuverlässig den Ball. In Bezug auf die Verwendung für autonome Roboter während des Spiels ist unsere Implementation aus effizienzgründen nicht brauchbar. Die erwarteten 30 fps werden momentan nicht erreicht. Die Kamera liefert zwar 30 Bilder/Sekunde jedoch können diese nicht verarbeitet werden. Eine Effizienzsteigerung ist möglich, wenn man nicht für jedes neue Bild ein Histogramm der Testbilder erstellt sondern dies nur im Fall neuer Testumgebungen und neuer Testbilder erstellt und dann nur für die Berechnung der Abweichung benutzt. Bei den praktischen durchgeführten Tests war der Roboter mit einem Rechner verbunden, auf dem die Ballverifikation berechnet wurde. Die Rechenleistung des Computers erreichte die Geschwindigkeit, sodass wir unsere Implementation erfolgreich einsetzen konnten.

9. Fazit

Das oben beschriebene Verfahren funktioniert sehr gut, kann aber in Zukunft durch eine kantenbasierte Ballerkennung ergänzt werden um die Robustheit gegenüber wechselnden Lichtverhältnissen zu erreichen.

10. Anhang

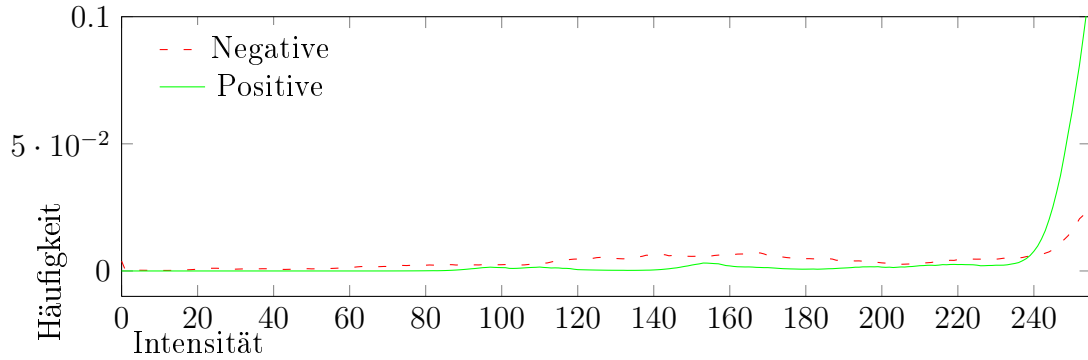


Abbildung 18: Durchschnittshistogramm der Ball-ROIs im RGB-Farbraum für den roten Kanal

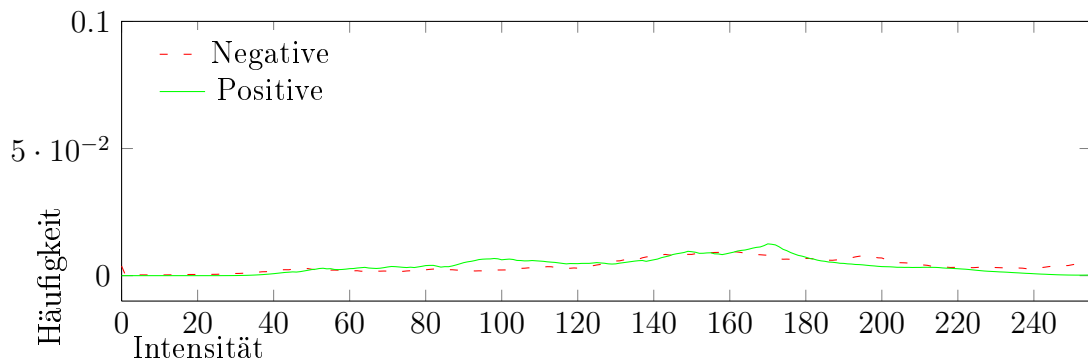


Abbildung 19: Durchschnittshistogramm der Ball-ROIs im RGB-Farbraum für den grünen Kanal

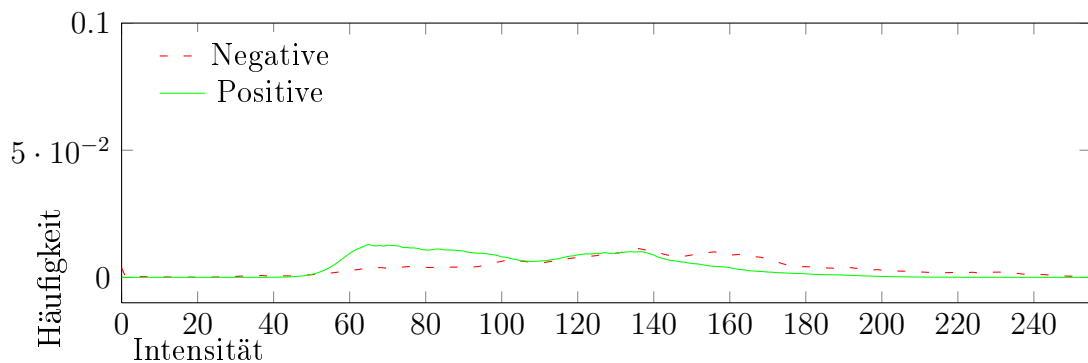


Abbildung 20: Durchschnittshistogramm der Ball-ROIs im RGB-Farbraum für den blauen Kanal

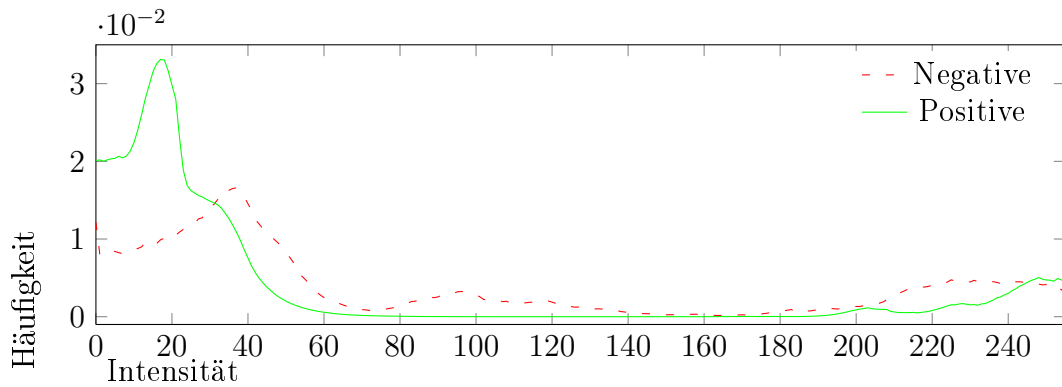


Abbildung 21: Durchschnittshistogramm der Ball-ROIs im HSV-Farbraum für den Hue-Kanal

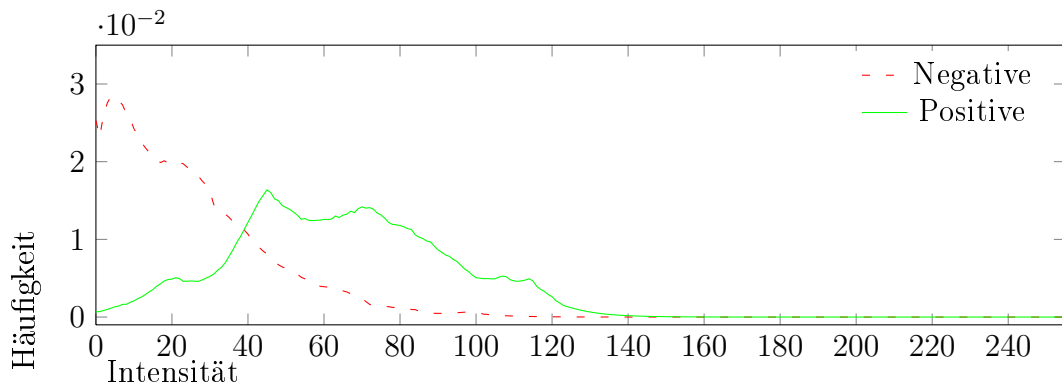


Abbildung 22: Durchschnittshistogramm der Ball-ROIs im HSV-Farbraum für den Saturation-Kanal

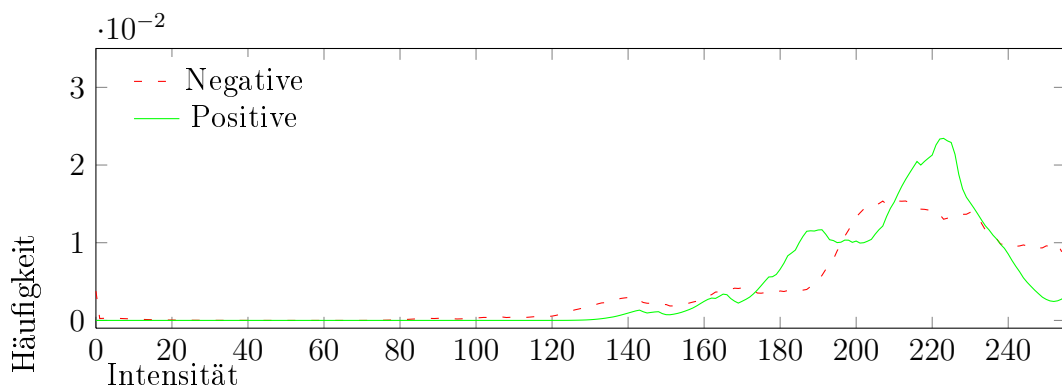


Abbildung 23: Durchschnittshistogramm der Ball-ROIs im HSV-Farbraum für den Value-Kanal

Abbildungsverzeichnis

1	SPL-Spielfeld mit Naos	4
2	Middle-Size Roboter auf ihrem Spielfeld	5
3	Mixed-Reality Roboter auf einem Bildschirm	6
4	Rescue-Roboter	6
5	Junior-League-Roboter	7
6	Nao-Roboter auf dem Testfeld	8
7	Beispiel-ROIs die von der Vorverarbeitung ausgewählt wurden	16
8	Durchschnittshistogramm der positiven Ball-ROIs im RGB-Farbraum . . .	17
9	Durchschnittshistogramm der negativen Ball-ROIs im RGB-Farbraum . . .	17
10	Durchschnittshistogramm der positiven Ball-ROIs im HSV-Farbraum . . .	19
11	Durchschnittshistogramm der negativen Ball-ROIs im HSV-Farbraum . . .	19
12	Visualisierung der Bewertung von zufälligen Ball-ROIs im RGB-Farbraum	21
13	Visualisierung der Bewertung von zufälligen Ball-ROIs im HSV-Farbraum mit Helligkeit	21
14	Visualisierung der Bewertung von zufälligen Ball-ROIs im HSV-Farbraum ohne Helligkeit	21
15	Ausschnitt Abbildung 12	25
16	Ausschnitt Abbildung 13	25
17	Ausschnitt Abbildung 14	25
18	Durchschnittshistogramm der Ball-ROIs im RGB-Farbraum für den roten Kanal	28
19	Durchschnittshistogramm der Ball-ROIs im RGB-Farbraum für den grünen Kanal	28
20	Durchschnittshistogramm der Ball-ROIs im RGB-Farbraum für den blauen Kanal	28
21	Durchschnittshistogramm der Ball-ROIs im HSV-Farbraum für den Hue- Kanal	29

22	Durchschnittshistogramm der Ball-ROIs im HSV-Farbraum für den Saturation-Kanal	29
23	Durchschnittshistogramm der Ball-ROIs im HSV-Farbraum für den Value-Kanal	29

Literatur

- [1] Dr. Ansgar Bredenfeld. Robocup GermanOpen. <http://www.robocupgermanopen.de>. [letzter Zugriff am 06.04.2011].
- [2] Steven Köhler. Das NodeFramework - Entwurf und Realisierung eines Software-Frameworks für robuste, höchstzuverlässige Anwendungen. Diplomarbeit, Universität der Freien und Hansestadt Hamburg, Fachbereich Informatik, Arbeitsbereich Technische Informatiksysteme, Juni 2011.
- [3] Janis Schönefeld. Real Time Object Recognition in the RoboCup Four Legged League. Baccalaureatsarbeit, Universität der Freien und Hansestadt Hamburg, Fachbereich Informatik, Arbeitsbereich Technische Informatiksysteme, März 2007.
- [4] Kristian Bredies und Dirk Lorenz. *Mathematische Bildverarbeitung: Einführung in Grundlagen und moderne Theorie*. Vieweg+Teubner Verlag, 2010.
- [5] Aldebaran Robotics. <http://www.aldebaran-robotics.com/en/node/1172>. [letzter Zugriff am 08.06.2011].
- [6] Generic Programming for Computer Vision. <http://hci.iwr.uni-heidelberg.de/vigra/>. [letzter Zugriff am 08.06.2011].
- [7] RoboCup Rules 2011 SPL. <http://www.tzi.de/spl/pub/Website/Downloads/Rules2011.pdf>. [letzter Zugriff am 08.06.2011].