

Erkennung von Toren beim RoboCup

Elena Noll
Sven-Hendrik Haase

24. Februar 2013

Inhaltsverzeichnis

Inhaltsverzeichnis	1
1 Einführung	2
2 Tore im RoboCup	2
2.1 Middle Size League	3
2.2 Humanoid League	4
2.3 Standard Platform League	6
3 Bilderkennung	7
3.1 Probleme	7
3.2 Erkennung mittels geometrischer Relationen	8
3.2.1 Farbkalibrierung YUV-Farbraum	8
3.2.2 Farbsegmentierung	9
3.2.3 Erkennung des Horizonts	13
3.2.4 Extraktion der Torpfosten und Modellierung	13
3.2.5 Beispiele	13
3.3 Erkennung mittels Hough-Transformation	17
3.3.1 Der HSV-Farbraum	18
3.3.2 Eckenfilter zum Erkennen von Torkonturen	19
3.3.3 Hough-Transformation zum Erkennen der Torsegmente	20
3.3.4 Aufspannen des Tormodells	22
3.3.5 Vorteile der Hough-Transformation an Hand von Bei- spielen	22
4 Fazit	24
5 Referenzen	25

1 Einführung

Diese Arbeit beschäftigt sich mit der Bilderkennung der Tore im RoboCup. Diese ist genau wie zum Beispiel die autonome Bewegung oder die Lokalisierung ein zentraler Punkt der Forschung. Wichtig ist, dass der Roboter wissen muss, nach was er eigentlich sucht. Das bedeutet, er muss wissen, welche Größe, welche Farbe und welche Form das Tor hat, damit er es erkennen kann. Außerdem ist zu sagen, dass der RoboCup in verschiedenen Ligen angeboten wird, die jeweils verschiedene Tore einsetzen. Dies wird später genauer erläutert. Es werden zudem zwei von vielen möglichen Verfahren zur Erkennung der Tore genauer vorgestellt.

2 Tore im RoboCup

Die Tore im RoboCup haben genau wie beim „normalen“ Fußball mit menschlichen Spielern die Funktion, Punkte zu werten. Allgemein bestehen alle Tore im RoboCup unabhängig von der Liga genau wie normale Fußballtore aus zwei Pfosten, einer Latte und einem Netz. Die Tore werden auf der Torlinie mittig zentriert aufgestellt, sodass die Torlinie und die Latte parallel zueinander sind.

Der RoboCup besteht aus verschiedenen Ligen, die grob ihrer Größe nach gestaffelt sind. In jeder Liga können die Tore eine unterschiedliche Größe sowie Farbe haben. Das Tor jeder Liga kann sich zudem jedes Jahr unabhängig von den anderen Ligen verändern. Der allgemeine Gedanke hierbei ist, dass die Anforderungen an die Roboter und somit die Teams jedes Jahr wachsen sollen und es jedes Jahr neue Herausforderungen geben soll. Ein Beispiel hierfür ist, dass im letzten Jahr durch in der SPL erstmals gleichfarbige Tore eingeführt wurden. Dies erschwert die Schwierigkeit der Selbstlokalisierung der Roboter erheblich.

2.1 Middle Size League

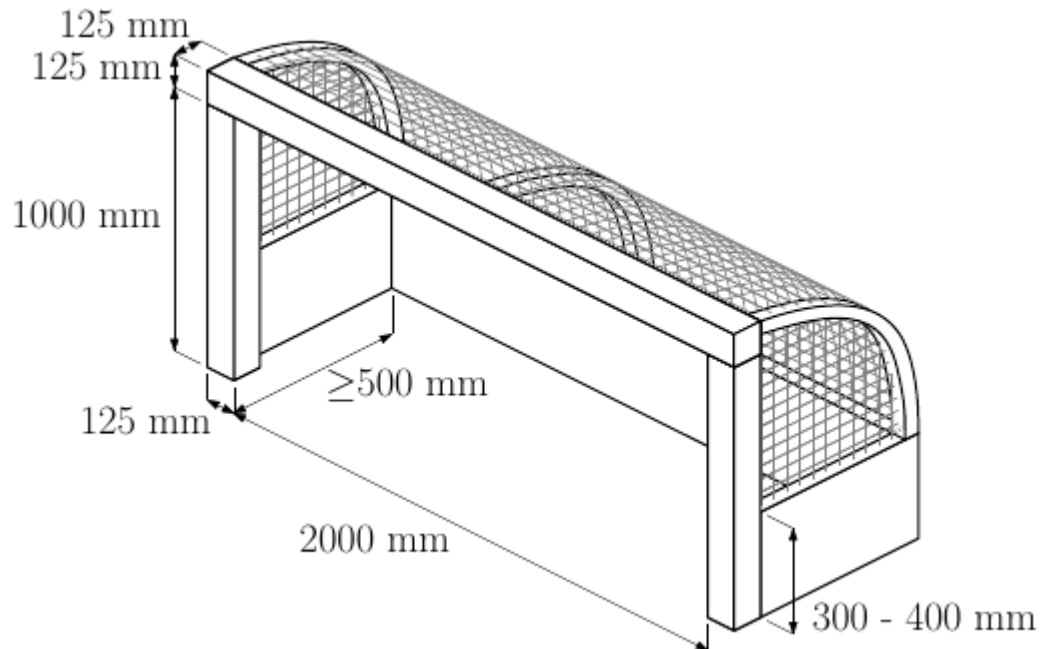


Abbildung 1: Tor in der Middle Size League

Die Grafik zeigt die Maße an. Die Farbe beider Tore bei der Middle Size League ist seit einigen Jahren weiß. Eine besondere Schwierigkeit bei der Bilderkennung stellt hierbei das ebenfalls weiße Netz dar, welches fälschlicherweise als Feldlinie erkannt werden könnte. Die Tore besitzen eine geschlossene Form. Das Netz ist solide und behält seine Form, was es von „echten“ Netzen beim Fußball unterscheidet.

2.2 Humanoid League

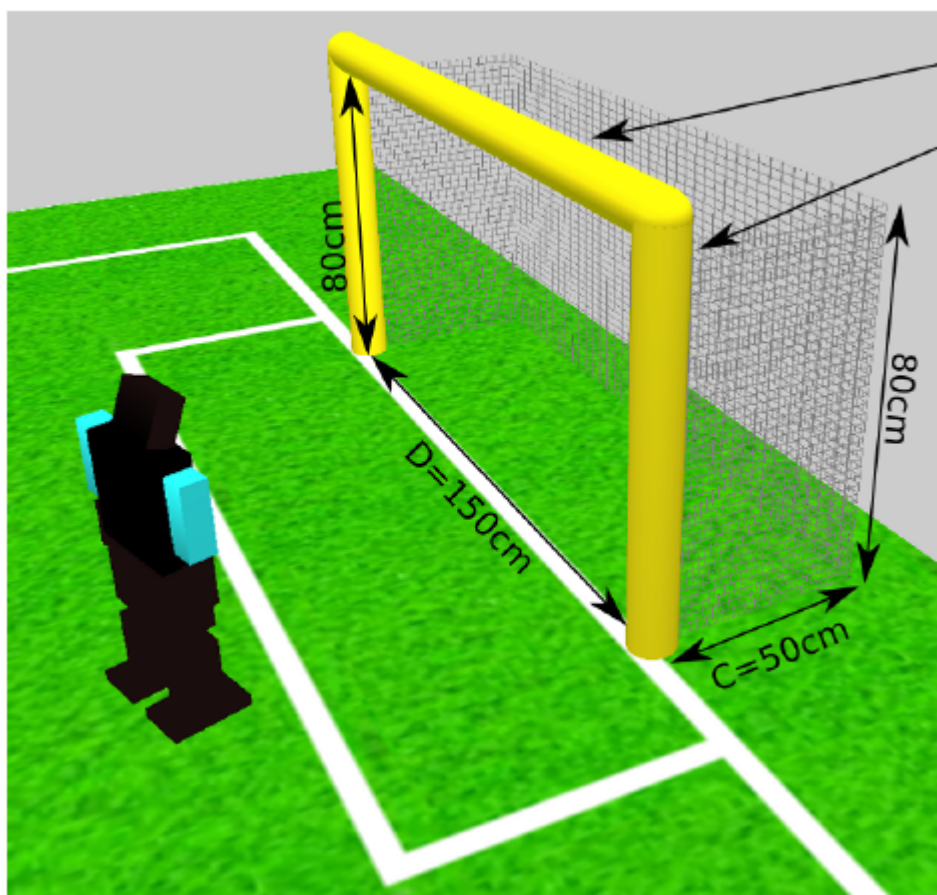


Abbildung 2: Tor in der Kid Size Humanoid League

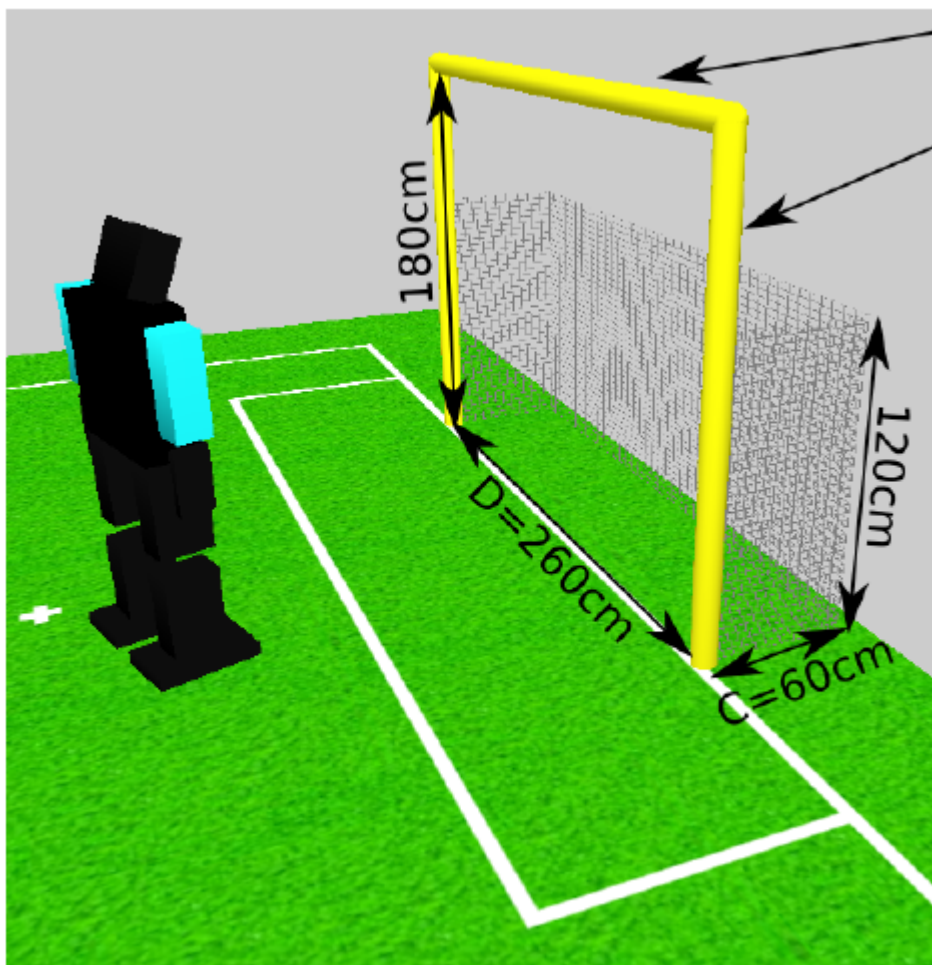


Abbildung 3: Tor in der Adult Size Humanoid League

Wie in den Grafiken dargestellt, gibt es für die beiden Größen der Humanoid League auch jeweils ein passendes Tor. Anders als bei der Middle Size League sind die Torfarben jedoch unterschiedlich: Eines ist blau, das andere ist gelb. Das Regelwerk von 2012 stellt in Aussicht, dass sich dies bald ändern könnte. Hinter dem Tor befindet ein engmaschiges, solides Netz mit dunkler Farbe.

2.3 Standard Platform League

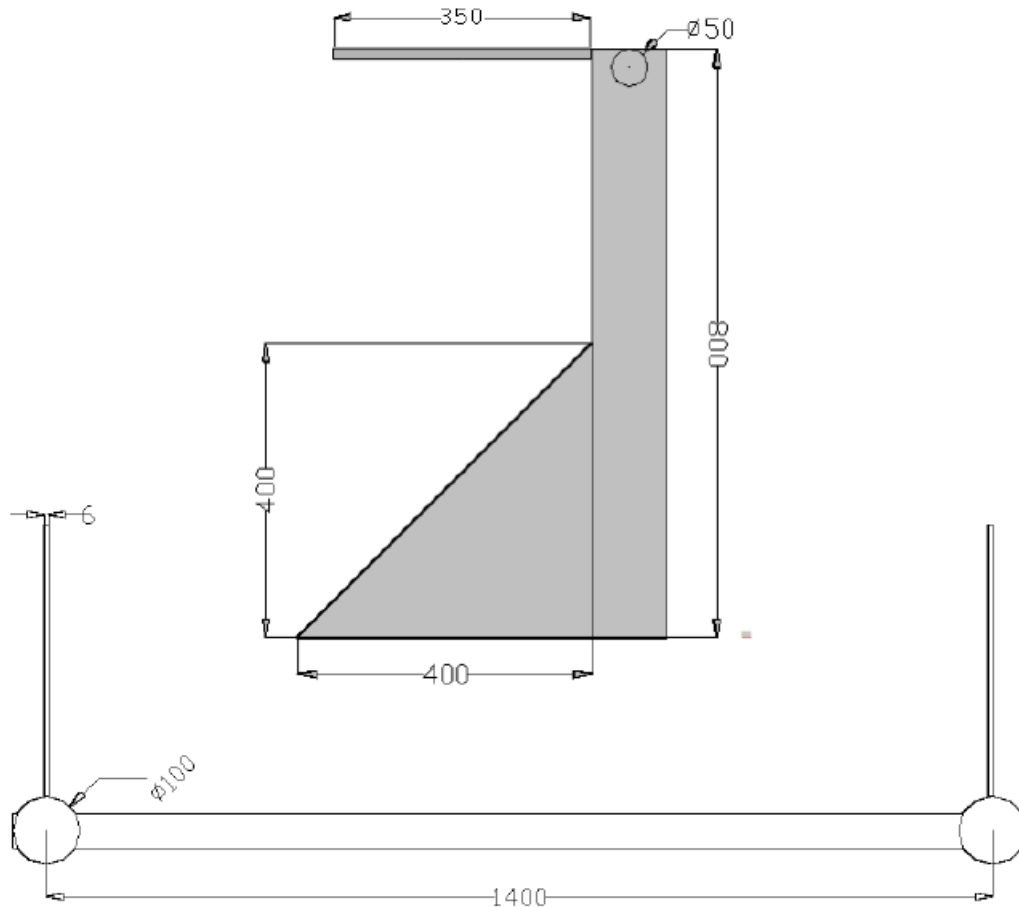


Abbildung 4: Tor in der Standard Platform League

Bei den Toren der Standard Platform League fallen die zusätzlichen Keile an Torpfosten auf, die sich nach hinten zur Befestigung des Netzes erstrecken. Diese Keile können zusätzlich bei der Bildverarbeitung dazu benutzt werden, den Roboter sich lokalisieren zu lassen, da sie mehr geometrische Informationen bieten, als ein einzelner Torpfosten. Zu bemerken ist auch, dass die Farbe beider Tore seit 2012 auf gelb festgelegt ist. Im Vorjahr war dies blau und gelb.

3 Bilderkennung

Die optische Erkennung von Toren im RoboCup ist ein wichtiger Bestandteil des Wettbewerbs. Obwohl Bilderkennung im Allgemeinen nichts Neues ist, ist das stetige Verbessern bestehender Verfahren und das Finden neuer Methoden Teil fortlaufender Forschung. Die sich ständig ändernden Regeln des RoboCups sorgen mitunter auch dafür, dass bestehende Algorithmen ihre Effektivität verlieren und jährlich revidiert werden müssen. In dieser Arbeit werden zwei von vielen Verfahren vorgestellt, da die beiden Verfahren die unterschiedlichen Herangehensweisen gut veranschaulichen.

3.1 Probleme

Die allgemeinen Probleme, die es bei der Bilderkennung für Tore zu lösen gilt, sind aufgrund der Umstände mannigfaltig: Durch die Mobilität des Roboters ändert sich die Perspektive ständig. Das führt dazu, dass das Bild des Tores, welches der Roboter erhält, teilweise stark verzerrt wird. Es kann beispielsweise vorkommen, dass der Roboter nur den unteren Teil eines einzigen Pfostens sieht. Dies führt zu großen Probleme bei der Lokalisation anhand dieses Bildes alleine, sodass das der optische Momenteindruck so nicht ausreicht, um festzustellen, wo sich der Roboter befindet. Das optische System muss also in ein umfassenderes System eingebettet werden. Die mittlerweile teils gleichfarbigen Tore kommen hierbei erschwerend hinzu.

Ein weiteres Probleme wurde bereits erwähnt: Tornetze können teilweise als Feldlinien erkannt werden. Außerdem können Teile vom Tor verdeckt sein (zum Beispiel von anderen Robotern) oder es könnten Hindernisse in der Schussbahn stehen. Letzteres ist nicht direkt ein Problem der Torerkennung, aber es ist damit eng verbunden, denn ein Torschussversuch kann nur erfolgreich sein, wenn der Schuss nicht vorher blockiert wird. Der Roboter muss sich also bewusst sein, wie er sich positionieren muss, damit ein Schussversuch gelingen kann.

Letztendlich muss die Lösung als Programm implementiert werden, welches auf der verwendeten Hardware des Roboters lauffähig ist. Diese Hardware ist häufig im Vergleich zu größeren Rechnern stark eingeschränkt, sodass die Lösung gut optimiert werden muss. Dies kann als versteckte Schwierigkeit aufgefasst werden, denn eine theoretische Lösung, die algorithmisch beziehungsweise mathematisch perfekt ist, läuft wahrscheinlich unpraktikabel langsam auf einem RoboCup Roboter.

3.2 Erkennung mittels geometrischer Relationen

Bei diesem Verfahren, das die geometrischen Strukturen der Elemente im RoboCup ausnutzt, werden vier Schritte ausgeführt:

1. Farbkalibrierung im YUV-Farbraum
2. Farbsegmentierung
3. Erkennung des Horizonts
4. Extraktion der Torpfosten und Modellierung

3.2.1 Farbkalibrierung YUV-Farbraum

Der YUV-Farbraum besteht aus drei Komponenten: Helligkeit und zwei Farbkomponenten. Er wurde ursprünglich für das analoge Farbfernsehen eingesetzt. Dies war aus zwei Gründen günstig: Zum einen war Kompatibilität zu alten Graustufen-Fernsehern gewährleistet, da lediglich die beiden neuen Farbkomponenten zum ursprünglichen Graustufenbild hinzukamen und zum anderen, da sich das menschliche Auge in diesem Farbraum besonders Fehler-tolerant verhält, sodass selbst bei schlechten Lichtverhältnissen die wichtigen Details erkennbar sind. Der letzte Punkt ist ausschlaggebend für die Verwendung in diesem Bilderkennungsverfahren, da auch Roboter von der separaten Helligkeitskomponente profitieren und besser bei wenig Licht arbeiten können.

Die folgende Grafik zeigt, wie sich ein Bild im YUV-Farbraum zusammensetzt:

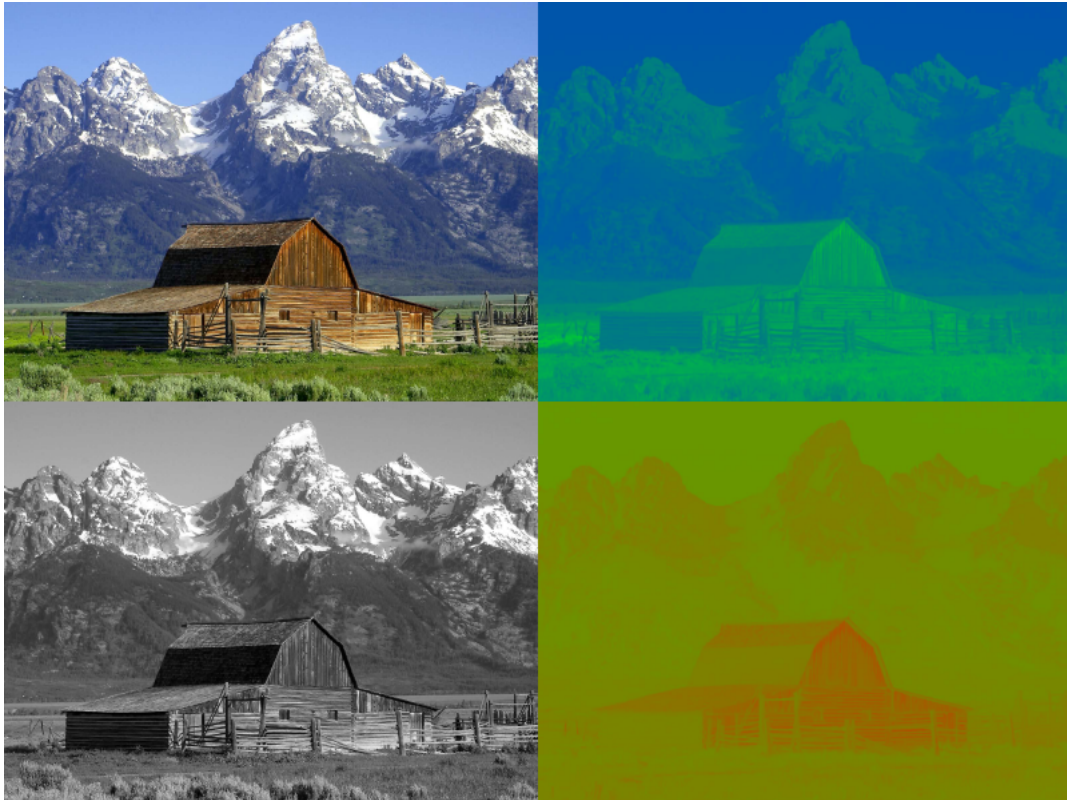


Abbildung 5: Beispiel eines Bildes im YUV-Farbraum

3.2.2 Farbsegmentierung

Bei der Farbsegmentierung geht es darum, Bereiche ähnlicher Farbe in einem Bild durch eine eindeutige Farbe zu markieren. Dies erleichtert die weitere Verarbeitung dieses Bildes enorm, da alle gleichfarbigen Bereiche in der Pixelmatrix nun den gleichen Wert haben. Wenn dieses Verfahren nicht angewendet würde, wären die Farbwerte selbst auf einem guten Bild nie eindeutig, da sich die Farbwerte sogar in einem Bild eines perfekt belichteten einfarbigen Gegenstandes unterscheiden.

Ein bekannter und einfacher Algorithmus für das Finden von Bereichen ähnlicher Farbe ist der Region Growing-Algorithmus. Hierbei wird ein beliebiger Punkt in einer Figur eines Bildes selektiert (manuell oder automatisch) und als Startpunkt festgelegt. Die Farbwerte in dem Startpunkt werden festgestellt und gespeichert. In der nächsten Iteration des Algorithmus wächst der selektierte Bereich ab Startpunkt um ein Pixel nach außen in alle Richtungen, solange der jeweilige neue Pixel einen Farbwert aufweist, der einen beliebigen Toleranzwert nicht übersteigt. Wenn der Toleranzwert an dieser Stelle überstiegen würde (zum Beispiel durch starke Farb- oder Helligkeitsunterschiede), dann wird an dieser Stelle nicht weiter gewachsen. Dieser Algorithmus iteriert solange, bis es in einer Iteration nicht mehr möglich ist, ein einziges Mal zu wachsen. Ab dann ist diese Fläche zu ende markiert. Die folgende Grafik soll den Algorithmus veranschaulichen:

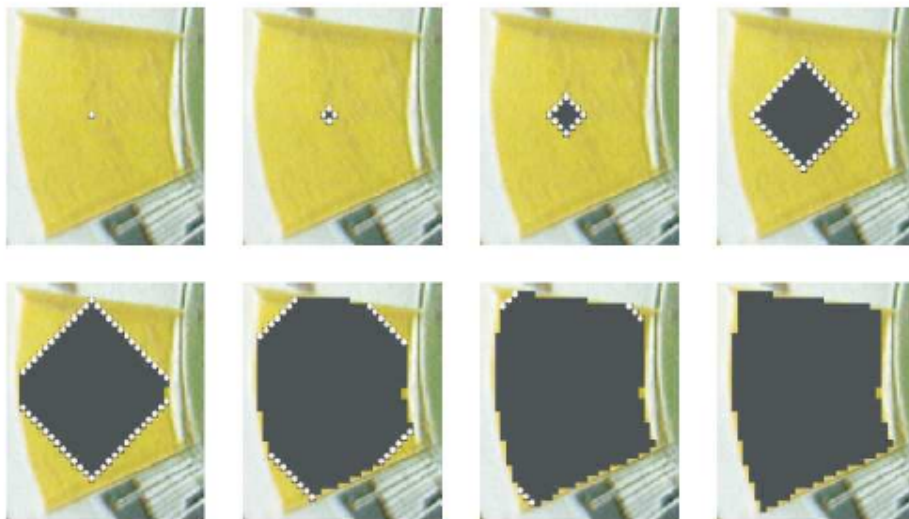


Abbildung 6: Vorgang der Farbsegmentierungsalgorithmus

Die folgende Grafik zeigt ein auf diese Weise markiertes Tor im RoboCup:

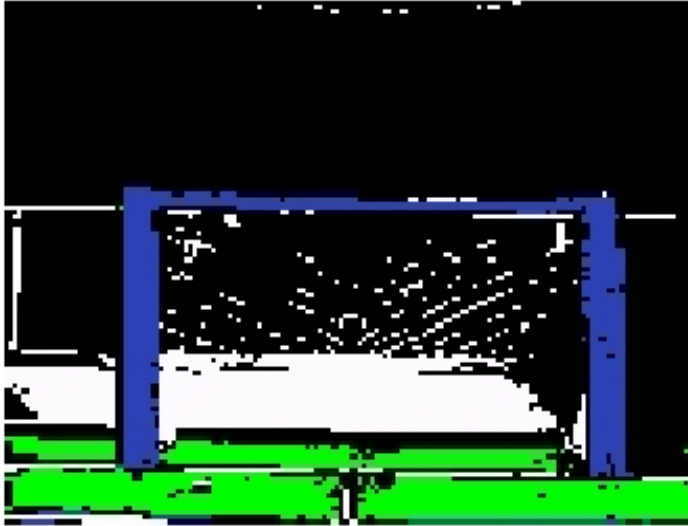


Abbildung 7: Farbsegmentiertes Bild eines RoboCup-Tores

Es ist einfach zu sehen, dass das gesamte Tor einfarbig blau markiert wurde, wodurch die Weiterverarbeitung stark vereinfacht wird.

Einige Programme existieren, um Robotern interaktiv verschiedene Farbflächen beizubringen. Dies erleichtert die Schwellenwertfindung und lädt zu Experimenten mit verschiedenen Lichtverhältnissen ein. Eines dieser Programme sieht so aus:

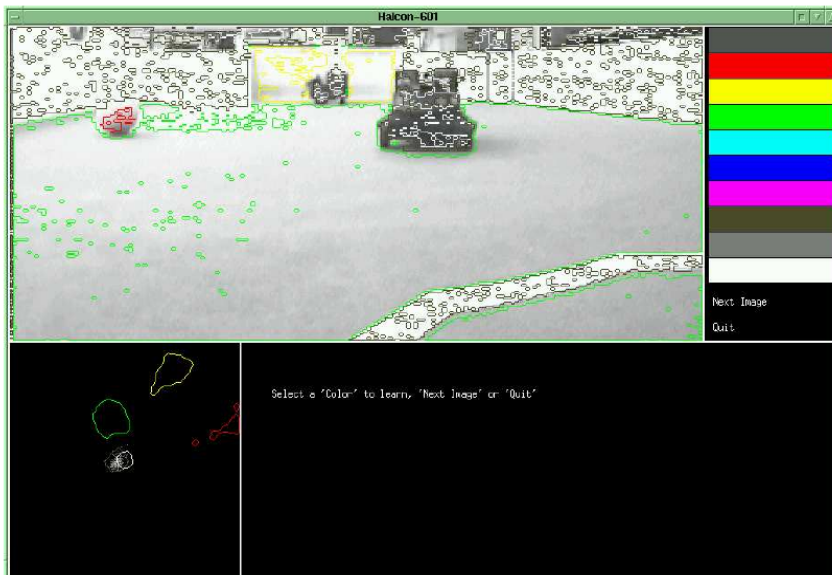


Abbildung 8: Programm zur interaktiven Echtzeit-Farbsegmentierung

Das Ergebnis mit diesem Programm sieht wie folgt aus:

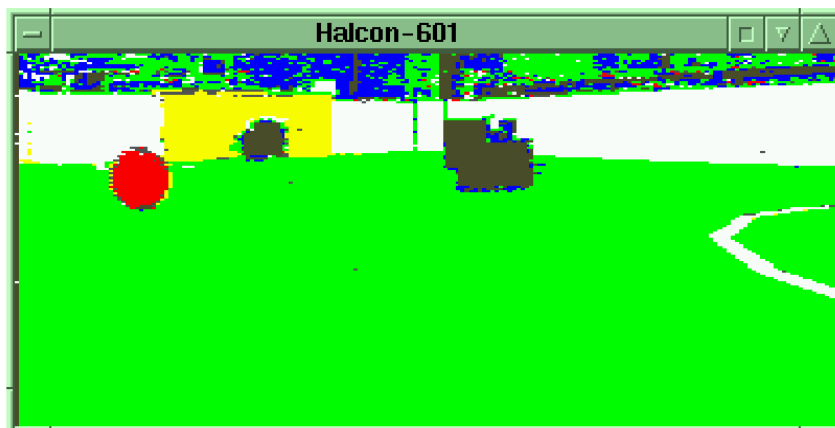


Abbildung 9: Segmentiertes Kamerabild

3.2.3 Erkennung des Horizonts

Als nächstes folgt die geometrische Erkennung, bei der sich zu Nutze gemacht wird, dass alle Pixel oberhalb der Torlatte unnötige Informationen darstellen. Deshalb wird die Torlatte als obere Grenze und Horizont festgelegt und das Bild darüber abgeschnitten. Zudem wird die Torlinie erkannt und als unterer Horizont gewertet.

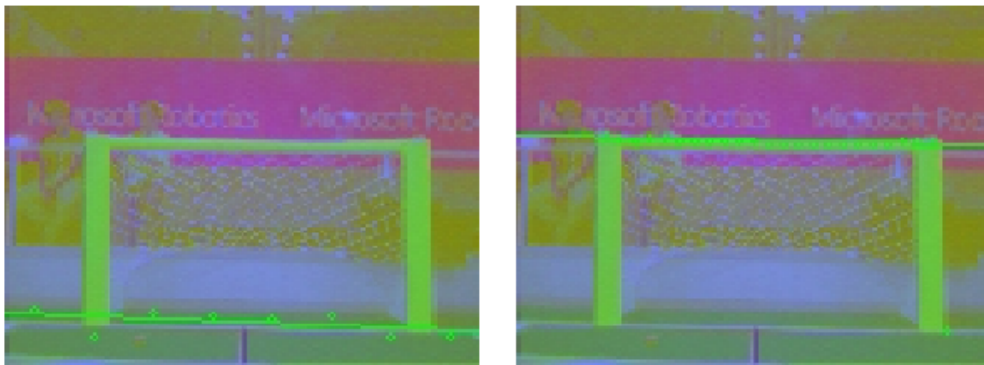


Abbildung 10: Geometrische Horizonterkennung

3.2.4 Extraktion der Torpfosten und Modellierung

Mithilfe von Scanlinien wird das inzwischen markierte und eingeschränkte Tor extrahiert und mittels Rechtecken modelliert.

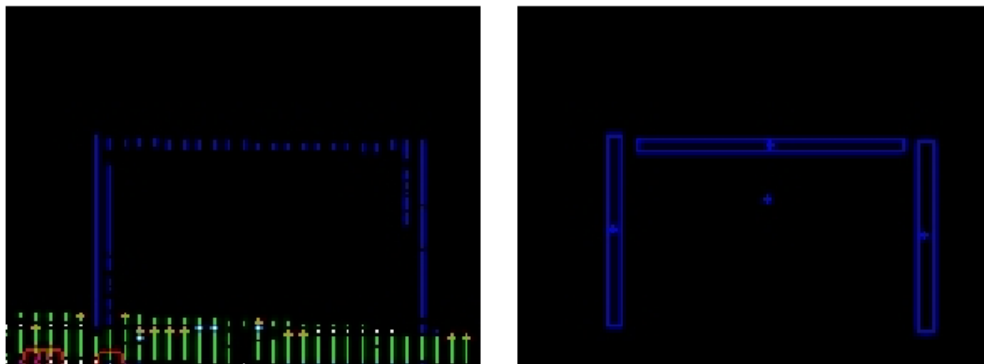


Abbildung 11: Links: Extraktion; Rechts: Modellierung

3.2.5 Beispiele

Nachfolgend ein paar anschauliche Beispiele der vier Schritte. Sie zeigen besonders die Stabilität und Verlässlichkeit des Algorithmus selbst bei schlech-

ten Bedingungen wie verdeckten Torpfosten und ungünstigen Perspektiven.

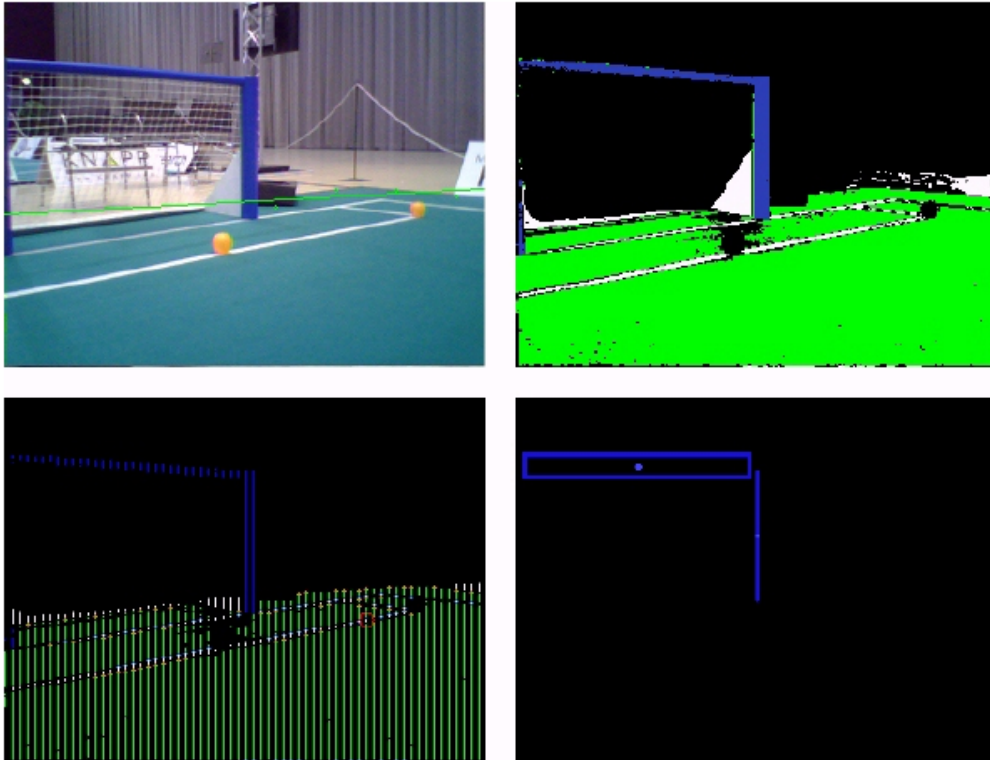


Abbildung 12: Beispiel des Verfahrens mit teils unsichtbarem Torpfosten

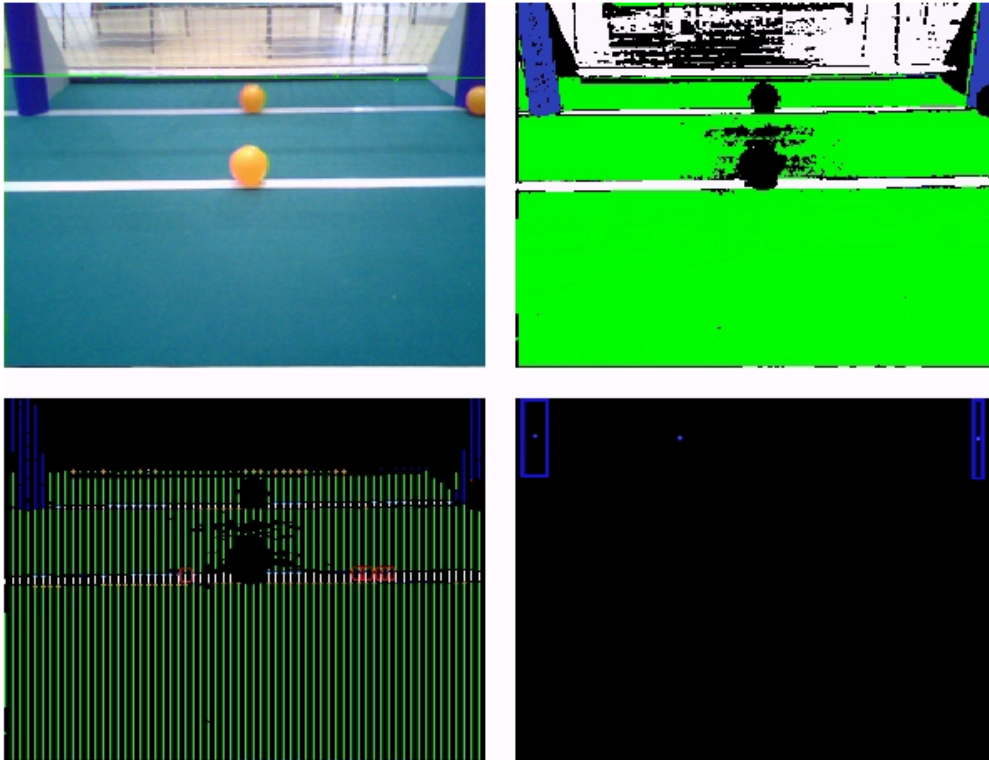


Abbildung 13: Beispiel des Verfahrens mit Bodenblick und wenig Torpfosten-
anteil im Bild

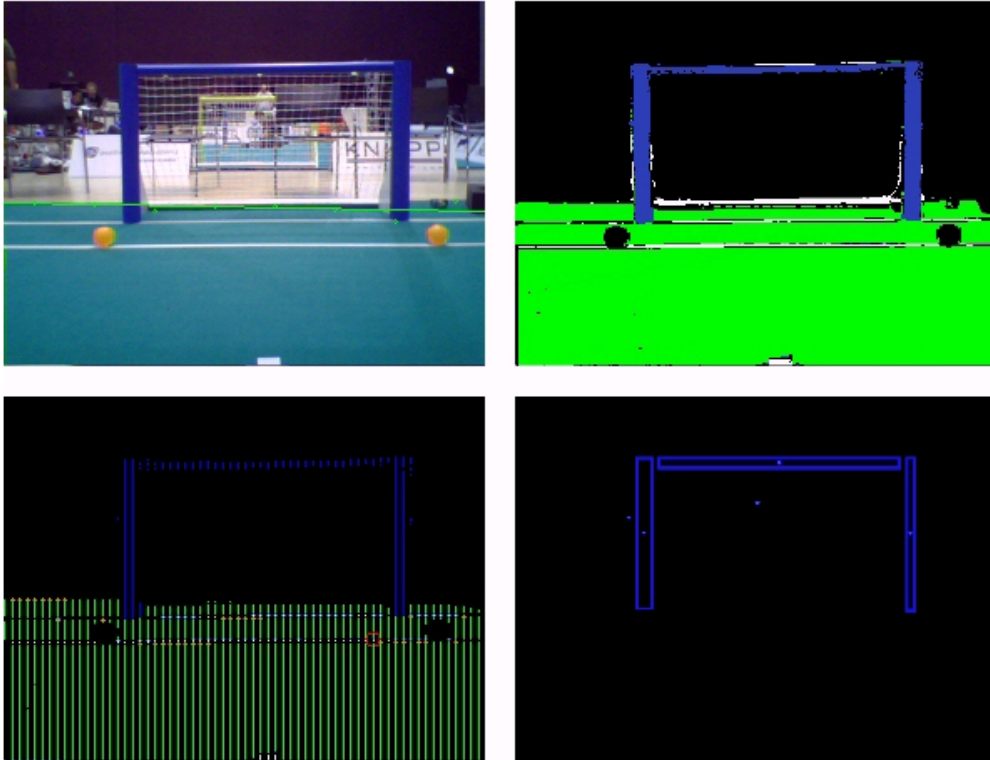


Abbildung 14: Beispiel des Verfahrens unter optimalen Bedingungen

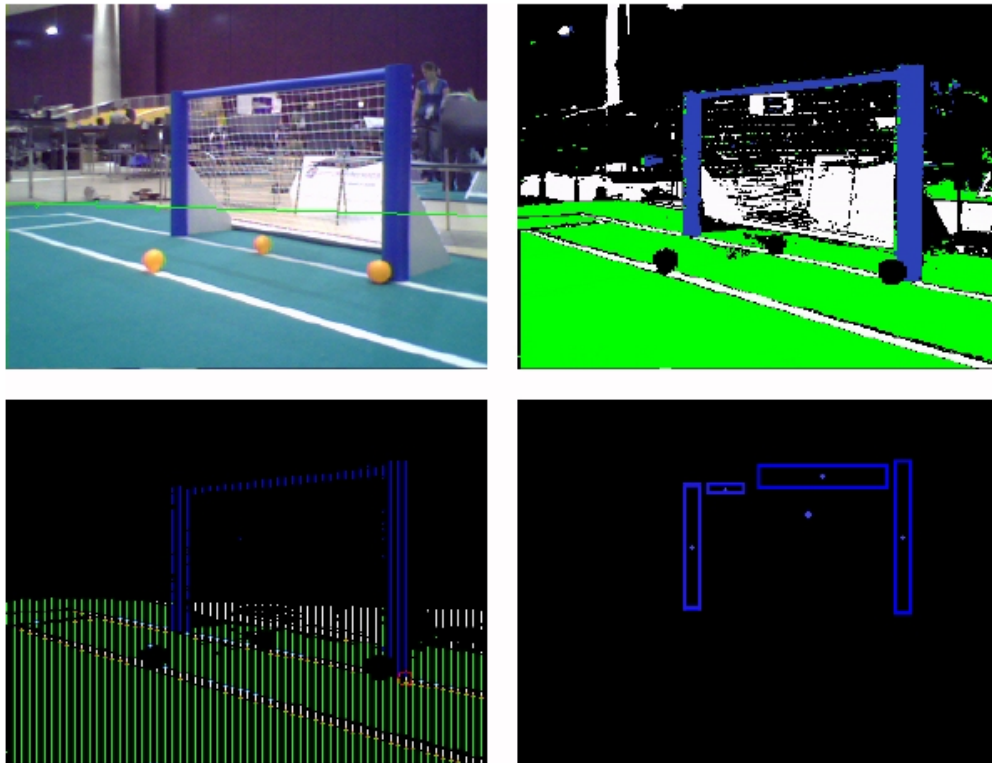


Abbildung 15: Beispiel des Verfahrens mit ungünstiger Perspektive

3.3 Erkennung mittels Hough-Transformation

Ein weiterer Weg zur Torerkennung im RoboCup ist die Hough-Transformation. Die Hough-Transformation wurde 1962 von P.V.C. Hough entworfen und von ihm unter dem Namen „Method and Means for Recognizing Complex Patterns“ patentiert. Später wurde sie von Duda und Hart weiterentwickelt und verbessert.

Die Hough-Transformation stellt ein Verfahren zur Erkennung von beliebigen, vorgegebenen geometrischen Strukturen in einem Gradientenbild dar. Angewendet wird dieses Verfahren nicht nur im RoboCup sondern häufig auch zu industriellen Zwecken. Beispielsweise um bestimmte Arbeitsvorgänge von Maschinen zu automatisieren oder zu verbessern.

Im Bezug auf die Torerkennung im RoboCup führt die Hough-Transformation alleine zu keinem Ergebnis. Zur endgültigen Erkennung des Tores mittels der Hough-Transformation sind vier wesentliche Schritte nötig. Hat der Roboter ein Bild eingefangen, finden zunächst zwei Schritte statt, um dieses

Bild auf die Hough-Transformation vorzubereiten. Der erste Schritt ist die Farbfilterung im HSV-Farbraum, der zweite Schritt ist die Anwendung eines Eckenfilters zum Erkennen der Torkonturen. Erst danach ist das Bild so weit aufgearbeitet und die eigentliche Hough-Transformation kommt zum Einsatz, um die Torsegmente zu erkennen. Zuletzt wird das entstandene Tormodell anhand von Eckpunkten modelliert und aufgespannt. Im Folgenden werden die einzelnen Schritte genauer erläutert.

3.3.1 Der HSV-Farbraum

Im HSV-Farbraum werden Farben über drei Komponenten definiert: Die H-(Hue)Komponente beschreibt den Farbton, die S-(Saturation)Komponente bestimmt wie satt bzw. blass die Farben dargestellt werden und mit der V-(Value)Komponente kann die Helligkeit definiert werden.

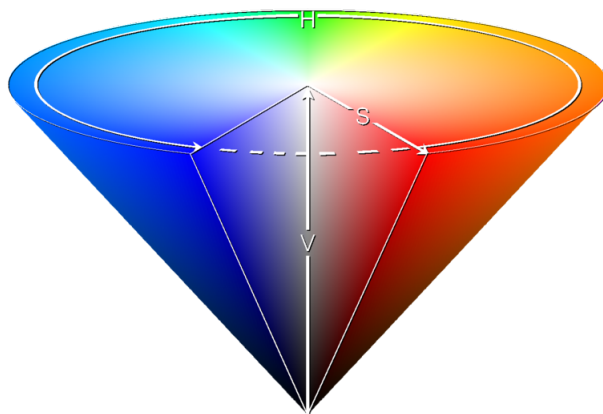


Abbildung 16: HSV-Farbraum als Kegel

Farbfilterung im HSV-Farbraum

Farbfilterung Unter Farbfilterung versteht man ein Verfahren, das eine bestimmte Farbe in einem Bild erkennt und nur diese herausfiltert. Alles andere im Bild wird vernachlässigt und nach der Farbfilterung nicht mehr dargestellt.

Warum Farbfilterung im HSV-Farbraum? Probleme bei der Farbfilterung entstehen dadurch, dass die Farbe des Tores z.B. durch unterschiedlichen Lichteinfall, Schatten oder Verschmutzung unterschiedlich wahrgenom-

men werden. Genau dort liegt der große Vorteil des HSV-Farbraumes. Ein Farbton kann unabhängig von Schwankungen der Helligkeit geprüft werden, indem man nur den Farbton zur Farbfilterung verwendet und die Sättigung und den Helligkeitswert vernachlässigt. Somit ist die Farbprüfung unabhängig von Schattenbildungen, Schmutz oder Schwankung in der Intensität einer Lichtquelle möglich.

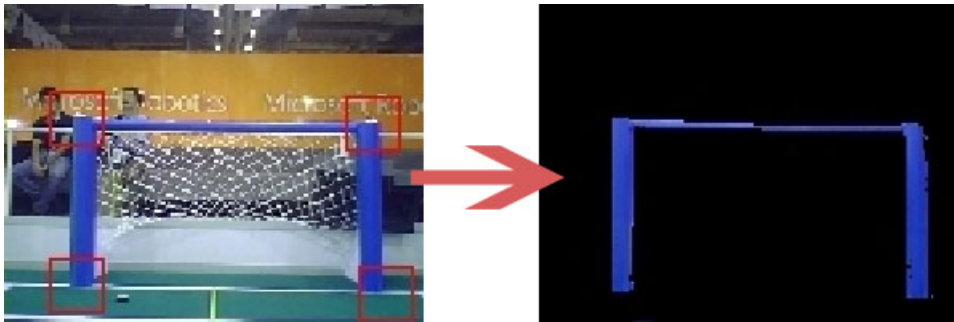


Abbildung 17: Bild nach Anwendung der Farbfilterung im HSV-Farbraum

3.3.2 Eckenfilter zum Erkennen von Torkonturen

Das Bild ist nach Anwendung des Farbfilters teilweise für die Hough-Transformation vorbereitet. Nun müssen nur noch mittels eines Eckenfilters die Torkonturen ausgearbeitet werden, da nur diese später relevant sind.

Das Erkennen der Konturen beruht auf der Tatsache, dass im Bereich einer Kontur eine sehr starke Helligkeits- bzw. Farbänderung vorliegt. In unserem Beispiel des blauen Tores haben wir die blaue Farbe herausgefiltert und der Rest des Bildes wird schwarz dargestellt. Nun gehen wir mit Hilfe eines Algorithmus von einem blauen Pixel aus in eine beliebige Richtung solange, bis eine starke Farb- bzw. Helligkeitsveränderung festgestellt wird. An den Stellen, an denen eine Veränderung erkannt wird, muss eine Kante vorliegen. Die Kanten werden nachgezeichnet und als einfache Linien dargestellt. Nach der Durchführung der ersten beiden Schritte liegt uns nun ein segmentiertes Bild vor.

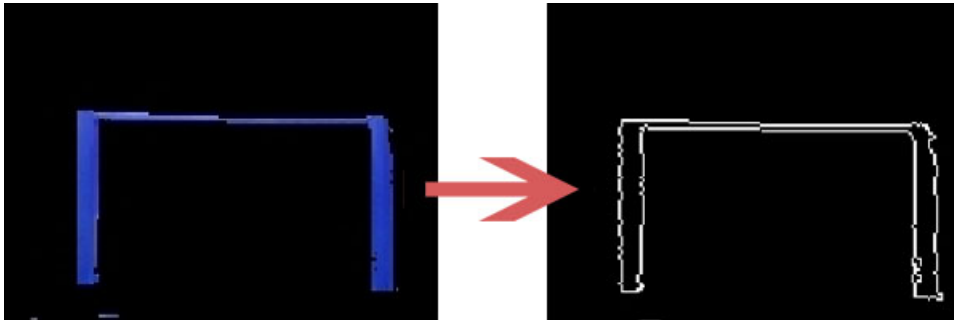


Abbildung 18: Bild nach Anwendung des Eckenfilters

3.3.3 Hough-Transformation zum Erkennen der Torsegmente

Nachdem das Bild nun fertig vorverarbeitet ist, können wir zum eigentlichen Schritt der Hough-Transformation selbst kommen. Ziel dieses Verfahrens ist es, wie Anfangs schon erwähnt, eine vorgegebene Referenzstruktur in einem Bild zu erkennen. In unserem Beispiel ist die Referenzstruktur eine Gerade, da das Tor aus Geraden (den zwei Pfosten und der Latte) besteht.

Geraden werden oft, wie in der Schule schon gelernt, durch die Form $y = m * x + b$ dargestellt. Die Idee hinter der Hough-Transformation ist es, alle Punkte in einem Bild in einen anderen Raum, der als Dualraum oder auch Hough-Raum bezeichnet wird, zu transformieren. Dazu müssen zunächst geeignete Parameter für eine Gerade gefunden werden. Dies ist in der Form $y = m * x + b$ nur bedingt möglich. Aus diesem Grund wandelt man die Gerade erst in die Hessesche Normalform um. Liegt uns die Gerade in der Form $d = x * \cos(\alpha) + y * \sin(\alpha)$ vor, wählen wir als unsere beiden Parameter den Abstand d und den Winkel α . Diese beiden Parameter dienen uns nun dazu, den neuen Raum aufzuspannen. Der Winkel wird die neue x-Achse und der Abstand die neue y-Achse, wie auch in Abbildung 19 zu sehen ist.

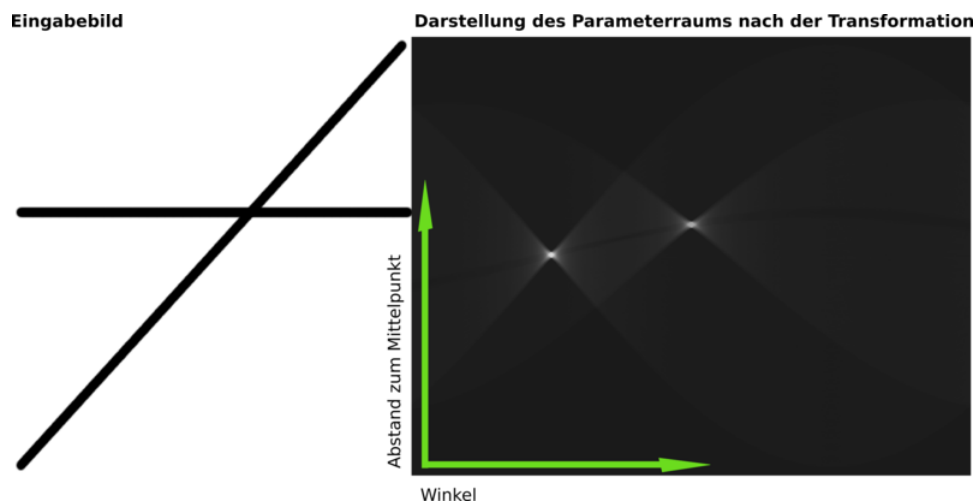


Abbildung 19: Beispiel für eine Hough-Transformation

Nachdem geeignete Parameter gewählt wurden. Werden alle Punkte im Ausgangsbild in den neuen Raum transformiert. Dies funktioniert folgendermaßen: Es wird ein Punkt ausgewählt und von diesem aus für alle möglichen Winkel der dazugehörige Abstand berechnet. Für alle Winkel und die dazugehörigen Abstände trägt man die so entstehenden Punkte im Dualraum ein. Dadurch entsteht für jeden Punkt im Ausgangsbild eine Kurve im Dualraum. Jede dieser Kurven in unserem Hough-Raum repräsentiert also einen Punkt im Ausgangsbild. Der Hough-Algorithmus überträgt nach eben beschriebem Verfahren alle Punkte in den Dualraum. Das Entscheidende ist nun, dass alle Kurven, die Punkte einer Geraden im Ausgangsbild repräsentieren sich in einem Punkt im Dualraum schneiden. Finden wir im Hough-Raum also solche Schnittstellen, wissen wir, dass an diesen Stellen eine Gerade im Originalbild vorliegt. Dies ist auch in Abbildung 1.4 zu sehen. Die Kurven im rechten Bild repräsentieren Punkte im linken Bild. An zwei Stellen im linken Bild sieht man hellere Punkte: Das sind die beiden Punkte an denen sich die Kurven schneiden und somit die Punkte, an denen Geraden vorliegen.

Anders kann man das Transformieren in den Dualraum auch als Matrix und nicht grafisch darstellen. Dies funktioniert genau wie oben beschrieben, nur dass man für alle Winkel und die dazugehörigen Abstände nicht einen Punkt im neuen Dualraum einträgt sondern in eine Matrix. Diesmal entsprechen die Spalten der Matrix den Winkeln und die Zeilen den Abständen. Ein Eintrag in der Matrix bedeutet, den Wert in der Matrix an dieser Stelle um Eins zu erhöhen. Zu Beginn sind alle Felder der Matrix mit dem Wert Null initialisiert. Findet an einer Stelle ein Eintrag statt, wird der Wert auf Eins

danach auf Zwei usw. erhöht. Aus diesem Grund wird die Matrix oft auch als Voting-Matrix bezeichnet. Hat man für alle Punkte gevotet, findet man die Geraden dieses Mal nicht anhand der Schnittpunkte der Kurven, sondern dadurch, dass alle Punkte, die auf einer Geraden liegen, ein gemeinsames Feld in der Matrix haben, und dieses somit am höchsten gevotet wurde. Das Feld, welches die höchste Votingzahl, hat repräsentiert also die Gerade im Ausgangsbild.

3.3.4 Aufspannen des Tormodells

Im letzten Schritt geht es darum, das Ergebnis der Hough-Transformation auszuwerten und nach Häufungspunkten zu suchen. Denn wie oben beschrieben, repräsentieren die Häufungspunkte die Geraden im Ausgangsbild. Wurden die Häufungspunkte gefunden, können diese wieder als Geraden dargestellt werden. Der Winkel und der Abstand können einfach abgelesen werden und die Gerade wieder im Ausgangsraum eingetragen werden. Ein Problem, welches sich nun ergibt, ist, dass unser Tor nicht aus Geraden, sondern aus endlichen Linien besteht. Mittels der Hough-Transformation werden aber nur die unendlichen Geraden erkannt und der Endpunkt unserer Linien kann nicht bestimmt werden. Um unser endgültiges Tormodell zu modellieren, müssen die Eckpunkte der Tore gefunden werden. Dies kann dadurch realisiert werden, dass man die Schnittpunkte der gefundenen Geraden betrachtet, und mit Hilfe derer die überflüssigen Teile der Geraden abschneidet und nur die für das Tormodell relevanten Teile modelliert.

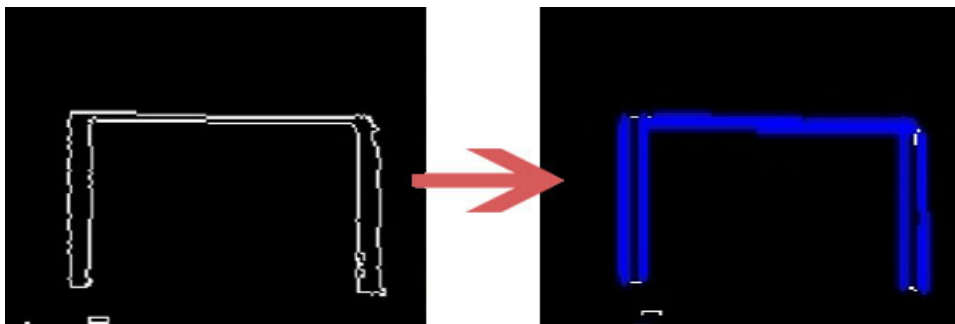


Abbildung 20: Bild nach Anwendung der Hough-Transformation und dem Aufspannen des Tormodells

3.3.5 Vorteile der Hough-Transformation an Hand von Beispielen

Wie zu Beginn erklärt, entstehen Probleme beim Erkennen des Tores z.B. dadurch, dass Teile des Tores verdeckt sind. Dies stellt, wie das folgende Bei-

spiel zeigt, kein Problem dar. Das Tor wird mittels der Hough-Transformation trotzdem einwandfrei erkannt.

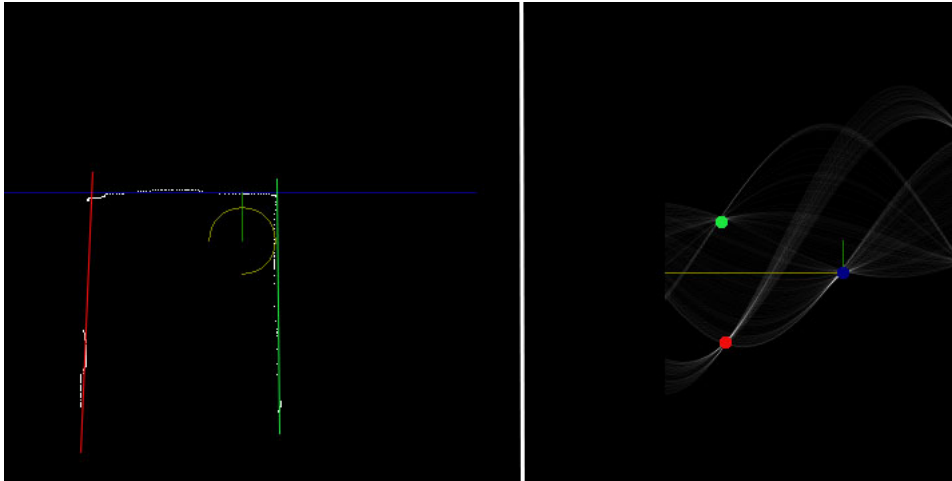


Abbildung 21: Transformation eines Teilweise verdeckten Tores in den Hough-Raum

Abbildung 21 zeigt wie ein teilweise verdecktes Tor nach dem Transformieren in den Hough-Raum aussieht. Die Häufungspunkte auf der linken Seite sind mit den drei farbigen Punkten gekennzeichnet und im Ausgangsbild sind die farbigen Geraden, die die Punkte repräsentieren, eingezeichnet. Es ist klar zu erkennen, dass es kein Problem ist, dass der linke Pfosten nur teilweise zu sehen ist. Der Häufungspunkt wird trotzdem erkannt und der Torpfosten kann komplett modelliert werden.

Ein weiteres Problem könnte eine schlechte Perspektive sein. Auch hiermit hat das Verfahren mittels Hough-Transformation keine Probleme. Abbildung 22 zeigt erneut die drei Geraden im Ausgangsbild mit den zugehörigen Punkten im Hough-Raum farbig markiert. Auch hier ist zu sehen, dass die schlechte Perspektive kein Problem darstellt und das Tor erkannt und modelliert werden kann.

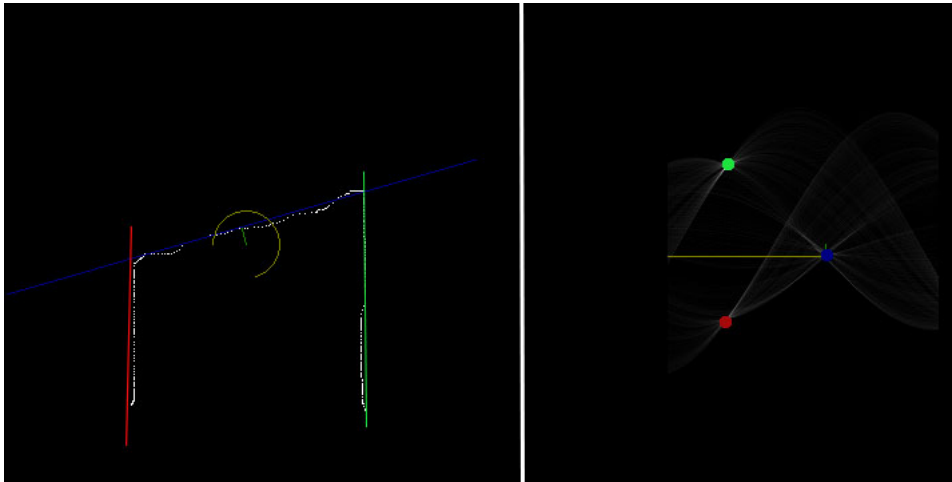


Abbildung 22: Transformation eines Tores aus einem schlechten Blickwinkel in den Hough-Raum

4 Fazit

Das Thema Torerkennung im RoboCup ist und bleibt ein spannendes Thema und wird noch lange einen Schwerpunkt in der Forschung im Bezug auf den RoboCup darstellen. Vor allem die häufigen Regeländerungen machen es erforderlich, nahezu jedes Jahr neue Methoden zu entwickeln oder alte zu verbessern. Mit zunehmender Verbesserung der Technik und des Wissensstandes werden auch die Verfahren immer effektiver werden.

5 Referenzen

- [1] Official RoboCup Standard Platform League Rules (http://wiki.robocup.org/wiki/Standard_Platform_League#Rules) 2010, 2011, 2012.
- [2] Official RoboCup Humanoid League Rules (http://wiki.robocup.org/wiki/Humanoid_League#Rules) 2009, 2010, 2011, 2012.
- [3] Official RoboCup Middle Size League Rules (http://wiki.robocup.org/wiki/Middle_Size_League#Rules) 2009, 2010, 2011, 2012.
- [4] Fast Image Segmentation, Object Recognition and Localization in a RoboCup Scenario (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.1388&rep=rep1&type=pdf>)
- [5] Recognition of Standard Platform RoboCup Goals (<http://gsyc.es/jmplaza/papers/jopha-2010.pdf>)
- [6] YUV-Farbmodell (<http://de.wikipedia.org/wiki/YUV-Farbmodell>)
- [7] RGB-Farbraum (<http://de.wikipedia.org/wiki/RGB-Farbraum>)
- [8] HSV-Farbraum (<http://de.wikipedia.org/wiki/HSV-Farbraum>)
- [9] k-Means-Algorithmus (<http://de.wikipedia.org/wiki/K-Means-Algorithmus>)
- [10] Die Hough-Transformation (http://page.mi.fu-berlin.de/alt/vorlesungen/sem04/10_6Die%20Hough.doc)
- [11] Die Hough-Transformation (<http://de.wikipedia.org/wiki/Hough-Transformation>)
- [12] Die Hough-Transformation (<http://www.malerczyk.de/applets/Hough/Hough.html>)
- [13] Die Hough-Transformation (<http://www.erkenntnishorizont.de/ki/bildverarb/hough.c.php?s>)
- [14] Hough-Transformation Applett (http://www.activovision.com/octavi/doku.php?id=hough_)