



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Matthias Wilms

Lokalisierung und Verfolgung von codierten Messmarken mit Hilfe von
Bildverarbeitung

Matthias Wilms

Lokalisierung und Verfolgung von codierten Messmarken mit Hilfe von
Bildverarbeitung

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Angewandte Informatik
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Andreas Meisel
Zweitgutachter: Prof. Dr. rer. nat. Kai von Luck

Abgegeben am 12. Dezember 2008

Matthias Wilms

Thema der Bachelorarbeit

Lokalisierung und Verfolgung von codierten Messmarken mit Hilfe von Bildverarbeitung

Stichworte

Codierte Messmarke, Bildverarbeitung , Kalman Filter, Kantenerkennung, Infrarot

Kurzzusammenfassung

Diese Arbeit befasst sich mit codierten Messmarken für den Einsatz in einem kamerabasierten Navigationssystem für mobile Systeme. Hierfür werden verschiedene Markentypen entwickelt bzw. erweitert, die für die Verwendung in diesem System geeignet sind. Die ebenso entwickelte/erweiterte Software zur Erkennung dieser Marken in Einzelbildern von digitalen Videosignalen ist mit einem Verfolgungsverfahren auf Basis des Kalman Filters ausgestattet.

Matthias Wilms

Title of the paper

Localisation and Tracking of Coded Targets Based on Image Processing

Keywords

Coded Target, Image Processing, Kalman Filter, Edge Detetction, Infrared

Abstract

This paper is about coded targets for the use in a camera based navigation system for mobile systems. Therefore different types of coded targets are developed or enhanced to meet the requirements for the use in this system. The also developed/enhanced software to recognize these targets in video frames is equipped with a Kalman Filter based tracking method.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	x
1. Einleitung	1
1.1. Motivation	1
1.2. Ziel und Aufgabenstellung der Arbeit	3
1.3. Aufbau der Arbeit	4
2. Allgemeine Grundlagen	6
2.1. Digitale Bildverarbeitung	6
2.1.1. Grauwertbilder	6
2.1.2. Kanten	6
2.2. Kamera und Beleuchtung	8
2.2.1. Verwendete Kamera	8
2.2.2. Abbildungsmaßstab und Linsengleichung	8
2.2.3. Fokussierung	9
2.2.4. Belichtung	10
2.2.5. Retroreflektierende Folie	11
2.2.6. Infrarotbeleuchtung	12
2.3. Verwendete Hard- und Software	15
2.3.1. Testrechner	15
2.3.2. LTI-Lib und Impresario	15
3. Codierte Messmarken	17
3.1. Einführung	17
3.2. Vorstellung verschiedener Markentypen	21
3.2.1. Kreisförmige Marken	21
3.2.2. Quadratische Marken	27
3.2.3. Selbstähnliche Marken	31
4. Verfolgung	36
4.1. Einleitung	36
4.2. Einfaches Verfahren	36
4.3. Kalman Filter	37
4.4. Beispiele	41
4.4.1. Bewegung 1	41

4.4.2. Bewegung 2	43
4.4.3. Bewegung 3	43
4.4.4. Bemerkungen	45
4.5. Ablauf der Verfolgung	45
5. Doppelverhältnismarke	49
5.1. Vorstellung	49
5.1.1. Aufbau	49
5.1.2. Konfiguration der Doppelverhältnisbereiche	51
5.1.3. Erkennung	55
5.2. Analyse	58
5.2.1. Materialien	58
5.2.2. Ort der Anbringung der Marken	61
5.2.3. Erkennungsentfernung	64
5.2.4. Erkennungsgeschwindigkeit	66
5.2.5. Sicherheit des Doppelverhältnisses zur Codierung der ID	66
5.3. Vorgenommene Veränderungen	68
5.3.1. Erweiterung der Marke um einen zusätzlichen Kreis	68
5.3.2. Verfolgung	70
5.4. Abschließender Test	71
5.5. Bewertung	72
6. Zylindermarke	77
6.1. Vorstellung	77
6.2. Verwandte Arbeiten	80
6.2.1. Zylinder	80
6.2.2. Kugel	82
6.3. Herstellung	83
6.4. Codierung	85
6.4.1. 1. Variante (2-aus-8)	86
6.4.2. 2. Variante (2-aus-10)	86
6.4.3. 3. Variante (2-aus-5)	87
6.5. Geometrische Eigenschaften	89
6.5.1. Gedrehte Abbildung des Zylinders	89
6.5.2. Einfluss perspektivischer Verzerrung auf den Barcode	89
6.5.3. Abbildung der Kugel	93
6.5.4. Sichtbarer Bereich der Kugel	95
6.6. Erkennung der Marken	97
6.6.1. Grundlagen	97
6.6.2. Kantenfilterung	101
6.6.3. Barcodefilterung	105
6.6.4. Markenextraktion	111
6.7. Impresario-Macro	121
6.7.1. Umsetzung	121

6.7.2. Parameter	121
6.8. Verfolgung	121
6.9. Tests	124
6.9.1. Barcodes	126
6.9.2. Gedrehte Abbildung des Zylinders	127
6.9.3. Neigung der Kamera	128
6.9.4. Klassifikation der Barcodebalken	128
6.9.5. Beleuchtung der Kugel	130
6.9.6. Identifikation eines anderen Elements im Bild als Marke	132
6.9.7. Fehlidentifikationen	134
6.9.8. Verteilung der Laufzeit	136
6.9.9. Vergleichstest	138
6.10. Bewertung	144
7. Kreisförmige Barcodemarke	147
7.1. Motivation	147
7.2. Konzept	147
7.2.1. Aufbau	147
7.2.2. Erkennung	149
7.2.3. Weitere Ideen	151
8. Fazit und Ausblick	153
8.1. Fazit	153
8.2. Ausblick	154
A. Anhang	156
Literaturverzeichnis	157
Glossar	162

Abbildungsverzeichnis

1.1. Shakey in Aktion.	2
1.2. Überblick über die Systemmodule.	4
2.1. Rampenkante und ein horizontales Grauwertprofil.	7
2.2. Bezeichnung der Größen des Abbildungsmaßstabs und der Linsengleichung.	9
2.3. Retroreflektierende Folie vom Typ SOLAS Grade.	12
2.4. Hinterspiegelte Glaskugeln, die einfallendes Licht in die Richtung aus der es kam reflektieren.	12
2.5. Illustration der erläuterten Begriffe, Einfallswinkel (Entrance Angle) und Beobachtungswinkel (Observation Angle) bei retroreflektierender Folie.	13
2.6. Bereich der Infrarotstrahlung im Vergleich zum sichtbaren Licht im elektromagnetischen Spektrum.	14
2.7. Vergleich zwischen manueller und automatischer Kameraeinstellung.	14
2.8. Impresario mit Prozessgraph für eine Canny-Edge Erkennung.	16
3.1. Crash-Test Szene mit runden schwarz-gelben Messmarken.	17
3.2. Verschiedene Typen codierter Messmarken.	18
3.3. Affine und perspektivische Transformationen.	20
3.4. Abbildung eines Kreises als Ellipse.	21
3.5. Darstellung der Abbildungsexzentrizität e	22
3.6. Marken mit 15-Bit Ringcode.	22
3.7. Marke der Firma AICON mit der ID 105.	23
3.8. Marke der Firma AICON mit 14 Einheiten.	24
3.9. Marken von Ahn u.a. mit verschiedenen Codelängen.	25
3.10. Fourier Tag mit der ID 210.	26
3.11. Fourier Tag mit der ID 210: Frequenzraum- und Binärdarstellung.	26
3.12. Oben Erkennung des Fourier Tag aus $3m$ Entfernung. Die unteren beiden Bilder zeigen die Abbildung der Bildzeile im Grauwertbereich (links) und im Frequenzspektrum (rechts).	27
3.13. Beispiele für die Nutzung von Marken in der AR.	28
3.14. Beispiel für ein interaktives Eingabegerät	28
3.15. Beispiele für ARToolkit-Marken.	29
3.16. Beispiele für ARTag-Marken. Abbildung aus [Fiala, 2005b]	30
3.17. Selbstähnliche Marke von Briggs und Scharstein.	32
3.18. Selbstähnliche Funktion mit $p = \frac{2}{3}$	32
3.19. Selbstähnliche kreisrunde Marke von Negre u.a..	34

4.1. Regelschleife des Kalman Filters.	39
4.2. Bewegung zu Kapitel 4.4.1.	42
4.3. Abweichungen der Vorhersagen von den Messwerten. (zu der Bewegung aus Abb. 4.2)	42
4.4. Bewegung zu Kapitel 4.4.2.	43
4.5. Abweichungen der Vorhersagen von den Messwerten. (zu der Bewegung aus Abb. 4.4)	44
4.6. Bewegung von links nach rechts zu Kapitel 4.4.3.	44
4.7. Grober Ablauf der Verfolgung.	47
5.1. Doppelverhältnismarke	50
5.2. Doppelverhältnis von 4 Punkten.	51
5.3. Aufbau des Doppelverhältnisbereichs.	53
5.4. Rasterverfahren zur Suche nach Doppelverhältnismarken.	56
5.5. Eckensuche auf der Aussenumrandung.	56
5.6. Beispiel für Kantensuche.	57
5.7. Beispiel für Eckensuche.	58
5.8. Struktur des erstellten Macros.	59
5.9. Anbringung der Marken an den Seitenwänden.	62
5.10. Anbringung der Marken direkt unter der Decke.	63
5.11. Anbringung der Marken direkt horizontal von der Decke hängend.	64
5.12. Erkennung aus ca. 15m.	65
5.13. Erkennung der Marke mit der ID 3.	68
5.14. Veränderte Doppelverhältnismarke mit fünf Kreisen.	69
5.15. Einordnung des Trackers in die vorhandenen Komponenten des Macros.	70
5.16. Einzelbild mit höchster Laufzeit ohne Tracking.	74
5.17. Versuchsaufbau (inkl. Beleuchtung) für den abschließenden Test der Doppelver- hältnismarken	75
6.1. 2D-Ansicht der Zylindermarke mit Beispielmaßen (2-aus-8-Code).	78
6.2. Foto einer Zylindermarke.	79
6.3. Genutzte Begriffe für die Kameraorientierung.	79
6.4. Zylindermarke mit 3 Bit Barcode aus [Arican, 2004]	80
6.5. Testparcours für die Zylindermarke aus [Arican, 2004]	81
6.6. Rocky 3.2 Roboter mit Zylinder.	82
6.7. Squidball in Aktion.	83
6.8. Kugelmarke mit roten und grünen Punkten aus [Bradley, 2005].	83
6.9. Zwei zusammengesetzte Kreisabschnitte zum Bekleben der Kugel.	85
6.10. 2D-Ansicht der Zylindermarke mit Beispielmaßen und der ID 86 in 2-aus-5-Code.	88
6.11. Barcode einer Zylindermarke mit maximalem Drehwinkel α und Scanlinie.	90
6.12. Neigung der Kamera.	90
6.13. Abbildung der Kugel als Kreis.	93
6.14. Abbildung der Kugel als Ellipse.	94
6.15. Illustration des sichtbaren Bereichs der Kugel.	96

6.16. Die grundlegenden Schritte zur Erkennung der Zylindermarke.	97
6.17. Vergleich der durch einen Kantenerfilter extrahierten Kanteneckenpunkte und der interpolierten Parabel.	100
6.18. Barcode mit Scanlinien.	101
6.19. Ablauf der subpixelgenauen Kantendetektion mit einer 3 x 3 Sobelmaske in y-Richtung.	105
6.20. Test	112
6.21. Darstellung der in Kapitel 6.6.4.1 verwendeten Bezeichnungen.	115
6.22. Exemplarische Darstellung des Kugelfensters.	115
6.23. Testmarke mit 2-aus-8 Code.	126
6.24. Testmarke mit 2-aus-10 Code.	127
6.25. Testmarke mit 2-aus-5 Code.	127
6.26. Marke mit 2-aus-8 Code in 3 m Entfernung mit maximaler Rotation der Codefläche.	128
6.27. Marke mit 2-aus-8 Code in 4 m Entfernung und maximaler Rotation der Codefläche durch Neigung der Kamera.	129
6.28. Marke mit 2-aus-8 Code und Neigung der Kamera um $\beta = 50^\circ$	129
6.29. Vergleich zwischen den Verfahren zur Ermittlung eines Basiswerts zur Klassifikation der Balken des Barcodes.	131
6.30. Extrembeispiel für die Schattenbildung durch einseitige Beleuchtung der Kugel.	132
6.31. Identifikation von Lamellen der Deckenbeleuchtung als korrekten 2-aus-8 Barcode.	133
6.32. Testbild zur Kugelgröße von Zylindermarken.	134
6.33. Testbild zur Problematik der Nähe von Zylindermarken.	135
6.34. Testbild zur Problematik bei auftreten mehrerer Kugeln im Kugelfenster.	135
6.35. Test zur Problematik überlappender Barcodes.	136
6.36. Testbild für Laufzeittest mit 3 Zylindermarken.	138
6.37. Verteilung der Gesamtlaufzeit der Erkennung auf die einzelnen Erkennungsschritte für Abbildung 6.36 ohne Verfolgung.	139
6.38. Verteilung der Gesamtlaufzeit der Erkennung auf die einzelnen Erkennungsschritte für Abbildung 6.36 mit Verfolgung.	139
6.39. Testbild für Laufzeittest mit 4 Zylindermarken, die fast das komplette Bild ausfüllen.	140
6.40. Verteilung der Gesamtlaufzeit der Erkennung auf die einzelnen Erkennungsschritte für Abbildung 6.39 ohne Verfolgung.	140
6.41. Pioneer Roboter mit Vorrichtung für Kamera und Infrarotscheinwerfer.	141
6.42. Frame aus dem abschließenden Vergleichstest der Zylindermarke.	142
7.1. Testumgebung aus [Manske, 2008a] mit Doppelverhältnismarken und parallel zur Decke ausgerichteter Kamera.	148
7.2. Kreisförmige Barcodemarke mit 2-aus-5 Code.	148
7.3. Kreisförmige Barcode-Marke mit 2-aus-5 Code, die von Scanlinien getroffen wird.	150
7.4. Konzentrische Ringmarken aus [Thomas u. a., 1997]	151

Tabellenverzeichnis

2.1. Einige Daten der verwendeten Kamera.	8
2.2. Einige Daten der verwendeten Infrarotscheinwerfer.	14
2.3. Verwendeter Rechner.	15
3.1. Einige geometrische Eigenschaften, die invariant unter den angegebenen Transformationen sind. (teilw. nach [Hartley und Zisserman, 2003])	20
4.1. Parameter für den Tracker.	48
5.1. Beispiel für den Doppelverhältnsbereich einer Marke (Alle Strecken in mm). Erkennungsgenauigkeit: 1 mm; Messgenauigkeit: 2 mm (teilw. aus [Wippich, 2008a])	53
5.2. Parameter des Macros CrossRatioTargets zur Erkennung von Doppelverhältnismarken.	60
5.3. In Kapitel 5.2 für Tests verwendete Marken.	61
5.4. In [Wippich, 2008a] empfohlene Werte für die Parameter des Erkennungsalgorithmus.	65
5.5. Vergleich der Marken mit der ID 1 und der ID 2 (siehe Tab. 5.3) bezogen auf die Abbildung der Strecken ihres Doppelverhältnisses auf die Anzahl von Pixeln auf dem dem Kamerasensor (siehe Tab. 2.1) bei der konfigurierten maximalen Erkennungsentfernung von 5363 mm)	67
5.6. In Kapitel 5.4 für Tests verwendete Marken.	73
6.1. Codierung von dezimalen Ziffern in 2-aus-5 Code.	87
6.2. Parameter des Macros CylinderTargets	122
6.3. Beispielwerte für die in Kapitel 6.3 vorgestellten Beispielmarken mit 2-aus-8 Barcode.	123

1. Einleitung

1.1. Motivation

Die mobile Navigation und die dafür notwendige Positionsbestimmung wird nicht nur bei uns Menschen immer wichtiger, man denke nur an Autonavigationsgeräte u.ä., sondern auch Teile der Robotik beschäftigen sich seit Jahren mit der Frage: Wie kann ein mobiler Roboter seine Position bestimmen und autonom in seiner Umgebung navigieren? In [Blanche und Lemarquand, 1992] wird diese Frage als das fundamentalste Problem beschrieben, welches gelöst werden muss, um einen mobilen Roboter mit autonomen Fähigkeiten auszustatten.

Erster mobiler Roboter, der Aktionen in seiner Umgebung u.a. mit Hilfe von Sensoren (z. B. einer TV-Kamera) und Großrechnern planen und durchführen konnte, war Anfang der 1970er Jahre der von SRI International (damals Stanford Research Institute) entwickelte Shakey (siehe [Shakey, 2008]). Das wohl bekannteste Problem, das Shakey lösen konnte, war das Verschieben von Klötzen (blocks) innerhalb, aber auch zwischen unterschiedlichen Räumen. Zu diesem Zweck wurden u.a. das Planungssystem STRIPS und für die Bildverarbeitung die Hough-Transformation eingesetzt.

Heutige Anwendungsbeispiele für solche Roboter bzw. Fahrzeuge sind z.B. Rollstühle, die den Behinderten bei der Suche nach (bzw. der Navigation auf) für ihre besonderen Bedürfnisse geeigneten Wegen helfen (siehe [Garcia Garcia u. a., 2002]). Genauso denkbar sind auch Roboter, die Wachaufgaben z.B. in großen Lagerhäusern übernehmen und bei der Entdeckung eines Eindringlings ein Alarmsignal inkl. Position der Person übermitteln. Ein aus den Medien bekanntes Beispiel für autonome Navigation ist die vom US-Verteidigungsministerium ausgeschriebene DARPA Urban Challenge¹, in der sich Fahrzeuge selbstständig auf einem vorgegebenen Kurs orientieren und diesen in einer vorher festgelegten Zeit abfahren müssen.

Für die genannten Aufgaben muss aber, die eingangs gestellte Frage geklärt werden, wie ein autonomer mobiler Roboter/autonomes Fahrzeug² seine Position (und auch seine Orientierung) feststellen kann, um dann in seiner Umgebung sicher navigieren zu können? Da ein Roboter seine Umwelt mit Hilfe von Sensoren wahrnimmt, stellt sich deshalb als erstes das Problem der Auswahl von Sensoren, die für eine Posenbestimmung³ mit dem Ziel der Navigati-

¹<http://www.darpa.mil/grandchallenge/>

²Die Ausdrücke autonomes Fahrzeug und autonomer mobiler Roboter sind innerhalb dieser Arbeit gleichbedeutend

³Kombination von Position und Orientierung im dreidimensionalen Raum



Abbildung 1.1. – Shakey in Aktion. Abbildung aus [Shakey, 2008]

on geeignet sind. Möglich sind z.B. Trägheitssensoren, bei denen sich auftretende Messfehler aber mit der Zeit ständig fortpflanzen. Weiterhin kann man das oft bei der Autonavigation verwendete satellitengestützte GPS⁴ nutzen, dessen Einsatz aber gerade im Indoor-Bereich nur schwer möglich ist (siehe [Manske, 2008a]). Aus diesen Gründen sind in der Literatur viele Ansätze dokumentiert, die mit (kalibrierten) digitalen (Video-)Kameras als Sensoren zur Erkundung der Umgebung arbeiten. Als Bezugspunkte in der Außenwelt werden zu diesem Zweck künstliche (z. B. [Briggs u. a., 2000]) oder teilweise auch natürliche Marken (z. B. [Yeh und Kriegman, 1995]) genutzt, die im Bild möglichst leicht und eindeutig zu identifizieren sind und einen Messpunkt zur Verfügung stellen. Mittels (mehrerer) dieser Marken, deren genaue Position im Raum bekannt ist, ist es möglich eine millimetergenaue Bestimmung der eigenen Position und Orientierung durchzuführen. Ähnliche Verfahren werden nicht nur in der Navigation und der dafür nötigen Posenbestimmung von autonomen mobilen Robotern eingesetzt, sondern sind z. B. auch in der Augmented Reality zu finden. Dort werden Marken und Kameras beispielsweise genutzt, um an bestimmten Stellen im von einem Raum aufgenommenen Bild (künstliche) Objekte einzublenden.

Da die Geschwindigkeit dieser Systeme direkt mit der Erkennung der verwendeten Marken zusammenhängt, existieren schon viele Lösungen für verschiedene Anwendungsbereiche und es werden weiterhin Möglichkeiten gesucht diese Marken zu optimieren oder neue, bessere zu erstellen.

⁴In Kombination mit anderen Techniken zur Bestimmung der Orientierung.

Auch an der Hochschule für Angewandte Wissenschaften Hamburg (HAW) wird dem Problem der kamerabasierten autonomen Navigation, und der dafür nötigen Posenbestimmung, von mobilen Systemen im Indoor-Bereich, innerhalb des Masterstudiengangs Informatik nachgegangen (siehe [Manske, 2008b; Manske und Jost, 2008; Manske, 2008a]). Bei dem dort genutzten Verfahren, welches auf einem photogrammetrischen 3D-Kameramodell basiert, ist es notwendig, dass zur Posenbestimmung mindestens 3 Marken im Bild zu identifizieren sind. Zur Unterstützung dieses Masterprojektes beschäftigte sich die Bachelorarbeit von Matthias Wippich [Wippich, 2008a] mit dem Thema der Markengestaltung und -erkennung. Aus dieser Arbeit ging ein eigener Markentyp inkl. Erkennungsalgorithmus hervor, der u.a. als Grundlage dieser Arbeit dient.

1.2. Ziel und Aufgabenstellung der Arbeit

Grundsätzliches Ziel dieser Arbeit ist es Marken und entsprechende Algorithmen zur Erkennung dieser in den Einzelbildern eines digitalen Videosignals prototypisch zu entwickeln. Diese Marken sollen in dem unter [Manske, 2008b; Manske und Jost, 2008; Manske, 2008a] beschriebenen System zur kamerabasierten Präzisionsnavigation zum Einsatz kommen.

Als Szenario für diese Arbeit wird ein auf dem Boden fahrendes Fahrzeug angenommen, dessen Kamera ebenfalls in Bodennähe befestigt ist und deren Pose durch das o.g. Verfahren bestimmt werden soll. Dieses Szenario wird ausgewählt, da es die wohl gängigste Verwendung eines solchen Systems darstellt und sich mit den an der HAW verfügbaren Ressourcen am einfachsten realisieren lässt. Generell sind die Erkenntnisse und Entwicklungen dieser Arbeit aber (u.U. mit Veränderungen) auch auf andere Szenarien übertragbar. Eine Teilaufgabe dieser Arbeit besteht darin den von Matthias Wippich entwickelten Markentyp genauer zu untersuchen und festzustellen, ob er für die Navigation und Posenbestimmung, für das o.g. Szenario geeignet ist. Ggf. müssen Veränderungen am Erkennungsalgorithmus oder der Marke selbst vorgenommen werden. Des Weiteren wird das Verfahren der Erkennung der Marken so ausgebaut werden, dass eine Verfolgung (Tracking) der Marken über mehrere Bilder möglich wird. Mittels der Verfolgung kann eine Vorhersage der Position der Marken im nächsten Bild erreicht werden, was in den meisten Fällen zu einer Verringerung der Laufzeit des Erkennungsalgorithmus führt.

Da sich diese Marken zur Zeit nur von einer Seite aus erkennen lassen, wird sich ein Schwerpunkt dieser Arbeit mit der Aufgabe beschäftigen einen Markentyp zu entwickeln, der eine Erkennung von mehr Standpunkten im Raum aus möglich macht. Dies kann z.B. in großen Räumen sehr nützlich sein, um Marken zu sparen. Außerdem befindet sich in dieser Arbeit noch ein Konzept zu einer konzentrischen Ringmarke, deren Notwendigkeit erst während der parallel abgeschlossenen Masterarbeit von Nico Manske (siehe [Manske, 2008a]) festgestellt wurde und deren praktische Umsetzung zeitlich nicht mehr erfolgen konnte. Des Weiteren gibt diese Arbeit einen groben Überblick über das Gebiet der codierten Messmarken mit einigen

Beispielen aus der Literatur.

Die zu implementierenden Algorithmen sollen in einem Modul (Macro) für die Testumgebung Impressario verfügbar sein, um sie einfach mit den anderen zu dem Gesamtsystem gehörenden Komponenten verbinden zu können (siehe Abb. 1.2).

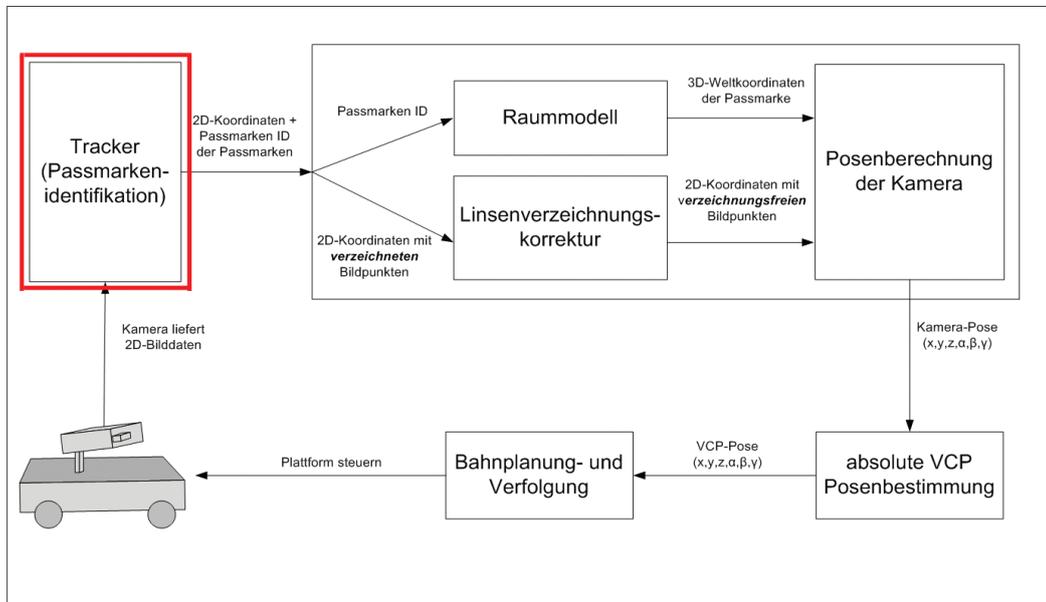


Abbildung 1.2. – Überblick über die Systemmodule⁵. Der rot markierte Bereich zeigt die Aufgabe dieser Arbeit. Abbildung aus [Manske, 2008a]

1.3. Aufbau der Arbeit

Nach der in diesem Kapitel vorgestellten Motivation und Zielsetzung dieser Arbeit liefert Kapitel 2 einen Überblick über allgemeine Grundlagen, die zum Verständnis dieser Arbeit notwendig sind.

Das Kapitel 3 soll die Frage klären, was codierte Messmarken überhaupt sind und außerdem verschiedene Anwendungsbereiche solcher Marken vorstellen. Dazu wird es einige exemplarische Beispiele konkreter Marken geben, um die in dieser Arbeit genutzten Marken und Verfahren einordnen zu können.

⁵Der Begriff Tracker, der von Manske verwendet wird, bezeichnet hier das Gesamtmodul, welches in dieser Arbeit entsteht und nicht das Teilmodul mit dem Namen Tracker, welches u.a. in Kapitel 4 beschrieben wird.

Kapitel 4 befasst sich mit dem Gebiet der Verfolgung und wie dieses zur Erhöhung der Erkennungsgeschwindigkeit in dieser Arbeit eingesetzt werden kann.

In Kapitel 5 werden die von Matthias Wippich entwickelten Marken analysiert und weiterentwickelt.

Das Kapitel 6 beschäftigt sich mit der neu entwickelten Zylindermarke und deren Erkennung.

In Kapitel 7 ein Konzept für eine einfache kreisförmige Barcodemarke zu finden, wohingegen in Kapitel 8 ein Fazit dieser Arbeit gezogen und ein Ausblick für mögliche Erweiterungen gegeben wird.

Im sich anschließenden Anhang A befindet sich eine Beschreibung der Daten, die sich auf der beiliegenden DVD-ROM befinden.

Hinweis: Die meisten in dieser Arbeit zu Zwecken der Illustration verwendeten Grafiken befinden sich im PDF-Dokument dieser Arbeit als Vektorgrafik. Deshalb ist es möglich bei den u.U. manchmal etwas kleinen Darstellungen in der Papier-Version die stufenlos vergrößerbare Version auf der in Anhang A beschriebenen DVD-ROM zu nutzen.

Der letztmalige Abruf der im Text aufgeführten URLs fand am 26.11.2008 statt.

2. Allgemeine Grundlagen

2.1. Digitale Bildverarbeitung

Ein digitales Bild ist rasterförmig als zweidimensionale Matrix organisiert, wobei die einzelnen Felder dieser Matrix als Bildpunkt (Pixel) bezeichnet werden. Der Zugriff auf diese Matrix erfolgt über die Angabe der Position des gewünschten Bildpunktes. Hierbei liegt ein Koordinatensystem zugrunde, dessen Ursprung sich in der oberen linken Ecke des Bildes befindet. Die x-Achse beschreibt somit die Spalten der Matrix und die y-Achse entsprechend die Reihen, wobei das Pixel am Koordinatenursprung entweder mit der Position (0,0) oder (1,1) angegeben wird. Zur Bestimmung des Pixelmittelpunktes (wichtig für die Arbeit mit Subpixelgenauigkeit) muss zur Spalten- und Zeilenkomponente der Pixelposition jeweils der Wert 0,5 hinzuaddiert werden. Häufig wird in der Bildverarbeitung bei dem Zugriff auf ein Pixel des Bildes, anders als gewohnt, die y-Koordinate (Zeilenkomponente) des Bildpunktes zuerst angegeben.

Der an der Stelle eines bestimmten Pixels gespeicherte Wert in der Matrix wird Pixelwert genannt und bestimmt z.B. bei Farbbildern die Farbe, die dieses Pixels im Bild besitzt. Die Codierung und Größe dieses Wertes hängen dabei vom verwendeten Farbmodell ab.

2.1.1. Grauwertbilder

Grauwertbilder sind Bilder, deren Inhalt ausschließlich aus Grauwerten (Helligkeit/Intensität eines Bildpunktes) besteht und die damit farzunabhängig sind. Die Grauwerte liegen normalerweise im Intervall von 0 (Schwarz) bis 255 (Weiß), was 256 verschiedene Werte (8 Bit) ergibt, mit denen die abzubildende Lichtmenge beschrieben werden kann. Grauwertbilder können entweder per Formel aus Farbbildern berechnet werden, oder direkt durch eine entsprechende Kamera aufgenommen werden. Diese Kameras teilen die Lichtmenge, mit der ein Pixel des Sensors während der Belichtungszeit bestrahlt wurde, in das beschriebene Intervall ein.

2.1.2. Kanten

Eine Grauwertkante in der digitalen Bildverarbeitung bezeichnet das lokale Konzept einer Diskontinuität im Verlauf der Grauwerte des Bildes. Es ist sinnvoll die Diskontinuität von Grauwerten zur Segmentierung von Bildern einzusetzen, da diese meist mehr Informationen beinhaltet, als eine kontinuierliche Fläche und auch unser Auge diese stärker gewichtet (siehe [Nischwitz und Haberäcker, 2004]).

In der Literatur sind verschiedene Kantenmodelle beschrieben (siehe z.B. [Gonzales und Woods, 2002]), wobei diese zumeist stark vereinfachen und in der Praxis mit Rauschen u.ä. kaum auftreten. Es lohnt sich aber dennoch das Modell der Rampenkante näher zu betrachten, um die grundsätzlichen Eigenschaften zu verstehen. Diese Rampe entsteht bei einem Übergang zwischen zwei Regionen. Die Länge richtet sich nach der Steigung, welche durch die Unschärfe bestimmt wird, was dazu führt, dass unscharfe Bilder breitere Kanten haben als scharfe Bilder. Betrachtet man nun eine eindimensionale horizontale Linie (Grauwertprofil) am Übergang zwischen zwei Bereichen, ergibt sich das Bild aus Abbildung 2.1. Es ist zu sehen, dass die erste Ableitung des Grauwertprofils an den Stellen der Steigung positiv ist. Die zweite Ableitung hingegen ist positiv beim Eintritt in die Steigung und negativ beim Austritt aus der Steigung, bei einem Übergang von hell nach dunkel wäre dies genau umgekehrt. Die bisherigen Betrachtungen bezogen sich auf eine einfache eindimensionale Bildzeile, das Konzept ist aber auch auf die Zweidimensionalität des Bildes anwendbar, wenn ein Grauwertprofil senkrecht zur Kantenrichtung gewählt wird (siehe [Gonzales und Woods, 2002]).

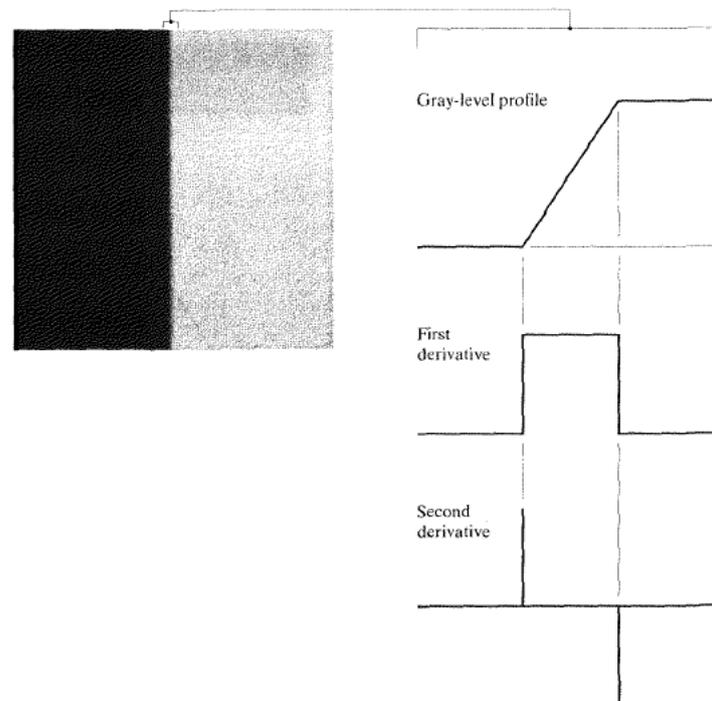


Abbildung 2.1. – Rampenkante sowie ein horizontales Grauwertprofil dieser Kante und die zugehörige 1. und 2. Ableitung. Abbildung aus [Gonzales und Woods, 2002]

2.2. Kamera und Beleuchtung

2.2.1. Verwendete Kamera

Für die Videoaufnahmen im Rahmen dieser Arbeit wurde eine Monochrom⁶-Kamera der Firma The Imaging Source Europe GmbH mit Objektiv der Firma Computar verwendet. (siehe Tab. 2.1) Diese Kamera (mit Objektiv) dient auch als Referenz für alle Berechnungen, die z.B. eine Brennweite voraussetzen.

Modell	DMK 41BF02
Video-Formate @ Bildrate	1280 x 960 Y800 @ 15; 7,5; 3,75 fps
Sensor	Sony ICX205AL CCD
Typ	Progressive Scan
Format	1/2"
Ausgabeauflösung in Pixel	H: 1280 V: 960
Pixel-Größe	H: 4,65 μm V: 4,65 μm
Belichtungszeit	$\frac{1}{10000}$ bis 30s
Anschluss	Firewire
Brennweite des Objektivs	5 mm
Bildwinkel	73.3°

Tabelle 2.1. – Einige Daten der verwendeten Kamera.

2.2.2. Abbildungsmaßstab und Linsengleichung

Möchte man wissen, wie groß ein Objekt im Bild abgebildet wird, ist hierzu ist die Einführung des Abbildungsmaßstabs (auch lateraler Abbildungsmaßstab) und der Linsengleichung (auch Gauß'sche Abbildungsgleichung) notwendig. Mittels der Linsengleichung kann die Größe der Abbildung auf dem Sensor bestimmt werden (Abkürzungen siehe Abb. 2.2). Der Abbildungsmaßstab β bezeichnet das Verhältnis der Größe eines Objekts in der Bildebene und dessen realer Größe (siehe [Jähne, 2005]):

$$\beta = \frac{G}{O} = \frac{b}{o} \quad (2.1)$$

Die Linsengleichung wird beschrieben durch (siehe [Jähne, 2005]):

$$\frac{1}{f} = \frac{1}{b} + \frac{1}{o} \quad (2.2)$$

⁶Schwarzweiß im Sinne von Grauwerten

Die Linsengleichung kann nun nach b aufgelöst und in 2.1 eingesetzt werden, was durch umstellen nach B die gesuchte Größe der Abbildung des Objekts auf dem Sensor ergibt:

$$B = \frac{f}{o-f} O \quad (2.3)$$

Wird B nun noch durch die Größe eines Pixels auf dem Sensor geteilt, ist die Anzahl der belegten Pixel für eine Strecke der Länge O aus einer Entfernung o mit der Brennweite b berechenbar. Es ist anzumerken, dass die Linsengleichung idealisiert, was bei den verwendeten Kameras nicht gegeben ist. dennoch erlauben mittels dieser Gleichung berechnete Werte in dieser Arbeit eine ausreichende Beurteilung der Situation.

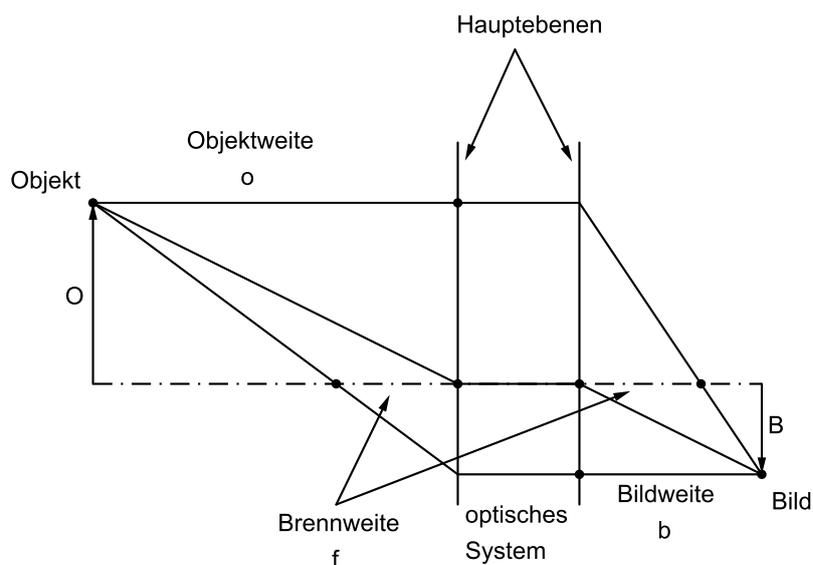


Abbildung 2.2. – Bezeichnung der Größen des Abbildungsmaßstabs und der Linsengleichung.

2.2.3. Fokussierung

Mit Fokussierung wird in der Fotografie die Anpassung der Einstellungen des Objektivs beschrieben, die eine scharfe Abbildung des aufzunehmenden entfernten Objekts erlaubt. Fokussieren im Sinne der Physik beschreibt das Treffen von Strahlen in einem Punkt. Dies trifft auch auf die Abbildung durch ein optisches System zu. Eine scharfe Abbildung eines Objektes wird erreicht, wenn sich die von einem Punkt dieses Objektes ausgesendeten Lichtstrahlen in einem Punkt auf der Bildebene der Kamera treffen. Zu diesem Zweck ist es notwendig den Abstand zwischen der bildseitigen Hauptebene des Objektivs und der Bildebene (Bildweite b) entsprechend einzustellen. Nicht immer ist es aber erwünscht ein exakt scharfes Bild zu haben: Bei einer subpixelgenauen Kantenbestimmung kann es von Vorteil sein, wenn das Bild ein wenig unscharf ist, da die Kantenposition so einfacher bestimmt werden kann.

Der Linsengleichung (siehe Gleichung 2.2) ist zu entnehmen, dass es für jede Entfernung eines Objekts zur Linse (Objektweite o) nur eine Einstellung der Bildweite geben kann, in der es zu einer scharfen Abbildung kommt. Für eine unendliche Entfernung o wäre dies die Objektivbrennweite f . Wird also der Abstand der Bildebene von der bildseitigen Hauptebene nicht exakt auf die Entfernung ausgerichtet, kommt es zu einer Unschärfe. Die Schärfentiefe definiert einen Bereich, in dem die Aufnahme dennoch als ausreichend scharf angesehen werden kann. Dieser Bereich wird durch den tolerierten Radius der entstehenden Unschärfekreise (Zerstreuungskreise) angegeben. Unschärfekreise entstehen dadurch, dass der zwischen bildseitiger Hauptebene und Bildebene, durch die von einem Objektpunkt kommenden Strahlen aufgespannte Kegel nicht exakt mit seiner Spitze auf die Bildebene trifft. Konzeptbedingt kann die Größe der Sensorelemente in der digitalen Fotografie ein Anhaltspunkt für den tolerablen Radius, da kleinere Elemente hardwareseitig nicht aufgelöst werden können (siehe [Jähne, 2005]). Die Größe des Bereichs der Schärfentiefe bei einer bestimmten Bildweite kann durch die Blende beeinflusst werden, eine große Blendenzahl führt zu einem kleinen Bereich und eine kleine Blendenzahl zu einem großen Bereich, der scharf abgebildet werden kann. In einem realen optischen System spielen neben der Fokussierung durch ein Verschieben der Bildebene und die durch Anpassung der Blende veränderbare Schärfentiefe u.a. auch noch die Beugung der Lichtstrahlen an der Blende und auftretende Abbildungsfehler eine Rolle, die hier aber nicht näher betrachtet werden.

2.2.4. Belichtung

Im Zeitraum der Belichtung entsteht das eigentliche Bild, da hier die Lichtstrahlen des aufzunehmenden Objekts auf den Sensor der Kamera treffen. Hierfür sind die Faktoren Empfindlichkeit, Blende und Zeit entscheidend. Die Empfindlichkeit bezieht sich auf den Sensor und gibt an wie lichtempfindlich dieser ist. Bei einer festgelegten Empfindlichkeit sind für die Aufnahme selbst dann die Blendenzahl und die Belichtungszeit entscheidend. Die Blende regelt die Menge an Licht, die auf den Sensor fällt, wohingegen die Belichtungszeit den Zeitraum festlegt für den der Sensor dem Licht ausgesetzt wird. Es ist klar, dass es hier äquivalenten Einstellungen geben muss: Lange Belichtungszeit große Blendenzahl (kleine Öffnung) entspricht einer kurzen Belichtungszeit und kleiner Blendenzahl (große Öffnung). Ein Bild ist zumeist dann korrekt belichtet, wenn alle abzubildenden Objekte den gleichen Kontrast und die Helligkeit aufweisen, wie in der Realität durch die Betrachtung mit dem bloßen Auge.

Die Belichtungszeit ist für diese Arbeit dahingehend wichtig, da um keine Bewegungsunschärfe bei hohen Geschwindigkeiten des Fahrzeugs zu erhalten, die Belichtungszeit möglichst niedrig sein sollte.

2.2.5. Retroreflektierende Folie

Zur Signalisierung wird bei allen während dieser Arbeit (weiter-)entwickelten Markentypen selbstklebende retroreflektierende Folie (Reflexfolie) der Firma 3M eingesetzt. Genauer handelt es sich um die Folientypen SOLAS Grade (siehe [3M, 2007]) und Series 3210 (siehe [3M, 2004a]). Die hinterspiegelten Glaskugeln, die in das Folienmaterial eingearbeitet sind, bewirken, dass einfallendes Licht größtenteils in die Richtung reflektiert wird, aus der es kam (siehe Abb. 2.4). Dieses Verhalten wird für die Arbeit dazu genutzt, um die Marken im Bild hervorzuheben. In Kombination mit Infrarotbeleuchtung und ausgewählter Belichtung, kann dies soweit führen, dass fast ausschließlich nur noch Marken im Bild zu sehen sind, was die Erkennung u.a. beschleunigt. Anwendung finden diese Folien sonst häufig zu Sicherheitszwecken, sei es auf Seenotrettungsmitteln wie Rettungswesten und Rettungsringen, oder im Straßenverkehr auf Warnwesten und Verkehrsschildern. Die Folien sind meistens in mehreren Farben verfügbar, für diese Arbeit wird ausschließlich Folie genutzt, die weiß wirkt, wenn sie angestrahlt wird. Bei der Arbeit mit Reflexfolie sind im wesentlichen drei verschiedene Begriffe wichtig:

- **Spezifischer Rückstrahlwert (Koeffizient der Retroreflexion), R' od. R_A :** Der spezifische Rückstrahlwert ist das Maß für die Retroreflexion der entsprechenden Folie. Er ist definiert als Leuchtdichte L am Punkt des Beobachters im Verhältnis zur Beleuchtungsstärke E_{\perp} , $R' = \frac{L}{E_{\perp}}$ (siehe [Bundesministerium für Verkehr, Bau und Stadtentwicklung, 1998]). Als Einheit ergibt sich Candela pro Quadratmeter pro Lux, $\frac{\text{cd}}{\text{m}^2 \cdot \text{lx}}$. Dieser Wert ist den technischen Begleitdokumenten der Folien zu entnehmen und ist je nach Kombination von Einfallswinkel und Beobachtungswinkel verschieden.
- **Einfallswinkel (Beleuchtungswinkel):** Der Einfallswinkel des Lichts beschreibt den Winkel, der sich zwischen dem einfallenden Lichtstrahl und der Normalen der Foliensoberfläche ergibt (siehe [Bundesministerium für Verkehr, Bau und Stadtentwicklung, 1998]). Normalerweise hat R' bei einem Einfallswinkel von 0° den höchsten Wert und verringert sich bei steigendem Einfallswinkel.
- **Beobachtungswinkel** Der Beobachtungswinkel ist der Winkel, der sich zwischen dem einfallenden Lichtstrahl und dem beobachteten Lichtstrahl ergibt (siehe [Bundesministerium für Verkehr, Bau und Stadtentwicklung, 1998]). Dieser Winkel darf nur sehr klein sein, da das von einem einfallenden Lichtstrahl reflektierte Licht fast ausschließlich in einem Kegel mit ca. 6° großem Öffnungswinkel reflektiert wird (siehe [3M, 2004b]).

Eine grafische Darstellung der beiden Winkel ist in Abbildung 2.5 zu finden.

Beispielwerte für SOLAS Grade Folie:

$$R' = 180 \text{ bei Einfallswinkel } 5^\circ \text{ und Beobachtungswinkel } 0,1^\circ$$

$$R' = 2,0 \text{ bei Einfallswinkel } 30^\circ \text{ und Beobachtungswinkel } 2^\circ$$



Abbildung 2.3. – Retroreflektierende Folie vom Typ SOLAS Grade. Abbildung aus [3M, 2008]

Beispielwerte für Series 3210 Folie:

$R' = 50$ bei Einfallswinkel 5° und Beobachtungswinkel $0,33^\circ$

$R' = 2,5$ bei Einfallswinkel 30° und Beobachtungswinkel 2°

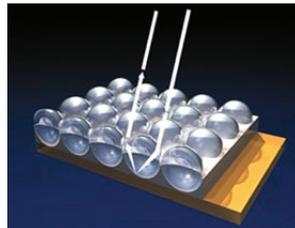


Abbildung 2.4. – Hinterspiegelte Glaskugeln, die einfallendes Licht in die Richtung aus der es kam reflektieren. Abbildung aus [3M, 2008]

2.2.6. Infrarotbeleuchtung

Zum Zwecke der Beleuchtung⁷ der Reflexfolie sollte für die im Rahmen dieser Arbeit (weiter-)entwickelten Markentypen Infrarotstrahlung eingesetzt werden. Dies hat in Kombination mit Infrarotfilter und einer manuell gewählten Blendenzahl, Helligkeits- und Kontrastanpassung (Gain) und Belichtungszeit den Vorteil, dass im Bild die von den Infrarotstrahlern bestrahlten Markenteile aus Reflexfolie als helle Bereiche sichtbar sind und die ganze Umgebung fast schwarz abgebildet wird. außerdem wird durch eine eigene Lichtquelle das Problem

⁷Die Begriffe Beleuchtung und Licht werden innerhalb dieser Arbeit auch für Infrarotstrahlung genutzt. Dies ist hier erwähnt, weil die Begriffe oftmals ausschließlich mit der für den Menschen sichtbaren Strahlung assoziiert werden.

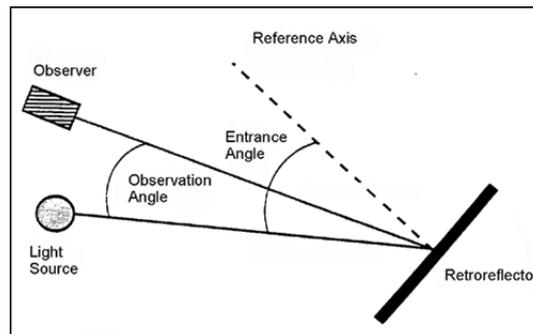


Abbildung 2.5. – Illustration der erläuterten Begriffe, Einfallswinkel (Entrance Angle) und Beobachtungswinkel (Observation Angle) bei retroreflektierender Folie. Abbildung aus [Federal Highway Administration, 2001]

der unterschiedlichen Beleuchtungssituationen umgangen.

Als Infrarotstrahlung wird die Strahlung im Bereich von 780nm bis 1 mm bezeichnet (siehe [Hornberg, 2006] und Abb. 2.6), dieser Bereich schließt direkt oberhalb an den für den Menschen sichtbaren Bereich der Strahlung an. Für die Tests während dieser Arbeit wurden 3 Infrarotscheinwerfer (siehe Tab. 2.2) der Firma MICROLIGHT Infrared Illuminators genutzt, deren Strahlung sich im Bereich um die 880nm bewegt (nahes Infrarot). Das sich diese Strahlung ausserhalb des sichtbaren Lichts befindet, hat auch den Vorteil, dass niemand in der Umgebung gestört wird (u.a. die Laserklasse des Strahlers muss beachtet werden) und somit quasi unsichtbar gemessen werden kann. Es ist aber besonders bei Überwachungszwecken zu beachten, dass bei Strahlung im unteren nahen Infrarotbereich der Scheinwerfer, für das menschliche Auge wahrnehmbar, schwach rot leuchtet.

Mit der verwendeten Kamera ist es möglich Infrarotstrahlung abzubilden, da der Sensor auch in diesem Bereich (wenn auch weniger) empfindlich ist. Das genutzte Infrarotfilter hilft dabei sichtbares Licht auszufiltern. Dennoch ist das Bild oftmals nicht komplett schwarz, da auch in Aussenräumen durch die Sonne, oder durch Lichtquellen wie Glühbirnen und Leuchtstoffröhren Infrarotstrahlung abgestrahlt wird. Mittels einer manuell gewählten Kombination aus Blendenzahl, Helligkeits- und Kontrastanpassung (Gain) und Belichtungszeit⁸ kann aber dafür gesorgt werden, dass fast ausschließlich das von der stark reflektierenden Reflexfolie (oder von anderen stark reflektierenden Materialien wie Stahl oder Spiegeln) reflektierte Infrarotlicht im Bild zu sehen ist. Das die Einstellungen manuell vorgenommen werden sollten hat den Sinn, dass durch den Wegfall anderer Lichtquellen in dem Bild nur einige helle Bereiche der Reflexfolie zu sehen sind und damit für die Kamera als stark unterbelichtet gilt. Die Kamera versucht dann z.B. durch eine längere Belichtungszeit und eine Helligkeits- und Kontrastanpassung das Bild zu verbessern. Dies führt dazu, dass das Bild stark rauscht, die Reflexfolie überbelichtet ist und auch wieder viele andere Objekte im Bild sichtbar werden (siehe Abb. 2.7).

⁸Genauere Informationen zu den verwendeten Kombinationen sind den jeweiligen Testkonfigurationen zu entnehmen.

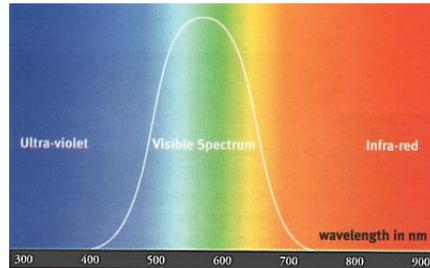


Abbildung 2.6. – Bereich der Infrarotstrahlung im Vergleich zum sichtbaren Licht im elektromagnetischen Spektrum. Abbildung aus [Perret Optic, 2008]



Abbildung 2.7. – Auswirkungen der manuellen Wahl der Kameraeinstellungen. Links: Manuelle Einstellung der Kamera. Rechts: Automatische durch die Kamera durchgeführte Einstellungen.

Typ	IR-56/20-880	IR-56/(60-40)-880	IR-30/80-880
Wellenlänge in nm	880 ± 20	880 ± 20	880 ± 20
Beleuchtungsweite m	40	18	5,5
Beleuchtungswinkel in ° (Öffnungswinkel des Kegels)	20 ± 2	$60 \pm 6 \times 40 \pm 4$	80 ± 6
Leistungsaufnahme in Watt	$\approx 7,2$	$\approx 7,2$	$\approx 12,0$

Tabelle 2.2. – Einige Daten der verwendeten Infrarotscheinwerfer.

2.3. Verwendete Hard- und Software

2.3.1. Testrechner

Alle im Rahmen dieser Arbeit nötigen Software-Tests und Auswertungen von Testvideos wurden auf dem in Tabelle 2.3 beschriebenen Rechner durchgeführt. Aus diesem Grunde beziehen sich alle angegebenen Laufzeiten immer auf diesen Testrechner.

Prozessor	AMD Athlon64 3200+ (2000 MHz)
Arbeitsspeicher	1,5 GB DDR-RAM
Betriebssystem	Windows XP SP2

Tabelle 2.3. – Verwendeter Rechner.

2.3.2. LTI-Lib und Impresario

Microsoft Visual Studio .NET 2003 diente als Entwicklungsumgebung für die erstellte Software, da einerseits die von Matthias Wippich (siehe [Wippich, 2008a]) erstellten Algorithmen in C++ vorliegen und auch die anderen Module des Gesamtsystems in dieser Sprache erstellt wurden/werden. Die Verwendung dieser alten Version hängt mit der Inkompatibilität zu neueren Versionen der Entwicklungsumgebung des nachfolgend erläuterten Programms Impresario und der LTI-Lib zusammen.

Die LTI-Lib ist eine am Lehrstuhl für Technische Informatik der RWTH Aachen entwickelte freie Bildverarbeitungsbibliothek für C++. Die Bibliothek kann unter <http://ltilib.sourceforge.net> kostenlos bezogen werden. Für die erstellten Macros wurde die Bibliothek in der Version 1.9 verwendet, da diese schon kompiliert Impresario beiliegt.

Impresario ist ein an der Rheinisch-Westfälischen Technischen Hochschule Aachen (RWTH Aachen) von Lars Libuda entwickeltes Programm zum einfachen Prototyping von Bildverarbeitungssystemen über eine graphische Nutzeroberfläche (GUI). Die aktuelle Version 1.3 stammt aus dem Jahr 2005 und konnte früher auf der Internetseite des Lehrstuhls für Technische Informatik (LTI) heruntergeladen werden. Seit dieser Lehrstuhl nicht mehr existiert gibt es keine kostenfreie Möglichkeit dieses Programm zu beziehen, weshalb sich eine Kopie auf dem beiliegenden Datenträger befindet. Bei der Nutzung können aus Macros, die Ein- und Ausgänge besitzen Prozessgraphen erstellt werden, in dem der Ausgang eines Macros mit dem Eingang eines anderen Macros verbunden wird (siehe Abb. 2.8). Für viele Anwendungen, wie dem Lesen von Bild- oder Videodateien, Kantenerkennung, Umwandlung von Bildtypen, ... sind schon vorgefertigte Macros vorhanden. Wenn ein Prozessgraph verbunden wurde kann das System sofort interaktiv getestet werden, d.h. man kann alle Ausgänge der einzelnen Macros

abgreifen und anzeigen lassen, als auch im Betrieb die Parameter der Macros verändern. Die vorgefertigten Macros nutzen Funktionen der LTI-Lib, das Programm selbst kann aber auch mit anderen Bibliotheken genutzt werden. Es ist möglich Impresario um eigene Macros zu ergänzen, was im Rahmen dieser Arbeit auch genutzt wurde. Diese Macros werden als DLL-Datei in C++ erstellt und dann automatisch bei Programmstart geladen.

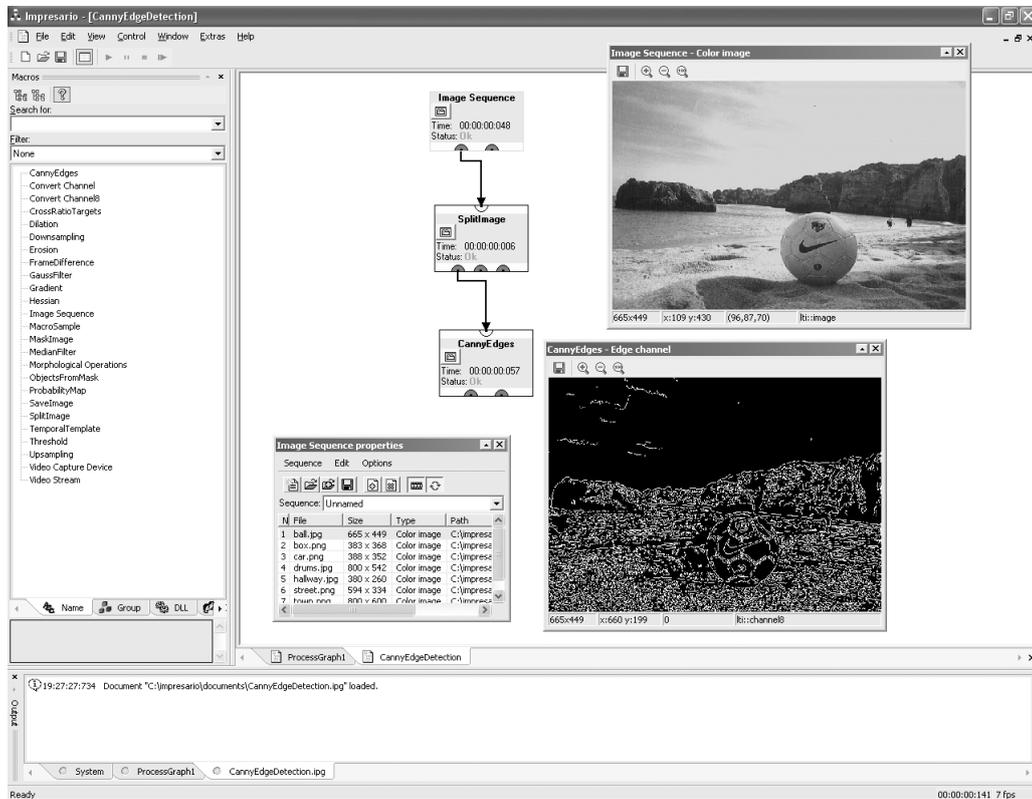


Abbildung 2.8. – Impresario mit Prozessgraph für eine Canny-Edge Erkennung.

3. Codierte Messmarken

3.1. Einführung

Messmarken tauchen in vielen unterschiedlichen Bereichen, die sich mit Messungen verschiedenster Art beschäftigen, auf. Beispiele dafür sind die (Nahbereichs-) Photogrammetrie, die industrielle Bildverarbeitung, die Robotik, Augmented Reality, . . . Sie dienen in Bildern dem Zweck der künstlichen Signalisierung von Messpunkten und werden verwendet, um Messungen einfacher automatisieren zu können, als dies mit natürlichen (meist schwerer zu identifizierenden) markanten Punkten des Messobjekts möglich ist. Um die Identifizierung auch in komplexen Bildern zu ermöglichen werden meist einfache Strukturen (Kreise o.ä.) verwendet, die sich genügend von der Umgebungstextur abheben müssen. Der eigentliche Messpunkt wird durch ein geeignetes Merkmal der Marke, wie z.B. dem Schwerpunkt eines Kreises oder dem Schnittpunkt mehrerer Geraden symbolisiert. Ein bekannter Anwendungsfall solcher Marken ist der in der TV-Werbung häufig anzutreffende Crash-Test bei Autos (siehe Abb. 3.1), bei dem Messmarken verwendet werden, die durch Aufkleben am Fahrzeug befestigt werden.



Abbildung 3.1. – Crash-Test Szene mit runden schwarz-gelben Messmarken. Abbildung aus [MSN, 2008]

Für eine Anwendung wie die der präzisen und sicheren Posenbestimmung reicht eine einfache Messmarke nicht aus, da die Marke in diesem Fall eine bekannte Position im Raum darstellen soll und somit eindeutig identifizierbar sein muss. Für Fälle dieser Art gibt es neben uncodierten Marken auch codierte Messmarken, die ein Muster mit einer codierten eindeutigen Identifikationsnummer (ID) tragen. Auch in der Vermessung mit mehreren Kameras/Kamerastandorten helfen diese Marken das sogenannte Korrespondenzproblem zu lösen, also das Finden des selben Punktes in aus verschiedenen Positionen aufgenommenen Bildern. Codierte Messmarken sind aber nicht auf den Bereich der Vermessung im weitesten Sinne beschränkt, auch zur Identifizierung von Objekten in der Stückgutsortierung oder im Postverkehr können solche Marken verwendet werden. Wie in Abbildung 3.2 zu sehen ist, kann der zur Codierung der ID genutzte Code verschiedenst gewählt werden (Barcode-Muster verschiedener Dimensionen, Koordinatenmuster, Schrift, ...).

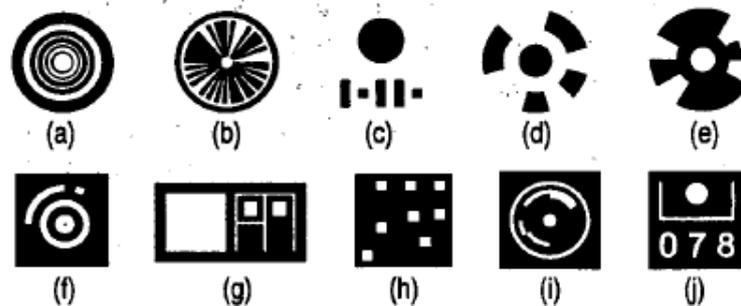


Abbildung 3.2. – Verschiedene Typen codierter Messmarken. (a)Russo,1972; (b)Trenkler,1981; (c)Wong,1988; (d)Schneider,1991; (e)Knobloch,1992; (f)van den Heuvel,1992; (g)Homainejad,1995; (h)Fraser,1997; (i)[Niederöst und Maas, 1997]; (j)Caesar,1997. Abbildung aus [Ahn u. a., 1999]

Da nicht in jedem Bereich, in dem Messmarken auftauchen, der Ausdruck “codierte Messmarke” verwendet wird, hier eine kleine (unvollständige) Übersicht bezüglich anderer Bezeichnungen:

- (codierte) Landmarke: Hauptsächlich in der Navigation/Posenbestimmung, mit der Unterscheidung künstlich eingebracht oder natürlich. (z.B. [Ahn u. a., 1999])
- codierte Zielmarke (coded target) (z.B. [Niederöst und Maas, 1997])
- Positionsmarke (z.B. [Wippich, 2008a])
- Passmarke: Hauptsächlich im Druckwesen, um bei Ungenauigkeiten des Drucks nachsteuern zu können.
- coded fiducial mark/marker bzw. tag: Häufig im Bereich der Augmented Reality. (z.B. [Fiala, 2005b])

Im weiteren Verlauf dieser Arbeit wird zur Vereinfachung nur noch der Begriff "Marke" verwendet.

Um für möglichst viele Anwendungsfälle einsetzbar zu sein, sollten die Marken u.A. folgende Kriterien erfüllen (siehe [Luhmann, 2000]):

- **Invarianz gegenüber Translation, Rotation, Skalierung und Scherung** (affine Transformationen) (siehe Abb. 3.3 und Tab. 3.1): Translations- und Rotationsinvarianz sind für den Anwender sehr von Vorteil, da er nicht darauf achten muss, dass die Marken in einer bestimmten Position oder Drehung im Bild erscheinen. Am einfachsten ist die Rotationsinvarianz durch kreisförmige Marken mit entsprechendem Code zu lösen. Die Skalierungsinvarianz stellt sicher, dass die Marken auch aus unterschiedlichen Entfernungen zur Kamera identifizierbar sind. In der Praxis ist die Skalierungsinvarianz natürlich nur begrenzt gegeben, da die Marken immer eine bestimmte Größe im Bild nicht unterschreiten dürfen, um eine genaue Messung zu ermöglichen.
- **Invarianz gegenüber perspektivischer Verzerrung** (siehe Abb. 3.3 und Tab. 3.1): Um auch perspektivisch verzerrte Marken erkennen zu können, ist außerdem zu fordern, dass die Marken gegenüber projektiven Transformationen (affine Transformationen + Perspektive) invariant sind. Oftmals reicht es laut Niederöst und Maas in [Niederöst und Maas, 1997] aus, dies durch Approximation mit affinen Transformationen sicherzustellen. Es gibt aber auch Marken, deren Code bewusst mit invarianten Merkmalen aufgebaut wird (siehe Kapitel 5).
- **Robustes Verfahren zur Decodierung der ID**: Die ID der Marke sollte auch unter schlechten Bedingungen (z.B. ungünstige Lichtverhältnisse) und bei Teilverdeckung zu decodieren sein. Hierzu ist es ggf. sinnvoll eine Fehlererkennung (z.B. Redundanz der Informationen) und eine Fehlerkorrektur (z.B. Forward Error Correction (FEC) (siehe z.B. [Fiala, 2005a]) einzubauen.
- **Präzise Bestimmbarkeit des Messpunktes**: Die präzise Bestimmbarkeit des Messpunktes (bzw. dessen Zentrums) hat erheblichen Einfluss auf die Genauigkeit der Messung, weshalb auf eine möglichst einfache Definition zu achten ist.
- **Möglichst große Anzahl an verschiedenen Marken**: Um auch für große Messobjekte oder die Navigation viele verschiedene Marken zur Verfügung zu haben, sollten mit dem genutzten Code möglichst viele IDs codierbar sein.
- **Kurze Verarbeitungszeit**: Enorm wichtig, um z.B. bei der Navigation in kurzen Zeitabständen präzise die aktuelle Position bestimmen zu können. Aber auch bei der 3D Messung möchte der Anwender nicht minutenlang auf ein Ergebnis warten. Deshalb ist es wichtig, dass die Marke möglichst schon so gestaltet wird, dass sie einfach und schnell erkannt werden kann.
- **Geringe Größe der Marke**: Große Marken sind zwar noch aus größeren Entfernungen zu identifizieren, verdecken aber u.U. auch schon einen großen Teil des Gegenstands,

der gemessen werden soll. Bei der Navigation/Posenbestimmung, möchte man die Marken auch so dezent wie möglich platzieren können. Denkbar sind hierfür z.B. Muster, die in Tapeten (siehe [Saito u. a., 2007]) o.ä. integriert sind.

- **Niedrige Herstellungskosten:** Es ist ein Unterschied, ob Marken mit einem handelsüblichen Laserdrucker selbst erstellt werden können, oder einer Extraanfertigung bedürfen. Die schon vorgestellte retroreflektierende Folie ist ein u.U. erheblicher Kostenfaktor.

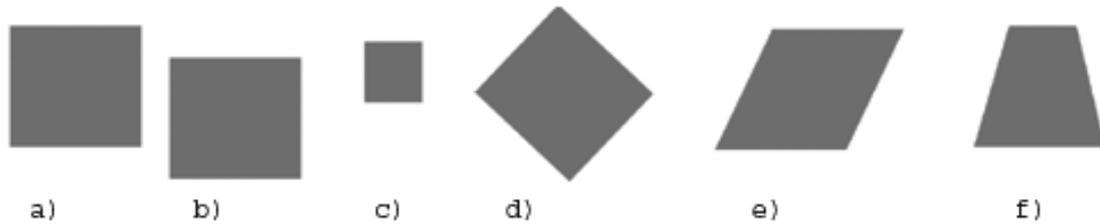


Abbildung 3.3. – b) bis e) zeigen Affine Transformationen: a) Original, b) Translation, c) Skalierung, d) Rotation, e) Scherung. f) zeigt eine perspektivische Transformation (auch die Transformationen b)-e) sind im projektiven Raum definiert)

<i>Transformation</i> \ <i>Invariant</i>	Fläche	Länge	Winkel	Parallelität	Doppelverhältnis ¹
Euklidisch	X	X	X	X	X
Ähnlichkeit		X	X	X	X
Affin				X	X
Projektiv					X

¹ Für eine genauere Erklärung des Doppelverhältnisses siehe Kapitel 5.1.1

Tabelle 3.1. – Einige geometrische Eigenschaften, die invariant unter den angegebenen Transformationen sind. (teilw. nach [Hartley und Zisserman, 2003])

Im folgenden Abschnitt sollen nun einige ausgewählte Markentypen exemplarisch vorgestellt werden. Wobei mit Absicht auf die Vorstellung von sogenannten “general purpose” Marken, wie DataMatrix⁹ o.ä. verzichtet wird, weil sich diese nur sehr begrenzt zu denen in dieser Arbeit behandelten Messzwecken eignen, da sie zur Identifizierung einen Großteil der Bildfläche einnehmen müssen. Die Beispiele erfüllen die meisten der genannten Anforderungen an codierte Messmarken, wobei von den jeweiligen Autoren die Invarianz gegenüber projektiven Transformationen nicht weiter beachtet wird, sondern sich oftmals ausschließlich auf affine Transformationen beschränkt wird. Trotzdem sind alle diese Marken auch bis zu einem gewissen Grad bei perspektivischer Verzerrung zu erkennen, wie den Bildern in den jeweiligen Arbeiten zu entnehmen ist.

⁹Findet z. B. Verwendung bei der STAMPIT Frankiersoftware der Deutschen Post AG.

3.2. Vorstellung verschiedener Markentypen

3.2.1. Kreisförmige Marken

Kreisförmige Marken eignen sich aufgrund des Vorteils, dass sie den Messpunkt schon flächenhaft verkörpern (meist noch durch eine spezielle Markierung in der Mitte der Marke unterstützt), sowie radialsymmetrisch und, bei der Wahl eines geeigneten Codes, rotationsinvariant sind, sehr gut zu Messzwecken. In der Praxis kann dieser Mittelpunkt deshalb per Schwerpunktberechnung, aber auch mit Ausgleichsrechnung, nach der Kleinsten-Quadrate-Anpassung und/oder verschiedenen anderen Operatoren/Methoden, die sich mit der Bestimmung von Ellipsenparametern beschäftigen, bestimmt werden. Ellipsenparameter sind von Bedeutung, da sich eine kreisförmige Marke unter perspektivischer Verzerrung in einer Ellipsenform in der Bildebene abbildet wird. Für nähere Informationen zu den Methoden der Parameterbestimmung bei Ellipsen sei z.B. auf [Luhmann, 2000] verwiesen. Bei der ellipsenfö-

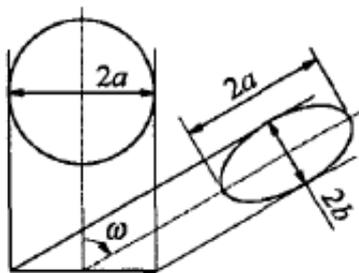


Abbildung 3.4. – Abbildung eines Kreises auf eine Ellipse, bei perspektivischer Verzerrung (hier Parallelprojektion). Abbildung aus [Ahn u. a., 1999]

migen Verzerrung muss beachtet werden, dass der (dann z. B. durch o.g. Methoden bestimmte) Ellipsenmittelpunkt nicht mit dem abgebildeten Mittelpunkt der Marke übereinstimmt. Dieses Verhalten wird als Abbildungsexzentrizität e bezeichnet und in Abbildung 3.5 illustriert. Diese Exzentrizität hängt u.a. von der Markengröße ab, weshalb die Größe des Durchmessers des Mittelpunktes (bzw. des Messpunktes) bei den hier betrachteten digitalen Bildern nicht zu groß werden, aber auch eine 5 Pixel-Grenze nicht unterschreiten sollte. Luhmann empfiehlt in [Luhmann, 2000] einen Durchmesser von 5 – 15 Pixel.

Grundsätzlich gibt es bei den kreisförmigen Mustern verschiedene Arten, die zur Codierung der IDs eingesetzt werden. Oftmals sind die Codeelemente hierbei in Ringform (Ringcode) konzentrisch um den Kreismittelpunkt angeordnet.

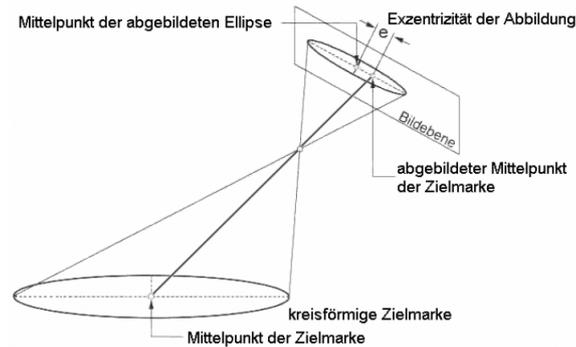


Abbildung 3.5. – Darstellung der Abbildungsexzentrizität e . Abbildung aus [Luhmann, 2000]

3.2.1.1. Ringcode

Ein Beispiel für Marken mit Ringcode ist das von Niederöst und Maas in [Niederöst und Maas, 1997] vorgestellte Muster.



Abbildung 3.6. – Marken mit 15-Bit Ringcode. Abbildung aus [Niederöst und Maas, 1997]

Der Messpunkt der Marke wird durch den in der Mitte befindlichen weißen Punkt abgebildet und von zwei, an den Stellen, wo Bits gesetzt sind, weißen Codekreisen umschlossen. Die Außenmarkierung der Marke bilden zwei so genannte Prüfkreise; der Weiße stellt die äußere Markierung der Marke dar und der innere (schwarz) dient zur Abgrenzung der Codes vom Außenkreis. Alle Kreise haben die selbe Breite.

Die Codekreise sind in insgesamt 18 Segmente aufgeteilt (12 außen und 6 im Inneren). Jedes dieser Segmente stellt ein Bit dar, wobei die Kombination von zwei benachbarten Außensegmenten und dem sich anschließenden Segment auf dem inneren Kreis nur einmal vorkommen darf, da es das Startsymbol des Codes ist. Die Bitwertigkeit ergibt sich aus der Position des Segments im Uhrzeigersinn vom Startsymbol aus gesehen. Die eindeutige ID lässt sich durch aufsummieren der Bitwerte aller gesetzten Bits und das hinzuaddieren einer 1 errechnen. Durch die Einschränkungen mit dem Startsymbol ergeben sich hier insgesamt 36768 verschiedene Marken.

Die Erkennung dieser Marken lässt sich in 4 Schritte einteilen:

- Detektion der Marken im Gesamtbild durch Binarisierung und Bestimmung des Schwerpunktes, der bei zwei Objekten (Mess- und Prüfkreis) gleich sein muss.
- Kleinste-Quadrate Methode zum genaueren Berechnen der jeweiligen Ellipsenparameter.
- Grauwertsampling zum Bestimmen der jeweiligen Codeparameter.
- Decodierung der IDs.

In der Praxis wird häufig eine von der Firma AICON entwickelte retroreflektierende Marke (siehe Abb. 3.7) eingesetzt, deren Code aus einem einfachen linearen Barcode in Ringform besteht. Diese Marke soll hier kurz vorgestellt werden, da sich Chen u.a. in [Chen u. a., 2007] mit einer alternativen Erkennung dieser Marken auseinandergesetzt haben, die auch auf andere kreisförmige Marken übertragbar ist.



Abbildung 3.7. – Marke der Firma AICON mit der ID 105. Abbildung aus [Chen u. a., 2007]

Der Mittelpunkt der Marke dient auch hier als Messpunkt und wird durch einen weißen Kreis signalisiert. Die Codierung der ID erfolgt hier im Gegensatz zu der vorher betrachteten Marke von Niederöst und Maas in nur einem Kreis. Dieser lässt sich je nach Konfiguration in 8 bis 14 Einheiten aufteilen (siehe Abb. 3.8), die jeweils ein Bit des Codes repräsentieren (1=Weiß;0=Schwarz). Dadurch lassen sich 25 bis 561 verschiedene IDs codieren. Da der Code kein explizites Startelement vorsieht wird zur Decodierung die Sequenz der Einheiten im Uhrzeigersinn aufgenommen. In Abbildung 3.7, wäre dies z.B. die Sequenz "w,s,w,w,s,w,s,w,s,s,s" (w=Weiß;s=Schwarz), unter der Voraussetzung man fängt bei der 12 Uhr Stellung an aufzunehmen. Wenn man diese Sequenz nun aus der Bitdarstellung in eine Dezimalzahl konvertiert, erhält man den Wert 2896 statt 105. Somit ist zu sehen, dass die Reihenfolge des Auslesens des Codes Einfluss auf die decodierte ID hat. Aus diesem Grunde zählt als eindeutige ID immer die niedrigste Dezimalzahl, die aus allen möglichen Sequenzen ermittelbar ist. Dies erklärt auch die im Vergleich eher kleine Anzahl von möglichen IDs, obwohl 14 Nutz-Bits zur Verfügung stehen.

Der interessante Ansatz den Chen u.a. wählen, ist die Erkennung der Marken mit Hilfe

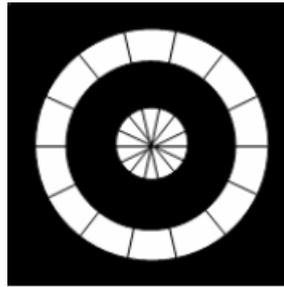


Abbildung 3.8. – Marke der Firma AICON mit 14 Einheiten. Abbildung aus [Chen u. a., 2007]

von Morphologischen Operatoren, was ihrer Meinung nach einfacher zu handhaben ist, als andere Methoden zur Erkennung. Die Morphologischen Operatoren entstammen der morphologischen Mathematik, die wiederum aus der Mengentheorie hervorging und eigentlich nur die Anwendung für Binärbilder beschreibt, aber auch auf Grauwertbilder übertragbar ist.

Die Erkennung läuft nun im wesentlichen folgendermaßen ab:

- Binarisierung des Originalbildes (Grauwertbild) anhand eines globalen Schwellwertes.
- Dilation dient dazu, die einzelnen Elemente einer Marke (Messpunkt und umliegende Code-Elemente) zu einem Objekt zu vergrößern. Dies ist der Punkt, mit dem die Güte des Verfahrens steht und fällt, weshalb Chen u.a. hier auch eine automatische Bestimmung anstreben. Zu wenige Dilationen führen dazu, dass nicht alle Elemente einer Marke zusammengebracht werden. Zu viele Dilationen haben zur Folge, dass eng benachbarte Marken vereinigt werden. Chen u.a. gehen aber davon aus, dass die Anzahl sehr viel leichter zu bestimmen ist, als die Parameter bei anderen Verfahren, da es sich hierbei um eine ganze Zahl handelt.
- Die nächsten beiden Schritte beschäftigen sich jetzt nur noch mit dem Auffüllen kleinerer Lücken (Closing) und der Extraktion der gefundenen Marken aus dem Originalbild.
- Der Rest läuft ähnlich wie bei Niederöst und Maas ab.

Die nun betrachteten Marken von Ahn u.a., die z.B. in [Ahn u. a., 1999] und [Ahn und Oberdorfer, 1998] vorgestellt werden, unterscheiden sich von den vorangegangenen dadurch, dass es sich bei dem verwendeten Ringcode um ein Punktmuster handelt. Die Punkte liegen in verschiedenen Winkeln auf konzentrischen Ringen um den Mittelpunkt der Marke. Der Messpunkt der Marke befindet sich im weiß umrandeten Mittelpunkt des Kreises. Der Code besteht aus kleineren (ebenfalls weißen) Kreisen, die den Mittelpunkt auf verschiedenen Umlaufbahnen umkreisen. Diese kleineren Kreise liegen alle auf einem großen schwarzen Kreis (Hintergrundkreis), um einen möglichst guten Kontrast zu erhalten. (siehe Abb. 3.9). Über die Art des in Abbildung 3.9 verwendeten Codes ist nichts dokumentiert. Zur Unterstützung der Fehlererkennung schlagen Ahn u.a. auch die Einführung einer Minimalbedingung für die Anzahl von Punkten vor (die Maximalbedingung ergibt sich aus der verwendeten Bitlänge) und

die Einführung eines Parity-Bits vor. Die Marken sind durch ihre Beschaffenheit sehr einfach

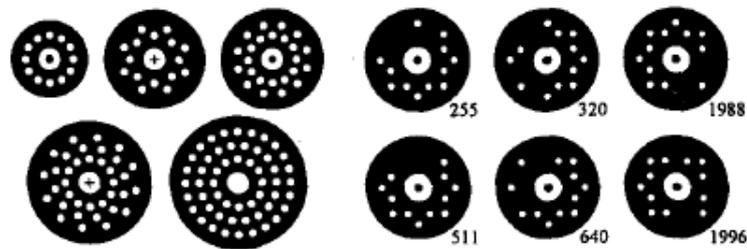


Abbildung 3.9. – Marken von Ahn u.a. mit verschiedenen Codelängen (links; 12-,16-,24-,36- und 54 Bit Code) und einige Beispielen (rechts) Abbildung aus [Ahn u. a., 1999]

auf verschiedene Entfernungen anpassbar, da ausschließlich der Durchmesser der Elemente erhöht werden muss. Zusätzlich kann, um z. B. die Messung im Nahbereich genauer zu gestalten, ein Kreuz oder kleiner Punkt in der Mitte des Messpunktes angebracht werden (siehe Abb. 3.9).

Der Ablauf der Erkennung ähnelt dem der vorangegangenen Marken:

- Binarisierung
- Entdeckung aller geeigneten Objekte
- Erkennung des Messpunktes und der Codepunkte
- Berechnung der Ellipsenparameter
- Neu-Ausrichtung des Codepunkte um den Messpunkt (Rücktransformation)
- Decodieren der ID

3.2.1.2. Fourier Tags

Fourier Tags (siehe Abb. 3.10) sind sehr spezielle Marken, deren IDs nicht wie bei den obigen Marken durch die Anordnung geometrischer Formen codiert werden. Es wird die diskreten Fourier-Transformation genutzt, um die eindeutigen IDs im Frequenzspektrum des Bildes codieren. Beschrieben sind diese von Sattar u.a. entwickelten Marken in [Sattar u. a., 2007]. Die Entwicklung wird damit begründet, dass alle bekannten Marken, ob nun mit oder ohne Fehlerkorrektur, ab einer bestimmten Entfernung nicht mehr identifizierbar sind, obwohl sie eigentlich noch im Bild sichtbar sind. Sattar u.a. nutzen Marken u.a. zur Kommunikation zwischen Tauchern und mobilen Unterwasser-Robotern, wobei es hierbei wichtig ist, dass die Marke auch über eine große Entfernung und unter schlechten Sichtbedingungen im Wasser erkannt werden können. Es kommt hierbei aber nicht unbedingt auf die korrekte Identifizierung der Marke an, sondern auch die Information, dass sich an einer Stelle im Bild eine Marke befindet ist für diese Anwendung schon ausreichend. Aus diesem Grunde haben Sattar u.a. die Fourier Tags

entwickelt, da diese auch aus großen Entfernungen erkennbar sind, wenn auch nicht immer eindeutig identifizierbar. Die Identifizierbarkeit nimmt mit der Entfernung ab, da die niederwertigen Bits des Codes in den hohen Frequenzen des Frequenzspektrums abgebildet werden, wohingegen die höherwertigen Bits die niedrigsten Frequenzen nutzen (siehe Abb. 3.11). Die

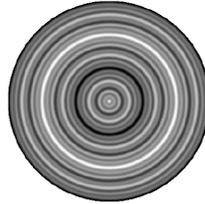


Abbildung 3.10. – Fourier Tag mit der ID 210. Abbildung aus [Sattar u. a., 2007]

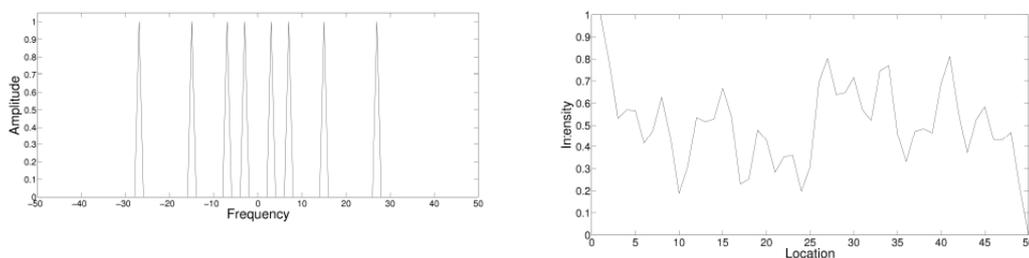


Abbildung 3.11. – Fourier Tag mit der ID 210. Links: Binärdarstellung im Frequenzraum (11010010). Rechts: Binärdarstellung als Intensitäten eines Grauwertbildes. Abbildung aus [Sattar u. a., 2007]

einzelnen Bits der IDs werden in gleichgroßen Abständen im reellen Frequenzspektrum symmetrisch um den Ursprung untergebracht. Zur Konstruktion der Marke wird der Code mit Hilfe der Formeln der zeit-diskreten Fourier-Transformation in eine diskrete Funktion verwandelt:

$$X(e^{i\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-i\omega n} \quad (3.1)$$

$$x[n] = \frac{1}{2\pi} \int_{2\pi} X(e^{i\omega}) \cdot e^{i\omega n} d\omega \quad (3.2)$$

Wobei Gleichung 3.1 von Sattar u.a. als Spektrum von Gleichung 3.2 bezeichnen. Durch die Festlegung auf reelle Werte werden ausschließlich Kosinusanteile genutzt, dies führt später dazu, dass weniger eindeutige IDs codiert werden können, ist aber einfacher zu decodieren, da nur noch auf Signalspitzen geachtet werden muss und keine Phasenverschiebung geschieht. Um auch im Bild eine reelle Funktion zu erhalten wird wie oben schon genannt eine Spiegelung am Ursprung vorgenommen, um eine symmetrische Funktion zu erhalten. Das diskrete Signal aus Gleichung 3.2 wird nun als Bildzeile definiert und ausgehend von einem Mittelpunkt rotierend um diesen aufgebracht, um die kreisförmige Marke zu erhalten.

Zur Erkennung dieser Marken in einem Bild wird zunächst eine Suche nach allen kreisförmigen Regionen durchgeführt und deren Mittelpunkt bestimmt. Die exakte Bestimmung des Mittelpunktes spielt neben Aspekten der Messgenauigkeit auch zur Decodierung der ID eine entscheidende Rolle, da von diesem Mittelpunkt ausgehend in beide Richtungen eine Bildzeile auf Symmetrie der Graustufen ausgewertet wird. Ist die Symmetrie, bestimmt durch festgelegte Schwellwerte, gegeben kann davon ausgegangen werden, dass es sich hierbei um ein Fourier Tag handelt. Dieser Bildzeile kann nun einer Fast Fourier-Transformation (FFT) unterzogen werden, um das Frequenzspektrum zu erhalten und die entsprechende ID festzustellen (siehe Abb. 3.12). Sollte sich die Marke nun zu weit weg befinden sind nicht mehr alle Bits lesbar, da die sehr hohen Frequenzen (niederwertige Bits) durch die geringen Auflösung verloren gegangen sind.

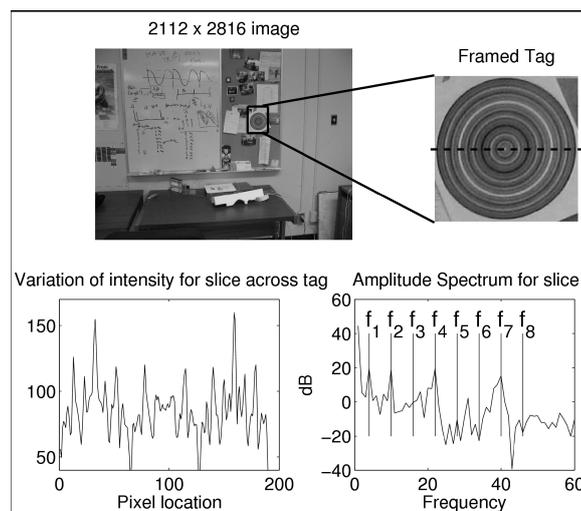


Abbildung 3.12. – Oben Erkennung des Fourier Tag aus $3m$ Entfernung. Die unteren beiden Bilder zeigen die Abbildung der Bildzeile im Grauwertbereich (links) und im Frequenzspektrum (rechts). f_1 bis f_8 zeigen die Stellen der einzelnen Bits der ID. Abbildung aus [Sattar u. a., 2007]

3.2.2. Quadratische Marken

In diesem Abschnitt soll auf die besonders in der Augmented Reality (AR) genutzten viereckigen (zumeist quadratischen) Marken eingegangen und 2 von ihnen beispielhaft vorgestellt werden. In der Augmented Reality werden diese Marken z.B. zum Zwecke der Einblendung von künstlichen Elementen in ein reales Umfeld (Bild, Video, ...) (siehe Abb. 3.13), oder zur Signalisierung auf neuartigen Eingabegeräten genutzt (siehe Abb. 3.14). Diese Marken sind aber nicht auf dieses Einsatzfeld beschränkt, sondern eignen sich auch für so ziemlich alle anderen Einsatzgebiete codierter Messmarken. Sie werden fast immer in einem kompletten Paket, bestehend aus Marken und Erkennungssoftware (häufig inkl. Verfolgung u.ä.), angeboten und

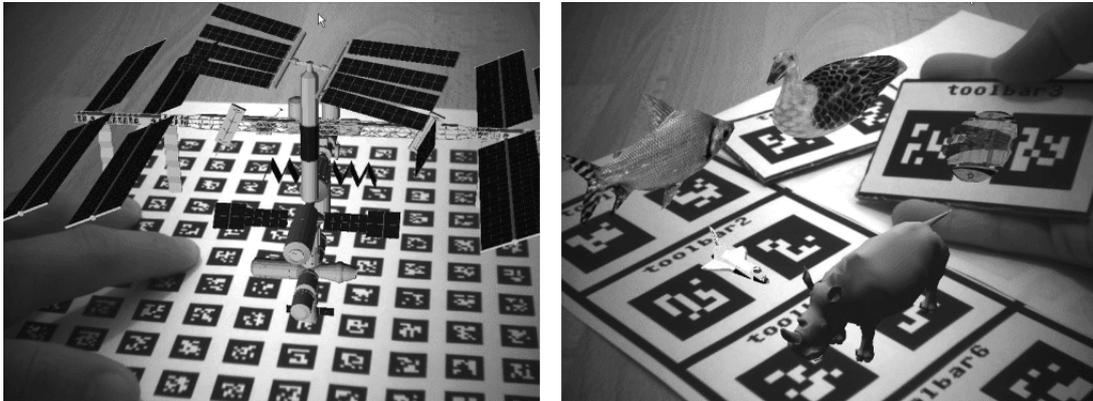


Abbildung 3.13. – Beispiele für die Nutzung von Marken in der AR. (hier: ARTag Magic Lens).
Abbildung aus [ARTag, 2008]

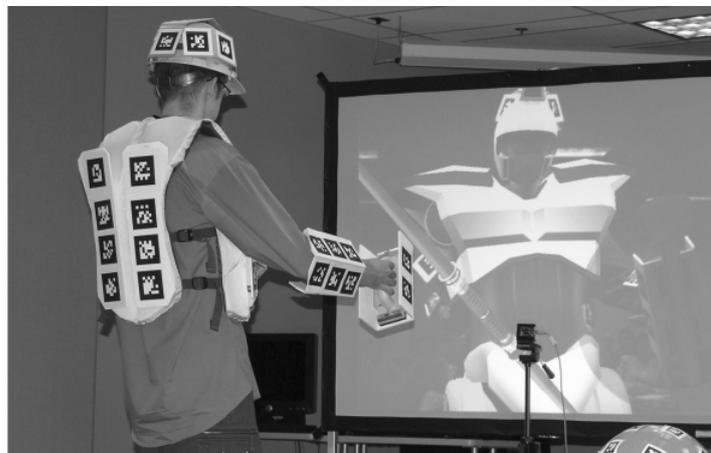


Abbildung 3.14. – Beispiel für ein interaktives Eingabegerät. (hier: ARTag Magic Mirror).
Abbildung aus [ARTag, 2008]



Abbildung 3.15. – Beispiele für ARToolkit-Marken. Abbildung aus [Fiala, 2005b]

einige sind auch im Internet frei verfügbar. Die Marken zeichnen sich dadurch aus, dass sie bereits 4 Messpunkte auf einer Marke besitzen, ihre Ecken. Damit ist es möglich schon anhand einer Marke die komplette Orientierung der Kamera zu berechnen. Diese Methode ist aber u.U ungenau, weshalb es bei allen hier vorgestellten Marken auch möglich ist den Schnittpunkt der Diagonalen des Vierecks als Messpunkt der Marke zu nutzen. Dieser Ansatz entspricht dann dem Mittelpunkt des Kreises bei den kreisförmigen Marken und hat laut Fiala und Shu in [Fiala und Shu, 2005] eine höhere Genauigkeit als die Nutzung der vier Eckpunkte, da sich so die Fehler bei der Bestimmung der Eckpunkte ausgleichen. Mit Verzicht auf die Nutzung der vier Eckpunkte als Messpunkte ist aber auch nur noch ein einziger Korrespondenzpunkt pro Marke möglich.

3.2.2.1. ARToolkit

ARToolkit ist eines der bekanntesten Marker-Systeme für Augmented Reality und im Internet für die nicht-kommerziellen Einsatz frei verfügbar¹⁰. In diesem Paket sind auch schon Eigenschaften wie die Verfolgung von einer oder mehrerer Marken eingebaut und müssen nicht extra entworfen werden.

Die Marken bestehen aus einem quadratischen schwarzen Rahmen, der sich auf einer weißen Fläche befindet. Innerhalb dieses Rahmens befindet sich das einzigartige Muster jeder Marke, welches über einen Vergleich/Korrelation mit in einer Datenbank befindlichen Vorlagen abgeglichen und so in eine ID decodiert wird (template matching). In der Gestaltung des Musters ist der Anwender komplett frei, auch Buchstaben, Zahlen und ähnliches können genutzt werden. Einige Beispiele sind in Abbildung 3.15 zu finden. Die ersten 4 Marken dieser Abbildung sind im Auslieferungszustand vordefinierte Marken, die letzten beiden stammen aus einer Arbeit von Owen u.a. [Owen u. a., 2002] und basieren auf Mustern, die mit Hilfe der diskreten Kosinus-Transformation erstellt wurden, um eines der größten Probleme, Fehlidentifikationen von Markern und der Erkennung anderer Objekte als Marker zu vermindern. Beide Fehler sind durch das Prinzip des Vergleichs mit einer Vorlage, die keinen bestimmten Anforderungen unterliegt, sehr problematisch, besonders wenn die Beleuchtung der Marken nicht besonders gut ist. Die Erkennung der Marken läuft nach einem sehr einfachen Verfahren ab:

¹⁰<http://www.hitl.washington.edu/artoolkit/>

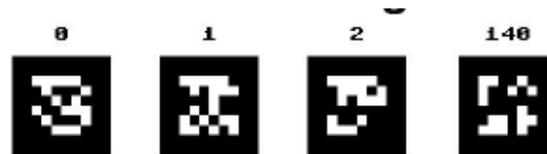


Abbildung 3.16. – Beispiele für ARTag-Marken. Abbildung aus [Fiala, 2005b]

- Binarisierung wird mit nur einem globalen Schwellwert, was das Verfahren sehr Beleuchtungsabhängig macht.
- Extraktion zusammenhängender Bereiche als Konturen und die subpixelgenaue Bestimmung der 4 Ecken.
- Template Matching des Innenraums der Kontur mit den zum Vergleich stehenden entsprechend skalierten Mustern.

Damit ist die Erkennung abgeschlossen und unter guten Lichtverhältnissen sind so auch gute Ergebnisse zu erreichen (siehe [Fiala, 2005b]).

3.2.2.2. ARTag

ARTag (beschrieben in [Fiala, 2005a]) wurde durch ARToolkit inspiriert und versucht dessen Probleme zu vermeiden. So wird zwar die Struktur der Marke beibehalten, aber die Decodierung findet nicht mehr über einen Abgleich mit Vergleichsmustern statt, sondern mittels Methoden aus der digitalen Signalverarbeitung (Cyclical Redundancy Check und Forward Error Correction).

In Abbildung 3.16 sind einige Beispiele für diese Marken zu finden. Die Marken bestehen aus einem 2 Einheiten breiten Rahmen, der entweder schwarz oder weiß sein kann und einem Innenbereich von $6 \cdot 6 = 36$ Einheiten. Insgesamt hat die Marke damit eine Größe von $10 \cdot 10$ Einheiten. Die codierte ID der Marke wird auf die 36 Einheiten des Innenbereichs aufgebracht, wobei es sich hierbei um einen 2D Barcode handelt. D.h. schwarze Einheit entspricht Bit gesetzt und weiße Einheit entspricht Bit nicht gesetzt. Insgesamt ließe sich so eine Codelänge von 36 Bit realisieren, welcher aber durch Redundanzen auf 10 Nutzbits zusammenschrumpft. Um die Decodierung auch bei Teilverdeckung und bei schlechten Lichtverhältnissen zu gewährleisten wird der Code durch Mittel aus der digitalen Signalverarbeitung aufgebaut. außerdem kann so auf einen bestimmten Startpunkt für die Decodierung verzichtet werden. Fiala beschreibt es besonders schön mit den Worten "The system can be abstractly described as a communication system, where a 10-bit ID is attempted to be sent through a medium of image capture to be received by the ARTag vision software." [Fiala, 2005a] Insgesamt stehen so 2002 verschiedene Marken zur Verfügung.

Die Erkennung der Marken erfolgt über eine Kantensuche im Bild, anhand derer quadratische Objekte gesucht werden. Der Innenbereich der Marken wird, wie bei ARToolkit, auf die Größe von $10 \cdot 10$ Einheiten gesampelt. Anders als bei ARToolkit erfolgt aber kein Vergleich mit anderen Mustern, sondern anhand eines lokalen Schwellwertes werden die einzelnen Bits extrahiert und die entsprechende ID decodiert. ARTag stellt somit ein ARToolkit ähnelndes System zur Verfügung, das aber keine Probleme mit der Fehlidentifikation von Markern hat.

Auch wenn es in der Literatur nicht extra erwähnt wurde, besitzen alle hier vorgestellten Marken eine gewisse Skalierungsinvarianz wohingegen die Rotationsinvarianz nur durch ein Vergleich/Auslesen des Codes mit/aus allen 4 möglichen Richtungen durch die jeweilige Erkennungssoftware realisiert wird. Über perspektivische Verzerrungen ist in der Literatur nichts vermerkt.

3.2.3. Selbstähnliche Marken

Nachdem bei den beiden vorherigen Beispielen (bis auf Fourier Tags 3.2.1.2) auf Marken eingegangen wurde, bei denen es während des Entwurfs eher auf eine Auswahl möglichst gut erkennbarer geometrischer Objekte ankam, soll hier nun ein komplett anderer Ansatz vorgestellt werden. Bei diesem Ansatz handelt es sich um eine rein mathematische Herangehensweise, die Marken die dabei herauskommen arbeiten mit der sogenannten Selbstähnlichkeit und wurden von Briggs und Scharstein (siehe [Briggs u. a., 2000]) zur Navigation eines autonomen mobilen Roboters entwickelt. Die Selbstähnlichkeit ist deshalb so gut für die Bildverarbeitung geeignet, da sie skalierungsinvariant ist und die Marke deshalb aus verschiedenen Entfernungen gut gefunden werden kann. Abbildung 3.17 zeigt eine dieser Marken mit selbstähnlichem Merkmal (links) und einem Barcode (rechts), der der eindeutigen Identifizierung dient. Auf den Barcode soll hier nicht weiter eingegangen werden, da hier auch andere Konzepte, z.B. mit explizitem Messpunkt (welcher hier fehlt), möglich sind.

Zur Definition der Selbstähnlichkeit wählen Scharstein und Briggs die p -Selbstähnlichkeit, definiert als:

$$f(x) = f(px) \quad \forall x > 0 \quad (3.3)$$

p beschreibt den Skalierungsfaktor und ist definiert für $0 < p < 1$. Die Gleichung 3.3 sagt aus, dass wenn eine Funktion p -selbstähnlich ist, sie um den Faktor p skaliert gleich ist. Das einfachste Beispiel für eine solche Funktion ist die konstante Funktion $f(x) = 3$. Auch für jeden Wert p liefert sie 3 als Ergebnis. Ein Beispiel für eine etwas kompliziertere Funktion ist in Abbildung 3.18 zu sehen, dort handelt es sich um eine Sinus-Funktion mit $p = \frac{2}{3}$. Zu beachten ist weiterhin, dass eine Funktion, die p -selbstähnlich ist auch gleichzeitig p^x -selbstähnlich für $x = 2, 3, 4, \dots$ ist. Besonders anschaulich ist die Eigenschaft der Selbstähnlichkeit in [Braun, 2005] beschrieben, dort heißt es: "Man kann sich die Selbstähnlichkeit veranschaulichen, wenn man die Funktion als elastische Feder betrachtet. Drückt man die im Intervall $[0, 1]$ definierte Feder auf das Intervall $[0, p]$ zusammen, so sieht die skalierte Funktion dort genauso

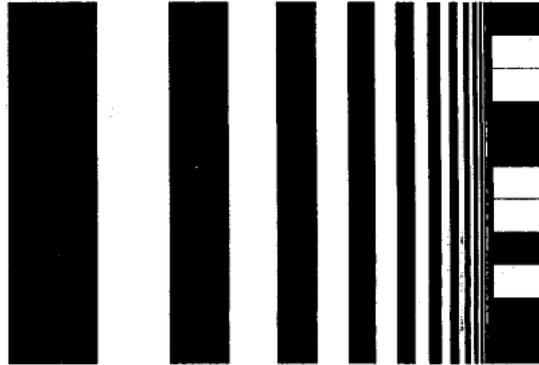


Abbildung 3.17. – Selbstähnliche Marke von Briggs und Scharstein mit Barcode am rechten Rand. Abbildung aus [Briggs u. a., 2000]

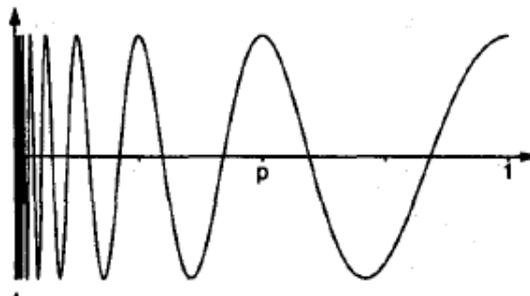


Abbildung 3.18. – Selbstähnliche Funktion mit $p = \frac{2}{3}$. Abbildung aus [Briggs u. a., 2000]

aus wie die unskalierte - sie ist nach einer Verkleinerung um p sich selbstähnlich."

Um diese Funktionen nun in einem Bild anhand ihres Musters (siehe Abb. 3.17) erkennen zu können stellen Briggs und Scharstein folgende Überlegung: Ein p -selbstähnliches Muster kann erkannt werden, indem das im Bild gefundene Muster mit einem p -skalierten Muster verglichen wird. Mathematisch wird dies durch folgendes Integral beschrieben:

$$d_{p,w}(f) = \frac{1}{w} \int |f(x) - f(px)| dx \quad (3.4)$$

Wenn $d_{p,w}$ den Wert 0 annimmt ist die Funktion $f(x)$ im Bereich w p -selbstähnlich. w bezeichnet dabei die Fensterbreite eines Suchfensters ist, welches über eine 1D Muster geschoben wird, weshalb der Wert von $d_{p,w}$ auch nur eine Aussage über die lokale Selbstähnlichkeit der Funktion $f(x)$ in diesem Bereich gibt. Soll jetzt ein Muster im Bild untersucht werden, wird ein Suchfenster über das Muster geschoben und wiederholt $d_{p,w}(f)$ berechnet, was dazu führt, dass dort, wo $d_{p,w}(f)$ den niedrigsten Wert besitzt die Selbstähnlichkeit am Größten ist.

Als Generatorfunktion für einer Muster einer Marke entscheiden Briggs und Scharstein für die Rechteckfunktion: $f(x) = [2(x - \lfloor x \rfloor)]$ (siehe Marke in Abb. 3.17), da diese Funktion nur für p - und p^x für $x = 1, 2, 3, \dots$ selbstähnlich ist und nicht für \sqrt{p} . Damit können bei einem Vergleich die o.g. konstanten Funktionen ausgeschlossen werden. Diese Generatorfunktion wird nun durch $Marke(x, y) = f(x)$ in ein 2D Muster überführt. Gleichungen zur Suche dieses selbstähnlichen Musters sehen wie folgt aus:

$$d_{\sqrt{p},w}(f) = 1 - \frac{1}{w} \int |f(x) - f(\sqrt{p}x)| dx \quad (3.5)$$

$$m_p(f) = d_{\sqrt{p},w}(f) - d_{p,w}(f) \quad (3.6)$$

$m_p(f)$ liefert nur für den Fall, dass das gefundene Muster ausschließlich p -selbstähnlich ist den Wert 1 und in allen anderen Fälle einen Wert um 0.

Der Algorithmus zur Erkennung leitet sich direkt aus den Gleichungen 3.5 und 3.6 ab, wobei zur Diskretisierung, mit einer selbst gewählten Schrittweite, die Integrale wegfallen und stattdessen Summen gebildet werden. Grauwerte für eventuell auftretende nicht ganzzahlige Koordinaten müssen durch Interpolation bestimmt werden.:

```
Schleife über alle Bildzeilen y
  Schleife über alle Bildspalten x
    Berechnung von m
    Wenn m==1 Marke gefunden
```

$m == 1$ wird in der Praxis selten auftreten, weshalb es ratsam ist nach lokalen Maxima zu suchen. Ist ein solches Maximum gefunden findet von dort aus beginnend die Suche nach dem Barcode statt.

Die fehlende Rotationsinvarianz der Marken, die ausgelöst wird durch den Erkennungsalgorithmus, der ausschließlich selbstähnliche Muster von links nach rechts erkennt und somit auf Rotationen von maximal 45° ausgelegt ist, lässt sich durch das dreimalige drehen des Bildes um 90° beheben. Dadurch würde sich allerdings die Anzahl der nötigen Operationen des Algorithmus vervierfachen.

Die Beschränkung auf die alleinige Beachtung von affinen Transformationen zeigt sich, wenn man sich die Testbilder in [Briggs u. a., 2000] anschaut, als nicht hinderlich, da auch perspektivisch verzerrte Marken gefunden werden können.

Eine Erweiterung der Idee von Briggs und Scharstein ist in [Negre u. a., 2007] zu finden, wo Negre u.a. eine selbstähnliche Marke für die Navigation eines Autonomous Underwater Vehicles (AUVs) nutzen. Zu diesem Zwecke haben Negre u.a. eine kreisrunde selbstähnliche Marke entwickelt (siehe Abb. 3.19), die damit auch die bei Briggs und Scharstein nicht vorhandene Rotationsinvarianz sicherstellt. Die Marke ist nicht codiert, dies ließe sich aber nachrüsten, sondern besteht aus einem großen kreisförmigen selbstähnlichem Muster, auf dem sich zwei kleinere befinden. Die Mittelpunkte der Kreise, die als Messpunkte genutzt werden können, sind in einem Dreieck angeordnet, um Ausrichtung und Abstand vom AUV zur Marke berechnen zu können. Der große Kreis dient zur Erkennung auch aus großen Entfernungen, wohingegen die beiden kleinen Kreise erst aus dem Nahbereich zu erkennen sind.



Abbildung 3.19. – Selbstähnliche kreisrunde Marke von Negre u.a. (links) mit ihren 3 lokalen Maxima (rechts). Abbildung aus [Negre u. a., 2007]

Die Marke hat das folgende Muster $Marke(\rho, \Theta) = f(\rho)$, wobei ρ und θ Polarkoordinaten sind, die in die auch bei Briggs und Scharstein genutzte Rechteckfunktion f eingesetzt werden. Hier kommt es nun auf die x - und y -Koordinaten an, wohingegen der Winkel θ im Muster keine Rolle spielt. Diese Marke ist komplett rotationsinvariant und lässt sich theoretisch mit der Matching-Funktion m_p aus Gleichung 3.6 berechnen, wobei es aufgrund von möglichen Verdeckungen oder anderen Störungen im Bild besser ist, wie von Negre u.a. vorgeschlagen, eine Bestimmung von m_p in 4 Richtungen durchzuführen (z.B. $-x, +x, -y$ und $+y$). Die komplette Funktion m_p hat dann die folgende Form:

$$m_p(x, y) = m_p^{-x}(x, y) + m_p^{+x}(x, y) + m_p^{-y}(x, y) + m_p^{+y}(x, y) \quad (3.7)$$

Nach der Berechnung von $m_p(x,y)$ bildet sich nun ein starkes lokales Maximum im Zentrum der Marke (siehe Abb. 3.19), dessen Bestimmung Negre u.a. mit Hilfe eines Schwellwertes vornehmen. Negre u.a. haben die Erkennung über und unter Wasser getestet und sie hat sich als robust gegenüber verschiedenen Lichtverhältnissen, ruckartigen Bewegungen der Kamera und, besonders unter Wasser wichtig, gegenüber Reflexionen in der Nähe der Wasseroberfläche. Um die Performanz der Erkennung zu steigern haben Negre u.a. außerdem ein Trackingverfahren auf Basis eines Partikelfilters entwickelt, der die Position der Marke im nächsten Einzelbild (Frame) ungefähr voraussagen kann und die Suche nach der Marke somit auf einen Bereich von 100×100 Pixel (bei einer Kameraauflösung von 640×480 Pixel) beschränkt werden kann. Die Geschwindigkeit wurde mittels des Trackings laut ihrer Beschreibung um den Faktor 3 verringert, wobei alle 10 Bilder eine volle Erkennung über das Bild ausgeführt wird, um z.B. eine neue Marke zu erkennen oder die alte Marke nicht zu verlieren.

Insgesamt scheint die Benutzung von selbstähnlichen Marken eine sehr robust und einfach zu sein. Durch die Entwicklung von Negre u.a. besitzt die Marke auch die wichtige Eigenschaft der Rotationsinvarianz. Interessant wäre mit Sicherheit, ob es möglich ist in die Mitte der kreisrunden Marke ein Codefeld einzubringen, um diese auch als codierte Messmarke nutzen zu können. Die Selbstähnlichkeit könnte dann, wie bei Briggs und Scharstein, zum einfacheren Finden einer im Inneren befindlichen codierten Marke (z.B. ARTag) genutzt werden.

4. Verfolgung

4.1. Einleitung

Unter dem Wort Verfolgung im Kontext dieser Arbeit ist die Verfolgung (im weiteren Verlauf u.a. auch Tracking genannt) der Position einer Marke über mehrere Einzelbilder eines Videosignals zu verstehen. Die Verfolgung soll zur Geschwindigkeitssteigerung der nachfolgenden Bildverarbeitungsoperationen führen, da es durch die Informationen aus den letzten Positionen der Marke möglich ist, eine Vorhersage der Position der Marke im nächsten Bild vorzunehmen. Diese Vorhersage wird dazu genutzt einen Bereich, bzw. Suchfenster, oder auch Region of Interest (ROI) genannt, zu definieren, der das Gebiet, in dem nach der Marke gesucht wird, eingrenzt. Diese Eingrenzung führt in den meisten Fällen zu einer deutlichen Verringerung der zu verarbeitenden Datenmenge und damit zur gewünschten höheren Verarbeitungsgeschwindigkeit des Erkennungsalgorithmus. Wobei sich in dieser Arbeit nicht, wie sonst üblich, der zu beobachtende Gegenstand bewegt, sondern die Kamera, was im Endeffekt aber keinen Unterschied macht.

Für diese Arbeit zwei verschiedene Verfahren zur Verfolgung genutzt, die im folgenden vorgestellt werden sollen. Bei dem ersten Verfahren handelt es sich um eine einfache offensichtliche Lösung dieses Problems, wohingegen das zweite Verfahren einen stochastischen Zustandschätzer beschreibt.

4.2. Einfaches Verfahren

Will man aus einer Menge von Positionsangaben (x - und y -Koordinate, hier Vektor \mathbf{x}) einer Marke aus vergangenen Bildern auf die Position im aktuellen Bild k schließen, bietet sich die einfachste und offensichtlichste Lösung an: Man errechnet aus der Position der Marke im letzten Bild (\mathbf{x}_{k-1}) und aus dem vorletzten Bild (\mathbf{x}_{k-2}) die zurückgelegte Distanz, womit sich anhand der Abtastrate/Zeitintervall Δt die Geschwindigkeit $\dot{\mathbf{x}}_{k-1}$ ermitteln lässt. Aus diesen Werten lässt sich nun unter der Annahme konstanter Geschwindigkeit die Position \mathbf{x}_k vorhersagen:

$$\dot{\mathbf{x}}_{k-1} = \frac{\mathbf{x}_{k-1} - \mathbf{x}_{k-2}}{\Delta t} \quad (4.1)$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \dot{\mathbf{x}}_{k-1} \Delta t \quad (4.2)$$

Wenn der Vektor \mathbf{x}_k berechnet wird, ist eine mögliche Position der Marke im aktuellen Bild bestimmt, um die herum oder mit deren Hilfe z.B. eine ROI definiert werden kann. Die Situation

der zeitweiligen Verdeckung der Marke kann durch die Annahme einer konstanten Geschwindigkeit über mehrere Bilder hinweg gelöst werden.

4.3. Kalman Filter

Das vorgestellte Verfahren ist laut Davies (siehe [Davies, 2005]) zu einfach, um die besten Ergebnisse zu liefern. Er macht dies u.a. daran fest, dass drei Dinge unterschieden werden sollten: Die Messung selbst, die beste Schätzung des Systemzustands vor der Messung und die beste Schätzung des Systemzustands nach der Messung. Der Systemzustand beinhaltet in diesem Fall die Position und die Geschwindigkeiten. außerdem sollten z.B. die durch Messungen auftretenden Ungenauigkeiten im System durch Rauschen mit in die Schätzung einbezogen werden können.

Diese genannten Bedingungen berücksichtigt das 1960 u.a. von Rudolf E. Kalman entwickelte Kalman Filter. Es ist seit längerem eine Standardmethode der Optimalen Regelungstechnik mit Zustandsbeobachter (siehe [Nischwitz und Haberäcker, 2004]) und wird (meist) zu Zwecken der Verfolgung auch in der Bildverarbeitung genutzt. Das Kalman Filter ist, u.a. unter der Bedingung von weißem, normalverteiltem System- und Messrauschen mit Erwartungswert 0, ein optimaler Schätzer für den Zustand eines linearen Systems. Die Optimalität ist im Sinne der Minimierung des mittleren quadratischen Fehlers zu verstehen. Aus der Schätzung des Systemvektors $\hat{\mathbf{x}}$ zum Zeitpunkt $k - 1$ ($\hat{\mathbf{x}}_{k-1}$) kann eine Schätzung für den Zustand des Systems zum Zeitpunkt k ($\hat{\mathbf{x}}_k$) erfolgen. Durch eine Messung \mathbf{y}_k wird die Schätzung $\hat{\mathbf{x}}_k$ des Zustands korrigiert, um auf Basis der neuen Informationen eine bessere Schätzung zu ermöglichen. Die Schätzung vor der Messung wird als a-priori Schätzung bezeichnet und durch $\hat{\mathbf{x}}_k^-$ ausgedrückt, wohingegen die durch die Messung korrigierte Schätzung $\hat{\mathbf{x}}_k^+$ a-posteriori genannt wird. Insgesamt ergibt sich ein rekursiver Ablauf, was den Vorteil hat, dass keine lange Historie von Schätzungen gespeichert werden muss, um eine Schätzung des Systemzustands zu erhalten.

Das System, dessen Zustand geschätzt werden soll, muss sich linear zeitdiskret wie folgt in einer Systemgleichung ausdrücken lassen:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (4.3)$$

Hierbei ist \mathbf{F} die Systemmatrix¹¹, die einen Zustand von $k - 1$ nach k überführt. Die beiden Variablen \mathbf{B}_k und \mathbf{u}_k werden als Eingabematrix bzw. Eingabevektor bezeichnet, spielen für diese Arbeit aber keine Rolle und werden deshalb ab jetzt weggelassen. \mathbf{w}_k ist das Prozessrauschen, welches wie schon erwähnt u.a. als weiß und normalverteilt angenommen wird. Um die

¹¹Es wird im Weiteren davon ausgegangen, dass die Systemmatrix \mathbf{F} , die Messmatrix \mathbf{H} , die Kovarianzmatrix des Messrauschens \mathbf{R} und die Kovarianzmatrix des Systemrauschens \mathbf{Q} konstant sind. Dies muss in der Praxis nicht unbedingt der Fall sein.

externen Messungen in die Schätzung einfließen lassen zu können, muss eine Messgleichung definiert werden, die den Zusammenhang zwischen Zustand und Messung festlegt:

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + v_k \quad (4.4)$$

\mathbf{H} ist die Messmatrix, welche die Beziehung zwischen der beobachteten Größe \mathbf{y}_k und dem Systemzustand \mathbf{x}_k beschreibt. Der Vektor \mathbf{v}_k dient hier zur Beschreibung des Messrauschens, für welches die gleichen Annahmen, wie bei \mathbf{w}_k gelten.

Das Kalman Filter arbeitet mit einer sog. Regelschleife, in der ein Prädiktor-Korrektor-Verfahren abläuft. Dabei gibt es in dieser Schleife, wie in Abbildung 4.1 zu sehen ist, zwei Phasen die durchlaufen werden:

- **Vorhersage:** Der Zustand des Systems zum Zeitpunkt k wird ohne den Einfluss von neuen Messungen vorhergesagt. D.h. es wird eine a-priori Schätzung $\hat{\mathbf{x}}_k^-$ vorgenommen.

$$\hat{\mathbf{x}}_k^- = \mathbf{F} \hat{\mathbf{x}}_{k-1} \quad (4.5)$$

$$\mathbf{P}_k^- = \mathbf{F} \mathbf{P}_{k-1}^+ \mathbf{F}^T + \mathbf{Q} \quad (4.6)$$

\mathbf{P} bezeichnet die Schätzfehlerkovarianzmatrix, über die der Fehler der Schätzungen ermittelt werden kann. \mathbf{Q} wiederum ist die Kovarianzmatrix für das Systemrauschen.

- **Korrektur:** Bei einer neuen Messung kann durch eine Korrektur mit den Messwerten ein neuer a-posteriori Systemzustand $\hat{\mathbf{x}}_k^+$ geschätzt werden.

$$\mathbf{K}_k = \frac{\mathbf{P}_k^- \mathbf{H}^T}{\mathbf{R} + \mathbf{H} \mathbf{P}_k^- \mathbf{H}^T} \quad (4.7)$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H} \hat{\mathbf{x}}_k^-) \quad (4.8)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- \quad (4.9)$$

\mathbf{K} wird als Kalman Gain bezeichnet und gibt den Faktor an, mit dem die Messung \mathbf{y}_k in die neue Schätzung einfließt. Dieser Faktor wird so optimiert, dass \mathbf{P}_k^+ minimiert wird. \mathbf{R} gibt die Kovarianz des Messrauschens an und \mathbf{I} bezeichnet die Einheitsmatrix. Für eine Herleitung der Gleichungen und weitere Informationen zum Kalman Filter sei auf [Welch und Bishop, 2006] und [Cuevas u. a., 2005] verwiesen. Ein gutes einführendes Beispiel zu diesem Thema ist in [Nischwitz und Haberäcker, 2004] zu finden.

Zur Verfolgung der Marken in dieser Arbeit werden die x - und y -Koordinaten von Punkten, die auch als Messungen dienen, und dessen Geschwindigkeiten (\dot{x} und \dot{y}) genutzt. Auf die Beschleunigung (\ddot{x} und \ddot{y}) wird hier bewusst verzichtet, da sich durch Tests zu dieser Arbeit ergeben hat, dass diese keinen positiven Einfluss auf die Zustandsschätzungen hat, was wohl darin begründet liegt, dass sie aus der schon nicht mehr messbaren Geschwindigkeit ermittelt werden muss.

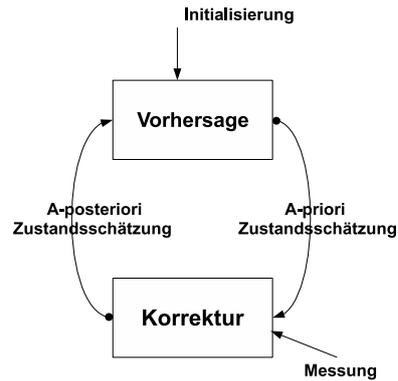


Abbildung 4.1. – Regelschleife des Kalman Filters.

Aus den bekannten Bewegungsgleichungen mit dem Zeitintervall Δt zwischen Zustand $k - 1$ und k

$$x_k = x_{k-1} + \dot{x}_{k-1} \Delta t \quad (4.10)$$

$$y_k = y_{k-1} + \dot{y}_{k-1} \Delta t \quad (4.11)$$

$$\dot{x}_k = \dot{x}_{k-1} \quad (4.12)$$

$$\dot{y}_k = \dot{y}_{k-1} \quad (4.13)$$

ergibt sich die folgende Systemgleichung

$$\begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + w_{k-1} \quad (4.14)$$

und die Messgleichung als

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} + v_k \quad (4.15)$$

Wie den Gleichungen 4.12 und 4.13 zu entnehmen ist, wird hier eine konstante Geschwindigkeit angenommen. Dies ist in der Realität sehr unwahrscheinlich, weshalb Änderungen bezüglich der Geschwindigkeit oder der nicht modellierten Beschleunigung im Systemrauschen \mathbf{Q} berücksichtigt werden müssen.

Das Kalman Filter ist in der für die softwaretechnische Umsetzung der Verfahren in dieser Arbeit genutzten LTI-Lib schon enthalten und muss demnach nicht mehr selbst implementiert werden. Ebenso befindet sich in dieser Bibliothek ein vorgefertigter Tracker, der die Systemgleichung 4.14 in ein Kalman Filter umsetzt. Es müssen nur noch die entsprechenden Werte

übergeben werden. Bei der Festlegung der initialen Fehlerkovarianz \mathbf{P}_0^- , der Messfehlerkovarianz \mathbf{R} und der Kovarianz des Systemrauschens \mathbf{Q} wird unkorreliertheit angenommen, sodass die Matrizen ausschließlich auf der Hauptdiagonalen einen von 0 verschiedenen Wert annehmen. Der initiale Systemzustand wird auf den Wert der ersten Messung gesetzt, woraus sich die folgenden Matrizen ergeben:

$$\mathbf{x}_0^- = \begin{bmatrix} x_{Pos} \\ y_{Pos} \\ 0 \\ 0 \end{bmatrix} \quad (4.16)$$

$$\mathbf{P}_0^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.17)$$

$$\mathbf{R} = \begin{bmatrix} 10^{-3} & 0 \\ 0 & 10^{-3} \end{bmatrix} \quad (4.18)$$

$$\mathbf{Q} = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix} \quad (4.19)$$

Die Werte¹² für \mathbf{x}_0^- und \mathbf{P}_0^- entsprechen den Standardwerten zur Initialisierung des genutzten Bibliotheksmoduls, wohingegen die Werte für \mathbf{R} und \mathbf{Q} experimentell anhand von mehreren aufgenommenen Videosequenzen ermittelt wurden. Diese Videosequenzen beinhalten hauptsächlich Bewegungen, die mit einer Kamera auf einem Rollwagen entstanden sind. Es sei aber darauf hingewiesen (siehe hierzu auch Kapitel 4.4.4), dass es mit mehr, als während der Bearbeitungszeit dieser Arbeit erlangten, Kenntnissen der Theorie des Kalman Filters und mehr praktischen Erfahrungen auf diesem Gebiet, mit Sicherheit noch bessere Ergebnisse erzielt werden können. Die genutzten Werte reichten aber in den Tests zu dieser Arbeit aus, um bis auf wenige Ausnahmen eine gute Verfolgung der Marken zu gewährleisten.

Wie schon erwähnt setzt dieses Filter voraus, dass sich das zu beobachtende System linear abbilden lässt, was gerade bei sehr komplexen Bewegungen zu Problemen führt. Alle Abweichungen von dieser Voraussetzung müssen theoretisch z.B. über Rauschen berücksichtigt werden. Dies betrifft z.B. auch den Fall einer nicht konstanten Beschleunigung, da diese nicht linear abgebildet werden kann. In diesem Fall ist es möglich das Erweiterte Kalman Filter (EKF) zu nutzen, bei dem die Systemgleichung und auch die Messgleichung nicht linear sein müssen. Ein Beispiel für die Nutzung und die damit verbundene verbesserte Verfolgung einer komplexen Bewegung ist in [Cuevas u. a., 2005] zu finden.

Ebenso wie bei dem vorgestellten einfachen Verfahren können auch mit dem Kalman Filter

¹²Die Einheit der Varianzen ist Pixel². Die Position wird mit der x - und y -Koordinate des jeweiligen Pixels angegeben.

zeitweise Verdeckungen der Marke überbrückt werden. Hierfür wird einfach in den betreffenden Zyklen der Schritt der Korrektur anhand neuer Messwerte weggelassen bzw. es werden so große Werte für das Messrauschen eingesetzt, dass die dennoch eingesetzten Messwerte praktisch nicht berücksichtigt werden. Dies führt im Endeffekt dazu, dass das Filter nur noch vorhersagt.

4.4. Beispiele

In diesem Abschnitt werden die beiden vorgestellten Verfahren anhand zweier Beispiele miteinander verglichen. Ein drittes Beispiel dient zur Illustration der kurzzeitigen Verdeckung des zu verfolgenden Punktes. Es wird jeweils ein Punkt, auf einer später noch vorgestellten Zylindermarke, über einen längeren Zeitraum verfolgt.

4.4.1. Bewegung 1

Dieses Beispiel (siehe Abb. 4.2¹³) zeigt eine durch Messungen aufgenommene Bewegung, die durch eine auf einem Rollwagen befindliche Kamera entstanden ist¹⁴. Der Rollwagen wurde per Hand ohne große Beschleunigungen im Schrittempo unter einer an der Decke hängenden Marke hindurchgefahren, wodurch eine relativ repräsentative Bewegung mit einigen Kurven erzeugt werden konnte. Die beiden weiteren Kennlinien der Grafik zeigen die von den beiden vorgestellten Verfahren vorhergesagten Bewegungen.

In Abbildung 4.3 sind für beide Verfahren die Abweichungen der Vorhersagen, von der jeweils gemessenen Position der einzelnen Bilder der Testsequenz, zu sehen. Die Berechnung der Abweichung wurde hierfür über den euklidischen Abstand der beiden Punkte (Vorhersage und Messung) durchgeführt. Bis auf einige wenige Ausnahmen ist das Kalman Filter bei seinen Vorhersagen genauer als das einfache Verfahren, was sich auch in der etwas kleineren durchschnittlichen Abweichung (ca. 2,19 Pixel zu ca. 2,33 Pixel) und der etwas kleineren durchschnittlichen quadratischen Abweichung (ca. 2,66 Pixel zu ca. 2,8 Pixel) ausdrückt.

An diesem Beispiel ist zu sehen, dass bei dieser relativ einfachen Bewegung beide Verfahren ziemlich gute Vorhersagen liefern, das Kalman Filter aber im Durchschnitt, trotz der nicht optimalen Parameter, knapp vorne liegt. Die sichtbar größten Abweichungen in Abbildung 4.3 resultieren dabei aus Situationen mit (kurzzeitigen) großen Beschleunigungen.

¹³Alle Abbildungen zu diesem Thema sind im PDF-Dokument dieser Arbeit auf der beiliegenden DVD-ROM als stufenlos vergrößerbare Vektorgrafiken zu finden.

¹⁴Das Video befindet sich auf der beiliegenden DVD-ROM (motion.avi).

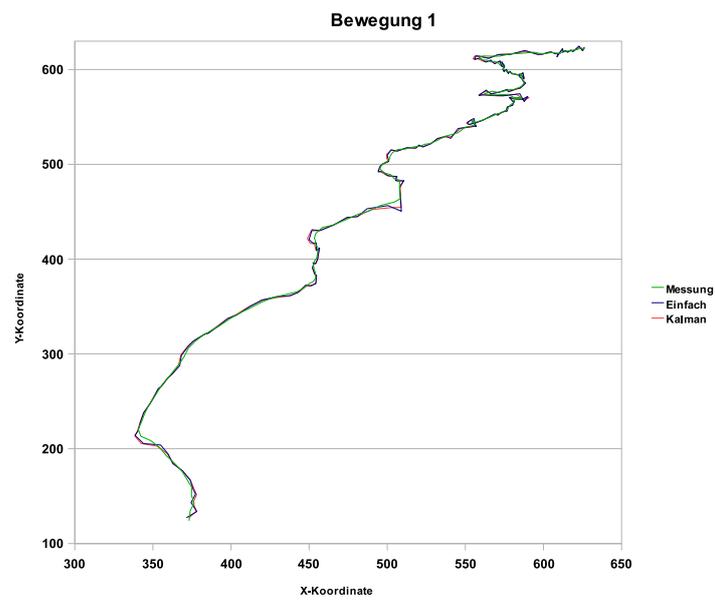


Abbildung 4.2. – Bewegung zu Kapitel 4.4.1. ($\Delta t \approx \frac{1}{7,5}$ s)

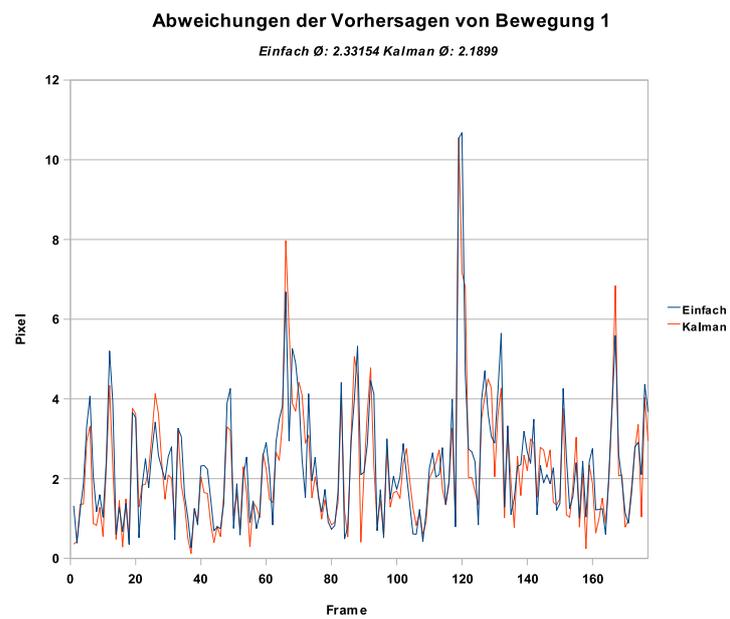


Abbildung 4.3. – Abweichungen der Vorhersagen von den Messwerten. (zu der Bewegung aus Abb. 4.2)

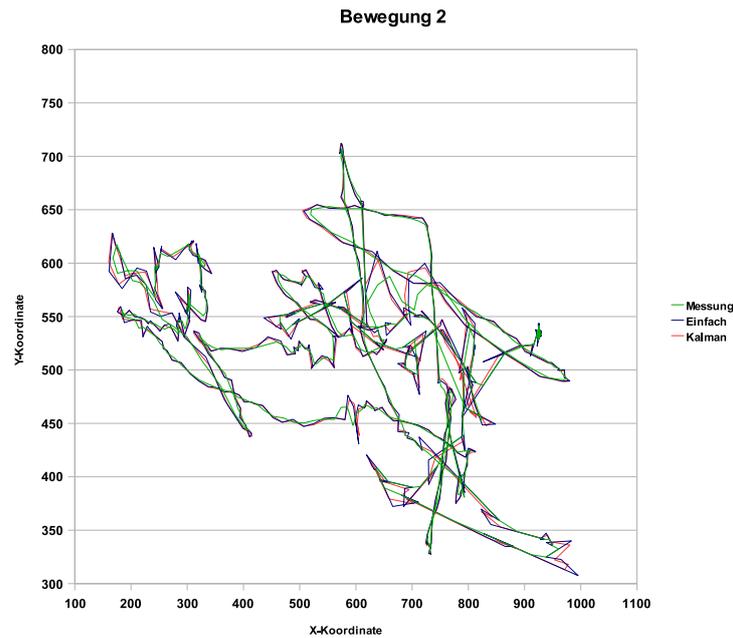


Abbildung 4.4. – Bewegung zu Kapitel 4.4.2. ($\Delta t \approx \frac{1}{15}$ s)

4.4.2. Bewegung 2

Abbildung 4.4 zeigt eine Bewegung, die so bei der gewöhnlichen Verwendung der Marken aus dieser Arbeit nur selten auftreten wird. Sie ist entstanden durch das schwenken der Kamera mit der menschlichen Hand, was zu einer sehr komplexen Bewegung, mit harten Kurven und damit verbunden starken Beschleunigungen geführt hat¹⁵. Dies sind auch die Stellen, an denen die Abweichungen (siehe Abb. 4.5) am größten sind. Generell kann gesagt werden, dass die Abweichungen sowohl im Durchschnitt (auf ca. 6,42 Pixel bzw. ca. 6,01 Pixel) als auch im Maximum stark angestiegen sind (quadratische mittlere Abweichung ca. 10,4 Pixel zu ca. 11,78 Pixel), das Kalman Filter aber wieder knapp vorne liegt.

4.4.3. Bewegung 3

Dieses Beispiel (siehe Abb. 4.6) zeigt eine Bewegung, bei der der verfolgte Messpunkt kurzzeitig (14 Einzelbilder bei $\Delta \approx \frac{1}{15}$ s) verdeckt ist. Diese ist aus Gründen der Einfachheit nicht durch eine Bewegung der Kamera, sondern durch die Bewegung der Marke entstanden¹⁶. Es ist zu sehen, dass das Kalman Filter mit seinen Schätzungen nicht soweit daneben liegt, wie das einfache Verfahren, welches aus den gemessenen Positionen eine Geschwindigkeit

¹⁵Das Video befindet sich auf der beiliegenden DVD-ROM (wildmotion.avi).

¹⁶Das Video befindet sich auf der beiliegenden DVD-ROM (occlusion.avi).

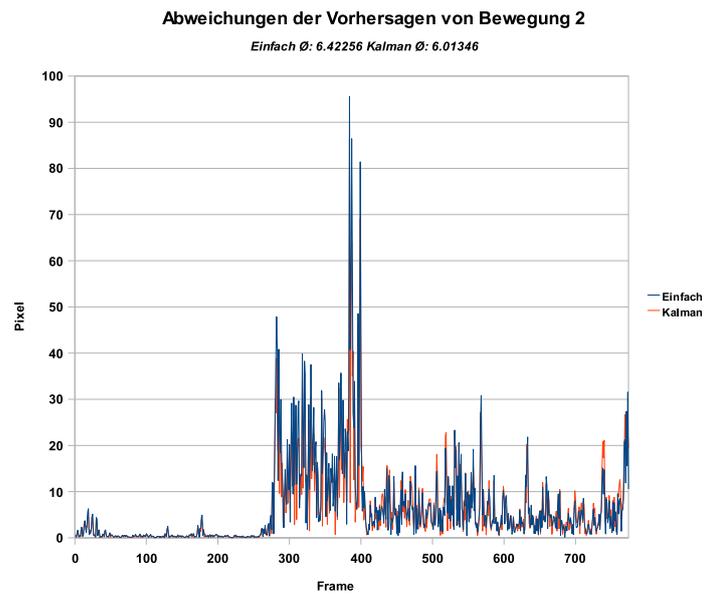


Abbildung 4.5. – Abweichungen der Vorhersagen von den Messwerten. (zu der Bewegung aus Abb. 4.4)

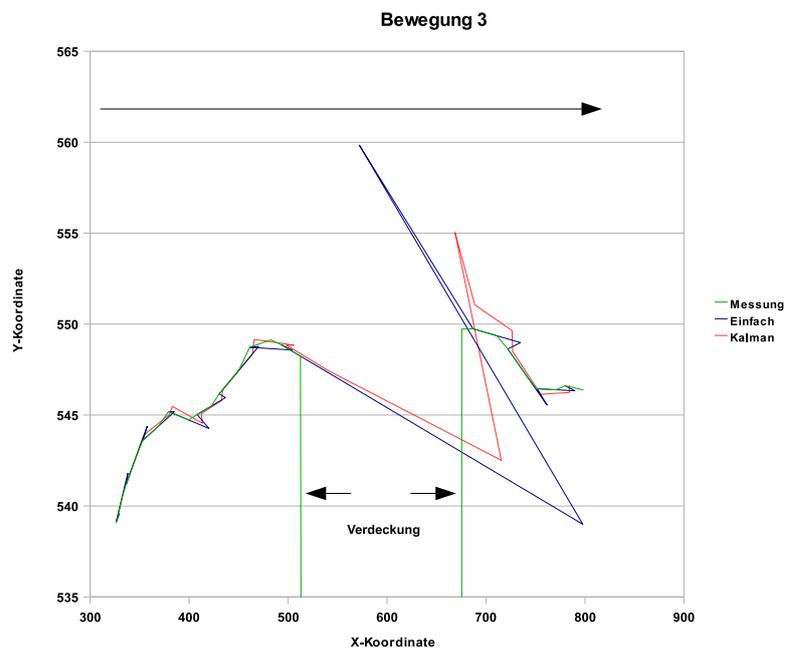


Abbildung 4.6. – Bewegung von links nach rechts zu Kapitel 4.4.3. ($\Delta t \approx \frac{1}{15} s$)

berechnet. Wobei anzumerken ist, dass bei der wiederkehr von Messwerten keine Korrektur (Geschwindigkeit wieder auf 0 setzen) des einfachen Verfahrens stattfindet, was zu dem sichtbaren starken Ausschlag in die falsche Richtung führt. Das Kalman Filter benötigt nachdem wieder Messwerte verfügbar sind einige Frames, um wieder bessere Vorhersagen zu treffen, was aber durch die Veränderung des Messrauschens für einen kurzen Zeitraum nach der Wiederkehr von Messwerten auch noch gezielter beeinflusst werden könnte.

4.4.4. Bemerkungen

Aus den Tests kann die Empfehlung abgeleitet werden eher das Kalman Filter zu nutzen, da es im Durchschnitt, wenn auch nur knapp, die besseren Ergebnisse liefert und auch die minimale (0,01 Pixel zu 0,04 Pixel) und maximale (86,53 Pixel zu 95,51 Pixel) Abweichung bei diesen Tests (ohne Beachtung von Bewegung 3) für das Kalman Filter spricht. Diese Werte können aber wie schon erwähnt sicher noch durch eine Optimierung der Parameter des Kalman Filters (z.B. auf ein bestimmtes Fahrzeug und/oder Markentyp) verbessert werden. U.u. kann dann auch die Beschleunigung gut in das Modell integriert werden, da diese hauptsächlich für die großen Abweichungen verantwortlich ist. außerdem ist durchaus sinnvoll die Kovarianzen des Rauschens dynamisch zu verändern, um zu verhindern, dass die Verstärkung durch das Kalman Gain immer (fast) gleich ist, was bei starken Abweichungen von Messung und Prädiktion ebenfalls zu Problemen führt. Hierfür sind in [Nischwitz und Haberäcker, 2004] z.B. einige sinnvolle Anregungen bezüglich der Veränderung des Systemrauschens, in Abhängigkeit der Geschwindigkeit mit der sich die Marken im Bild bewegen, zu finden. Ein weiterer Schritt zur Erweiterung könnte die Integration von Werten aus der Odometrie des Fahrzeuges in das Filter sein, um so bessere Anhaltspunkte für Geschwindigkeit und Beschleunigung für die Vorhersage zu erhalten.

Ein Nachteil des Kalman Filters besteht darin, dass die Berechnungen mehr und aufwendiger sind, als bei dem einfachen Verfahren. Die Auswirkungen dieses Nachteils auf die Laufzeit eines Erkennungsalgorithmus ist in Kapitel 6.9.8 zu finden.

4.5. Ablauf der Verfolgung

Da die Verfolgung bei allen im Rahmen dieser Arbeit verwendeten/entworfenen Markentypen eingesetzt werden soll, wird hier ein grundsätzlicher Ablauf skizziert, der bei allen noch vorzustellenden Marken gleich ist.

Grundsätzlich besteht die Verfolgung darin, dass mehrere Punkte pro Marke im Bild gemessen und deren Positionen für das nächste Bild vorausgesagt werden. Dies wird zyklisch wiederholt. Dieser Zyklus ist durch die Nutzung des Kalman Filters schon vorgegeben und lässt sich so auch als Basis für einen generalisierten Ablauf nutzen. Aus den Vorhersagen, die das gewählte Verfahren liefert, wird ein minimales Rechteck berechnet, in dem sich alle

Punkte befinden und welches parallel zu den Seiten des Bildes orientiert ist. Dieses Rechteck wird entsprechend für jede Marke errechnet und dem eigentlichen Erkennungsalgorithmus übergeben. Dieses nutzt diese Rechtecke nun als Suchfenster für die Marken. Die Form des Rechtecks hilft hierbei, da diese nun einfach aus dem großen Gesamtbild extrahiert und einzeln untersucht werden können. Bei der Bestimmung der Rechtecke sollten aber Toleranzen berücksichtigt werden, um ungenaue Schätzungen begrenzt abfangen zu können. In den Suchfenstern gefundene Marken werden dementsprechend an das Verfahren zurückgemeldet, woraufhin dieses z.B. im Falle des Kalman Filters eine Korrektur durchführt. Die Verwaltung der Verfolgung geschieht hierbei für alle Marken zentral in einem Tracker, welcher das wirklich eingesetzte Verfahren kapselt und den Ablauf vorgibt.

Damit Marken überhaupt verfolgt werden können ist es nötig in Abständen das ganze Bild zu durchsuchen, hierdurch können mit der Zeit verlorene Marken wieder- und neue Marken erstmals gefunden werden, die somit auch zur Verfolgung zur Verfügung stehen. Es sind insgesamt noch zwei spezielle Situationen zu beachten, auf die gesondert reagiert werden muss:

- **Marke im letzten Bild zum ersten Mal gefunden:** Ist die Marke erst im letzten Bild zum ersten Mal gefunden worden, kann noch keine Aussage über die Geschwindigkeit der Punkte getroffen werden. Aus diesem Grund müssen hier besondere Toleranzen für die Größe des Suchfensters angenommen werden. Im Extremfall, bei zu hohen zu erwartenden Geschwindigkeiten und einer zu niedrigen Bildrate der Kamera, kann hier auch das ganze Bild durchsucht werden.
- **Marke im letzten Bild nicht gefunden:** Durch vielfältige Störungen (z. B. kurzzeitige Verdeckungen, zu schnelle Bewegung der Kamera, . . .) oder dadurch dass die Marke den Bildbereich der Kamera verlässt kann es passieren, dass eine Marke nicht, in dem durch die Schätzungen aufgespannten Suchfenstern, gefunden wurde. Das hier implementierte Verfahren toleriert die Nichterkennung auf einer bestimmten, durch einen Schwellwert (z. B. 5 Bilder), einstellbaren Folge von Bildern. Das Verfahren läuft somit nur noch im Prädiktionsmodus.

Zusätzlich bietet der Tracker, welcher jeweils softwaretechnisch in den entsprechenden Impresario-Macros der Markenerkennung umgesetzt ist, außerdem die Möglichkeit solange immer das ganze Bild zu durchsuchen, bis eine bestimmte Anzahl von Marken im Bild gefunden wurde. Dies würde für diese Arbeit z.B. 3 Marken bedeuten, da ohne diese Anzahl eine Posenbestimmung nicht möglich ist. Hierbei gibt es zwei Möglichkeiten für den Fall, dass zwar für genügend Marken eine Position im Bild vorausgesagt wurde, diese aber nicht alle gefunden werden konnten. Die erste Möglichkeit besteht darin im nächsten Durchlauf das komplette Bild zu durchsuchen oder zweitens dies schon auf das aktuelle Bild anzuwenden. In dieser Arbeit wird der zweite Fall angenommen, was für diese Situation zwar einen höheren Zeitaufwand für dieses Einzelbild nach sich zieht, der aber vertretbar erscheint. Bei zu wenigen Marken, für die eine Position im Bild vorhergesagt werden kann, wird allerdings gleich das ganze Bild durchsucht, da die Wahrscheinlichkeit, dass sich in den verbleibenden Suchfenstern noch andere als die erwarteten Marken befinden im Durchschnitt als sehr gering angesehen werden

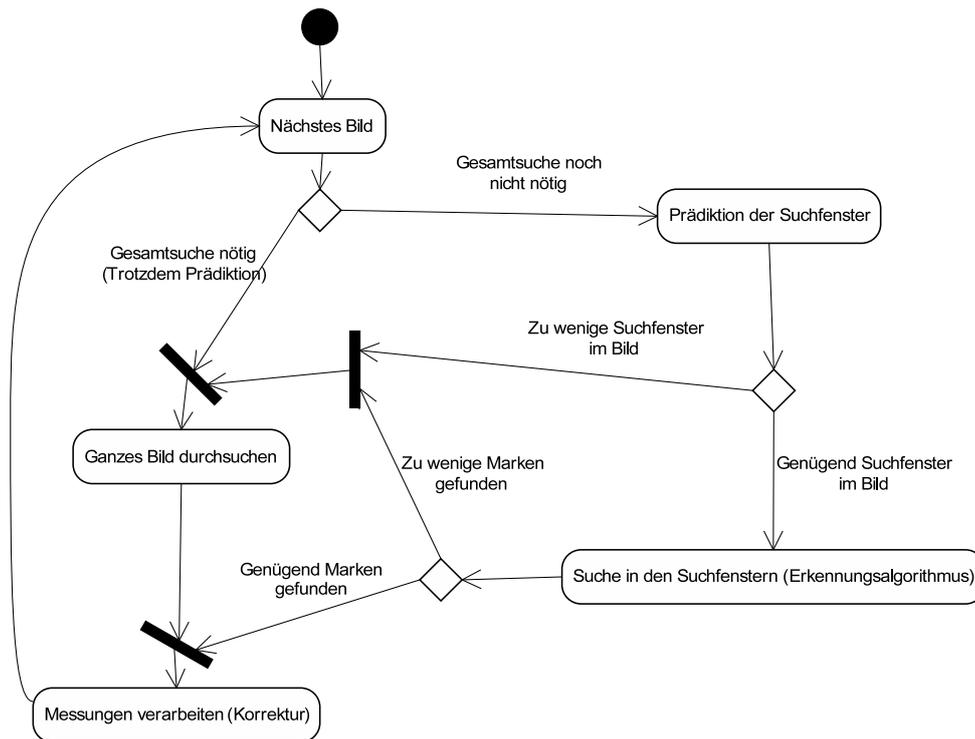


Abbildung 4.7. – Grober Ablauf der Verfolgung.

kann. Eine andere Problematik betrifft die Situation, dass sehr viele oder sehr große Marken im Bild verfolgt werden, in diesem Fall ist es möglich, dass die Verfolgung u.a. durch mehrfach durchsuchte Bildteile nicht mehr effizient ist. Dies sollte in der Praxis aber nur selten auftreten, da die Marken normalerweise so im Raum angebracht werden, dass möglichst wenige benötigt werden und somit beide Fälle selten werden. Die genauen Parameter des Trackers sind in Tabelle 4.1 zu finden und über die Benutzerschnittstelle der Impresario-Macros anpassbar. Der hier besprochene Ablauf ist in Abbildung 4.7 grafisch aufbereitet.

Parameter	Bedeutung	Beispielwerte f. Schrittgeschwindigkeit
<i>minNumberOfTargets</i>	Minimale Anzahl von Marken, die erkannt sein müssen, um nur auf Basis der Vorhersagen zu suchen.	3
<i>trackingMode</i>	Bietet die Möglichkeit von Frame auf zeitbasiertes Tracking umzustellen, was z.B. nötig ist, wenn über einen Framegrabber gearbeitet wird.	frame
<i>windowSizeFactorWidth</i>	Breite des genutzten Suchfensters um eine Marke herum im Verhältnis zur Breite des vorhergesagten Suchfensters.	2
<i>windowSizeFactorHeight</i>	Höhe des genutzten Suchfensters um eine Marke herum im Verhältnis zur Höhe des vorhergesagten Suchfensters.	2
<i>windowSizeFactorFirstFrameWidth</i>	Breite des genutzten Suchfensters um eine Marke herum im Verhältnis zur Breite des vorhergesagten Suchfensters, wenn noch keine Geschwindigkeitsinformationen vorliegen.	2
<i>windowSizeFactorFirstFrameHeight</i>	Höhe des genutzten Suchfensters um eine Marke herum im Verhältnis zur Höhe des vorhergesagten Suchfensters, wenn noch keine Geschwindigkeitsinformationen vorliegen.	2
<i>notInFrameThreshold</i>	Schwellwert für die Anzahl von kommenden Einzelbildern, in denen nach einer nicht mehr gefundenen Marke noch gesucht werden soll.	5
<i>fullSearch</i>	Die Anzahl von Einzelbildern nach der das Bild wieder komplett durchsucht werden soll, um neue Marken zu finden.	10

Tabelle 4.1. – Parameter für den Tracker.

5. Doppelverhältnismarke

In diesem Kapitel wird der von Matthias Wippich während seiner Bachelorarbeit (siehe [Wippich, 2008a]) an der HAW unter dem Namen "Doppelverhältnis-Positionsmarke" zum Zwecke der Navigation/Posenbestimmung entwickelte Markentyp beschrieben und die im Rahmen dieser Arbeit vorgenommenen Erweiterungen und Tests erläutert. Bei weiteren Fragen zu Kapitel 5.1 sei hiermit direkt auf [Wippich, 2008a] verwiesen, wo eine genauere Erklärung dieser Themen zu finden ist.

5.1. Vorstellung

5.1.1. Aufbau

Die Marke hat den folgenden Aufbau (siehe Abb. 5.1):

- Schwarzer Hintergrund zur besseren Kontrastbildung.
- Weiße rechteckige Außenmarkierung, die vorzugsweise aus retroreflektierender Folie hergestellt wird.
- 4 Kreise mit dem gleichen Durchmesser, die nach dem aus der Geometrie bekannten Doppelverhältnis (engl. cross-ratio) auf einer Gerade in der Mitte der Marke verteilt sind. Der Mittelpunkt der Kreise dienen als Messpunkte des Doppelverhältnisses. Dieses Doppelverhältnis wird zur Codierung der ID genutzt und wurde vorwiegend deshalb ausgesucht, da es invariant gegenüber projektiven Transformationen ist. Auch diese Kreise sollten aus dem selben Material wie die Außenmarkierung hergestellt werden.
- Der Mittelpunkt des unteren Kreises dient als Messpunkt, weshalb die Marke nicht rotationsinvariant ist und immer vertikal zur Kameraorientierung (bei der Posenbestimmung von autonomen Fahrzeugen meist gleich dem Boden) aufgehängt werden muss.
- Alle Abstände, außer die des Doppelverhältnisses, und Breiten sind entweder gleich dem Durchmesser der Kreise oder ein vielfaches davon.

Das Doppelverhältnis von 4 verschiedenen Punkten, die sich auf einer Geraden befinden, ist definiert als reelle Zahl, welche das Verhältnis zweier Teilverhältnisse mit jeweils 3 Punkten abbildet (siehe Abb. 5.2). Es ist also ein Verhältnis von Verhältnissen.

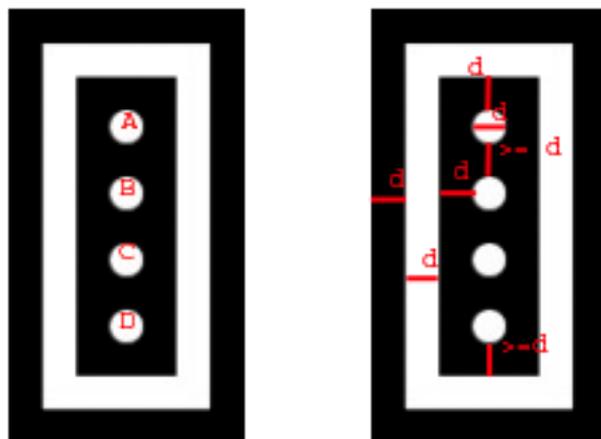


Abbildung 5.1. – Doppelverhältnismarke; links ohne, rechts mit Abstandsmäßen (d =Kreisdurchmesser) Abbildung teilw. aus [Wippich, 2008a]

Das Teilverhältnis λ von 3 kollinearen Punkten A , B und C , wobei B die Strecke \overline{AC} teilt, ist definiert als (siehe z.B. [Luhmann, 2000]):

$$\lambda = R(A, B, C) \text{ mit } \lambda \in \mathbb{R} \quad (5.1)$$

$$\lambda = \frac{\overline{AC}}{\overline{BC}} \quad (5.2)$$

$$(5.3)$$

Daraus folgt das Doppelverhältnis τ von 4 kollinearen Punkten A , B , C und D , wobei die Strecke \overline{BC} die Strecke \overline{AD} teilt, als:

$$\tau = CR(A, B, C, D) \text{ mit } \tau \in \mathbb{R} \quad (5.4)$$

$$\tau = \frac{R(A, B, C)}{R(A, B, D)} \quad (5.5)$$

$$\tau = \frac{\overline{AC} \cdot \overline{BD}}{\overline{BC} \cdot \overline{AD}} \quad (5.6)$$

Es gibt mehrere verschiedene Definitionen des Doppelverhältnisses, wobei sich diese nur in der Reihenfolge der Punkte unterscheiden. Insgesamt lassen sich durch Permutation der Punkte 24 Doppelverhältnisse bilden, wobei aber alle die auf dem Austausch von zwei Punktpaaren basieren den gleichen Wert haben. Dies trifft jeweils auf 4 der 24 Permutationen zu, weshalb maximal 6 verschiedene Doppelverhältnisse möglich sind.

Das Doppelverhältnis ist invariant gegenüber projektiven Transformationen, bei denen die einfachen Längenverhältnisse, im Unterschied zu affinen Transformationen, nicht erhalten bleiben. Aus diesem Grund eignet es sich sehr gut für den Einsatz bei Messmarken, da die ID

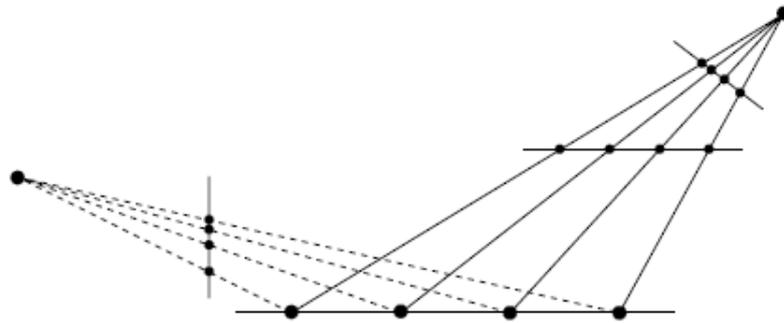


Abbildung 5.2. – Projektive Transformationen einer Linie mit 4 Punkten. Doppelverhältnis dieser Punkte bleibt gleich. Abbildung aus [Hartley und Zisserman, 2003]

so auch unter der häufig vorkommenden perspektivischen Verzerrung eindeutig ausgelesen werden kann. Gleiche Invariante gilt auch für 4 Geraden, die sich in einem Punkt schneiden, da sich durch eine Gerade, die diese 4 Geraden schneidet, 4 kollineare Schnittpunkte bilden, deren Doppelverhältnis wieder invariant ist. Für weitere Informationen zum Doppelverhältnis sei auf [Hartley und Zisserman, 2003] und [Luhmann, 2000] verwiesen.

5.1.2. Konfiguration der Doppelverhältnisbereiche

Um unterschiedliche Exemplare dieses Markentyps herstellen zu können müssen diese unterschiedliche Doppelverhältnisse aufweisen. Aus diesem Grund wurde von Wippich ein Java-basiertes Konfigurationsprogramm erstellt, dessen Algorithmus und die nötigen Vorüberlegungen hier kurz zusammengefasst dargestellt werden.

Der Konfigurationsalgorithmus hat den Zweck Marken vorgegebener Größe mit eindeutigen IDs zu berechnen. Dies ist ohne Automatisierung schlecht machbar, da darauf geachtet werden muss, dass die Doppelverhältnisse wirklich unterschiedlich sind und auch der Verschnitt, der unweigerlich auftritt minimiert wird.

Die unterschiedlichen Doppelverhältnisse werden auf Basis einer vorgebbaren Markengesamtgröße (Maße des schwarzen Untergrunds) und einer Mindestgröße der Kreisdurchmesser in Pixel (am Besten nicht unter 5 Pixel) auf einer gewünschten maximalen Erkennungsentfernung berechnet. Zu diesem Zweck müssen die internen Daten der Kamera und des Objektivs (Pixelgröße auf dem Chip, Pixelanzahl, Brennweite) bekannt sein, da auf Basis dieser Parameter die einzelnen unterscheidbaren Doppelverhältnisse errechnet werden.

Da es laut Wippich (siehe [Wippich, 2008a]) weder für Menschen noch für Maschinen möglich

ist die Marken exakt zu fertigen, arbeitet der Algorithmus mit folgenden einstellbaren Toleranzen:

- **Genauigkeit zu produzierender Größen:** Gibt an, wie genau die einzelnen Elemente (Kreisdurchmesser, Breite der Außenmarkierung,...) hergestellt werden können und sorgt somit dafür, dass nur herstellbare Streckenlängen u.ä. errechnet werden.
- **Erstellungsgenauigkeit:** Maß für die Genauigkeit des Aufklebens der Kreise, da durch eine Abweichung ein anderes Doppelverhältnis entsteht.
- **Messgenauigkeit:** Genauigkeit der Erkennung des Doppelverhältnisses im Bild.

Durch die Definition dieser 3 Toleranzen ergeben sich 5 verschiedene Doppelverhältnisse pro Marke, wobei sich das angestrebte Doppelverhältnis in der Mitte dieser Verhältnisse befindet:

- **Minimal erkanntes minimal erstelltes Doppelverhältnis:** Dieser Wert stellt das minimale noch erkannte Doppelverhältnis dar, welches sich aus der Addition der Toleranzen zur Erstellung und Erkennung ergibt.
- **Minimal erstelltes Doppelverhältnis :** Minimales Doppelverhältnis, welches durch die Erstellungstoleranzen entstehen kann.
- **Angestrebtes Doppelverhältnis:** Das Doppelverhältnis, welches herauskommen soll, wenn die Marke exakt hergestellt und exakt erkannt wird.
- **Maximal erstelltes Doppelverhältnis:** Maximales Doppelverhältnis, welches durch die Erstellungstoleranzen entstehen kann.
- **Maximal erkanntes maximal erstelltes Doppelverhältnis:** Maximales Doppelverhältnis, welches erkannt wird. Folgt ebenfalls aus der ungenaueren Erkennung des maximal erstellten Doppelverhältnisses.

Als Doppelverhältnis und damit die codierte eindeutige ID einer Marke wird der komplette Bereich zwischen "Minimal erkanntem minimal erstelltem Doppelverhältnis" und "Maximal erkanntem maximal erstelltem Doppelverhältnis" angesehen (siehe Abb. 5.3).

Ein Beispiel für die verschiedenen Doppelverhältnisse ist in Tab. 5.1 zu finden. Aus den Ergebnissen dieser Tabelle lässt sich ersehen, dass die Strecke \overline{BC} den größten Einfluss auf den Wert des Doppelverhältnisses hat, da der Zähler $\overline{AC} \cdot \overline{BD}$ des Doppelverhältnisbruchs in diesem Beispiel immer den Wert 900 annimmt und es somit bei den unterschiedlichen Werten auf den Nenner ankommt. Dieser wird, aus dem Produkt von \overline{BC} und \overline{AD} gebildet (siehe Gl. 5.6), weshalb klar ist, dass \overline{BC} den größten Einfluss hat, da diese Strecke auch den Wert von \overline{AD} mitbestimmt. Aus diesem Grund wird \overline{BC} von Wippich als ein Ausgangspunkt des Algorithmus genutzt. Der Konfigurationsalgorithmus soll nun Maße von Marken erzeugen. Dafür werden ausgehend von der Marke mit dem kleinsten Doppelverhältnis alle Streckenlängen durchprobiert, bis eine Nachfolgemarke gefunden wurde, die den folgenden Regeln genügt:

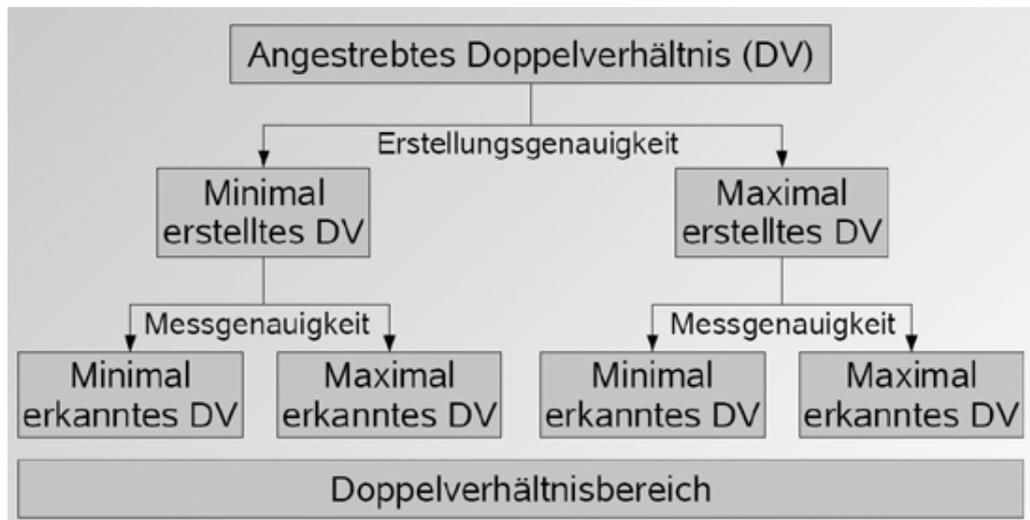


Abbildung 5.3. – Aufbau des Doppelverhältnisbereichs. Abbildung aus [Wippich, 2008b]

	\overline{AB}	\overline{BC}	\overline{CD}	Doppelverhältnis
Minimal erkanntes minimal erstelltes Doppelverhältnis	4	26	4	1,01809
Minimal erstelltes Doppelverhältnis	8	22	8	1,07655
Angestrebtes Doppelverhältnis	10	20	10	1,125
Maximal erstelltes Doppelverhältnis	12	18	12	1,19047
Maximal erkanntes maximal erstelltes Doppelverhältnis	16	14	16	1,39751

Tabelle 5.1. – Beispiel für den Doppelverhältnisbereich einer Marke (Alle Strecken in mm). Erkennungsgenauigkeit: 1 mm; Messgenauigkeit: 2 mm (teilw. aus [Wippich, 2008a])

- Die Streckenlängen müssen gültig sein. Dies bedeutet, dass die Streckenlängen so gewählt werden, dass sie auf die Marke passen. Aber auch die Einbeziehung der Erstellungsgenauigkeit und der Genauigkeit der zu produzierenden Größen ist wichtig, um zu verhindern, dass Elemente mit einer Genauigkeit hergestellt werden müssen, die nicht erreicht werden kann.
- Das minimal erkannte minimal erstellte Doppelverhältnis der neuen Marke muss größer sein als das maximal erkannte maximal erstellte Doppelverhältnis der Vorgänger-Marke. D.h. eine Überlappung der Bereiche ist nicht erlaubt.
- Das neue maximal erkannte maximal erstellte Doppelverhältnis soll so klein wie möglich sein, um eine möglichst große Anzahl von unterschiedlichen Marken erstellen zu können.

Bei der folgenden aus [Wippich, 2008a] entnommenen Beispielrechnung ergeben sich 28 verschiedene Marken mit einem Kreisdurchmesser von 25 mm, einer Abbildungsgröße dieser Kreise von 5 Pixel auf einer Entfernung von 5636 mm und Werten für die Doppelverhältnisse zwischen 1,03106 und 2,20580:

- Daten der genutzten Kamera:
 - Pixelgröße auf dem Chip (μm): 5,6
 - Brennweite (mm): 6,0
 - Pixelanzahl (Breite x Höhe): 640 x 480
- Positionsmarkenhöhe (mm): 500
- Betrachtungsentfernung (mm): 5363
- Bildgröße des Kreisdurchmessers (Pixel): 5
- Genauigkeit zu produzierender Größen (mm): 1
- Erstellungsgenauigkeit (mm): 0,3
- Messgenauigkeit (%): 3,3

Die Unterschiede der Doppelverhältnisse von Nachfolgemarken sind oftmals erst in der 3. Nachkommastelle zu finden.

5.1.3. Erkennung

In diesem Abschnitt wird die Funktionsweise des in [Wippich, 2008a] vorgestellten Algorithmus, zur Erkennung aller Doppelverhältnismarken in einem Bild, erläutert. Grundsätzlich läuft die Suche nach den Marken im kompletten Bild darauf hinaus, dass das Bild mittels einer Art Rasterverfahren untersucht wird (siehe Abb. 5.4). Das genutzte Raster besitzt dabei, wie in Abbildung 5.4 zu sehen ist, zwei Schrittweiten, eine in y-Richtung (Zeilen), diese entspricht der minimalen Höhe einer Marke, berechnet aus dem kleinsten Kreisdurchmesser, und eine andere Schrittweite in x-Richtung (Spalten), welche dem kleinstmöglichen Kreisdurchmesser entspricht. Für jedes so getroffene Pixel wird geprüft, ob es sich wegen seiner Helligkeit mit großer Wahrscheinlichkeit auf der weißen Außenmarkierung (oder einem Kreis) einer Marke befindet. Ziel dieses Verfahrens ist es einen Punkt auf der linken Seite der Außenmarkierung zu finden. Wurde ein solcher Punkt gefunden, wird mit Hilfe des in Kapitel 5.1.3.2 beschriebenen Verfahrens zur Kantendetektion versucht die obere linke Ecke der Umrandung zu finden. Von dieser Ecke geht dann die Suche im Uhrzeigersinn weiter, um die verbleibenden 3 Ecken zu finden (siehe Abb. 5.5). Sind diese gefunden wird innerhalb dieses Rechtecks wiederum ein Rasterverfahren durchgeführt, dessen horizontale und vertikale Schrittweite dem minimalen Kreisdurchmesser entspricht. Dieser Durchlauf dient der Suche nach den 4 Kreisen der Marke, die untereinander zu finden sein müssen. Sollte einer der Kreise, d.h. ein helles Pixel, gefunden werden, wird von diesem Punkt ausgehend eine Eckpunktsuche in 8 Richtungen (jeweils 45° verschoben) durchgeführt. Aus dem Schnittpunkt den Positionen der Eckpunkte wird mit Hilfe des arithmetischen Mittels eine subpixelgenaue Mittelpunktbestimmung durchgeführt. Anhand der 4 Mittelpunkte der Kreise kann nun die Decodierung der ID bzw. die Bestimmung des Doppelverhältnisses vorgenommen werden. Der errechnete Wert wird dann auf das Enthalten sein im Doppelverhältnissbereich einer erstellten Marke geprüft. Ist dieser Tests bei einer Marke positiv, wurde sie erfolgreich gefunden und identifiziert. Durch das Vorgehen per Rasterschritt und der Suche nach der linken oberen Ecke, wird auch klar, dass dieser Algorithmus keine rotierten oder stark verzerrten Marken finden kann. Aus diesem Grunde gibt es einige veränderbare Parameter, sodass auch wenig verzerrte und gedrehte Marken gefunden werden können.

Wippich hat sich wohl aufgrund der von vermuteten Geschwindigkeitsvorteilen dazu entschlossen sowohl einen eigenen Kantenerkennungsalgorithmus als auch einen eigenen Algorithmus zur Bestimmung von Ecken zu entwickeln. Ob die Identifikation der Marken durch diese Eigenentwicklung wirklich schneller läuft, als mit Standardverfahren der Bildverarbeitung, ließe sich erst durch eine Vergleichsimplementation endgültig klären.

5.1.3.1. Kantenerkennungsalgorithmus

Das in [Wippich, 2008a] vorgestellte Verfahren ist in der Lage sowohl harte (direkt zwei benachbarte Pixel) als auch weiche Kanten (über mehrere Pixel) zu erkennen. Dazu wird von einem Anfangswert (*init*) aus, der dem Verfahren zusammen mit einer Suchrichtung (8 sind möglich) vorgegeben wird, ein möglicher Startwert gesucht. Dieser Startwert wird durch eine Iteration in

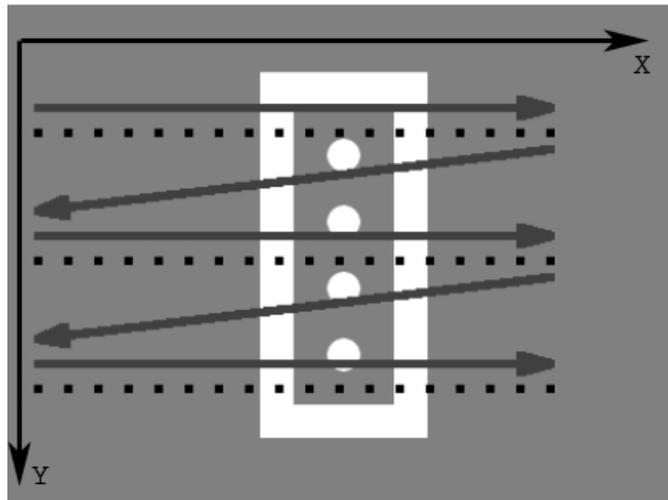


Abbildung 5.4. – Rasterverfahren zur Suche nach Doppelverhältnismarken. Abbildung aus [Wippich, 2008a]

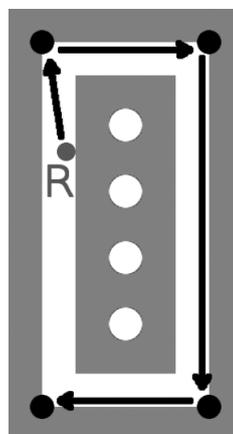


Abbildung 5.5. – Eckensuche auf der Außenumrandung. Abbildung aus [Wippich, 2008b]

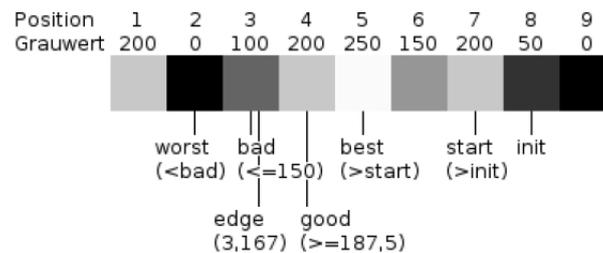


Abbildung 5.6. – Beispiel für Kantensuche. (festgelegte Grauwert-Differenz von *best* und *worst*: 75%) Abbildung aus [Wippich, 2008b]

und entgegengesetzt der Suchrichtung bestimmt, indem jedes Pixel mit dem vorangegangenen per Grauwert verglichen wird. Der hellste (also größte) Grauwert, der nach dem Anfangswert, egal in welcher der beiden Richtungen folgt, ist dann automatisch der Startwert (*start*). Von diesem Startwert ausgehend wird in Suchrichtung nach einem Pixel gesucht, das größer oder gleich dem Startwert ist. Dieser Wert wird in [Wippich, 2008a] als *best* bezeichnet. Ausgehend von diesem Wert wird ebenfalls in Suchrichtung ein Pixel gesucht, dessen Grauwert eine vorher einstellbare Differenz zum Grauwert des *best*-Pixel überschreitet. In diesem Fall ist eine Kante gefunden und das entsprechende Pixel *bad* genannt. Zusätzlich wird noch ein schlechtester Wert *worst* bestimmt, um so per Bildung eines gewichteten Arithmetischen Mittels über die Pixel von *best* bis *worst* eine subpixelgenaue Bestimmung der Kante vorzunehmen (siehe Abb. 5.6). Dies ist für die exakte Bestimmung der Kreisrandpunkte wichtig. Das Pixel *worst* selbst wird gefunden, indem von *bad* ausgehend das letzte Pixel bestimmt wird, das einen kleineren Grauwert als *bad* hat. Durch einen vorher festgelegten Prozentsatz der Grauwert-Differenz von *best* und *worst* kann so ausgehend von *worst* eine pixelgenaue Position (*good*) der Kante bestimmt werden. Die nur pixelgenaue Bestimmung ist z.B. bei den Kantenpositionen auf der Außenmarkierung völlig ausreichend.

Mit diesem Verfahren können allerdings nur Kanten von weiß nach schwarz gefunden werden, was für dieses Einsatzgebiet aber auch ausreicht, da immer von einem Punkt auf der Außenmarkierung oder einem der Kreise ausgegangen wird.

5.1.3.2. Bestimmung von Ecken

Der in Kapitel 5.1.3.1 vorgestellte Kantenerkennungsalgorithmus wird in [Wippich, 2008a] nun für die Bestimmung von Eckkanten, also der Kante, die am weitesten in einer Suchrichtung liegt, eingesetzt. Ausgehend von einem Startpunkt wird senkrecht zur Suchrichtung der Ecke (es sind wieder 8 möglich) nach einer Kante gesucht. Wurden Kanten gefunden wird von dem gefundenen Pixel und vom Startpunkt aus versucht in Suchrichtung weiter voranzukommen. Ist es von keinem Punkt aus mehr möglich weiterzukommen ist eine Eckkante gefunden, sonst werden diese Schritte weiter durchgeführt, bis eine erreicht ist. (siehe Abb. 5.7) Dieses Verfahren wird auch zur Kreisbestimmung eingesetzt, in dem in allen 8 Richtungen nach dem

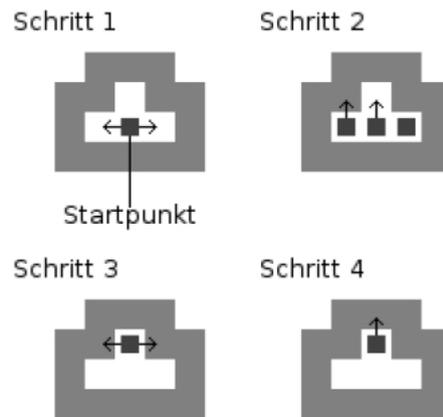


Abbildung 5.7. – Beispiel für Eckensuche. Suchrichtung: oben. Abbildung aus [Wippich, 2008b]

jeweils am weitesten außen liegenden Punkt gesucht wird. Diese 8 Punkte beschreiben dann den Kreis.

5.2. Analyse

5.2.1. Materialien

Zur Analyse der Nutzungseignung der Marke, für die Posenbestimmung eines auf dem Boden fahrenden Fahrzeugs, wurden einige Testvideos im Flur des 7. Stockwerks des E/I-Hochhaus der HAW Hamburg erstellt, aus denen Schlüsse bezüglich einer Erweiterung/Veränderung der Marke und ihrem Erkennungsalgorithmus gezogen werden sollen. Einige der Aufnahmen aus diesem Kapitel entstammen auch der Arbeit von Matthias Wippich (siehe [Wippich, 2008a]), diese sind entsprechend mit Quellennachweis gekennzeichnet.

Die von Wippich durchgeführten Aufnahmen wurden mit einer nicht genauer genannten Kamera, deren Daten im Berechnungsbeispiel in Abschnitt 5.1.2 zu finden sind, durchgeführt. Die Aufnahmen für diese Analyse wurden mit 15 fps bei voller Auflösung der in Tabelle 2.1 beschriebenen Kamera, sowie automatischer Belichtungszeit und Helligkeits- und Kontrastanpassung erstellt und unkomprimiert im Y800 Format gespeichert. Die Auswertung der Aufnahmen wurde auf dem in Tab. 2.3 näher spezifizierten Rechner vorgenommen.

5.2.1.1. Anpassung der Erkennungssoftware von Wippich

Da die von Wippich während seiner Arbeit (siehe [Wippich, 2008a]) erstellte Software als in C++ geschriebene Konsolenanwendung vorliegt, muss hier eine Verbindung zu dem von

Impresario zur Macro-Erstellung mitgelieferten Klassen geschaffen werden. Hierzu wird das Entwurfsmuster Fassade genutzt, welches die technischen Klassen des Konsolenprogramms kennt und diese über eine einheitliche Schnittstelle der von Impresario generierten Macro-Klasse zur Verfügung stellt (siehe Abb. 5.8). Dadurch ist eine lose Kopplung zwischen dem eigentlichen Macro und dem Erkennungsalgorithmus gewährleistet. Es ist somit kein Zugriff der Macro-Klasse auf interne Strukturen der Erkennung möglich/nötig, da dies alles von der Fassade geregelt und entsprechend weitergeleitet wird. Es sind dennoch einige Änderungen des Originalcodes von Wippich nötig, da die Aufgaben, welche sonst vom Hauptprogramm der Konsolenanwendung wahrgenommen werden, nun durch die Fassade, oder die Macro-Oberfläche durchgeführt werden müssen:

- Die erstellten Marken mit allen Daten¹⁷ werden in einer XML-Datei gespeichert. Die Aufgabe des Einlesens bei Programmstart übernimmt nun die Fassade.
- Für die Konsolenanwendung existiert eine zweite XML-Datei, in der Parameter des Erkennungsalgorithmus abgelegt sind (siehe Tab. 5.2). Diese werden nun nicht mehr als XML-Datei eingelesen, sondern sind auf der Oberfläche des Macros für den Anwender jederzeit änderbar und werden bei Veränderung an die Fassade weitergegeben.
- Da nun keine Bilder mehr aus dem Dateisystem eingelesen werden müssen, sondern mit den durch andere Impresario-Macros, z.B. aus einem Videosignal, bereitgestellten Einzelbildern gearbeitet wird, fällt dies komplett weg. Als Eingabebild wird das Bild genutzt, welches über den entsprechenden Eingang des Macros geliefert wird und dann an die Fassade weitergeleitet wird.
- Die Ausgabe beschränkt sich nun nicht mehr, wie beim alten Konsolenprogramm, auf die erkannten Marken und deren Daten, sondern liefert zur optischen Kontrolle des Ergebnisses, über einen Macro-Ausgang, ein Bild zurück, auf dem die Marken rot markiert sind.



Abbildung 5.8. – Struktur des erstellten Macros. Zugriff auf Erkennungsalgorithmus nur über Fassade möglich.

5.2.1.2. Marken

Für die Tests standen 3 Marken, die von Matthias Wippich während seiner Arbeit hergestellt wurden, zur Verfügung. Diese sind mit den Werten der Beispielrechnung aus Kapitel 5.1.2 konfiguriert worden, genauere Angaben sind in Tab. 5.3 zu finden. Dass die in dieser Arbeit

¹⁷Der Doppelverhältnissbereich und dessen Zuordnung zu einer ID.

Parameter	Bedeutung
<i>circleWidthMinDefault</i>	Minimale erlaubte Breite eines zu erkennenden Kreises in Pixel.
<i>circleHeightMinDefault</i>	Minimale erlaubte Höhe eines zur erkennenden Kreises in Pixel. Die Unterscheidung zwischen <i>circleWidthMinDefault</i> und <i>circleHeightMinDefault</i> erfolgt aufgrund der Abbildung der Kreise als Ellipse bei perspektivischer Verzerrung.
<i>reflectiveSheetGrayScaleValueMin</i>	Minimaler Wert eines Pixels welches auf retroreflektierenden Folie liegt.
<i>stripeSlopeVerticallyMax</i>	Maximale erlaubte vertikale Neigung der Außenmarkierung in $^{\circ}$.
<i>stripeSlopeHorizontallyMax</i>	Maximale erlaubte horizontale Neigung der Außenmarkierung in $^{\circ}$.
<i>slopeDifferenceMaxPX</i>	Toleranzwert der Winkelbestimmung in Pixel.
<i>cornerErrorMaxPX</i>	Maximaler Abstand zweier Pixel um noch als selbe Ecke zu gelten in Pixel.
<i>edgeDifferenceMin</i>	Differenz zwischen 2 Grauwerten, damit der Kantenkennungsalgorithmus eine Kante feststellt.
<i>edgeGoodRatio</i>	Prozentsatz der Grauwertdifferenz zwischen den Endpixeln der Kante, die reicht, um den Punkt als pixelgenaue Bestimmung der Kante anzusehen.

Tabelle 5.2. – Parameter des Macros CrossRatioTargets zur Erkennung von Doppelverhältnismarken.

verwendete Kamera eine doppelt so hohe Auflösung hat führt zu keinen Problemen bei der Erkennung, da die auf die Daten dieser Kamera konfigurierten Marken, bis auf Rundungsfehler, exakt die selben wären. Dies wird durch die kleinere Brennweite und die kleinere Größe der Pixel auf dem Sensor möglich. Durch die höhere Auflösung wird eine Vervierfachung des Bildausschnitts erreicht werden (Verdopplung horizontal und vertikal), bei gleicher Anzahl von Pixel, durch die eine Marke im Bild dargestellt wird. Als Nachteil steht dem auch die vierfache Datenmenge pro Einzelbild gegenüber.

Die Marken haben die fortlaufenden IDs 1 bis 3 und der Hintergrund sind aus schwarzer

	<i>Marke 1</i>	<i>Marke 2</i>	<i>Marke 3</i>
Angestrebte Streckenlänge AB in mm	50,0	53,0	55,0
Angestrebte Streckenlänge BC in mm	225,0	216,0	209,0
Angestrebte Streckenlänge CD in mm	50,0	54,0	60,0
Min. erk. min. erst. Doppelverhältnis	1,03106	1,03749	1,04479
Minimal erstelltes Doppelverhältnis	1,03334	1,04006	1,04767
Angestrebtes Doppelverhältnis	1,03418	1,04102	1,04873
Maximal erstelltes Doppelverhältnis	1,03504	1,04198	1,04980
Max. erk. max. erst. Doppelverhältnis	1,03747	1,04473	1,05286
Gesamtbreite der Marke in mm	175	175	175
Gesamthöhe der Marke in mm	500	500	500
Kreisdurchmesser in mm	25	25	25
Konfig. max. Erkennungsentfernung in mm	5363	5363	5363
Kreisgröße bei max. Erkennungsentfernung in Pixel	5	5	5

Tabelle 5.3. – In Kapitel 5.2 für Tests verwendete Marken.

Pappe gefertigt. Die weiße Außenumrandung besteht aus retroreflektierender Folie vom Typ SOLAS Grade (siehe Kapitel 2.2.5). Sowohl bei diesen als auch bei den in [Wippich, 2008a] beschriebenen Tests diente eine normale, hinter der Kamera platzierte, Schreibtischlampe als Lichtquelle ohne Infrarotfilter. Die Werte für die Parameter des Erkennungsalgorithmus sind Tabelle 5.4 zu entnehmen.

5.2.2. Ort der Anbringung der Marken

Im Rahmen dieser Arbeit wurden 3 verschiedene Orte für die Markenbringung im Flur der HAW ausprobiert.

5.2.2.1. An den Seitenwänden

Die Anbringung der Marken an den Seitenwänden (siehe Abb. 5.9) ist keine praktikable Lösung, da im Beispielbild nur die rechten beiden Marken erkannt werden. Auch durch die Veränderung der Toleranzen bezüglich der Verzerrungswinkel des Erkennungsalgorithmus ist die linke Marke aus dieser Position nicht erkennbar. D.h. alle Marken an den Seitenwänden müssten in den Raum hinein gehängt werden und könnten nicht mit ihrer größten Fläche auf die Wand geklebt werden, da sie dort nicht mehr erkannt werden. Abhilfe würde u.U. ein Objektiv mit einem noch größeren Bildwinkel schaffen, dessen Abbildung wäre dann aber noch stärker verzeichnet, weshalb dies hier keine gute Lösung darstellt. Außerdem bleibt das Problem, dass bei starker Verzerrung die mögliche Erkennungsentfernung sinkt. Dies müsste dann in einer großen Anzahl von Marken enden, die für die Bestückung des kompletten Flures benötigt werden. Das in den Raum hängen der Marken stellt auch keine besonders gute Alternative dar, da hierdurch die Nutzbarkeit des Flures durch Personen eingeschränkt wäre.

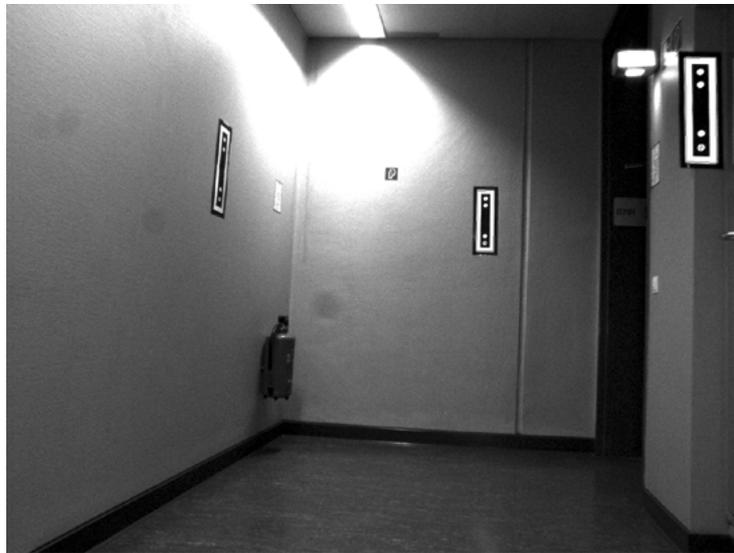


Abbildung 5.9. – Anbringung der Marken an den Seitenwänden. Erkannt werden nur die beiden rechten Marken.

5.2.2.2. Direkt unter der Decke

Eine andere Möglichkeit stellt die horizontale (wahlweise auch die vertikale) Anbringung der Marke mit ihrer größten Fläche direkt an der Flurdecke dar (siehe Abb. 5.10). Aber auch dies führt zu Problemen, da in diesem Beispielbild nur 2 der 3 Marken erkannt werden, weil die dritte Marke zu stark verzerrt ist und dadurch zu klein abgebildet wird. Diese Verzerrung könnte man auflösen, indem die Kamera direkt Richtung Decke gerichtet wird, was aber bei einer Entfernung von der Kamera zur Decke von ca. 2,50 m zu einem sehr kleinen Bereich führen würde, der auf dem Bild zu sehen wäre. Auch hier bräuchte man ein Objektiv mit einem größeren Bildwinkel, welches stark verzeichnet.

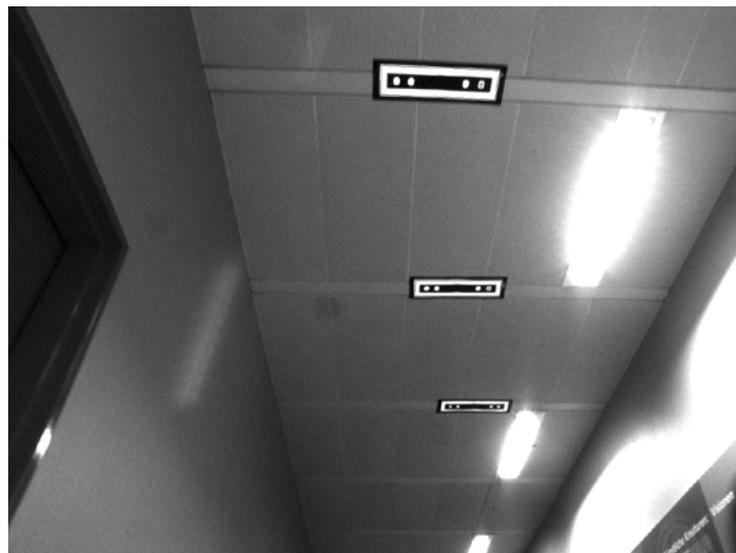


Abbildung 5.10. – Anbringung der Marken mit ihrer größten Fläche direkt unter der Decke. Erkannt werden nur die beiden vorderen (im Bild oberen) Marken.

5.2.2.3. Horizontal von der Decke hängend

Die beste Möglichkeit die Marke anzubringen ist das horizontale unter die Decke hängen, so dass die Marke horizontal orientiert im Bild erscheint¹⁸ und die größte Fläche der Marke damit senkrecht zur Decke ausgerichtet ist (siehe Abb. 5.11). So kann die gesamte maximale Erkennungsentfernung der Marken ausgenutzt werden und auch die möglichen Verzerrungen halten sich in Grenzen. Zur Erkennung der Marken ist aber eine Veränderung des Erkennungsalgorithmus notwendig, da dieser nur auf die Erkennung von vertikal im Bild abgebildeten Marken ausgelegt ist. Dies erfordert aber keinerlei große Umbauten und soll hier deshalb auch nicht

¹⁸Längste Seite des Bildes ist (ungefähr) parallel zur längsten Seite der Marke.

weiter thematisiert werden, da es genügt das Bild intern um 90° zu drehen. So ist es nun wieder möglich den alten Algorithmus zu nutzen. Alternativ könnte man in diesem Flurszenario auch die Bildebene selbst um 90° drehen, da der horizontale Bildwinkel größer als der vertikale Bildwinkel ist und damit bei einer Neigung der Kamera in Richtung Flurdecke u.U. mehr Marken im Bild abgebildet werden.

Um Marken zu sparen sollte die Marke außerdem von ihrer Vorder- und Rückseite erkennbar sein. Zu diesem Zweck muss ein fünfter Kreis auf der Marke installiert werden, um die Eindeutigkeit des Messpunktes bei der Erkennung von beiden Seiten zu gewährleisten.



Abbildung 5.11. – Anbringung der Marken direkt horizontal von der Decke hängend.

5.2.3. Erkennungsentfernung

Bei den Tests im Rahmen dieses Kapitels wurden, wenn nicht anders angegeben, die in [Wip-pich, 2008a] empfohlenen Werte für die Parameter des Erkennungsalgorithmus genutzt (siehe Tab. 5.4). Bei den Tests erwies sich insbesondere *circleHeightMinDefault* mit dem Wert 5 als zu restriktiv. Durch diesen Wert soll eigentlich gesichert werden, dass die erkannten Kreise mindestens die Anzahl an Pixeln im Durchmesser auf der maximalen Erkennungsentfernung haben, die bei der Konfiguration angegeben wurde. Mit dem Wert 4 ließen sich dort aber durchgängig bessere Ergebnisse erzielen. Durch das Herabsetzen des Parameters auf den Wert 2 kann sogar eine Erkennung aus ca. 15m erreicht werden (siehe Abb. 5.12), obwohl dies nicht zu empfehlen ist, da dann zu wenig Pixel zur Verfügung stehen, um eine korrekte und genaue Messung der Kreismitte durchführen zu können. Diese Messung ist aber immerhin noch so genau, dass das Doppelverhältnis mit einem Wert von 1,05273 korrekt erkannt und die Marke

Parameter	Wert
<i>circleWidthMinDefault</i>	2
<i>circleHeightMinDefault</i>	5
<i>reflectiveSheetGrayScaleValueMin</i>	75
<i>stripeSlopeVerticallyMax</i>	15
<i>stripeSlopeHorizontallyMax</i>	40
<i>slopeDifferenceMaxPX</i>	1,2
<i>cornerErrorMaxPX</i>	3
<i>edgeDifferenceMin</i>	50
<i>edgeGoodRatio</i>	0,75

Tabelle 5.4. – In [Wippich, 2008a] empfohlene Werte für die Parameter des Erkennungsalgorithmus.

als Marke Nr. 3 identifiziert wird. außerdem erhöht sich die Zeit, die für die komplette Auswertung des Bildes benötigt wird (von ca. 400 ms auf ca. 1 s), da die Abmessungen des kleinsten möglichen Kreises sinken und somit die Schrittweiten des Rasters kleiner werden.

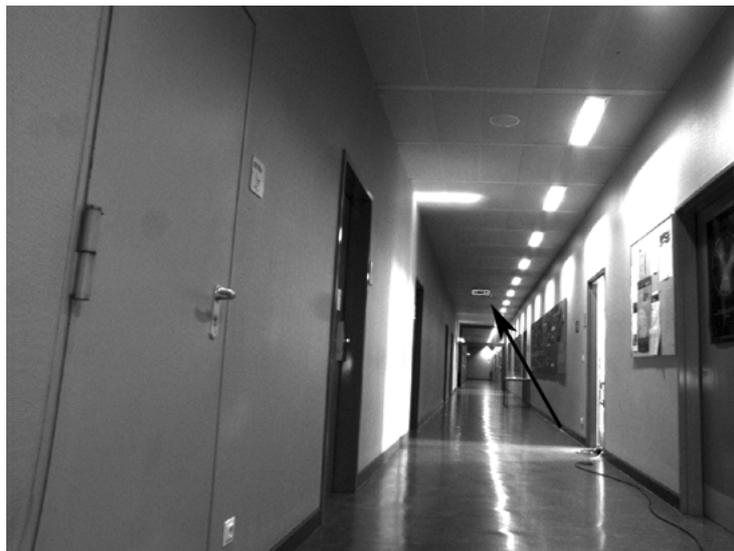


Abbildung 5.12. – Erkennung durch Veränderung der Parameterwerte sogar aus ca. 15 m möglich.

5.2.4. Erkennungsgeschwindigkeit

Da sich die Anzahl der Pixel durch die höhere Auflösung der genutzten Kamera vervierfacht hat, dauert auch die Erkennung ca. viermal so lange, wie bei den Aufnahmen aus [Wippich, 2008a]. Die Erkennung von 5.9 dauert auf dem in Tab. 2.3 beschriebenen Testrechner, mit den Werten aus Tabelle 5.2 ca. 300 ms (bei *circleWidthMinDefault* = 4 erhöht sich die Laufzeit noch weiter). Dies ist für die in [Manske, 2008b] angestrebte Abtastrate von 10 Bildern pro Sekunde für das Gesamtsystem zu viel. Die Geschwindigkeit des Rasterverfahrens hängt, wie in Kapitel 5.1.3 gesehen, maßgeblich von der Anzahl vermeintlich zur retroreflektierenden Folie der Marke gehörenden Pixel ab. Diese Zugehörigkeit wird über den Wert Parameters *reflectiveSheetGrayScaleValueMin* gesteuert. In [Wippich, 2008a] wird hier ein Wert von 75 vorgeschlagen, da bei den dortigen Tests keine zu einer Marke gehörende Stelle der Folie einen geringeren Grauwert besaß. Durch eine Erhöhung auf den Grauwert 100 können bei der Auswertung von Abbildung 5.9 schon ca. 100 ms gespart werden. Eine weitere Komponente, die die Geschwindigkeit verringert ist, wie in Kapitel 5.2.3 gesehen, die kleinste Abmessung der Kreise nach denen gesucht wird, da hierdurch die Größe der Schrittweiten des Rasters bestimmt werden. Je kleiner die Kreise werden dürfen, desto langsamer die Erkennung. Wobei diese beiden Parameter beim Auffinden eines Kreises kontrolliert werden und zu kleine Kreise abgelehnt werden, womit entscheidend zur Sicherheit bei der Identifizierung des Doppelverhältnisses beigetragen wird.

Als Erweiterung des Erkennungsalgorithmus kommt die schon in Kapitel 4 vorgestellte Verfolgung der Marken über mehrere Bilder zum Einsatz, um nicht immer das gesamte Bild durchsuchen zu müssen. Eine weitere Verbesserung könnte über die Nutzung der Infrarotstrahler inkl. entsprechendem Filter für die Kamera aus Kapitel 2.2.6 erzielt werden.

5.2.5. Sicherheit des Doppelverhältnisses zur Codierung der ID

Da das Doppelverhältnis bei diesen Marken zur Codierung der ID dient, muss weitgehend ausgeschlossen werden, dass sich durch die Betrachtung aus unterschiedlichen Entfernungen oder Winkeln das erkannte Doppelverhältnis so stark verändert, dass es außerhalb seines konfigurierten Bereiches landet. Würde dies passieren kommt es zu einer Fehlidentifikation, die neben der Identifikation eines anderen Elements im Bild als Marke der folgenschwerste Fehler ist, der auftreten kann. Die Identifikation einer eines anderen Bildelements als Marke ist durch die Struktur der Marke aber sehr unwahrscheinlich und bis jetzt noch nicht aufgetreten. Die Fehlidentifikation einer Marke konnte bei den bisherigen Tests ausschließlich durch das Zulassen zu kleineren Werten für die Parameter *circleWidthMinDefault* und *circleHeightMinDefault* (siehe. Tab. 5.2), wodurch die Messung der Kreismitten sehr ungenau wurde und auch noch stärker von der Beleuchtung abhing. Dadurch kam es vereinzelt zu Fehlidentifikationen verursacht durch ein falsch gemessenes Doppelverhältnis. Die o.g. beiden Parameter sollten immer so gewählt werden, dass der Parameter ,welcher den Kreisradius in Richtung der Messung des Doppelverhältnisses angibt (*circleHeightMinDefault*), mind.

einen Wert von 4-5 Pixel hat, da sich diese Werte in den Tests als sicher erwiesen haben. Der andere Parameter kann dann auch kleinere Werte annehmen, um perspektivische Verzerrungen in einer Richtung auszugleichen, aber auch hier sollten möglichst Werte zwischen 4 und 5 Pixel verwendet werden.

Es kommt also bei der Identifikation auf die Funktionsfähigkeit der subpixelgenauen Bestimmung der Kreismitten an, die wie in Abbildung 5.2.3 zu sehen ist auch noch aus einer Entfernung, die fast dreimal der konfigurierten maximalen Erkennungsentfernung entspricht korrekte Ergebnisse liefert. Dennoch sollte diese potentielle Gefahr wie o.g. nicht unterschätzt werden, da es leicht zu Fehlidentifikationen kommen kann, was durch ein Beispiel (siehe Tab. 5.3) mit zwei Marken, basierend auf der in Kapitel 2.2.2 eingeführten Linsengleichung, verdeutlicht werden soll. Die Problematik der Beleuchtung wird nicht berücksichtigt aber auch so kann die Gefahr illustriert werden. An diesem Beispiel (siehe Tab. 5.5) ist zu sehen, dass sich bei dieser Konfiguration nur noch ein Unterschied der Streckenlängen im Bereich unter einem Pixel ergibt.

	Marke ID 1	Marke ID 2
Pixel für Strecke \overline{AB}	2,0049	2,1252
Pixel für Strecke \overline{BC}	9,0223	8,6614
Pixel für Strecke \overline{CD}	2,0049	2,1653

Tabelle 5.5. – Vergleich der Marken mit der ID 1 und der ID 2 (siehe Tab. 5.3) bezogen auf die Abbildung der Strecken ihres Doppelverhältnisses auf die Anzahl von Pixeln auf dem dem Kamerasensor (siehe Tab. 2.1) bei der konfigurierten maximalen Erkennungsentfernung von 5363 mm)

Neben der Entfernung stellt die perspektivische Verzerrung ein anderes Extrem dar, welches für die Änderung des Doppelverhältnisses einer Marke verantwortlich sein kann. Das Doppelverhältnisses ist zwar invariant gegenüber perspektivischen Verzerrungen, was aber nicht die zur Messung genutzten Kreise miteinbezieht, sodass die in Kapitel 3.2.1 beschriebene Abbildungsexzentrizität eine Rolle spielen kann, diese in der Praxis aber eher zu vernachlässigen ist, da sich die bodennahe Kamera zu weit von der an der Decke befestigten Marke entfernt befindet. Zu dieser Problematik wurden einige Testaufnahmen gemacht (siehe Abb. 5.13) und es konnte auch hier kein Winkel konstruiert werden, aus dem die Marke falsch identifiziert wurde. Bei der perspektivischen Verzerrung spielt also eher das grundsätzliche Problem der Verkleinerung der Kreis eine Rolle, als deren Abbildungsexzentrizität.

Generell kann festgehalten werden, dass die konfigurierten Parameter eingehalten werden müssen, da es sonst leicht zu Verschiebungen der Doppelverhältnisbereiche der einzelnen Marken kommen kann und eine Fehlidentifikation möglich wird. Insbesondere ist hierbei auf die korrekte und präzise Anbringung der Kreise zu achten, da sich jede Veränderung an dieser Stelle auf das Doppelverhältnisbereich der Marke auswirkt. außerdem wird hier empfohlen die erstellten Marken vor ihrem ersten Einsatz auf die Genauigkeit der Herstellung zu untersuchen,

um mögliche Fehlidentifikationen früh erkennen zu können. Auch sollten die verwendeten Marken für die Verwendung mit der genutzten Kombination aus Kamera und Objektiv konfiguriert sein, da sonst auch hier ein Fehlerpotential vorliegt.



Abbildung 5.13. – Erkennung der Marke mit der ID 3 (siehe Tab. 5.3 bei Beleuchtung durch Sonnenlicht) aus einer Entfernung von 2,65 m und einem Winkel von ca. 30° zwischen optischer Achse und Boden. Doppelverhältnis wird auch unter diesem Winkel mit 1,05244 korrekt gemessen.

5.3. Vorgenommene Veränderungen

Dieser Abschnitt dient der Beschreibung der im Rahmen dieser Arbeit vorgenommenen Veränderungen und Ergänzungen der Doppelverhältnismarke und des Erkennungsalgorithmus.

5.3.1. Erweiterung der Marke um einen zusätzlichen Kreis

Wie in Kapitel 5.2.2.3 schon ausgeführt ist es für die horizontale Anbringung der Marke unter der Decke und die gleichzeitige Erkennbarkeit von der Vorder- und Rückseite aus, notwendig die Marke um einen fünften Kreis zu erweitern. Damit gibt es von beiden Seiten aus gesehen mit dem mittleren Kreis einen eindeutigen Messpunkt der Marke, der bei nur vier Kreisen nicht bestimmt werden kann. Der fünfte Kreis wird, wie in Abbildung 5.14 zu sehen, am Ende der Marke eingefügt. Der Mindestabstand zum vierten Kreis entspricht wie üblich dem Durchmesser der Kreise, wobei auch ein Maximalabstand von doppelter Länge des Durchmessers definiert ist. Dadurch ergibt sich ein Bereich (in Abb. 5.14 durch die beiden horizontalen Balken

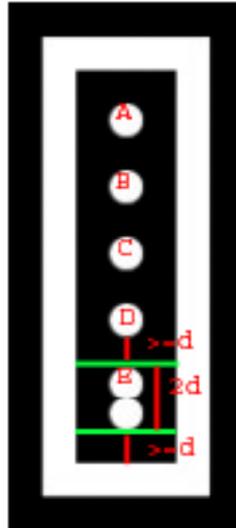


Abbildung 5.14. – Veränderte Doppelverhältnismarke mit fünf Kreisen. d entspricht dem Kreisdurchmesser.

markiert), in dem sich der fünfte Punkt befinden darf. Dieser Bereich dient dazu den fünften Kreis in die Codierung mit einzubeziehen und somit mehrere Möglichkeiten zu bekommen. Mit den fünf Kreisen ist es nun möglich zwei verschiedene Doppelverhältnisse zu bilden, deren Kombination zur Codierung der ID dient (es sind natürlich noch mehr Doppelverhältnisse möglich, diese werden hier aber nicht betrachtet).

Der Konfigurationsalgorithmus muss nur dahingehend geändert werden, dass nach Berechnung des Doppelverhältnisses der oberen vier Kreise (bezogen auf Abb. 5.14), das nur noch einen Teil der gesamten Markenfläche zur Verfügung hat, ein zweites Doppelverhältnis $D(B, C, D, E)$ berechnet wird. Dieses setzt sich dann aus den Strecken \overline{BC} , \overline{CD} und \overline{DE} zusammen, wobei ausschließlich die Länge der Strecke \overline{DE} gewählt werden kann, da die anderen Strecken schon durch das erste Doppelverhältnis $D(A, B, C, D)$ festgelegt sind. Durch den schon angesprochenen Bereich für die Platzierung des fünften Kreises auf der Marke ist es nun möglich für jedes Doppelverhältnis $D(A, B, C, D)$ u.U mehrere Doppelverhältnisse $D(B, C, D, E)$ zu erstellen und somit die Anzahl der codierbaren IDs zu erhöhen. Dies ist nicht immer möglich, da es nur erlaubt ist noch nicht erstellte Kombinationen der beiden Doppelverhältnisse zu erzeugen, weil sonst die Eindeutigkeit verloren gehen würde. Es ist also z. B. nicht möglich bei der Existenz der Kombination $D(A, B, C, D) = 1$ und $D(B, C, D, E) = 2$ zu dem Doppelverhältnis $D(A, B, C, D)$ mit dem Wert 2 ein $D(B, C, D, E) = 1$ zu erzeugen, da dann diese Kombination doppelt existieren würde.

In der Realität hat das Doppelverhältnis einer Marke, wie in Kapitel 5.1.2 beschrieben, keinen eindeutigen Wert sondern besteht aus Doppelverhältnissbereichen. Der Konfigurationsalgorithmus achtet deshalb darauf, dass die Bereiche sich nicht überschneiden und somit die

Eindeutigkeit der Kombination erhalten bleibt.

Als Beispiel für die Konfiguration von Exemplaren mit 5 Kreisen dienen die nach den Tests in Kapitel 5.2 erstellten Marken. Diese Marken wurden für die Verwendung im Flur des 7. Stockwerks des E/I-Hochhaus der HAW Hamburg mit der Kamera aus Tabelle 2.1 konfiguriert. Sie sind von den Ausmaßen her doppelt so groß (Höhe: 1000mm, Breite: 350mm u. Kreisdurchmesser: 50mm), wie die Marken aus Tabelle 5.3, da sie für eine doppelt so große Erkennungsentfernung von 10626mm konfiguriert wurden. In dieser Konfiguration lassen sich 32 eindeutig identifizierbare Marken erstellen. Die gleiche Anzahl an Marken kann auch konfiguriert werden, die gleiche Markenfläche mit nur vier Kreisen genutzt wird. Grundsätzlich muss beachtet werden, dass sich mit den 5 Kreisen nicht immer mehr eindeutige IDs codieren lassen als mit 4 Kreisen. Manchmal liegt die Zahl der möglichen eindeutigen IDs mit 5 Kreisen auch unter der Anzahl, die bei 4 Kreisen und gleicher Markengröße möglich ist. Dies hängt hauptsächlich mit der Größe der Kreise im Verhältnis zur Gesamtmarke zusammen, da bei der 5er Marke das erste 4er Doppelverhältnis auf einem kleineren Bereich konfiguriert werden muss, was dazu führen kann, dass bei sehr großen Kreisen nur sehr wenige Doppelverhältnisse gebildet werden können. Somit hängt alles von den möglichen Kombinationen mit dem zweiten Doppelverhältnis ab, wobei hier auch noch auf eindeutige Kombinationen dieser beiden Verhältnisse geachtet werden muss.

Man kann sich als nicht bei jeder Wahl von Parametern für die Konfiguration sicher sein, dass mit dem neuen Ansatz mehr eindeutige Marken zur Verfügung stehen als mit dem Alten. Der entscheidende Vorteil dieser Art der Marke ist also nicht unbedingt die Erhöhung der Anzahl der möglichen eindeutigen IDs, sondern, wie auch Matthias Wippich im Ausblick seiner Arbeit (siehe [Wippich, 2008a]) anmerkt, die Rotationsinvarianz dieser Marke, da nun der Messpunkt immer klar durch den mittleren der fünf Kreise definiert ist.

5.3.2. Verfolgung

Wie schon bei den Tests bezüglich der Erkennungsgeschwindigkeit (siehe. Kapitel 5.2.4) angemerkt wird hier der in Kapitel 4 vorgestellte Tracker genutzt, der es ermöglicht die Geschwindigkeit der Erkennung der Marke in den Einzelbildern eines Videosignals zu erhöhen. Der Tracker wird als Komponente zwischen der Fassade und dem Erkennungsalgorithmus eingefügt (siehe Abb. 5.15). Dies ist sinnvoll, weil der Tracker dem Erkennungsalgorithmus so die zu durchsuchenden Bereiche (Suchfenster) übergeben kann, ohne das die Erkennung etwas vom Tracker weiß.

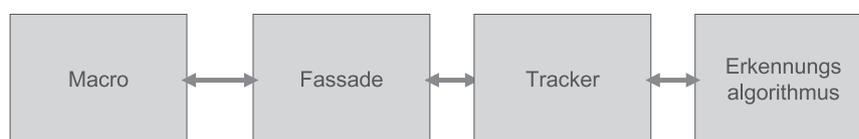


Abbildung 5.15. – Einordnung des Trackers in die vorhandenen Komponenten des Macros.

Als Position der Marke, die verfolgt werden soll, dient nicht der Messpunkt, sondern die Koordinaten der vier Ecken der Außenumrandung. Dies hat gegenüber der Nutzung des Messpunktes den Vorteil, dass so schon die Orientierung der Marke vorliegt und nicht extra berechnet werden muss. Außerdem ist so eine einfache Verfolgung der Marke selbst bei perspektivischen Verzerrungen möglich, da bei jeder Schätzung jeder der vier Eckpunkte isoliert betrachtet wird.

Eine weitere Möglichkeit die Geschwindigkeit der Erkennung zu optimieren, die nicht unbedingt mit dem Trackingverfahren zusammenhängt, aber trotzdem in diesem Abschnitt beschrieben werden soll, ist die Nutzung von Vorwissen über die ungefähre Position der Marken. In Abbildung 5.11 ist zu sehen, dass eine Marke die horizontal von der Decke hängend angebracht ist je nach gewählter Ausrichtung der Kamera u.U. nicht in der unteren Hälfte des Bildes auftreten kann. Aus diesem Wissen kann dann gefolgert werden, dass dieser Bereich des Bildes auch nicht durchsucht werden muss. Dies ist in der vorliegenden Arbeit am Beispiel der unter der Decke hängenden Marken implementiert worden, in dem es möglich ist das Bild nach einen variabel einstellbaren Prozentsatz zu zerteilen, wobei der untere Teil des Bildes eben nicht mehr durchsucht wird.

Des Weiteren lässt sich auch durch dynamische Werte für die Schrittweiten des Rasters im inneren Rechteck der Marke Zeit sparen, indem die Werte auf Basis der bekannten Länge und Breite dieses Rechtecks angepasst werden. Bei der Originalimplementation wird hier auch bei sehr großen Marken mit den eingestellten kleinsten möglichen Kreisabmessungen gearbeitet.

5.4. Abschließender Test

Dieser Test soll die grundsätzliche Tauglichkeit dieser veränderten Marke mit 5 Kreisen bei horizontaler Anbringung unter der Decke zeigen. Die Marken wurden zu diesem Zweck aber nur von einer Seite beklebt, um den Aufwand so gering wie möglich zu halten, da die Herstellung der Marken einige Zeit in Anspruch nimmt. Auch dieser Test wurde in den gleichen Räumlichkeiten durchgeführt, wie die Tests im Kapitel 5.2. Es wurde wieder die gleiche Kamera verwendet (siehe. Tab. 2.1), diesmal aber mit anderen Parametern. Es wurde kein Auto-Gain mehr genutzt, sondern eine manuelle Einstellung auf den Wert 460 sowie eine manuelle Belichtungszeit von $\frac{1}{120}$ s. außerdem wurde bei diesem Test nur eine Framerate von 7,5 fps genutzt, um das Datenaufkommen zu verringern. Insgesamt ist somit ein unkomprimiertes Video¹⁹ aus 463 Einzelbildern entstanden. Da dieser Tests mit den unter 2.2 beschriebenen Infrarotscheinwerfern gemacht wurde, was den Einsatz des ebenfalls schon genannten Infrarotfilters sinnvoll macht, wurde ebenfalls eine manuelle Blendeneinstellung von $f/2$ vorgenommen, damit trotz der kurzen Belichtungszeit noch genügend Licht auf den Sensor fällt. Der Fokusring des Objektivs wurde so eingestellt, dass die Marken in allen Entfernungen einigermaßen scharf abgebildet wurden.

¹⁹Das Video befindet sich auf der beiliegenden DVD-ROM (crfinaltest.avi).

Als Grundmaterial für die Marken wurde auch hier schwarze Pappe genutzt, welche zur Stabilisierung mit einem der Größe der Marke entsprechenden Stück 2mm dicker Graupappe verstärkt wurde. Als Reflektionmaterial wurde diesmal die Reflexfolie mit der Bezeichnung Series 3210 (siehe Kapitel 2.2.5) genutzt, da diese durch die Hochschule einfacher/günstiger beschafft werden konnte, als die SOLAS Folie. Es wurden insgesamt 5 Marken hergestellt, deren Daten in Tabelle 5.6 zu finden sind. Die Kamera und die 3 Infrarotscheinwerfer befanden sich auf einem Rollwagen, welcher unter den an die Decke angebrachten Marken manuell in Schrittgeschwindigkeit hindurch gefahren wurde. Die Marken waren in einem Abstand von 3,60m horizontal in 3m Höhe unter der Decke befestigt, wobei die gesamte Breite von 2,10m des Flures durch das versetzte ankleben der Marken ausgenutzt wurde. Die hilft den Abstand zwischen den Marken zu maximieren, was die Berechnung der Pose genauer macht. Genauere Informationen zum Versuchsaufbau sind auch der Abbildung 5.17 zu entnehmen. Bei diesem Test wurden für die Parameter die Werte aus Tabelle 5.4 verwendet, mit der Änderung von *circleWidthMinDefault* auf 4 und *circleHeightMinDefault* auf 4. Diese Einstellungen, und durch die Beleuchtung bedingt die Änderung von *edgeDifferenceMin* auf 35 und *reflectiveSheetGrayValue* auf 65, haben ausgereicht, um in jeder Phase 3 Marken im Bild identifizieren zu können. Wie schon in Kapitel 5.2.5 angedeutet, sollte der Wert von 4 entweder bei *circleHeightMinDefault* oder *circleWidthMinDefault* nicht unterschritten werden, um keine Fehlidentifikationen zu riskieren. Als weiterer Parameter wurde der in dieser Arbeit neu eingeführte Parameter *cut* auf den Wert von 0,6 gesetzt, womit nur die oberen 60% des Bildes durchsucht werden, was bei dieser Kameraneigung ausreicht, um immer mindestens 3 identifizierbare Marken im durchsuchten Bereich zu haben²⁰. Hierbei ergab sich auf dem in Tabelle 2.3 genannten Rechner eine Laufzeit²¹ von ca. 27 ms bis ca. 33 ms für das Erkennungsmacro unter Impresario pro Einzelbild. Der höchste Wert entstand, wenn 4 Marken erkannt werden. Die Größe der Marke ist, durch die dynamischen Werte für die Schrittweiten des inneren Rasters, nicht besonders wichtig. Ein Beispiel aus dem Testvideo ist in Abbildung 5.16 zu finden.

Bei einem Durchlauf mit eingeschaltetem Tracking-Algorithmus mit den in Tabelle 4.1 aufgeführten Beispielwerten ergab sich eine Laufzeit ca. 12 ms bis ca. 18 ms (exkl. Einzelbilder, die komplett durchsucht wurden), wobei auch hier der höchste Wert wieder durch Situationen verursacht wurde, in denen 4 Marken im Bild zu erkennen waren. D.h. das Tracking bringt hier eine Ersparnis in der Laufzeit von ca. 45% bis ca. 55%.

5.5. Bewertung

Als Fazit kann mitgenommen werden, dass mit den in dieser Arbeit vorgenommenen Veränderungen, wie 5. Kreis auf der Marke und dem Trackingverfahren, ein funktionierender und

²⁰Drehungen des Fahrzeugs in Richtung Wand sind mit dieser Konfiguration aber ausgeschlossen.

²¹Laufzeittests wurden durchgeführt ohne ein Ausgabebild zu erzeugen, da dies ca. 4ms Zeit kostet.

	Marke 1	Marke 2	Marke 3	Marke 26	Marke 27
<i>Doppelverhältnis Nr. 1</i>					
Angestrebte Streckenlänge AB in mm	100,0	100,0	104,0	171,0	183,0
Angestrebte Streckenlänge BC in mm	300,0	300,0	287,0	132,0	122,0
Angestrebte Streckenlänge CD in mm	100,0	100,0	108,0	196,0	192,0
Min. erk. min. erst. Doppelverhältnis	1,06126	1,06126	1,07236	1,47713	1,54272
Minimal erstelltes Doppelverhältnis	1,06581	1,06581	1,07747	1,50381	1,57364
Angestrebtes Doppelverhältnis	1,06666	1,06666	1,07842	1,50883	1,57947
Maximal erstelltes Doppelverhältnis	1,06752	1,06752	1,07939	1,51390	1,58536
Max. erk. max. erst. Doppelverhältnis	1,07234	1,07234	1,08479	1,54264	1,61885
<i>Doppelverhältnis Nr. 2</i>					
Angestrebte Streckenlänge AB in mm	300,0	300,0	287,0	132,0	122,0
Angestrebte Streckenlänge BC in mm	100,0	100,0	108,0	196,0	192,0
Angestrebte Streckenlänge CD in mm	100,0	100,0	100,0	125,0	100,0
Min. erk. min. erst. Doppelverhältnis	1,55213	1,76253	1,49514	1,17282	1,14129
Minimal erstelltes Doppelverhältnis	1,59236	1,80931	1,53021	1,18378	1,15155
Angestrebtes Doppelverhältnis	1,6	1,04102	1,53684	1,18583	1,15348
Maximal erstelltes Doppelverhältnis	1,60772	1,82716	1,54356	1,18790	1,15542
Max. erk. max. erst. Doppelverhältnis	1,65194	1,87852	1,58187	1,19955	1,16638
Gesamtbreite der Marke in mm	350,0	350,0	350,0	350,0	350,0
Gesamthöhe der Marke in mm	1000,0	1000,0	1000,0	1000,0	1000,0
Kreisdurchmesser in mm	50,0	50,0	50,0	50,0	50,0
Konfig. max. Erkennungsentfernung in mm	10626,0	10626,0	10626,0	10626,0	10626,0
Kreisgröße bei max. Erkennungsentfernung in Pixel	5,0	5,0	5,0	5,0	5,0

Tabelle 5.6. – In Kapitel 5.4 für Tests verwendete Marken.

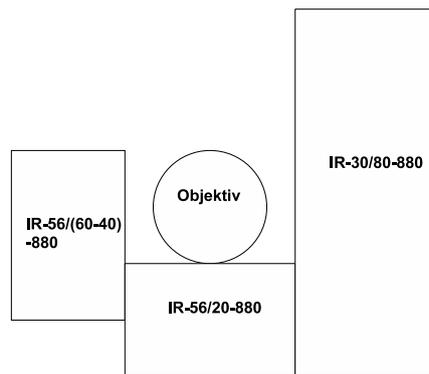
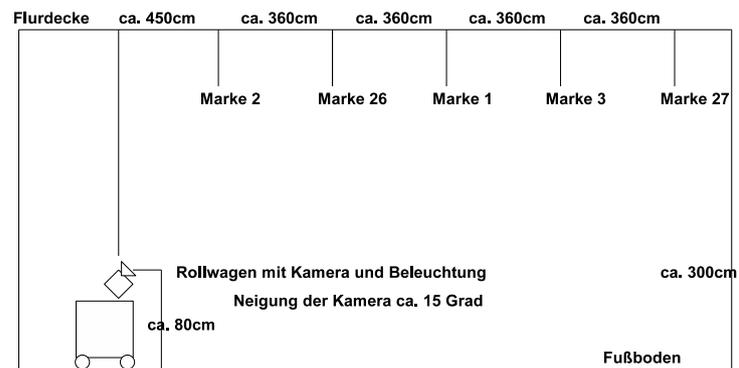


Abbildung 5.16. – Einzelbild mit höchster Laufzeit (ca. 33 ms) ohne Tracking und mit *cut* bei 0,6.

praxistauglicher Markentyp inkl. Erkennung vorliegt. Dieser Typ eignet sich, wie gesehen, sehr gut für lange Flure, aber aufgrund der Markenform grundsätzlich auch für Situationen, in denen eine Marke so an der Wand befestigt werden muss, dass sie nicht in den Raum ragt oder besonders extreme Perspektiven zu erwarten sind²², die bei der Nutzung anderer Markentypen u.U. nicht möglich sind. Auch die Nutzung in anderen Räumlichkeiten/Situationen ist selbstverständlich möglich.

Als größter Vorteil dieser Marke kann die Invarianz der zur Codierung der ID genutzten Doppelverhältnisse gegenüber projektiven Transformationen genannt werden. Diese Marke kann also theoretisch auch bei extremer perspektivischer Verzerrung sicher identifiziert werden. Zur Sicherheit des Doppelverhältnisses kann gesagt werden, dass die Tests zu dieser Arbeit gezeigt haben, dass bei einer ungefähren Einstellung der Software auf die konfigurierte Minimalgröße der Kreise im Bild eine sichere Erkennung auch bei starker perspektivischer Verzerrung und großen Entfernungen möglich ist, ohne dass mit einer Fehlidentifikation gerechnet werden muss. Dass ein anderes Objekt als Marke erkannt wird ist ebenso fast ausgeschlossen, da diese Struktur wohl in den meisten Umgebungen einzigartig ist. Verstärkt wird diese Einzigartigkeit noch durch den Einsatz der Infrarotbeleuchtung. Durch die genannte perspektivische Verzerrung können allerdings sehr große Einfallswinkel an der Reflexfolie entstehen, was zu Problemen bei der Erkennung führt. Die perspektivische Verzerrung führt außerdem dazu, dass für eine Richtung des Rasters sehr kleine Werte angenommen werden müssen, sodass der Algorithmus dadurch u.U. sehr laufzeitintensiv wird. Aus diesem Grund sollte bei zu erwartenden perspektivischen Verzerrungen bei der Konfiguration der Marken lieber ein etwas

²²Hierfür wäre allerdings u.U. die Nutzung anderer Reflexfolie/Materialien nötig.



Ungefähre Anordnung der Infrarotscheinwerfer

Abbildung 5.17. – Versuchsaufbau (inkl. Beleuchtung) für den abschließenden Test der Doppelverhältnismarken

größerer Wert für den Messkreis angenommen werden, um dann in der Erkennung nicht unter 4 Pixel gehen zu müssen.

Als Nachteil kann die relativ geringe Anzahl an eindeutigen IDs die codiert werden können angesehen werden. Die Anzahl variiert je nach Größe der Marke bzw. der angestrebten Pixelgröße der Kreise im Bild. Als weiterer Nachteil kann die Tatsache gewertet werden, dass in anderen Räumlichkeiten als Fluren eine u.U. sehr große Anzahl von Marken nötig ist, da diese maximal von 2 Seiten aus erkannt werden können.

Bei einem abschließenden Test konnte ein guter Kompromiss zwischen Größe der Marke und der Erkennungsentfernung gefunden werden, sodass die Marken mit einer Höhe von 35 cm von der Decke hingen und dadurch die normale Nutzung des Flurs kaum eingeschränkt wird. Die Erkennungsentfernung von mindestens 10 m scheint für den Innenraumgebrauch auch angemessen zu sein, sodass bei gleicher Anforderung und Kamerahardware diese Größe der Marken zu empfehlen ist, sofern man mit den zur Verfügung stehenden 32 eindeutigen IDs auskommt. Sollte man mehr Platz für die Marken haben oder noch weniger eindeutig identifizierbare Marken benötigen ist es aber auch empfehlenswert die Mindestgröße des Messkreises von 5 Pixel zu erhöhen, um die Erkennungsgeschwindigkeit zu beschleunigen.

Insgesamt liegt mit Hilfe der Infrarotbeleuchtung und dem Trackingverfahren ein schnelles System vor, welches auf dem genutzten Testrechner auch bei einer Auflösung von 1280 x 960 Pixel, die angestrebte Erkennungsrate von 10 fps (siehe [Manske, 2008b]) ohne Probleme einhält. Generell kann die Software beide Markentypen (mit 4 und 5 Kreisen) verarbeiten, weshalb es auch weiterhin möglich ist, z.B. in Fällen wo keine Rotationsinvarianz benötigt wird, die 4er Variante zu verwenden. Die angesprochene Rotationsinvarianz der Marke schlägt sich zur Zeit noch nicht in der Software nieder, da hierfür ein komplett neuer Erkennungsalgorithmus implementiert werden müsste, was den Umfang dieser Arbeit aber sprengen würde. Der aktuelle Algorithmus konnte zwar testweise so umgebaut werden, dass er auch beliebig rotierte Marken erkennt (ausgenommen sind Marken die um ca. 45° rotiert sind), dies geht aber zu Lasten der Laufzeit, weshalb dies als wenig geeignet für die hier behandelte Problematik angesehen werden kann. Bei Tests zur Masterarbeit von Nico Manske (siehe [Manske, 2008a]) wurde diese testweise Umsetzung aber eingesetzt (siehe auch Kapitel 7).

Für weitere praktische Erfahrungen mit dieser Marke sei auf die Masterarbeit von Nico Manske (siehe [Manske, 2008a]) verwiesen, da diese dort zu Testmessungen mit einer Genauigkeit von 2 – 3 cm eingesetzt wurde. Die relativ ungenaue Messung ist hierbei laut Manske aber nicht auf die Marken oder das Erkennungsmodul zurückzuführen.

6. Zylindermarke

6.1. Vorstellung

Mit der in Kapitel 5 vorgestellten Doppelverhältnismarke lassen sich für das Szenario der Posenbestimmung/Navigation eines auf dem Boden fahrenden autonomen Fahrzeugs alle gängigen Räumlichkeiten, wenn auch teilweise mit starken Einschränkungen, abdecken. Auffallend ist hierbei das Problem der u.U. hohen Anzahl von benötigten Marken, da diese nur maximal zwei Seiten haben, anhand derer sie zu identifizieren sind. Dies fällt z.B. bei größeren Räumen ins Gewicht, da dort, um von möglichst allen Kamerastandorten aus immer (bzw. häufig) drei Marken im Kamerablickfeld zu haben, viel mehr Marken benötigt werden, als Messpunkte nötig wären, wenn alle Marken aus allen Betrachtungsrichtungen identifizierbar wären. Um dieses Problem mit der horizontal von der Decke hängenden Doppelverhältnismarke zu verringern, könnte man aus drei Markenoberflächen ein Dreieck bilden. Diese Form hätte den Vorteil, dass sie aus mehr Richtungen erkannt werden kann, als die zweiseitige Variante. Durch diese Änderung würde die Marke aber ihren eindeutigen Messpunkt verlieren, was diese Idee unpraktikabel macht. Problematisch sind bei dieser Idee auch weiterhin die teilweise sehr großen Einfallswinkel des Lichts auf der Reflexfolie, die auftreten, wenn die Oberfläche der Doppelverhältnismarke und die Bildebene der Kamera nicht parallel sind. Diese Einfallswinkel treten durch die Richtung Decke geneigte Kamera sowieso auf, vergrößern sich aber noch, wenn die Bildebene zur Markenoberfläche auch noch verdreht (siehe Abb. 6.3) wird. Das Problem der vielen benötigten Marken soll nun (wenigstens teilweise) mit einem im Rahmen dieser Arbeit neuentwickelten Markentyp gelöst werden. In Abbildung 6.1 ist eine zweidimensionale Sicht dieser neuen Marke zu sehen, wohingegen Abbildung 6.2 eine reale Marke zeigt.

Als Ausgangskörper für eine aus möglichst vielen Richtungen sichtbare Marke scheint eine runde Oberfläche am Besten geeignet zu sein. Die beste Lösung wäre eine Kugel, welche mangels praktikabler einfach zu identifizierender Codierung aber ausscheidet. Die Wahl fiel deswegen auf einen Zylinder, dessen Mantelfläche ist, bis auf die Ansicht von oben oder zu weit unterhalb der Marke, aus allen Betrachtungsrichtungen zu erkennen und wird im Kamerabild idealisiert als Rechteck abgebildet, welches immer einen (fast) gleich großen Abschnitt der Mantelfläche zeigt.

Zur Codierung der ID dienen im weiteren noch genauer vorgestellte eindimensionale Barcodes, die in horizontalen Balken auf die Mantelfläche des Zylinders aufgebracht werden. Dies

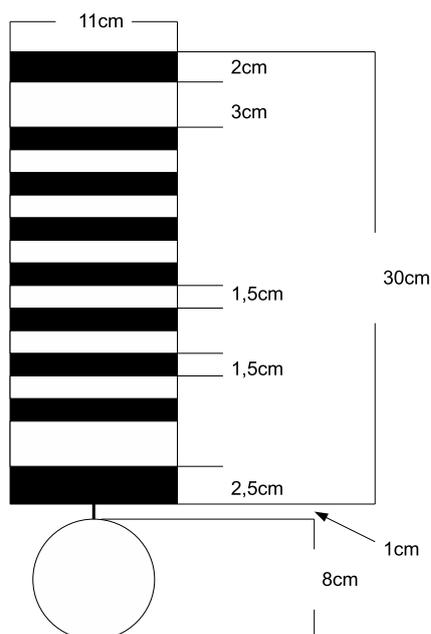


Abbildung 6.1. – 2D-Ansicht der Zylindermarke mit Beispielmessungen (2-aus-8-Code).

hat den Vorteil, dass bei von der oberen bis zur unteren Kante des Bildes durchgezogenen Scanlinien (Abtaststreifen) aus allen Richtungen, aus denen die komplette Höhe der Mantelfläche zu sehen ist, auch immer der komplette Barcode abgetastet werden kann.

In dieser Arbeit wird vorausgesetzt, dass der Zylinder, ähnlich den Doppelverhältnismarken, unter der Decke, befestigt wird. Es sind auch andere Anbringungsorte denkbar, welche aber u.U. zu Lagen der Marke im Bild führen können, die nicht von dem hier vorgestellten Erkennungsalgorithmus erfasst werden. Da der Körper des Zylinders u.U. teilverdeckt sein kann und sich Kreise zur genauen Bestimmung eines Messpunktes besser eignen, wird zur Darstellung dieses Messpunktes eine Kugel unter den Zylinder gehängt. Diese Kugel wird auf dem Kamerabild, bis auf Extremperspektiven, als Kreis abgebildet, dessen Mittelpunkt dem Mittelpunkt der Kugel entspricht und somit einen für die Posenbestimmung benötigten Messpunkt liefert.

Die Balken des Barcodes und die Kugel bestehen aus Reflexfolie, bzw. sind mit dieser beklebt, was den Einsatz der auch bei der Doppelverhältnismarke genutzten Infrarotbeleuchtung möglich macht, um schon durch die Beleuchtung eine weitestgehende Trennung der Marken von anderen Objekten im Bild durchführen zu können.

In den folgenden Abschnitten dieses Kapitels sollen die Herstellung, die Eigenschaften, die Codierung und die Erkennung der Marke genauer erläutert. außerdem erfolgt ein kurzer Überblick über ähnliche Arbeiten.



Abbildung 6.2. – Foto einer Zylindermarke.

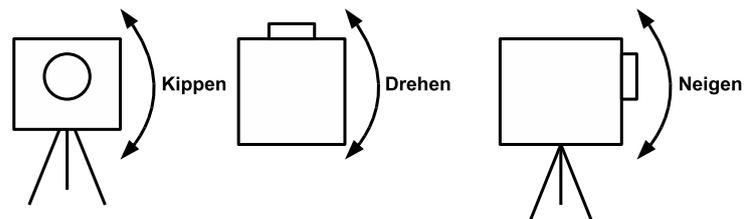


Abbildung 6.3. – Genutzte Begriffe für die Kameraorientierung.



Abbildung 6.4. – Zylindermarke mit 3 Bit Barcode aus [Arıcan, 2004]

6.2. Verwandte Arbeiten

6.2.1. Zylinder

Die Idee einen Zylinder zu Zwecken der Messung o.ä. zu nutzen ist nicht neu, weshalb hier zwei verwandte Arbeiten vorgestellt werden.

In der Arbeit von Zafer Arıcan (siehe [Arıcan, 2004]) wird ein ähnlicher mit einem Barcode versehener Zylinder zur Posenbestimmung eines Fahrzeugs genutzt. Der Zylinder besteht, wie in Abbildung 6.4 zu sehen ist, aus weißem Plastik mit einem aufgeklebtem schwarzen Barcode. Da er jedoch ohne Kugel als Messpunkt ausgestattet ist, ist die Messpunktbestimmung darauf angewiesen den kompletten Zylinder unverdeckt sehen zu können, da der Messpunkt über den Mittelpunkt des sich abbildenden Rechtecks bestimmt wird. Der Code selbst besteht aus zwei Begrenzungsbalken oben und unten, die die Marke von anderen Elementen im Bild unterscheidbar machen sollen. Innerhalb dieses Fensters befindet sich ein 3-Bit Code, dessen Balken in konstantem Abstand aufgetragen werden. Ein fehlender Balken bedeutet hierbei, dass das betreffende Bit nicht gesetzt ist, womit dann insgesamt acht verschiedene Marken möglich sind. Als weitere Einschränkungen wird noch die Annahme getroffen, dass die Bildebene der Kamera immer parallel zum Zylinderlängsschnitt orientiert ist und es somit zu keinen Verzerrungen bezüglich der Balkenbreiten kommt. Dies beschränkt zusammen mit dem kleinen Code die Nutzbarkeit der Marke auf kleine Testparcours (siehe Abb. 6.5).

Eine weiterer Unterschied zu den Aktivitäten an der HAW besteht darin, dass für die Posenbestimmung eine Stereokamera auf dem mobilen Roboter zur Verfügung steht. Die Erkennung selbst erfolgt über ein Template Matching anhand von zwei 4×12 Pixel großen Schablonen, die einen schwarz-weiß-Übergang bzw. einen weiß-schwarz Übergang ausdrücken. Anhand dieser Schablonen wird für jedes Pixel ein Ähnlichkeitsmaß (normalisierte Summe der Quadratdifferenzen) mit den Schablonen festgelegt. Bereiche im Bild mit hoher Übereinstimmung mit den Schablonen werden als potentiell zugehörig zum Barcode gezählt und per Konturanalyse

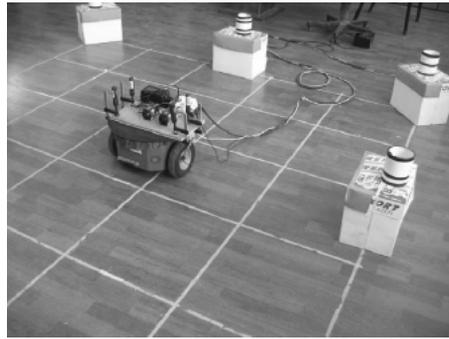


Abbildung 6.5. – Testparcours für die Zylindermarke aus [Arıcan, 2004]

zu einzelnen Balken des Barcodes zusammengeschlossen. Die einzelnen Balken werden dann über die Entfernung zueinander und ihre Größe auf die Möglichkeit ihrer Zugehörigkeit zum Code getestet. Wurde ein Code erfolgreich gefunden, wird für den oberen und unteren Begrenzungsbalken ein linker und rechter Endpunkt des Balken gesucht. Über die Mitte der beiden Begrenzungsbalken kann dann der Mittelpunkt des Zylinders berechnet werden, der als Messpunkt der Marke dient.

Ein anderes Beispiel für die Verwendung eines Zylinders in Fragen der Bestimmung der Position und Orientierung bei mobilen Fahrzeugen stellt ein Ansatz aus dem Pathfinder Mars-Programm der NASA dar (siehe [Volpe u. a., 1995]). Da für die Erkundung ein mobiler Roboter zum Einsatz kommen sollte, der aber trotzdem vom Landemodul aus lokalisiert werden muss, wurde die Posenbestimmung mit Hilfe eines auf dem Dach des Roboters befestigten Zylinder getestet (siehe Abb. 6.6). Ziel dieser Tests war es die Posenbestimmung über Radsensoren u.ä. jedenfalls teilweise zu ersetzen.

Die Oberfläche des Zylinders ist vertikal in gleichgroße Quadranten aufgeteilt worden, welche jeweils mit einer Farbe eingefärbt sind, die nicht in einem angrenzenden Quadranten genutzt wird. So kann durch die Bestimmung der vom Landemodul aus gesehenen Farbflächen und deren Größe die Orientierung und Position des Roboters über ein Stereo-Kamera-System bestimmt werden. Die Erkennung wird über die Klassifikation einzelner Pixel anhand ihrer Farben und deren Zusammenfassung zu größeren einfarbigen Bereichen durchgeführt. Wenn diese einfarbigen Bereiche genau nebeneinander liegen ist der Zylinder lokalisiert und die Größe der einzelnen (maximal 3) sichtbaren Farbquadranten kann bestimmt werden. Anhand der Größe der einzelnen Farbquadranten im Bild und dem bestimmten Schwerpunkt des Zylinders ist es nun möglich die Pose zu bestimmen. Bei diesem Zylinder wird der "Code" also nicht zur Unterscheidung von anderen Fahrzeugen o.ä. eingesetzt, sondern dient ausschließlich dazu die Orientierung und Position des Roboters in Relation zum unbeweglichen Landemodul zu bestimmen.

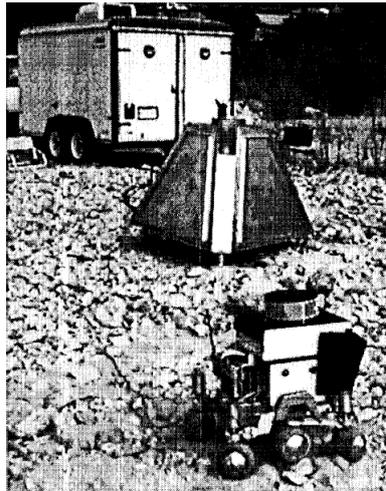


Abbildung 6.6. – Im Vordergrund: Rocky 3.2 Roboter mit Zylinder. Abbildung aus [Volpe u. a., 1995]

6.2.2. Kugel

Auch Kugeln werden aufgrund ihrer idealisierten Abbildung als Kreis auf der Bildebene der Kamera und ihrer (fast) vollen Sichtbarkeit aus allen Blickwinkeln häufig zu Signalisierungs- und Messzwecken eingesetzt. Auch hierzu werden nun zwei Arbeiten kurz vorgestellt.

Breite Verwendung finden mit Reflexfolie beklebte Kugeln bei Virtual Reality Systemen, wie sie z.B. zum Motion Capture (Bewegungserfassung) bei Computerspielen oder Filmen eingesetzt werden. Als ein Beispiel soll hier die Entwicklung eines Motion Capture basierten Spiels, für große Hallen dienen. Bei diesem Squidball genannten Spiel (siehe [Bregler u. a., 2005]) handelt es sich um ein in Echtzeit stattfindendes interaktives Videospiel für bis zu 4000 Spielern in großen Hallen, das mit Hilfe von Motion Capture Techniken funktioniert.

Bei diesem Spiel geht es darum, dass die Spieler mit Hilfe der Bewegung von zwölf großen Wetterballons die im Raum befindlichen virtuellen und über eine große Leinwand sichtbaren Ziele zerstören. Zur Orientierung dienen den Spielern die, über das kommerziell verfügbare Motion Capture System der Firma Vicon, ebenfalls auf der Leinwand abgebildeten aktuellen Positionen der Wetterballons (siehe Abb. 6.7). Diese mit Reflexfolie bezogenen Ballons dienen hier als physikalische Eingabe für das Spiel und ihre Position (der Mittelpunkt des Ballons) wird mit Hilfe der Ausgaben des Motion Capture System visualisiert. Diese System besteht hierbei aus mehreren Kameras, die mit Infrarotbeleuchtung ausgestattet sind und die Kugeln bzw. Ballons aus mehreren Richtungen aufnehmen, um daraus die jeweilige Raumposition zu errechnen.

In [Bradley, 2005] stellen Bradley und Roth ein Pendant zu den in Kapitel 3.2.2 erläuterten Marken für Augmented Reality Aufgaben vor, die aus allen Richtungen sichtbar sein soll, also



Abbildung 6.7. – Squidball in Aktion. Links ist im Hintergrund die Leinwand zu sehen. Abbildung aus [Bregler u. a., 2005]



Abbildung 6.8. – Kugelmarke mit roten und grünen Punkten aus [Bradley, 2005].

ein ähnliches dem in dieser Arbeit behandelten Problem. Wobei die Marke von Bradley und Groth keine eindeutige ID besitzt, um sie von anderen Marken unterscheiden zu können. Die Marke besteht aus einem blauen Ball, der mit 16 roten und 16 grünen Punkten beklebt ist (siehe Abb. 6.8). Diese Punkte sind zufällig über die Oberfläche verteilt, müssen aber immer einen Mindestabstand haben, um nicht ineinander überzugehen. Für die Erkennung sind die Positionen der einzelnen Punkte auf der Kugel in Polarkoordinaten ebenso bekannt, wie der Winkel, den die Punkte paarweise relativ zum Zentrum der Kugel aufspannen. Mit Hilfe dieser Informationen und der Messung der Entfernung des Kugelmittelpunkts von der Kamera, anhand des Durchmessers des abgebildeten Kreises, lässt sich die Pose des Balls im Raum berechnen.

Als besonderen Vorteil gegenüber quadratischen Marken geben Bradley und Roth neben der Eigenschaft, dass die Kugel aus allen Richtungen zu sehen ist an, dass die Kugelform außerdem die Erkennung auch bei Teilverdeckung ermöglicht.

6.3. Herstellung

Die Grundlage für den Zylinder (Beispiel in Abb. 6.1) der im Rahmen dieser Arbeit erstellten Prototypen bildet ein Pappzylinder, der aus einer Teppichrolle stammt. Es sind selbstverständlich auch Zylinder aus anderen Quellen und Materialien möglich. Die Größe des Zylinders wird

auf ca. 30 cm festgelegt, sodass mit der untergehängten Kugel eine Marke entsteht, die mit Aufhängung nicht mehr als 40 – 50 cm von der Decke hängt, um es in den 3 m hohen Räumen der HAW auch großen Menschen zu ermöglichen problemlos unter der Marke hindurchzugehen. Der verwendete Zylinder hat einen Durchmesser von 11 cm. Es macht durchaus Sinn hier auch einen größeren Durchmesser zu nutzen, da dies z.B. größere Drehwinkel der Marke im Bild ermöglicht. Auch die Höhe des Zylinders und der Durchmesser der Kugel sind quasi beliebig veränderbar, da der Erkennungsalgorithmus ausschließlich auf Verhältnissen arbeitet, die an das jeweilige Anwendungsszenario angepasst werden.

Der gesamte Zylinder wird mit schwarzer Pappe bezogen oder alternativ mit schwarzer Farbe bestrichen, wobei diese nicht-reflektierend sein sollte. Die schwarze Farbe soll einen möglichst großen Kontrast zu den aus Reflexfolie bestehenden und horizontal aufgetragenen Balken des Barcodes bilden. Als Reflexfolie wird hier die in Kapitel 2.2.5 vorgestellte Series 3210 von 3M genutzt. Die genaue Nutzung der Zylinderfläche wird durch die Wahl des Barcodes festgelegt und ist somit variabel. Der offene Boden wird ebenfalls mit einem dem Durchmesser des Zylinders entsprechenden Kreis verdeckt, wobei dieser in der Mitte ein Loch haben muss, um einen Befestigungsstab für die Kugel aufzunehmen. Der Abstand zwischen Kugel und Zylinderboden entspricht in der Beispielkonfiguration ca. 1 cm. Die Befestigung über einen Stab, im Gegensatz zum direkten Kleben der Kugel an den Zylinderboden, hat den Sinn, dass hiermit die durch die Befestigung verdeckte Fläche der Kugel minimiert wird, wobei diese Verdeckung in der Praxis nur in wenigen Situationen Auswirkungen auf die Erkennung hat. Der größere Abstand zwischen Zylinder und Kugel hat sich aber auch in der Erkennung des Barcodes von Vorteil erwiesen, da sich so Kugel und Streifen besser voneinander abgrenzen lassen. Generell deutet diese Befestigung unterhalb des Zylinders aber an, dass die Marke nur von unterhalb oder ungefähr auf gleicher Höhe zur Markenfläche angebrachten Kameras identifiziert werden kann, da sonst die Kugel nicht vollständig sichtbar ist. Dies ist beabsichtigt und entspricht den Anforderungen an diese Marke zur Navigation/Posenbestimmung von bodennahen Fahrzeugen mit Kamera.

Die obere Öffnung des Zylinders wird zur Befestigung genutzt, entweder durch das Ankleben von Winkeln oder eine Aufhängung aus Fäden. Die letztere Möglichkeit birgt aber die Gefahr, dass sich die Marke u.U. schon bei leichten Luftzügen, welche auch in Innenräumen auftreten können, bewegt und ihre Position somit nicht mehr der ausgemessenen entspricht.

Als Kugel wird in dieser Arbeit eine Styroporkugel mit einem Durchmesser von 8 cm aus dem Bastelbedarf genutzt. Da die Größe der Kugel Einfluss auf die maximale Erkennungsentfernung hat, ist auch eine größere Kugel möglich. Wie schon ausgeführt ist die Kugel zur Unterscheidung vom Hintergrund im Bild mit Reflexfolie beklebt. Dass es sehr problematisch ist diese Folie glatt auf eine Kugel aufzubringen ist in [Bregler u. a., 2005] beschrieben. Dort wurde u.a. auch mit ebenfalls von 3M angebotenen Lack (Sprühfarbe), welcher die zur Retroreflexion benötigten verspiegelten Kugeln enthält, experimentiert. Dieser Lack hat aber deutlich geringere Reflexionseigenschaften als die verfügbaren selbstklebenden Folien und stellt deshalb keine wirkliche Alternative dar, obwohl er wesentlich einfacher aufzubringen ist. In dieser Arbeit wird, ähnlich wie bei einem Strandball und in [Bregler u. a., 2005], die Kugel aus

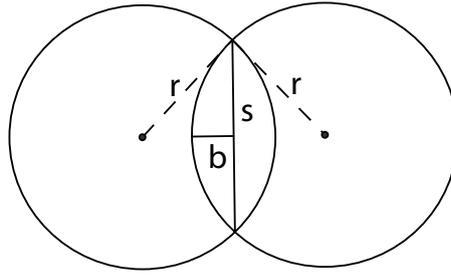


Abbildung 6.9. – Zwei zusammengesetzte Kreisabschnitte zum Bekleben der Kugel.

zwei an ihrer Längsseite zusammengesetzten 0,5 cm breiten Kreisabschnitten beklebt (siehe Abb. 6.9). Bei einem Radius von 4 cm werden hierfür 26 dieser Streifen benötigt. Breitere Streifen sind für diesen Kugeldurchmesser nicht zu empfehlen, da sie das Aufkleben stark erschweren und zu einer deutlich höheren Blasenbildung führen. Kleinere Ungenauigkeiten dagegen sind durch die Bestrahlung ohnehin (kaum) im Bild nicht zu sehen. Der Radius r_{kreis} für die benötigten Kreise zur Herstellung der Streifen lässt sich z.B. über die Formel für Kreissegmente berechnen (siehe z.B. [Stöcker, 1995]):

$$r_{kreis} = \frac{\frac{s^2}{2} + b^2}{2b} \quad (6.1)$$

$$s = r_{kugel} \pi \quad (6.2)$$

b : Halbe Breite des Streifens

s : Sehnenlänge

6.4. Codierung

Um die einzelnen Exemplare dieses Markentyps voneinander unterscheiden zu können, wird zur Codierung der eindeutigen IDs ein Barcode eingesetzt. Ein Barcode besteht üblicherweise aus alternierenden parallelen schwarzen Balken und weißen Zwischenräumen verschiedener Breite. In dieser Arbeit wird dieses System dahingehend verändert, dass die Balken durch die weiße Reflexfolie und die Zwischenräume durch die schwarze Zylinderoberfläche repräsentiert werden (siehe Abb. 6.1). Es hat sich bei Tests herausgestellt, dass um eine Erkennung mit der genutzten Kamera (siehe Tab. 2.1) auch aus einer Entfernung von maximal 7 – 10,50 m jedenfalls teilweise zu gewährleisten, die Balken nicht schmaler als 1 cm werden dürfen, was bei einer Höhe von ca. 30 cm zu deutlichen Einschränkungen in der Codeauswahl führt. Die schwarzen Zwischenräume werden aus Gründen der Einfachheit und dem u.a. noch nicht getesteten Einfluss einer möglichen Überstrahlung der beiden unterschiedlichen Balkenbreiten aus Reflexfolie auf die Größe der Zwischenräume nicht zur Codierung genutzt. Diese Interleaved genannte Form von Barcodes ist aber auf jeden Fall eine Option, um noch mehr Informationen pro Fläche codieren zu können. Eine weitere Option bietet der von Arican in

[Arıcan, 2004] genutzten Code (siehe 6.2.1), welcher hier aber u.a. aufgrund der zu erwartenden Probleme bei der eindeutigen Erkennung, abseits der dort dargestellten optimalen Bedingungen, nicht genutzt wird. Als gutes Maß für diese Zwischenräume konstanter Größe haben sich 1,5 cm erwiesen.

Aufgrund der genannten Einschränkungen ist es nicht möglich einfach einen in der Praxis häufig verwendeten Barcode, wie z.B. UPC (Universal Product Code siehe [Pavlidis u. a., 1990]), zu nutzen. UPC bietet zwar die Codierung einer zwölfstelligen Dezimalziffernkette an, benötigt aber für jede Ziffer 7Bit. Jedes Bit ist repräsentiert durch die minimale Breite eines schwarzen Balkens oder weißen Zwischenraums. Es sind Balken/Zwischenräume bis zur vierfachen Breite des minimalen Balkens/Zwischenraums möglich. Die binäre Ziffer 0 wird durch den Zwischenraum einfacher Breite dargestellt, dementsprechend entspricht der Balken einfacher Breite der binären Ziffer 1. Zusätzlich zu den reinen Ziffern beinhaltet der Code noch Start-, Stop- und Mitteltrennzeichen sowie Prüfsummen u.ä.. Es ist einzusehen, dass Codes dieser Größe bei der angestrebten Zylindergröße und Erkennungsentfernung nicht verwendet werden können. Stattdessen werden hier nun drei speziell für die Anforderungen des genutzten Zylinders (teilweise) selbstentwickelte Codes vorgestellt, die sich in ihrer Größe, der Anzahl der möglichen codierbaren IDs und der maximalen Erkennungsentfernung unterscheiden. Als Vorteil kommt bei der geplanten Nutzung auf dem Zylinder aber zum tragen, dass der Barcode durch die geplante Anbringung, von der Decke hängend nicht, rotationsinvariant sein muss. Die Richtung aus der der Code ausgelesen wird ist immer die gleiche.

6.4.1. 1. Variante (2-aus-8)

Für diese Variante wird die Breite eines schmalen Balkens auf 1,5 cm festgelegt, welche auch der Breite der schwarzen Zwischenräume entspricht. Der Code besteht aus 8 Balken, wobei 2 dieser Balken doppelt so breit sind (3 cm, Symbol W), wie die schmalen Balken (Symbol N). Wenn am Anfang und am Ende des Barcodes noch jeweils ein Zwischenraum zum Zylinderrand hin eingeplant wird, ergibt dies eine Gesamtlänge von 28,5 cm, womit der Zylinder fast perfekt ausgenutzt wird, wie in Abbildung 6.1 zu sehen ist. Die einzelnen IDs der Marken werden direkt aus den Balken gebildet, indem jeder Balken 1 Bit einer Binärzahl repräsentiert. Daraus ergibt sich von oben (niederwertigstes Bit) nach unten gelesen eine 8 Bit Binärzahl, wobei der schmale Balken der binären Ziffer 0 und der breite Balken der binären Ziffer 1 entspricht. Es lassen sich mit diesem Code insgesamt 28 IDs codieren (niederwertigstes Bit rechts): $WNNNNNNW \rightarrow 129$, $NWNNNNWN \rightarrow 66$, $NNNWNWNN \rightarrow 20$, ...

6.4.2. 2. Variante (2-aus-10)

Diese Variante entspricht bis auf die folgenden Änderungen dem Code aus Kapitel 6.4.1:

- Erweiterung um 2 schmale Balken.
- Schmale Balken haben eine Größe von 1 cm.

- Breite Balken haben die dreifache Größe eines schmalen Balkens (3 cm).

Dieser Code nutzt die ca. 30cm Höhe des Zylinders optimal aus und es lassen sich mit ihm 45 IDs codieren (niederwertigstes Bit rechts): $WNNNNNNNNW \rightarrow 513$, $NWNNNNWNNN \rightarrow 264$, $NNNWNNWNN \rightarrow 68$, ...

6.4.3. 3. Variante (2-aus-5)

Diese Variante nutzt, anders als die Vorherigen, den auch industriell eingesetzten Barcode 2-aus-5, indem eine zweistellige Dezimalzahl mit diesem Code codiert wird, wodurch insgesamt 100 eindeutige IDs möglich sind (00-99). Allerdings muss der Zylinder dafür um 4,5cm auf 34,5cm erhöht werden, da sonst der Platz nicht ausreicht. Jede Dezimalziffer wird bei diesem Code durch 5 Bits codiert (siehe [U.S. National Communications System Technology & Standards Division, 1996]). Die auf dem Zylinder zu oberst angebrachten 5 Balken bilden in dieser Arbeit von oben nach unten gelesen die niederwertige Dezimalstelle ab, wobei der oberste Balken jeder Dezimalstelle das niederwertigste Bit darstellt. Die Gewichtung der Balkenpositionen entspricht dabei 0-1-2-3-6. Auf ein eindeutiges Start- und Endesymbol sowie die Prüfziffer wird hier aus Platzgründen bewusst verzichtet. Wie der Name schon schließen lässt, bestehen zwei dieser 5 Bits aus der binären Ziffer 1 während die anderen 0 sind. Die binäre Ziffer 1 wird wie auch schon bei 6.4.1 und 6.4.2 durch einen breiten Balken (Symbol W) und die Ziffer 0 durch einen schmalen Balken (Symbol N) repräsentiert. In dieser Arbeit wird die übliche Relation der schmalen und breiten Balken genutzt (breiter Balken dreimal größer als schmaler Balken). Die Abmessungen der Balken entsprechen dabei denen aus Kapitel 6.4.2. Die genaue Abbildung der Dezimalziffern auf die Balken ist in der Tabelle 6.1 zu finden und eine Beispielcodierung ist in Abbildung 6.10 zu sehen.

Ziffer	Balken(0-1-2-3-6) (W=breit,N=schmal)
0	NWWNN
1	WWNNN
2	WNWNN
3	WNNWN
4	NWNWN
5	NNWWN
6	WNNNW
7	NWNNW
8	NNWNW
9	NNNWW

Tabelle 6.1. – Codierung von dezimalen Ziffern in 2-aus-5 Code.

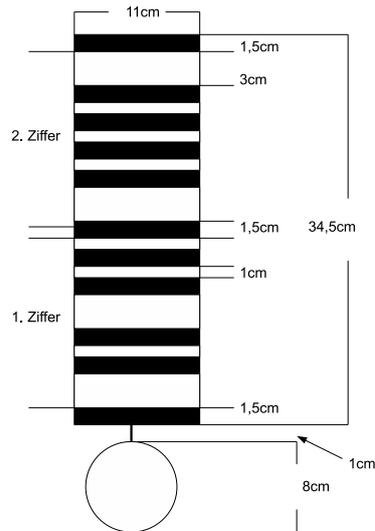


Abbildung 6.10. – 2D-Ansicht der Zylindermarke mit Beispielmaßen und der ID 86 in 2-aus-5-Code.

Die Tatsache, dass auf ein Start-, Mittel- und Stoppsymbol sowie auf eine Prüfziffer bei allen 3 Varianten verzichtet wird, ist vor dem Hintergrund, dass der Barcode neben der Kugel das einzige Erkennungsmerkmal der Marke ist unglücklich. Aufgrund von Platzproblemen lässt sich dies aber nicht ohne eine u.U. deutliche Vergrößerung der Zylinderhöhe oder einer Verringerung der maximalen Erkennungsentfernung erreichen, da z.B. der 2-aus-5-Code für Start- und Stoppsymbol jeweils zwei Breiten und einen schmalen Balken benötigt. Obwohl sie keine Prüfsymbole u.ä. enthalten besitzen die 3 vorgestellten Codes aber alle die Eigenschaft der Fehlererkennung, die zumindest soweit geht, dass alle Muster aussortiert werden können die nicht die benötigte Gesamtanzahl an Balken bieten oder keine 2 breiten Balken besitzen. Weiterhin sichern die drei Varianten durch ihre konstante Anzahl von breiten und schmalen Balken die für die spätere Erkennung wichtige Eigenschaft der konstanten Größe. Probleme können in der Praxis dort auftreten, wo dem Barcode ähnliche Muster vorhanden sind, in diesem Fall dient nur noch die Kugel als Unterscheidungsmerkmal. Eine weitere potentielle Gefahr stellen zwei im Bild übereinander abgebildete Marken mit Barcode dar, weil es auch hier potentiell zu Verwechslungen kommen kann. Dies sollte aber eigentlich dadurch ausgeschlossen sein, dass die Marken relativ weit voneinander entfernt angebracht werden und es somit z.B. zu erheblichen Größenunterschieden in der Balkenbreite kommt und auch die Fehlererkennung des Barcodes viele Fälle verhindert. Weiterhin sind auch Fehlidentifikationen durch fehlerhafte Messungen der Balkenbreiten z.B. ausgelöst durch Störungen oder Probleme durch die perspektivische Verzerrung des Barcodes nicht ausgeschlossen. Diese ließen sich, sollte dies zu Problemen führen und auch durch die Fehlererkennung der Barcodes nicht erkannt werden, durch den Einsatz eines Prüfsymbols verringern.

6.5. Geometrische Eigenschaften

Dieser Abschnitt zeigt einige wichtige geometrischen Eigenschaften des Zylinders und der Kugel auf, um z.B. entscheiden zu können, ob sie bei der Erkennung der Marke wichtig sind oder vernachlässigt werden können.

6.5.1. Gedrehte Abbildung des Zylinders

Obwohl der Zylinder senkrecht von der Decke hängt und die Kamera entsprechend so ausgerichtet sein sollte, dass mit spaltenweisen Scanlinien über das Bild der gesamte Barcode getroffen wird, ist diese ideale Situation in der Realität nicht immer herzustellen. Eine gedrehte Abbildung der Marke bzw. des Barcodes im Bild kann z.B. bei einer nicht perfekten Ausrichtung des Rollwinkels der Kamera (kippen siehe Abb. 6.3) auftreten. Eine andere Situation tritt durch das Neigen (Veränderung des Nickwinkels) der Kamera, um die unter der Decke hängenden Marken besser erfassen zu können, auf. Hierbei kommt es zu einer perspektivischen Verzerrung (Bildebene und Zylinderlängsschnittfläche sind nicht parallel - siehe auch Kapitelneigung) und die Marke erscheint, je weiter außen im Blickfeld sie sich befindet, u.U. stark gedreht im Bild.

Wenn man den Zylinder als ein in die Bildebene projiziertes Rechteck ansieht, entspricht der maximale Drehwinkel unter dem der komplette Barcode durch noch mindestens eine Scanlinie getroffen wird, dem in Abbildung 6.11 durch α beschriebenen Drehwinkel. Die Berechnung von α ist denkbar einfach:

$$\tan \alpha = \frac{b}{h} \quad (6.3)$$

Aus Gleichung 6.3 folgt, dass wenn ein größerer Drehwinkel benötigt wird ein breiterer Zylinder genutzt werden muss. Für den bei den Tests zu dieser Arbeit hauptsächlich genutzten Zylinder mit 2-aus-8 Code und $b = 11 \text{ cm}$; $h \approx 25,5 \text{ cm}$ folgt ein maximaler Drehwinkel von $\alpha = 23,33^\circ$. Zu große Drehwinkel ($\gg 45^\circ$) sind aber aufgrund des noch vorzustellenden Erkennungsalgorithmus nur eingeschränkt möglich. Es ist zu beachten, dass sich der Winkel α bei perspektivischer Verkürzung u.U. vergrößert.

6.5.2. Einfluss perspektivischer Verzerrung auf den Barcode

Wie schon in Kapitel 6.5.1 erwähnt, führt die Neigung der Kamera (Rotation um die Querachse) in Richtung der von der Decke hängenden Marke dazu, dass es zu einer Verkippung²³ der Bildebene gegenüber der ehemals parallelen Zylinderlängsschnittfläche²⁴ kommt. Hierdurch entsteht u.a. eine perspektivische Verzerrung des Barcodes (siehe Abb. 6.12). Dies hat zur

²³Nicht zu verwechseln mit dem kippen der Kamera aus Abbildung 6.3.

²⁴Wenn im weiteren Verlauf dieser Arbeit vom Längsschnitt des Zylinders die Rede ist, ist damit die Längsschnittfläche gemeint, die ohne Neigung der Kamera parallel mit der Bildebene ist.

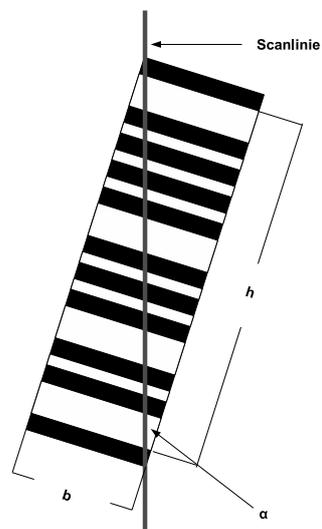


Abbildung 6.11. – Barcode einer Zylindermarke mit maximalem Drehwinkel α und Scanlinie.

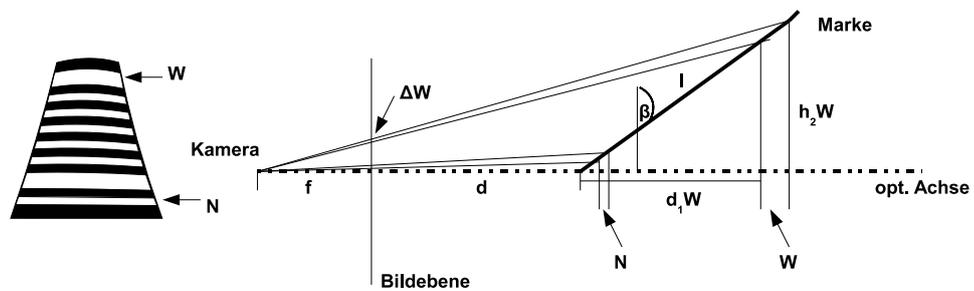


Abbildung 6.12. – Neigung der Kamera wird hier durch die Verkippung der auf der optischen Achse stehenden Zylindermarke in Bezug zur Bildebene dargestellt.

Folge, dass sich der untere Teil des Barcodes näher an der Bildebene befindet als der obere Teil, wodurch sich die oberen Balken im Bezug zu den unteren Balken in der Größe verändern, was bei einer großen Entfernung (bezogen auf die Markengröße) zur Marke aber kein Problem darstellt. Im schlimmsten Fall führt dies im Nahbereich aber dazu, dass ein schmaler unterer Balken (Symbol N) größer oder genauso groß ist, wie ein breiter Balken (Symbol W) im oberen Teil des Codes. Dadurch sollte es aufgrund der gewählten Barcodes normalerweise nicht zu einer Fehlidentifikation der Marke kommen, dennoch sollte diese Problematik zumindest einmal theoretisch betrachtet werden.

In einer Beispielrechnung soll nun die Relevanz dieses Problem für die zu erwartenden Praxisanforderungen der Marke untersucht werden. Die Bezeichnungen beziehen sich auf die in Abbildung 6.12 verwendete Beschriftung.

Die minimale Entfernung zur Marke wird auf $d = 2000$ mm festgelegt, da dieser Wert durch die Deckenmontage wohl nicht unterschritten wird. Als Neigung wird von $\beta = 50^\circ$ ausgegangen, da dies ungefähr dem experimentell festgestelltem Wert für den Einfallswinkel der Reflexfolie, ab dem die Reflexion mit den genutzten Scheinwerfern und Beobachtungswinkeln merklich nachlässt, entspricht. Dieser Wert ist allerdings u.a. abhängig von der Stärke der Beleuchtung und letztendlich auch von den Einstellungen der nachfolgend noch erläuterten Kantenfilterung. Er ist generell als ziemlich willkürlich gewählt anzusehen und kann bei entsprechender Beleuchtung auch verändert werden, was aber u.U. Nachteile bei der Erkennungsentfernung u.ä. nach sich zieht. Die für die Rechnung genutzte Marke entspricht den Maßen nach der aus Abbildung 6.1, allerdings sind die untersten beiden Balken vertauscht. Für die Brennweite wird die Kamera aus Tabelle 2.1 genutzt, $f = 5$ mm.

Die Rechnungen basieren auf dem Strahlensatz wodurch die Ergebnisse als idealisiert anzusehen sind, da z.B. keine Einflüsse, die durch die Abbildung mit einem optischen System entstehen berücksichtigt werden. Dennoch eignen sich die Werte sehr gut als Anhaltspunkte für die in der Praxis zu erwartende Situation.

Wie in Abbildung 6.1 zu sehen ist, befindet sich der Drehpunkt im Berührungspunkt des Fußes des Zylinders und der optischen Achse.

$$h_x X = \sin(90^\circ - \beta) l_x X \quad (6.4)$$

$$d_x X = \cos(90^\circ - \beta) l_x X \quad (6.5)$$

$$\Delta W = \left(\frac{f}{(d + d_2 W)} h_2 W \right) - \left(\frac{f}{(d + d_1 W)} h_1 W \right) \quad (6.6)$$

$$\Delta N = \left(\frac{f}{(d + d_2 N)} h_2 N \right) - \left(\frac{f}{(d + d_1 N)} h_1 N \right) \quad (6.7)$$

- | | |
|--|--|
| $h_x X$: Entfernung der Kante x des Streifens X von der opt. Achse. | $d_x X$: Entfernung der Kante x des Streifens X vom Fußpunkt des Zylinders. |
| d : Entfernung des Fußpunktes des Zylinders von der Kamera. | ΔW : Größe des breiten Streifens W in der Bildebene. |
| ΔN : Größe des schmalen Streifens N in der Bildebene. | |

Aus den Gleichungen 6.4 bis 6.7 folgt mit den o.g. Werten für den breiten Balken $\Delta W \approx 39,7 \mu\text{m}$ und für den schmalen Balken $\Delta N \approx 23,5 \mu\text{m}$, was ein Verhältnis von $\frac{\Delta W}{\Delta N} \approx 1,68$ ergibt. Die errechneten Werte lassen darauf schließen, dass es hier bei extrem großen Winkeln und extrem kurzen Entfernungen zu Problemen kommen könnte, dies in der Praxis allerdings nicht zu erwarten ist. Zumal es mit den anderen zur Verfügung stehenden Code-Varianten zu einem Verhältnis von 1:3 zwischen schmalen und breiten Balken kommt, anstatt 1:2 in diesem Beispiel, womit die Situation entschärft werden kann. Es muss angemerkt werden, dass die im Erkennungsalgorithmus zur Klassifikation genutzten Intervalle die Problematik theoretisch nochmals verschärfen, was in der Praxis aber normalerweise zu keiner Fehlidentifikation führt. Dies ist, wie schon erwähnt, darauf zurückzuführen, dass in diesen Fällen die Fehlererkennung des Barcodes greift, weil durch die in der Praxis genutzten Intervalle zwar z.B. schmale Balken am Fuß des Zylinders schneller als breite Balken erkannt werden, im Gegenzug werden aber auch breite Balken am Kopf des Zylinders länger als solche klassifiziert. Dies führt normalerweise dazu, dass es zu viele breite Balken gibt und somit die Fehlererkennung des Barcodes den gefundenen Code als nicht gültig ansieht. Generell ist die Fehlidentifikation aufgrund perspektivischer Verzerrung nicht ausgeschlossen, aber sehr unwahrscheinlich und die Wahrscheinlichkeit ließe sich, wie schon erwähnt, durch ein Prüfsymbol weiter verringern.

Zusammenfassend kann festgestellt werden, dass die Neigung der Kamera/Marke in der Praxis keine wirkliche Relevanz für die Unterscheidung der breiten und schmalen Balken hat. Die Neigung der Kamera besitzt aber dahingehend Relevanz, dass sie dafür verantwortlich ist, dass sich das Verhältnis aus der abgebildeten Markenhöhe und Markenbreite verändert, was später noch zu beachten sein wird.

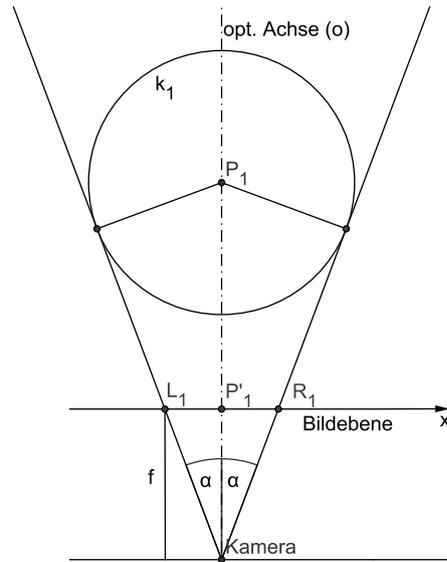


Abbildung 6.13. – Abbildung der Kugel als Kreis.

6.5.3. Abbildung der Kugel

Wie schon erwähnt, wird die Kugel in der Bildebene immer als Kreis abgebildet, dies ist aber nur eine idealisierte Annahme, da es zwischen zwei Fällen zu unterscheiden gilt:

- 1. Optische Achse geht durch den Kugelmittelpunkt:** Dieser Fall ist in Abbildung 6.13 illustriert und es ist zu sehen, dass in diesem Fall die Kugel als Kreis abgebildet wird. Die beiden tangentialen Abbildungsstrahlen berühren die Kugel in der gleichen Entfernung von der Bildebene (in Bezug auf d. opt. Achse) und sind ebenfalls um den gleichen Winkel α von der optischen Achse gedreht. In diesem Fall gelten die folgenden Gleichungen:

$$\overline{L_1 P'_1} = \overline{P'_1 R_1} \quad (6.8)$$

$$x_{P'_1} = \frac{x_{L_1} + x_{R_1}}{2} \quad (6.9)$$

x_{L_1} : x-Koordinate des Punktes L_1 x_{R_1} : siehe x_{L_1}
 $x_{P'_1}$: siehe x_{L_1}

P'_1 wird genau in der Mitte der Strecke $\overline{L_1 R_1}$ abgebildet.

- 2. Optische Achse geht nicht durch den Kugelmittelpunkt:** In diesem Fall wird die Kugel als Ellipse in der Bildebene der Kamera abgebildet, da die beiden tangentialen Abbildungsstrahlen durch die runde Form der Kugel diese nicht in der gleichen Entfernung zur

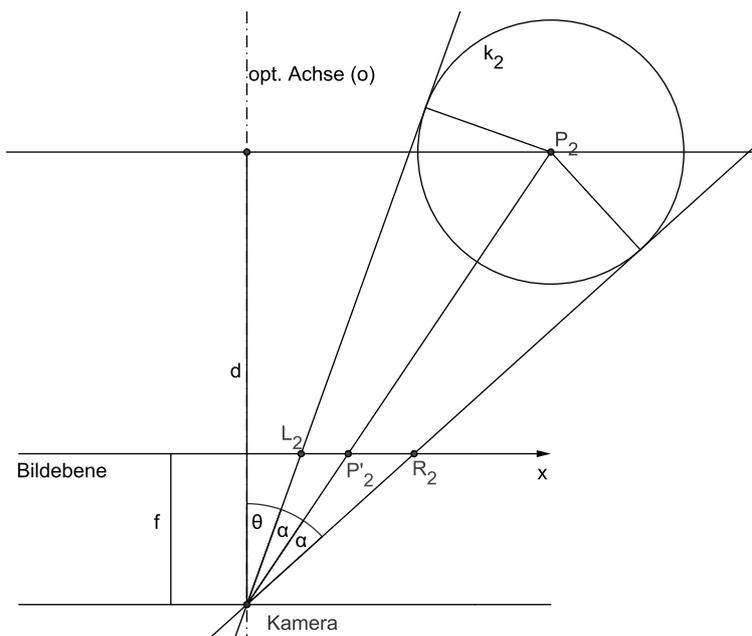


Abbildung 6.14. – Abbildung der Kugel als Ellipse.

Bildebene (in Bezug auf d. opt. Achse) berühren (siehe Abb. 6.14 und [Lin und Chang, 2005]). Dies führt, genauso wie bei der Abbildung eines Kreises auf eine Ellipse bei einer nicht senkrecht ausgerichteten Kamera auch, zu einer Abbildungsexzentrizität e (siehe Kapitel 3.2.1), da die automatische Messung in der Mitte der Strecke $\overline{L_2R_2}$ stattfindet. Ausgelöst wird dies wie schon angedeutet durch den gleich Winkel α zwischen den Tangenten und dem Abbildungstrahl des Mittelpunktes und dem durch die Drehung entstandenen Winkel θ . Dies führt dazu, dass bei einem Abstand d zwischen Kamera und der Ebene der Kugelmitte folgendes gilt:

$$\overline{L_2P'_2} - \overline{P'_2R_2} = \begin{cases} 0 & d \rightarrow \infty \\ \neq 0 & \text{otherwise} \end{cases} \quad (6.10)$$

$$x_{P'_2} = \begin{cases} \frac{x_{L_2} + x_{R_2}}{2} & d \rightarrow \infty \\ > \frac{x_{L_2} + x_{R_2}}{2} & \overline{L_2P'_2} > \overline{P'_2R_2} \wedge d < \infty \\ < \frac{x_{L_2} + x_{R_2}}{2} & \overline{L_2P'_2} < \overline{P'_2R_2} \wedge d < \infty \end{cases} \quad (6.11)$$

In der Praxis spielen weder die leichte Ellipsenbildung noch die Abbildungsexzentrizität eine Rolle. Die Ellipsenbildung ist bei den bisherigen Tests nicht sichtbar gewesen, bzw. konnte nicht von durch andere Umstände ausgelöste Verformungen unterschieden werden. Die Abbildungsexzentrizität spielt, wie schon in Kapitel 3.2.1 ausgeführt, ab einem Abstand von einigen Kreis- bzw. Kugeldurchmessern keine Rolle mehr, was durch eine Beispielrechnung verdeutlicht werden soll:

Bei einer Entfernung von 2000mm und einem Kugeldurchmesser von 80mm ergibt

$\alpha = 0,95^\circ$. $\theta = 29,7^\circ$ stellt sicher, dass der komplette horizontale Bildwinkel der verwendeten Kamera ausgenutzt wird. Es lassen sich nun folgende einfache Gleichungen aufstellen:

$$x_{L_2} = \tan \theta f \quad (6.12)$$

$$x_{R_2} = \tan(\theta + 2\alpha)f \quad (6.13)$$

$$x_{P'_2} = \tan(\theta + \alpha)f \quad (6.14)$$

$$e = x_{P'_2} - \left(x_{L_2} + \frac{x_{L_2} - x_{R_2}}{2}\right) \quad (6.15)$$

Aus einer Brennweite von $f = 5 \text{ mm}$ ergeben sich folgende Werte für den o.g. Fall:

$$x_{L_2} = 2,740 \text{ mm}$$

$$x_{R_2} = 2,959 \text{ mm}$$

$$x'_{P_2} = 2,848 \text{ mm}$$

$$e = -1,5 \mu\text{m}$$

Der Wert von $e = -1,5 \mu\text{m}$ entspricht weniger als $\frac{1}{3}$ eines Pixels der Kamera und wenn man den Wert wieder auf die Kugelebene zurückrechnet ergibt sich ein Fehler von ca. $0,7 \text{ mm}$, bei dem in der Praxis entschieden werden muss, ob er im Toleranzbereich der Messgenauigkeit liegt. Diese Rechnung ist aufgrund der extremen Vereinfachung der Abbildung durch ein reales optischen Systems ungenau, sie veranschaulicht aber trotzdem gut die zu erwartenden Werte.

6.5.4. Sichtbarer Bereich der Kugel

Bei Ansicht von Abbildung 6.15 ist zu erahnen, dass aufgrund der durch die gekrümmte Oberfläche der Kugel ausgelösten perspektivischen Verkürzung, diese nicht immer mit ihrem kompletten Durchmesser im Bild abgebildet wird. Dies wird klar, wenn man beachtet, dass der Einfallswinkel δ des Abbildungsstrahls t , der die Oberfläche der Kugel berührt, nicht größer als 90° werden kann. Im Fall von $\delta = 90^\circ$ bezeichnet man t als tangentialen Abbildungsstrahl. Die Kugel wird mit ihrem kompletten Durchmesser nur bei einer unendlichen Entfernung der Kugelmittle zum Projektionszentrum abgebildet. Aber wie bei den anderen, durch Perspektive ausgelösten Eigenschaften, schon beschriebenen spielt auch diese nur bei einer starken Perspektive überhaupt eine Rolle. In dieser Arbeit ist es eigentlich schon deshalb zu vernachlässigen, da keine Entfernungswerte der Kugel von der Kamera benötigt werden.

Was bei der hier genutzten Kugel aber etwas erschwerend hinzukommt, ist die Verringerung des Einfallswinkel δ auf den maximalen Einfallswinkel, aus dem die verwendete Reflexfolie noch genügend Licht zurückwirft. Dies führt dazu, dass die Kugel immer kleiner abgebildet wird und dadurch weiter entfernt erscheint, als sie es eigentlich ist. Wenn man also idealisiert davon ausgeht, dass der Kamerastandpunkt dem der Lichtquelle entspricht kann z.B. über die

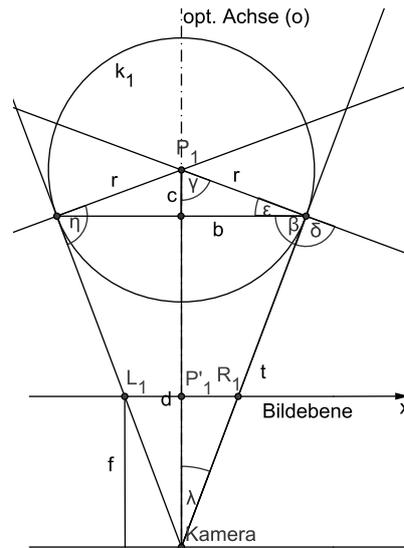


Abbildung 6.15. – Illustration des sichtbaren Bereichs der Kugel.

folgenden Formeln bspw. für einen Winkel $\delta = 50^\circ$ und einer Entfernung $d = 2000\text{ mm}$ der sichtbare Bereich der Kugel ($r = 40\text{ mm}$) $2b$ berechnet werden.

$$\eta = 180^\circ - \delta \quad (6.16)$$

$$\sin \lambda = \frac{\sin \eta r}{d} \quad (6.17)$$

$$\gamma = 180^\circ - \eta - \lambda \quad (6.18)$$

$$b = r \sin \gamma \quad (6.19)$$

Bei den genannten Beispielwerten ergeben sich folgende Ergebnisse:

$$\eta = 130^\circ$$

$$\lambda = 0,878^\circ$$

$$\gamma = 49,121^\circ$$

$$b = 30,244\text{ mm}$$

Es ist in diesem Beispiel nur eine Kugel mit einem Radius von ca. $30,244\text{ mm}$ anstatt von 40 mm , die sie real besitzt, zu sehen. Wird der Einfallswinkel aber auf $\delta = 90^\circ$ gesetzt ergibt sich für b ein Wert von $39,99\text{ mm}$, was belegt, dass diese Verkürzung an sich in der Praxis kaum eine Rolle spielt. Die Benutzung von Reflexfolie hingegen verringert die maximale Erkennungsentfernung der Kugel etwas, was aber im Endeffekt u.a. auch stark von der Kombination aus genutzter Folie und Beleuchtung abhängig ist.

Das hier beschriebene gilt natürlich ebenfalls für den Zylinder mit seiner gekrümmten Oberfläche.

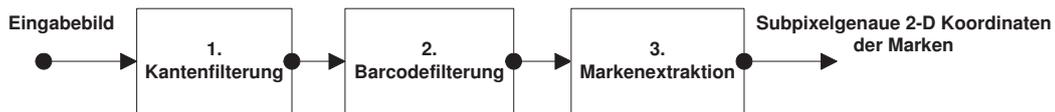


Abbildung 6.16. – Die grundlegenden Schritte zur Erkennung der Zylindermarke. Die Schritte 1 und 2 werden einmal für das gesamte Bild ausgeführt, bevor Schritt 3 für jede Marke einzeln erfolgt.

6.6. Erkennung der Marken

Dieser Abschnitt behandelt ausführlich die einzelnen Schritte, die zur Erkennung der Marken in einem Bild notwendig sind. Die Organisation der Unterabschnitte orientiert sich dabei an den in Abbildung 6.16 aufgeführten grundlegenden Schritten.

Der Erkennungsalgorithmus ist größtenteils so entworfen worden, dass eine zeitlich sehr schnelle Erkennung aller im Bild befindlichen Marken möglich ist. Hierzu wurden einige sonst in der Bildverarbeitung u.U. übliche Verfahren weggelassen bzw. nur vereinfacht angewendet. Nähere Informationen dazu sind immer an der entsprechenden Stelle zu finden.

Die softwaretechnische Umsetzung des Erkennungsalgorithmus steht auch bei diesem Markentyp als Impresario-Macro zur Verfügung. Die Parameter des Macros werden im Laufe dieses Abschnitts ebenfalls erklärt werden und sind zum Nachschlagen auch in Tabelle 6.2 zu finden. In den einzelnen Abschnitten zu den Schritten werden keine Empfehlungen für Werte der verwendeten Parameter gegeben, da diese mit Erklärungen besser in einem eigenen Abschnitt 6.7.2 aufgehoben sind.

6.6.1. Grundlagen

Dieser Unterabschnitt vermittelt die für die Erkennung der Marken notwendigen Grundlagen.

6.6.1.1. Sobel-Operator

Der Sobel-Operator dient der Kantenfilterung und nutzt hierzu die diskrete erste Ableitung von Grauwerten, wobei diese senkrecht zur Kante geglättet wird. Mittels je einer 3×3 Faltungsmaske für die x- und y-Richtung wird für einen Bildpunkt der sog. Gradient G gebildet. Der Gradient ist ein Vektor, dessen Betrag der Stärke des Helligkeitssprungs in Richtung des steilsten Helligkeitsanstiegs entspricht. Die Richtung des Gradienten steht dabei senkrecht zum Verlauf der Kantenlinie. Aus den für jeden Punkt berechneten Gradienten ergibt sich ein Gradientenbild. Orte des größten Helligkeitssprungs (Kantenpunkte) in einer Richtung des Eingabebildes (E) werden durch lokale Maxima des Betrags der Gradienten in der selben Richtung repräsentiert.

$$\begin{aligned}
 G_x &= \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * E \\
 G_y &= \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * E \\
 G &= \sqrt{G_x^2 + G_y^2} \\
 \Theta &= \arctan\left(\frac{G_y}{G_x}\right)
 \end{aligned}$$

6.6.1.2. Subpixelgenaue Kantenbestimmung

Mit dem in Kapitel 6.6.1.1 vorgestellten Sobel-Operator lassen sich über lokale Maxima des Gradientenbildes Kantenpositionen bestimmen. Ein Nachteil hierbei ist, dass diese Kantenpositionen nur pixelgenau sind. Wie z.B. in Kapitel 5 gesehen, macht es bei einigen Anwendungen aber Sinn eine höhere als die durch die Kamera vorgegebene Auflösung nutzen zu wollen. D.h. es soll eine subpixelgenaue Position der Kante bestimmt werden.

Zu diesem Zwecke kann ein existierender Ansatz zur subpixelgenauen Kantenbestimmung, anhand der Information die das Gradientenbild bietet, genutzt werden. Ein häufig in der Literatur aufgeführtes Verfahren (siehe z.B. [Devernay, 1995] oder [Hornberg, 2006]) nutzt zu diesem Zwecke ein quadratisches Polynom, mit dem die Kante in der Nähe ihres Scheitelpunkts interpoliert wird (siehe Abb. 6.17). Der Scheitelpunkt selbst beschreibt in diesem Fall die subpixelgenaue Kantenposition, die es näherungsweise zu bestimmen gilt. Da in dieser Arbeit nur die y-Werte (aus Sicht des Bildes) der Kantenpositionen einer subpixelgenauen Bestimmung bedürfen, werden sich die weiteren Ausführungen ausschließlich auf diese beziehen. Dies bedeutet auch, dass es für den Gradienten in diesem eindimensionalen Fall nur 2 Richtungen gibt.

Das in in Kapitel 5.1.3 vorgestellte Verfahren aus [Wippich, 2008a] zur subpixelgenauen Kantenbestimmung nutzt das gewichtete arithmetische Mittel, welches hier u.a. aus Gründen des potentiell höheren Zeitbedarfs nicht verwendet wird.

Als Stützstellen der Interpolation dienen nach [Devernay, 1995] das lokale Maximum $|\nabla(y,x)|$ des Gradientenbetrages, in der Richtung, in die er zeigt, sowie die 2 benachbarten Werte $|\nabla(y-1,x)|$ und $|\nabla(y+1,x)|$. Zur Suche nach dem Interpolationspolynom $p(y) = \sum_{i=0}^2 n_i y^i$

lässt sich nun mit der Annahme $y = 0$ folgendes Gleichungssystem aufstellen

$$n_0 - n_1 + n_2 = a \quad (6.20)$$

$$n_0 = b \quad (6.21)$$

$$n_0 + n_1 + n_2 = c \quad (6.22)$$

welches zur Lösung $n_0 = b$, $n_1 = \frac{c-a}{2}$, $n_2 = a - b - \frac{c-a}{2}$ führt. Eingesetzt in die Formel zur Bestimmung der y -Koordinate des Scheitelpunktes einer Parabel ergibt sich die Lösung y_s :

$$y_s = \frac{-n_1}{2n_2} \quad (6.23)$$

$$y_s = \frac{a - c}{2(a - 2b + c)} \quad (6.24)$$

Aus Gleichung 6.24 folgt bei einem lokalen Maximum b : $y_s \in [-0.5, 0.5]$. Dies bedeutet, dass sich bei der Definition der Koordinate y als Zentrum dieses Pixels die subpixelgenaue Position als $y_{subpixel} = y + y_s$ ausdrücken lässt.

An einem einfachen Beispiel soll nun die Korrektheit der Methode gezeigt werden: Gegeben sei eine ideale Stufenkante mit den Grauwerten $f(y - 2, x) = f(y - 1, x) = f(y, x) = 0$ und $f(y + 1, x) = f(y + 2, x) = 255$. Es ist ersichtlich, dass sich die exakte Kantenposition bei $y + \frac{1}{2}$ befinden muss. Für eine erste Ableitung über 3 Pixel ergeben sich aus den angegebenen Grauwerten die folgenden Gradientenbeträge: $|\nabla(y - 1, x)| = 0 = a$, $|\nabla(y, x)| = 255 = b$ und $|\nabla(y + 1, x)| = 255 = c$. Werden diese Werte in Gleichung 6.24 eingesetzt, lässt sich $y_s = \frac{1}{2}$ bestimmen, was zur korrekten Kantenposition $y_{subpixel} = y + \frac{1}{2}$ führt.

6.6.1.3. Formfaktor

Mittels des Formfaktors, der eine dimensionslose Größe darstellt, ist es möglich eine gefundene Form auf ihre Identität hin zu überprüfen. Oftmals wird der Formfaktor auch als Kompaktheitsmaß bezeichnet.

Der in dieser Arbeit genutzte Formfaktor bezieht sich auf die Kreisförmigkeit. Mit diesem Formfaktor F ist es möglich festzustellen, wie nahe eine Form einem Kreis kommt. Dieser kann u.a. definiert werden als (siehe [Ahn u. a., 1999])

$$F = \frac{Umfang^2}{4 \cdot \pi \cdot \text{Fläche}} \quad (6.25)$$

F gibt dabei das Verhältnis zwischen dem Flächeninhalt eines Kreises mit dem Umfang der zu testenden Form und der Fläche dieser an. Für einen Kreis hat dieses Verhältnis den Wert 1, wohingegen sich z.B. für ein Rechteck mit einer unendlichen Länge und minimaler Breite $F \mapsto \infty$ ergibt.

Da der Formfaktor innerhalb dieses Kapitels genutzt werden soll, um zu überprüfen ob eine durch ein Achteck approximierte Kontur der Kugel (Kreis) der Marke entspricht (Verfahren siehe Kapitel 5.1.3.2), muss geklärt werden, wie sich das Achteck auf den Formfaktor auswirkt.

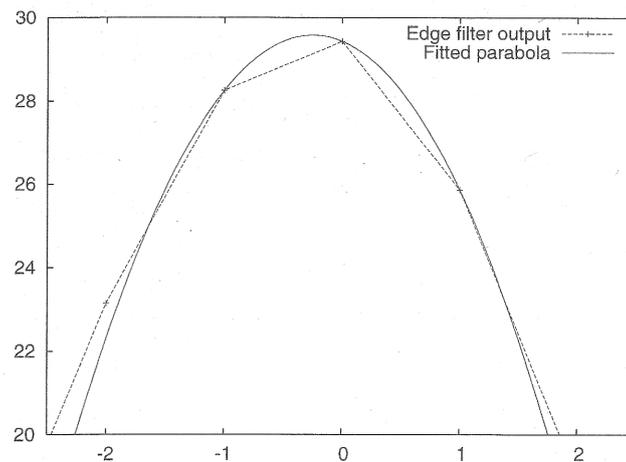


Abbildung 6.17. – Vergleich der durch einen Kantenfilter extrahierten Kantenpunkte und der interpolierten Parabel, deren Scheitelpunkt der subpixelgenauen Kantenposition entspricht. Abbildung aus [Hornberg, 2006]

Dies ist nötig, um einen Idealwert zu haben, an dem ein Schwellwert orientiert werden kann. Die Abweichung eines regelmäßigen Achtecks von einem Kreis ist mit $F = \frac{8}{\pi \cdot (1 + \sqrt{2})} \approx 1,055$ sehr gering, was bedeutet, dass die Approximation schon sehr genau ist und gleichzeitig mit einem Schwellwert in der Nähe dieses Idealwerts eine große Anzahl nicht-kreisförmiger Objekte effektiv ausgeschlossen werden kann. Dies ist möglich, da davon auszugehen ist, dass sich im Blickfeld der Kamera nicht viele als fast regelmäßiges Achteck approximierbare Formen befinden, die keine Kreise bzw. Kugeln sind. Wobei zu beachten ist, dass selbst bei Kreisen u.a. durch den genutzten Erkennungsalgorithmus nicht sichergestellt werden kann, dass es sich immer um ein regelmäßiges Achteck handelt und deshalb zur Berechnung der Fläche A des Achtecks mit den Punkten $(x_0, y_0) \dots (x_7, y_7)$ die Gauß'sche Trapezformel genutzt werden muss, da diese kein regelmäßiges Polygon voraussetzt:

$$A = \frac{1}{2} \sum_{i=0}^7 (y_i + y_{i+1})(x_i - x_{i+1}) \quad (6.26)$$

$$x_8 = x_0 \quad (6.27)$$

$$y_8 = y_0 \quad (6.28)$$

Nun kann, wie schon erwähnt, über einen Schwellwert entschieden werden, ob eine gefundene Form, die durch ein Achteck approximiert wurde, ausreichende²⁵ Kreisförmigkeit aufweist, um als Kugel der Marke anerkannt zu werden. Hierbei sind Toleranzen speziell für die mögliche Ellipsenförmigkeit aber auch Mess- bzw. Rechenungenauigkeiten zu berücksichtigen. Das genaue Intervall kann zum Beispiel durch Versuche mit Testdaten bestimmt werden.

²⁵Idealwert wäre $F = \frac{8}{\pi \cdot (1 + \sqrt{2})}$.

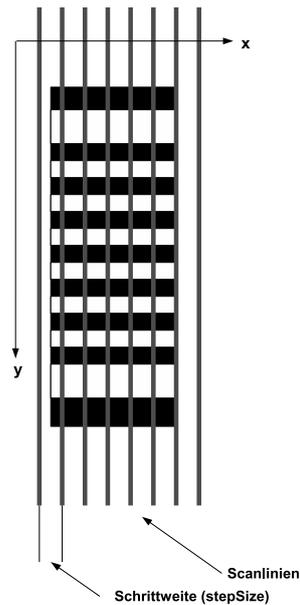


Abbildung 6.18. – Barcode mit Scanlinien.

6.6.2. Kantenfilterung

Das von der Kamera aufgenommene Grauwertbild wird in diesem Verarbeitungsschritt einer Kantenfilterung unterzogen, die auf einem einfachen Scanlinienverfahren basiert. Daraus ergibt sich, dass mit Abtaststreifen, den Scanlinien, die in einem Abstand von *stepSize* Spalten spaltenweise über das Bild gelegt werden (siehe Abb. 6.18), die Barcodes der Marken bis zu einer in Kapitel 6.5.1 erläuterten Drehung des Zylinders erkannt werden können. Es ist zu erkennen, dass es hierbei also nur auf eine Erkennung der Balken des Barcodes in y-Richtung des Bildes ankommt. Aus diesem Grund erscheint es sinnvoll auch eine entsprechende Kantenfilterung, die diese Balken aus dem Bild extrahieren soll, nur in dieser Richtung anzuwenden, da hiermit Zeit gespart werden kann. Es ist allerdings auch klar, dass dadurch mit dem Sobel-Operator keine Orientierung des Barcodes bestimmt werden kann und es bei großen Drehwinkeln ($\gg 45^\circ$), so diese überhaupt möglich sind, zu (sehr) kleinen Werten als Maskenergebnis kommen kann.

Auf das Eingabebild wird nun ohne vorherige Vorverarbeitung die y-Maske des Sobel-Operators (siehe Kapitel 6.6.1) angewendet, was für jedes Pixel der untersuchten Spalten passiert. Eine Vorverarbeitung mittels Rauschfilter erschien durch die zu dieser Arbeit durchgeführten Tests als unnötig, da sich das Rauschen bei den verwendeten Kameraeinstellungen in tolerablen Grenzen hielt und nur sehr geringen Einfluss auf die Erkennung hatte, was ein Weglassen dieses Verarbeitungsschrittes, auch gerade im Hinblick auf die dadurch gewonnene Zeit, sinnvoll erscheinen lässt. Bei anderen Kameras oder Aufnahmebedingungen kann aber u.U. ein solcher Rauschfilter nötig sein, weshalb dieser dann im Einzelfall nachgerüstet werden

müsste. Dieser Filter sollte aber auch sehr kleine Kanten (1-2 Pixel) erhalten können, um die Erkennung auch auf weitere Entfernungen zu ermöglichen.

Dem Sobel-Operator wird hier der Vorrang gegenüber z.B. dem in [Arıcan, 2004] durchgeführten Template Matching gegeben, da die Faltung nicht besonders rechenintensiv ist und durch den gewichteten Mittelwert der Maske auch eine Glättung senkrecht zum Gradienten vorgenommen wird. Einer der Nachteile des Sobel-Operators, die u.U. ungenaue Bestimmung der Gradientenrichtung spielt für diese Arbeit keine Rolle, da nur der Wert der y-Maske genutzt wird.

Die y-Maske des Sobel-Operator wird nun auf jeden Pixel der ausgewählten Spalte angewendet und das Ergebnis wiederum als Bild gespeichert, wobei sich dafür aus der Maske die folgende Rechenvorschrift für das Eingabebild f ergibt:

$$g(y,x) = (f(y+1,x-1) + 2f(y+1,x) + f(y+1,x+1)) - (f(y-1,x-1) + 2f(y-1,x) + f(y-1,x+1)) \quad (6.29)$$

Auf die Multiplikation von $g(y,x)$ mit dem Wert $\frac{1}{4}$, die sonst oftmals zu Zwecken der Normierung durchgeführt wird, wird hier verzichtet. Dadurch wird zwar für das Gradientenbild mehr Speicher verbraucht, dies hat sich aber nicht nachteilig auf die Laufzeit ausgewirkt, was sich bei anderen Plattformen aber anders darstellen kann.

Aus Gleichung 6.29 ist zu sehen, dass sich bei einem Wechsel von Schwarz auf Weiß²⁶ ein Wert von $g(y,x) < 0$ und bei einem Wechsel von Weiß auf Schwarz ein Wert von $g(y,x) > 0$ ergibt. Dieser Effekt ist in sofern für die Erkennung des Barcodes nützlich, da dieser sich aus weißen Balken auf schwarzem Grund zusammensetzt und es somit zu aufeinanderfolgenden Paaren von Schwarz-Weiß- und Weiß-Schwarz-Übergängen kommt, was die Erkennung des Barcodes vereinfacht.

Das so erstellte neue Bild g enthält nun die Ableitungen in y-Richtung für jedes Pixel einer Spalte x des Eingangsbildes. Um aus diesem die genauen Kantenpositionen zu extrahieren werden die lokalen Maxima der Spalten gesucht. Hierzu wird über einen festen Schwellwert ein minimaler Ableitungsbetrag von $derivThreshold$ ²⁷ festgelegt, sodass wenn dieser unterschritten wird angenommen wird, dass es sich hierbei um keine Kantenposition handeln kann. Der Schwellwert wird auf einen festen Wert gesetzt und nicht adaptiv bestimmt, da davon ausgegangen wird, dass durch die mitgeführte Infrarotbeleuchtung immer ein festes Mindestmaß an Kontrast bezüglich des Barcodes und der Umgebung vorhanden ist. Mit diesem Schwellwert werden also durch Störungen und schlechten Kontrast falsche oder verfälschte Kanten aussortiert. Um nun die pixelgenauen Kantenpositionen zu erhalten werden die verbliebenen Pixel einer sog. lokalen Non-Maxima-Suppression unterzogen, bei der alle nicht lokalen Maxima unterdrückt werden und am Ende (hauptsächlich) nur noch Kantenpositionen übrigbleiben. Diese Operation wird auch nur auf jede $stepSize$ -te Spalte angewendet, wobei

²⁶Aus der Abtastrichtung von der Oberkante zur Unterkante des Bildes gesehen.

²⁷Erklärung aller verwendeter Parameter im Text oder kompakt in Tabelle 6.2

alle Pixel, die nicht der folgenden Bedingung genügen unterdrückt werden²⁸:

$$|g(y,x)| \geq |g(y-1,x)| \quad (6.30)$$

$$|g(y,x)| > |g(y+1,x)| \quad (6.31)$$

Durch die beiden genannten Operationen sind nun die pixelgenauen Kantenpositionen bestimmt, was sich aber zur Identifizierung des Barcodes als oftmals nicht ausreichend erwiesen hat, besonders wenn sich durch die Entfernung zur Marke die Balken nur noch 1-5 Pixel groß sind. Aus diesem Grund wird aufbauend auf den gefundenen Kantenpositionen eine subpixelgenaue Bestimmung der Kante in y-Richtung durchgeführt. Hierzu wird das in den Grundlagen in Kapitel 6.6.1.2 erläuterte Verfahren zur subpixelgenauen Kantenbestimmung genutzt, welches zur Interpolation mit einem quadratisches Polynom arbeitet. Die drei benötigten Werte ergeben sich für den Bildpunkt $g(y,x)$ aus dem darüber- und darunterliegenden Pixel, also $g(y-1,x)$ und $g(y+1,x)$ und dem lokalen Maximum selbst.

Nun liegen für jede Kantenposition zwei verschiedene Informationen vor:

- In y-Richtung subpixelgenaue Kantenposition (x- und y-Koordinate)
- Ergebnis der Sobelmaske (z.B. 0=Schwarz-nach-Weiß, 1=Weiß-nach-Schwarz), der genaue Wert ist für die weiteren Schritte nicht mehr interessant, weshalb er durch eine Konstante ersetzt wird.

Es ist in diesem Stadium nicht mehr sinnvoll diese Informationen jeweils in einem Bild bzw. großen Array zu speichern, dessen Größe dem Eingangsbild entspricht, da die Anzahl der Kantenpositionen im Vergleich zu der Gesamtanzahl der Pixel des Eingangsbildes meistens gering ist. Aus diesem Grunde werden nur noch die subpixelgenaue Kantenposition und die Codierung des Übergangs in einer speziellen Datenstruktur gespeichert, um die Laufzeit der nachfolgenden Schritte zu verringern. Dies ist möglich, da das Iterieren über ein komplettes Bild mit zu überprüfenden Bedingungen meistens mehr Zeit in Anspruch nimmt, als der Aufbau und die Nutzung einer Datenstruktur, die nur noch die benötigten Daten enthält. Als Datenstruktur bietet sich z.B. eine Hashmap mit folgendem Aufbau an:

DATENSTRUKTUR

1 *Map* < *X – Coord*, List of Container < *Y – Coord*, *Transition* >>

Im folgenden können die hier beschriebenen Operationen nochmal als Pseudocode betrachtet werden, ebenfalls befindet sich in Abbildung 6.19 eine Illustration des Vorgehens:

²⁸Die Gleichungen 6.30 und 6.31 stellen eine Vereinfachung dar, denn in der Realität muss auf die Vorzeichen der einzelnen Werte geachtet werden.

EDGEDETECTION(I, E)

▷ I Eingangsbild; E Datenstruktur (Map) für die Kantenpositionen und den Übergang

```

1  for y ← 1 to height[I] do
2      for x ← 1 to width[I] do
3          if (x ÷ STEPSIZE = 0) then
4              S[y][x] ← YSOBEL(I, y, x)
5          endif
6      endfor
7  endfor
8  for y ← 1 to height[S] do
9      for x ← 1 to width[S] do
10         if (x ÷ STEPSIZE = 0) then
11             if (|S[y][x]| > DERIVTHRESHOLD) then ▷ Unterdrückung zu kleiner Werte
12                 if (S[y][x] > 0) then ▷ Übergang Schwarz->Weiß
13                     if (S[y][x] ≥ S[y-1][x] ∧ S[y][x] > S[y+1][x]) then ▷ Non-Maxima-Suppression
14                         ▷ Einfügen der neuen Kantenposition an das Ende der Liste für die jeweilige Spalte x.
15                         E[x] ← E[x] + new CONTAINER(SUBPIXELINTERPOLATION(S, x, y), BLACKWHITE)
16                     endif
17                 else ▷ Übergang Weiß->Schwarz
18                     if (S[y][x] ≤ S[y-1][x] ∧ S[y][x] < S[y+1][x]) then ▷ Non-Maxima-Suppression
19                         ▷ Einfügen der neuen Kantenposition an das Ende der Liste für die jeweilige Spalte x.
20                         E[x] ← E[x] + new CONTAINER(SUBPIXELINTERPOLATION(S, x, y), WHITEBLACK)
21                     endif
22                 endif
23             endif
24         endif
25     endfor
26 endfor
27 return

```

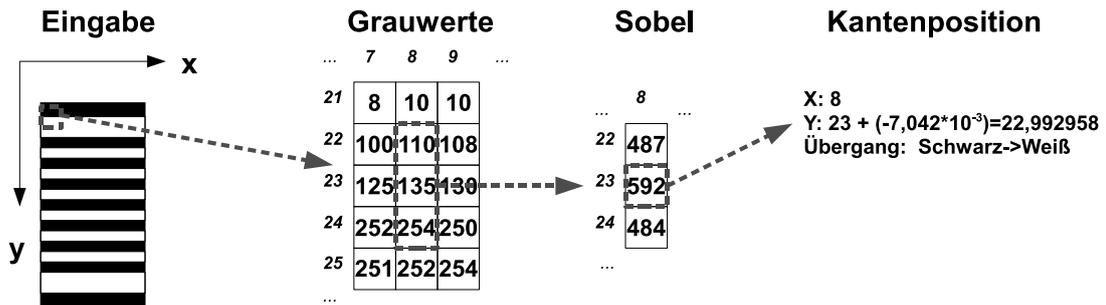


Abbildung 6.19. – Ablauf der subpixelgenauen Kantendetektion mit einer 3 x 3 Sobelmaske in y-Richtung.

Wie dem Pseudocode zu entnehmen ist, wird die sonst häufig durchgeführte Prozedur des Zusammenfügens einzelner nicht zusammenhängender Kantenteile zu einer zusammenhängenden Kantenlinie (Edge Linking) hier nicht durchgeführt, da es für die weiteren Verarbeitungsschritte nicht notwendig ist.

6.6.3. Barcodefilterung

Wie kommt man von den im letzten Schritt für jede *stepSize*-te Spalte des Bildes gesammelten Informationen zur Detektion der im Bild befindlichen Barcodes? Diese Frage ist die bedeutendste im Rahmen der Erkennung dieser Marke, da das initiale Suchen dieser ausschliesslich über den Barcode verläuft.

Aus dem vorangegangenen Schritt liegen für jede *stepSize*-te Spalte des Bildes Informationen über die Übergänge (Kanten) zwischen schwarzen und weißen Bereichen mit ihrer Polarität und deren subpixelgenauer y-Position vor. Ein Balken des genutzten Barcodes erscheint im Bild hell (weiß) wohingegen der schwarze Untergrund dunkel abgebildet wird. Er lässt sich also aus einem Schwarz-Weiß- und einem darauffolgenden Weiß-Schwarz-Übergang beschreiben. Die Information welcher Übergang vorliegt ist über die gespeicherte Polarität gegeben. Aus diesen Informationen lässt sich anhand der y-Position der aufeinanderfolgenden Übergänge die subpixelgenaue Breite für jeden vermeintlich zu einem Barcode²⁹ gehörenden Balken ermitteln. Es ist weder sichergestellt, dass jeder gefundene Übergang zu einem Balken des Barcodes gehört, noch schreiben die verwendeten Codes explizit vor welche Balken wo stehen müssen. Dies ist z.B. bei UPC-Barcodes anders, hier ist gibt es in der Mitte, am Start und am Ende des Barcodes Balken mit fester Breite (meist schmal), anhand derer sich das Lesegerät kalibrieren kann.

D.h. für diese Arbeit muss ein Ansatz gewählt werden, bei dem das Verfahren anhand der

²⁹Auch wenn es sich bei diesen aufeinanderfolgenden Übergängen nicht immer um Balken des Barcodes handeln muss, wird dieser Begriff trotzdem zur Erläuterung der verwendeten/entwickelten Verfahren genutzt.

vorliegenden Informationen automatisch einen Basiswert errechnet, um die Balken jeweils als schmal oder breit klassifizieren zu können. Dabei helfen unterstützend die berechneten Breiten der einzelnen Balken und das Wissen um die Anzahl der verschiedenen Balken, die der verwendete Barcode definiert. Die zur Veranschaulichung in diesem Abschnitt genutzten Beispiele beziehen sich alle auf den Code 2-aus-8 (siehe Kapitel 6.4.1), für die andere Barcodes gilt das Erläuterte analog.

Der Ablauf der Suche nach einem Barcode beginnt nun in jeder Spalte mit der Suche nach dem ersten Vorkommen eines Schwarz-Weiß-Übergangs, auf den ein Weiß-Schwarz-Übergang folgen muss. Da bekannt ist, dass der 2-aus-8 Barcode genau 8 Balken besitzt ist nach der Detektion von 8 Übergangspaaren ein möglicher Code gefunden. Nun muss, wie oben schon erläutert, durch ein Verfahren bestimmt werden, welcher der Balken breit oder schmal ist. Zu diesem Zwecke wurden zwei verschiedene Verfahren ausprobiert:

1. **Maximum-Likelihood-Verfahren:** (Likelihood) Dieses Verfahren stammt aus der Arbeit von Coughlan u.a. (siehe [Coughlan u. a., 2006]) und arbeitet mit einem sehr einfachen Maximum-Likelihood-Verfahren. Coughlan u.a. gehen davon aus, dass es bei dem von ihnen genutzten Barcode, wie in dieser Arbeit auch, keine festen Balken zur Kalibrierung des Verfahrens gibt. Dies wäre eigentlich noch kein Problem, da man auch einfach den schmalsten Balken des Barcodes zur Kalibrierung nutzen könnte, dies ist aber aufgrund fehlender Prüfbits des Barcodes u.ä. nicht besonders zu empfehlen. Aus diesem Grunde muss ein anderes Verfahren zur Bestimmung einer Basisbalkenbreite eines schmalen Balkens gefunden werden, anhand derer die Balken klassifiziert werden können.

Maximum-Likelihood-Verfahren dienen u.a. dazu die Kennwerte einer Stichprobe zu bestimmen, wenn diese von einem unbekanntem Parameter abhängen. Das Maximum-Likelihood-Verfahren ist also ein Schätzverfahren, das den zu bestimmenden Parameter so schätzt, dass die Wahrscheinlichkeit, dass er zur Stichprobe gehört maximiert wird. Als Parameter wird in diesem Fall die Breite der schmalen Balken gewählt, da von ihr die Breite der breiten Balken abhängt. Diese Abhängigkeit kann für den 2-aus-8 Code als Gleichung wie folgt ausgedrückt werden:

$$W = 2N \quad (6.32)$$

Um nicht mit Ableitungen von Dichtefunktionen arbeiten zu müssen wählen Coughlan u.a. eine nichtnormalisierte negative logarithmierte Likelihood-Funktion, was dazu führt dass der Wert für N gefunden werden muss, für den L minimal ist:

$$L(N) = \sum_{i=1}^8 \min(|B(i) - N|, |B(i) - 2N|) \quad (6.33)$$

Der Parameter $B(i)$ in Gleichung 6.33 bezeichnet dabei die Breite des i -ten Balkens des Barcodes. Die Werte für N können einem vorher festgelegten Intervall entspringen, das sich z.B. an der minimalen und maximalen Breite der im Barcode befindlichen Balken orientiert. Dieses wird dann schrittweise durchlaufen, wobei sich bei der subpixelgenauen

Betrachtung in dieser Arbeit eine Schrittweite von 0,1 als ausreichender Kompromiss zwischen Genauigkeit und Ausführungsgeschwindigkeit erwiesen hat. Nach dem Durchlauf des Intervalls wird der Wert N für den L sein globales Minimum erreicht als der Wert angesehen, anhand dessen die Balken klassifiziert werden. Das ausgewählte N ist also der Schätzwert (Basiswert oder auch Klassenmitte) für einen schmalen Balken des untersuchten Barcodes. Es ist zu erkennen, dass dieses Verfahren, gerade bei sehr breiten Balken, laufzeitintensiv ist. Dies kann auch nicht durch die einfache Suche nach dem ersten lokalen Minimum von L optimiert werden, da dieses lokale Minimum nicht dem globalen Minimum entsprechen muss. Andere Möglichkeiten der Optimierung müsste man sich ggf. noch überlegen.

2. **Bestimmung des Durchschnitts der 6³⁰ schmalsten Balken:** (narrowAvg) Inspiriert durch das Maximum-Likelihood-Verfahren baut dieses Verfahren schon auf dem Vorwissen über das Aussehen von korrekten Barcodes auf. Aus den 8 gefundenen Balken werden die 6 schmalsten Balken (N_1 bis N_6) bestimmt. Durch Mittelwertbildung ergibt sich dann der Schätzwert N zum klassifizieren der Balken:

$$N = \frac{N_1 + N_2 + N_3 + N_4 + N_5 + N_6}{6} \quad (6.34)$$

Die zwei zur Verfügung stehenden Verfahren können nun genutzt werden, um eine Klassifizierung der gefundenen Kanten blockweise vorzunehmen.

Da nicht davon ausgegangen werden kann, dass alle Balken einer Klasse gleich groß sind und dass die breiten und schmalen Balken sich genau um den durch den Code vorgegebenen konstanten Faktor unterscheiden, müssen Toleranzen geschaffen werden. Für die Festlegung der Toleranzen sind die Extrema der Unterschiede zwischen den Größen der beiden Balkenbreiten von maßgebender Bedeutung. In [Coughlan u. a., 2006] wird z.B. die obere Toleranzgrenze des schmalen Balkens mit dem Faktor 1,5 angegeben. In dieser Arbeit können aber auch z.B. durch ein Training mit Testdaten bestimmte Intervalle genutzt werden (siehe Tabelle 6.2 und Kapitel 6.9.4). Dies kann nützlich sein, da über das enge ziehen der Grenzen möglichst viele Muster, die keinen zu einer Marke gehörenden Barcode darstellen, ausgeschlossen werden sollen.

Als Vergleich der beide Verfahren lässt sich festhalten dass beide das gewünschte Ergebnis, nämlich einen Basiswert anhand dem die Balken des Barcodes klassifiziert werden können, liefern. Das Maximum-Likelihood-Verfahren liefert sehr präzise Ergebnisse und es ist möglich viele falsche Codes, die z.B. durch Größenunterschiede auffallen, effektiv auszusortieren. Bei dem einfachen Verfahren, welches nur auf Mittelwertbildung der 6 schmalsten Balken basiert, kann es schneller zu Fehlschätzungen und damit einer Fehlidentifikation des Codes kommen. Dies hat damit zu tun, dass immer davon ausgegangen wird, dass die 6 kleinsten Balken auch wirklich schmale Balken des Barcodes darstellen. In dieser Arbeit wird das einfache Verfahren trotzdem eingesetzt, weil es theoretisch weniger Laufzeit verbraucht (siehe 6.9.8), in den

³⁰Bei Barcodes mit mehr mehr/weniger schmalen Balken ist hier eine andere Anzahl zu wählen.

gemachten Tests sehr ähnliche Ergebnisse für den Schätzwert bei korrekten Barcodes lieferte (siehe Kapitel 6.9.4) und immer die gleiche Anzahl an Marken gefunden hat. Dies hängt auch damit zusammen, dass durch den gewählten Barcode (immer 8 Balken) und eine später noch beschriebene Überprüfung der Größe der Zwischenräume, kaum andere Muster in die engere Auswahl als Barcode einer Marke kommen. Aber besonders hier könnte das Maximum-Likelihood-Verfahren seinen Vorteil ausspielen. Sollte es aber in der Praxis zu Problemen mit der einfachen Variante kommen, ist eine Nutzung des Maximum-Likelihood-Verfahrens zu empfehlen.

Andere Verfahren, wie nur den schmalsten oder breitesten Balken zur Kalibrierung bzw. Schätzung zu nutzen, wurden auch getestet, aber wegen zu ungenauer Ergebnisse nicht weiter beachtet.

Bei Benennung der Toleranzgrenzen (Intervalle) mit *narrowMin* und *narrowMax* für die Unter- bzw. Obergrenze der schmalen Balken, sowie *wideMin* und *wideMax* bei den breiten Balken lässt sich die Klassifikation $C(S)$ eines Streifen B mit dem geschätzten Wert N eines der o.g. Verfahren C wie folgt beschreiben:

$$C(B) = \begin{cases} N & \text{if } narrowMin \cdot N \leq B \wedge narrowMax \cdot N \geq B \\ W & \text{if } wideMin \cdot N \leq B \wedge wideMax \cdot N \geq B \\ Error & \text{otherwise} \end{cases} \quad (6.35)$$

(6.36)

Für die Erkennung eines korrekten Barcodes spielen auch die einzelnen Abstände zwischen den Balken eine Rolle. Deshalb werden auch diese bei der Überprüfung mit einbezogen, da auch sie sich über ein Verhältnis zu N ausdrücken lassen. Im Fall des Codes 2-aus-8 entspricht es genau 1, wobei auch hier Toleranzen (*spaceMin* u. *spaceMax*) eingebaut werden müssen, da die Zwischenräume dazu neigen kleiner abgebildet zu werden, als die Balken aus Reflexfolie.

Nach dem klassifizieren der einzelnen Balken und der Überprüfung der Zwischenräume des gefundenen Codes kann nun festgestellt werden, ob es sich um einen korrekten Barcode handelt, indem die korrekte Anzahl der schmalen und breiten Balken überprüft wird. Sollte ein Barcode als korrekt erkannt werden, wird er in einer geeigneten Datenstruktur, vorzugsweise einer Hashmap, als String von Breitenangaben der Balken (z.B. WNNNNNNW) und den zugehörigen Anfangs- und Endkoordinaten des Barcodes gespeichert:

DATENSTRUKTUR

1 *Map* $\langle Code, ListofContainer \langle Start - Coord, End - Coord \rangle \rangle$

Da ein Code aufgrund der Redundanz die ein eindimensionaler Barcode mitbringt mehrfach auftreten kann, wird hier eine Liste der Anfangs- und Endkoordinaten der jeweiligen Fundstellen gehalten. Diese Redundanz ist gewollt und wird später genutzt, um zu verifizieren, dass

es sich bei dem gefundenen Muster tatsächlich um einen Barcode handelt, der zu einer Marke gehört. Wird das gefundene Muster nicht als Barcode erkannt, oder es gibt noch weitere Balken unterhalb der untersuchten in dieser Spalte, werden mit den unteren 7 Balken des aktuellen Musters und dem nächsten weiter unten gelegenen Balken der Spalte eine weitere Untersuchung durchgeführt. Ausgeschlossen hiervon sind Muster, bei denen aufgrund eines definierten Abstands (breitester Balken in der Auswahl $\cdot spaceMax >$ Balken) die einzelnen Balken klar zu weit auseinander liegen und somit keinen korrekten Barcode bilden können. Bei allen anderen Mustern wird eine genauere Überprüfung vorgenommen. Dieses Vorgehen stellt sicher, dass alle Barcodes, die durch die Kantendetektion übriggeblieben sind, auch tatsächlich die Chance haben erkannt zu werden.

Dieses Verfahren wird solange durchgeführt, bis keine für einen korrekten Barcode ausreichende Anzahl von untereinanderliegenden Balken, die noch nicht zusammen untersucht wurden, mehr zur Verfügung steht. Wird dies für alle Spalten, die in der Datenstruktur der Kantendetektion vorhanden sind gemacht, ist dieser Schritt abgeschlossen und es stehen alle korrekt erkannten Barcodes für eine genauere Bestimmung der einzelnen Marken zur Verfügung. Es ist bei diesem Verfahren aber auch klar, dass ein Balken durch unglückliche Umstände zu zwei korrekt erkannten Barcodes gehören kann. Dies kann nicht ausgeschlossen werden, da (an dieser Stelle der Erkennung) nicht sicher ist, welcher der Richtige wäre. Der folgende (an einigen Stellen vereinfachende) Pseudocode soll die in diesem Schritt vorgenommenen Operationen verdeutlichen:

BARCODEFILTER(E, B)

```

  ▷  $E$  Datenstruktur (Map) für die Kantenpositionen und den Übergang
  ▷ Precondition: Die Listen  $E[x]$  sind sortiert nach der Y-Koordinate ihrer Elemente
  ▷  $B$  Datenstruktur (Map) für die gefundenen Barcodes

  ▷  $keys[E]$  entspricht den Spalten, die Kanten enthalten
  ▷ CODESIZE entspricht der Anzahl der Balken des Barcodes
1  forall  $x \in keys[E]$  do
2       $wbCount \leftarrow 0$                                 ▷ Zählt die Weiß-Schwarz Übergänge
3       $bwCount \leftarrow 0$                                 ▷ Zählt die Schwarz-Weiß Übergänge
4       $searchFor \leftarrow BLACKWHITE$  ▷ Übergang, der als nächstes gefunden werden muss
      ▷  $e$  entspricht einem Übergangcontainer
5      forall  $e \in E[x]$  do
6          if ( $transition[e]=searchFor$ ) then                ▷ ges. Übergang gefunden
7               $searchFor \leftarrow -searchFor$ 
              ▷  $wb$  und  $bw$  speichern die Y-Koord. der gef. Übergänge
8              if ( $transition[e]=BLACKWHITE$ ) then
9                   $bw[bwCount] \leftarrow YCoord[e]$ 
10                  $bwCount \leftarrow bwCount + 1$ 
11                 else  $wb[wbCount] \leftarrow YCoord[e]$ 
12                  $wbCount \leftarrow wbCount + 1$ 
13             endif
14         elseif ( $transition[e]=BLACKWHITE$ ) then ▷ "falscher" Übergang gefunden
15              $wbCount \leftarrow 0$ 
16              $bwCount \leftarrow 1$ 
17              $bw[0] \leftarrow YCoord[e]$ 
18         else
19              $wbCount \leftarrow 0$ 
20              $bwCount \leftarrow 0$ 
21         endif
22          $refused \leftarrow FALSE$ 
23         if  $bwCount=wbCount=CODESIZE$  then ▷ mögl. korr. Barcode gefunden
24              $N_{Est} \leftarrow ESTIMATE(bw, wb, CODESIZE)$ 
25              $nCount \leftarrow 0$ 
26              $wCount \leftarrow 0$ 
              ▷ Schleife über alle Balken
27             for  $i \leftarrow 0$  to  $i < CODESIZE$  do
28                  $bar \leftarrow wb[i]-bw[i]$                 ▷ Größe des Balkens
              ▷ Größe des Zwischenraums prüfen
29                 if ( $i > 0 \wedge (bw[i]-wb[i-1] < N_{Est}*SPACEMIN$ 
                     $\vee bw[i]-wb[i-1] > N_{Est}*SPACEMAX)$ ) then
30                      $refused \leftarrow TRUE$ 
              ▷ Schmäler Balken
31                 elseif ( $bar \leq N_{Est}*NARROWMAX$ 
                     $\wedge bar \geq N_{Est}*NARROWMIN$ ) then
32                      $nCount \leftarrow nCount + 1$ 
33                      $code[i] \leftarrow N$                 ▷ code enthält den Schlüssel für B

```

```

34         ▷ Breiter Balken
           elseif ( $bar \leq N_{Est} * WIDEMAX$ 
35                 $\wedge bar \geq N_{Est} * WIDEMIN$ ) then
36                  $wCount \leftarrow wCount + 1$ 
37                  $code[i] \leftarrow W$     ▷ code enthält den Schlüssel für B
38             else  $refused \leftarrow TRUE$ 
39             endif
40         endfor
           if  $refused = FALSE \wedge wCount = WINCODE$  then
41             ▷ Korrektter Barcode gefunden, Position in B eintragen
              $B[code] \leftarrow B[code] + \text{new CONTAINER}(\text{POINT}(x, bw[0]),$ 
42                  $\text{POINT}(x, wb[CODESIZE - 1]))$ 
           endif
           ▷ Versuch den nächsten Barcode mit den CODESIZE-1
           ▷ unteren Balken zu bilden
43         for  $j \leftarrow 1$  to  $j < CODESIZE - 1$  do
44              $bw[j] \leftarrow bw[j + 1]$ 
45              $wb[j] \leftarrow wb[j + 1]$ 
46         endfor
47          $bwCount \leftarrow CODESIZE - 1$ 
48          $wbCount \leftarrow CODESIZE - 1$ 
49     endif
50 endforall
51 endforall

```

6.6.4. Markenextraktion

Aus dem letzten Schritt der Erkennung sind Listen mit allen korrekten vertikalen Barcodes und deren Start- und Endpositionen entstanden, die jeweils einen Pixel breit sind und im Sinne der Verständlichkeit im weiteren Verlauf Barcode-segmente genannt werden, da sie einem Segment des kompletten eindimensionalen Barcodes einer Marke entsprechen. Wie schon gezeigt wurde sind diese Listen für jede ID getrennt angelegt. Nun gilt es aus diesen Barcode-segmenten jene herauszufiltern, die zu einer Zylindermarke gehören.

Es kann nicht von Anfang an davon ausgegangen werden, dass alle gefundenen Barcode-segmente zu einer Marke gehören, da diese Muster u.U. auch durch andere im Bild abgebildeten Gegenstände entstanden sein können. Des Weiteren ist es auch möglich, dass Teile eines zu einer Marke gehörenden Barcode-segments und andere sich im Bild befindliche Gegenstände zusammen wieder ein korrektes Barcode-segment ergeben.

Um die gefundenen Barcode-segmente endgültig zu verifizieren ist es nötig die unter den Zylinder gehängte Kugel, deren Mittelpunkt den Messpunkt darstellt, zu erkennen. D.h. ein Barcode-segment kann nur zu einer gültigen Marke gehören, wenn es eine passende Kugel

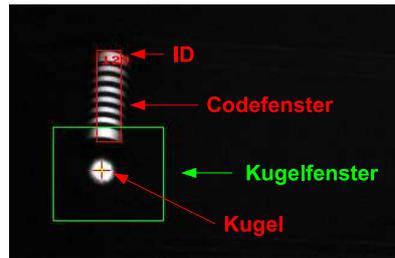


Abbildung 6.20. – Exemplarische Abbildung einer Zylindermarke mit Codefenster, Kugelfenster, Kugelmitte und ID.

für ihn gibt. Dabei ist es nicht so einfach wie bei der Doppelverhältnismarke diesen Messpunkt zu finden, da es aufgrund der gewollten Sichtbarkeit aus (fast) allen Richtungen keine Umrahmung für die Kugel geben kann, die beim Auffinden hilfreich wäre.

Um die Marke zu erkennen kann die Redundanz des eindimensionalen Barcodes genutzt werden. Was bedeutet, dass es sinnvoll ist einen Schwellwert festzulegen, der eine Mindestredundanz an Barcodesegmenten gewährleistet, die mindestens eingehalten werden muss, um überhaupt die Suche nach einer zugehörigen Kugel zu beginnen. Gleichzeitig muss neben der reinen Anzahl der gefundenen Barcodes, die die gleiche ID codieren, auch eine Analyse der Positionen dieser stattfinden. Zu diesem Zwecke wird ein sog. Codefenster aufgebaut, welches auch die Grundlage zur Berechnung eines geeigneten Fensters für die Suche nach der Kugel bildet. Hauptsächlich kann dieser Schritt nun auf drei Teilschritte reduziert werden, die im folgenden genauer beschrieben werden. Dies sind die Bestimmung des Codefensters, des Kugelfensters und der Kugelmitte, welche in Abbildung 6.20 exemplarisch zu sehen sind.

6.6.4.1. Bestimmung des Codefensters

Der Zweck dieses Fensters besteht darin gefundene Barcodesegmente, die die gleiche ID codieren, zu einer Marke zusammenzufassen, um diese genauer lokalisieren zu können.

Ausgehend vom ersten Barcode in der Datenstruktur, wird das erste Segment dieser Liste zur Bestimmung einer maximalen Höhe und Breite der Marke genutzt, um die weiteren Segmente dieser Liste auf Zusammengehörigkeit prüfen zu können. Voraussetzung hierfür ist die Sortierung der einzelnen Barcodelisten nach aufsteigender x-Koordinate (Spalte).

Bei der Bestimmung der maximalen Höhe des Codefensters handelt es sich um ein Toleranzmaß zur Einordnung der Segmente, da aufgrund einer leicht gedrehten Abbildung der Marke (siehe Kapitel 6.5.1) o.ä., nicht alle Segmente die gleichen Werte für die y-Koordinaten ihres Start- und Endpunkts haben.

Ausgehend vom ersten Segment der Liste, welches als Basis $((xStart_{Base}, yStart_{Base})$ ³¹ u. $(xEnd_{Base}, yEnd_{Base})$) dient, kann die maximale Höhe, bzw. die dazugehörigen y-Koordinaten ($minYCode$ u. $maxYCode$), folgendermaßen ausgedrückt werden:

$$codeHeight = (yEnd_{Base} - yStart_{Base}) \quad (6.37)$$

$$minCodeHeight = codeHeight - (codeHeight \cdot codeSizeTolerance) \quad (6.38)$$

$$maxCodeHeight = codeHeight + (codeHeight \cdot codeSizeTolerance) \quad (6.39)$$

$$minYCode = yStart_{Base} - (codeHeight \cdot codeYTolerance) \quad (6.40)$$

$$maxYCode = yEnd_{Base} + (codeHeight \cdot codeYTolerance) \quad (6.41)$$

Weiterhin wird hierbei auch die minimale bzw. maximale Höhe der Segmente bestimmt ($minCodeHeight$ u. $maxCodeHeight$), um nur Segmente ungefähr gleicher Größe zusammenzufassen. Aus Gleichung 6.40 ist zu entnehmen, dass die Berechnung des Intervalls $[minYCode, maxYCode]$ allein auf Basis eines vorgegebenen Parameters $codeYTolerance$ geschieht. Dies ist nötig, da genauere Werte nicht zur Verfügung stehen, weil hierfür z.B. die Orientierung der Marke bekannt sein müsste.

Für die maximale Breite, bzw. deren zugehörigen x-Koordinaten ($minXCode$ u. $maxXCode$), findet eine Schätzung mit Bezug auf das Verhältnis aus Breite und Höhe des gesamten Barcodes der realen Marke ($width_{Barcode}$) und ($height_{Barcode}$) statt³²:

$$minXCode = xStart_{Base} \quad (6.42)$$

$$maxXCode = xStart_{Base} + ((yEnd_{Base} - yStart_{Base}) \cdot \frac{width_{Barcode}}{height_{Barcode}}) \quad (6.43)$$

Das Verhältnis $\frac{width_{Barcode}}{height_{Barcode}}$ entspricht nicht unbedingt den in der Realität durch die Abbildung der Kamera gegebenen Bedingungen, wie z.B. in Kapitel 6.5.2 schon beschrieben wurde, da dieses Verhältnis z.B. nicht Invariant gegenüber der auftretenden perspektivischen Verzerrung ist. Deshalb muss für diesen Parameter ($codeRatio$) ein Wert angenommen werden, der Toleranzen für diese Situationen berücksichtigt. Für jedes weitere Segment dieser Liste wird dann geprüft ob es in die beiden Intervalle $[minXCode, maxXCode]$ und $[minYCode, maxYCode]$ passt und die Mindest- bzw. Maximalgröße ($minCodeHeight$ u. $maxCodeHeight$) erfüllt. Die beiden Intervalle spannen also ein Suchfenster auf.

Sollte nach einem Durchlauf über alle Segmente der Liste nicht die festgelegte Mindestanzahl an Segmenten $minCodeSegments$ zusammengekommen sein, die sich innerhalb der errechneten Intervalle befinden, findet eine weiterer Durchlauf mit dem zweiten Segment der Liste als Basis statt. D.h. mit dem ersten Segment als Basis war keine Marke zu finden, weshalb jetzt eine andere Basis versucht wird. Dieses Vorgehen findet nicht nur dann statt, wenn $minCodeSegments$ nicht erfüllt ist, sondern auch, wenn keine Kugel gefunden wurde

³¹Es sei noch einmal darauf hingewiesen, dass alle Parameter der Erkennung kompakt beschrieben in Tabelle 6.2 zu finden sind. Eine Visualisierung der in diesem Schritt verwendeten Parameter ist auch in Abbildung 6.21 und 6.22 zu finden.

³²Mit $width_{Barcode}$ ist hier der Durchmesser des Zylinders gemeint.

oder andere Fehler auftreten. In der Praxis hat sich dieser Ansatz aber als zu zeitaufwendig herausgestellt, sodass alle Segmente, die schon einmal als gültige Segmente einer Basis verifiziert wurden nicht mehr selbst Basis werden können. Dies wird nur in extremen Ausnahmefällen (gleichhohe Segmente durch Störung neben echtem Barcode) dazu führen, dass einzelne Marken nicht erkannt werden können, sodass dieses Risiko zum Vorteil der besseren Zeiteffizienz eingegangen werden kann.

Insgesamt wird bei diesem Vorgehen davon ausgegangen, dass wenn ein Bereich, der viele gleiche Barcode-segmente enthält und eine dazu passende Kugel gefunden wurde, die Marke mit dieser ID korrekt erkannt wurde, sodass u.U. noch in der Liste verbliebene Segmente, die nicht in die Intervalle passten vernachlässigt werden können, da es nur eine Marke mit dieser ID im Bild geben kann. Dies könnte zur Sicherheit aber auch noch überprüft werden, um zwei mit der selben ID ausgestattete Marken als Fehler zu ermitteln. Es ist außerdem anzumerken, dass bei diesem Vorgehen die zu einer Marke gehörenden Barcode-segmente nicht genau aneinanderliegen (Pixel an Pixel) müssen, sondern hier auch größere Lücken entstehen können, was sehr hilfreich ist, um auch teilweise verdeckte Marken erkennen zu können.

Das gesuchte Codefenster ist, wie in Abbildung 6.20 zu sehen, ein Rechteck mit zwei beschreibenden Punkten (links oben u. rechts unten bzw. $cw1$ und $cw2$). Der obere Punkt wird über die x-Koordinate des Startsegments (Basis) gebildet, die y-Koordinate entspricht der kleinsten y-Koordinate aller zugehörigen Segmente. Der untere Punkt wird über die x-Koordinate des Segments mit dem größten Wert dieser gebildet, wohingegen die y-Koordinate der größten y-Koordinate aller zugehörigen Segmente entspricht. In Abbildung 6.21 ist eine vereinfachte graphische Beschreibung der meisten hier verwendeten Begrifflichkeiten zu finden.

6.6.4.2. Bestimmung des Kugelfensters

Das Kugelfenster stellt den Bereich des Bildes dar, welcher nach der Kugel einer Marke durchsucht werden soll (siehe Abb. 6.22). Um dieses Fenster bestimmen zu können, muss man sich klarmachen, wo die Kugel zu finden sein muss: Die Kugel hängt immer in der Mitte unter dem Zylinder. Um die Kugel lokalisieren zu können, müsste man also die genaue Breite und Orientierung der Marke, dessen Barcode gefunden wurde, kennen, um die Mitte des Zylinders ermitteln zu können. Ausserdem wird noch der Durchmesser der Kugel benötigt, um ein entsprechendes Suchfenster aufspannen zu können. Die Orientierung soll bei diesem einfachen Verfahren nicht bestimmt werden, weshalb wie bei *codeRatio* mit einfachen Schätzungen gearbeitet wird.

Es wird aus der im Bild gefundenen Höhe ($height_{cw}$) und der Breite ($width_{cw}$) des Barcodes, hier zur Vereinfachung stellvertretend bei beiden Werten die Maße des Codefensters³³, ein Verhältnis errechnet, welches mit einem z.B. vorher an der Marke ausgemessenen Quotienten

³³Alternativ wäre hier z.B. statt $height_{cw}$ auch die Nutzung des größten Barcode-segments möglich, bzw. u.U. manchmal auch sinnvoller (breiter aber niedriger Zylinder).

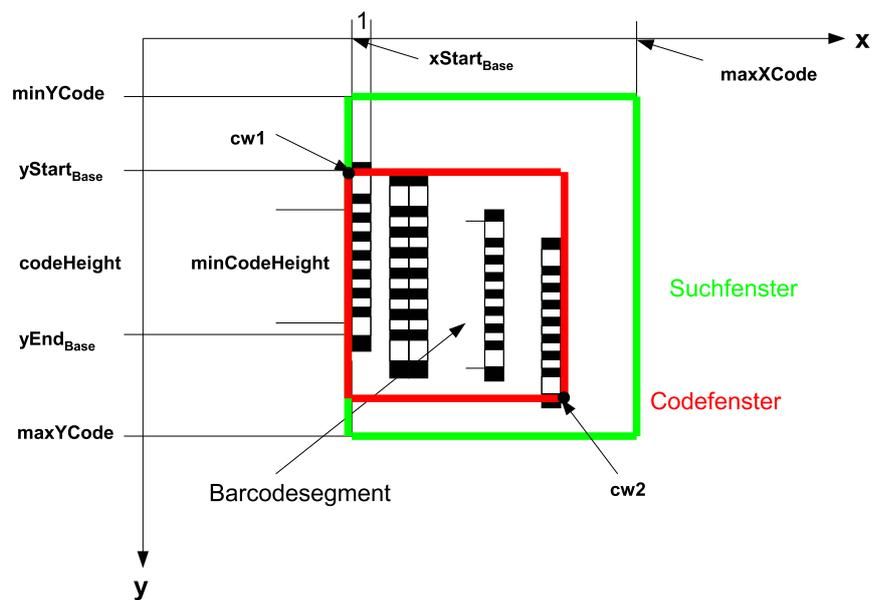


Abbildung 6.21. – Darstellung der in Kapitel 6.6.4.1 verwendeten Bezeichnungen.

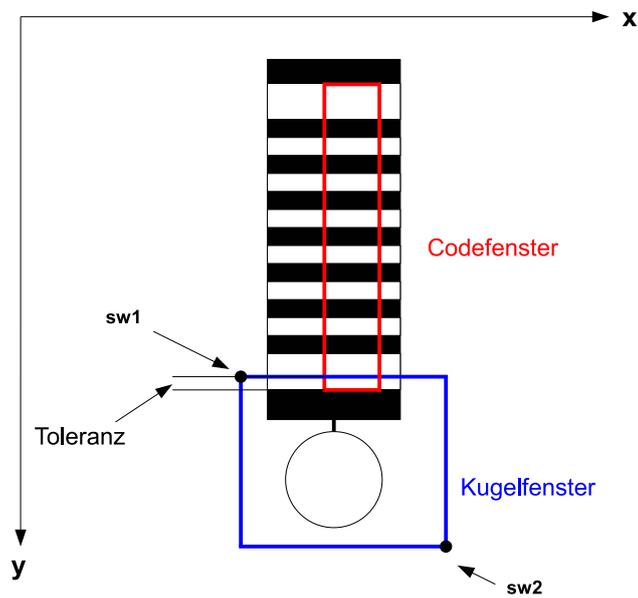


Abbildung 6.22. – Exemplarische Darstellung des Kugelfensters.

verglichen werden kann. Bei der Wahl dieses Quotienten ist nur in Fällen bei denen der Durchmesser der Kugel geringer als der des Zylinders ist, die Verwendung des Zylinderdurchmessers sinnvoll bzw. nötig. In anderen Fällen muss hier mit dem Durchmesser der Kugel gearbeitet werden.

Es kann nun folgende Gleichung aufgestellt werden:

$$height_{cw} = cw2_y - cw1_y \quad (6.44)$$

$$width_{cw} = cw2_x - cw1_x \quad (6.45)$$

$$\frac{height_{cw}}{width_{cw} \cdot factor_{gap}} = codeSphereRatioX \quad (6.46)$$

$$(6.47)$$

Der Parameter *codeSphereRatioX* in Gleichung 6.46 gibt das genannte Verhältnis aus Höhe des Barcodes und dem Durchmesser der Kugel³⁴ an. *codeSphereRatioX* muss, wie auch *codeRatio*, außerdem genügend Toleranz für perspektivisch verkürzte und gedreht abgebildete Marken bieten, da sich diese in ihrer Höhe bzw. Breite gegenüber der (fast) immer gleich abgebildeten Kugel verändern. Wird die Gleichung 6.46 nach *factor_{gap}* aufgelöst, kann errechnet werden, um wieviel der gefundene Barcode/Zylinder in der Breite vom vorgegebenen Verhältnis abweicht. Das Kugelfenster wird auf die Größe der gemessenen Breite *width_{cw}* festgelegt und um den fehlenden Anteil *factor_{gap}* erweitert. Da aber nicht bekannt ist, an welcher Seite der Marke etwas fehlt, muss dieser Anteil an beiden Seiten der Marke hinzuaddiert, bzw. subtrahiert werden, sodass sich die x-Koordinaten des Fensters wie folgt ergeben:

$$sw1_x = cw1_x - (width_{cw} \cdot factor_{gap}) \quad (6.48)$$

$$sw2_x = cw2_x + (width_{cw} \cdot factor_{gap}) \quad (6.49)$$

Als Ausgangswert für die zugehörigen y-Koordinaten dient die untere Kante des Codefensters. *sw2_y* lässt sich durch die Addition eines Produktes, welches aus dem vorher zu definierenden Verhältnis *codeSphereRatioY* und der Höhe des Barcodes (bzw. des Codefensters) besteht, zu dem Wert *cw2_y* bilden. *codeSphereRatioY* gibt hierbei das Verhältnis des Abstands der Unterkante der Kugel vom Barcode und der Höhe des Barcodes (bzw. des Codefensters) an. Für *sw1_y* könnte eigentlich der Wert von *cw2_y* genutzt werden. Hierbei ist wieder zu beachten, dass diese Werte so gewählt werden müssen, dass auch Marken die einer Rotation oder perspektivischer Verzerrung unterliegen erkannt werden können. Wird die Marke im Bild gedreht abgebildet, kommt es zu der Situation, dass die Kugel in das Codefenster hineinragt, dies wird durch die Toleranz *sphereYTolerance* berücksichtigt. Die y-Koordinaten des Kugelfensters lassen sich nun bestimmen als:

$$sw2_y = cw2_y + (height_{cw} \cdot codeSphereRatioY) \quad (6.50)$$

$$sw1_y = cw2_y - (height_{cw} \cdot sphereYTolerance) \quad (6.51)$$

³⁴In dieser Arbeit wird der Durchmesser des Zylinders genutzt, weil dieser größer als der Durchmesser der Kugel ist und dieses Verfahren sonst schon bei einfachen Fällen ohne Perspektive oder Rotation nicht funktionieren würde. Dies hängt damit zusammen, dass dann nicht sichergestellt werden könnte, dass die Mitte unter dem Zylinder wirklich vom Suchfenster abgedeckt wird.

6.6.4.3. Bestimmung der Kugelmitte

Mit dem Kugelfenster ist der Bereich, in der die zum Zylinder gehörende Kugel liegen kann hinreichend genug eingegrenzt, um eine Kreis- bzw. Ellipsensuche durchzuführen. Hierfür wird die schon in Kapitel 5.1.3.2 vorgestellte Methode zur Kreiserkennung genutzt, bei der die Kontur des Kreises durch ein Achteck approximiert wird. Dieses Verfahren wird um den in Kapitel 6.6.1 beschriebenen Formfaktor zur Überprüfung der gefundenen Form erweitert. Die Schrittweite des Rasters, welches dieses Verfahren zur Suche der Kugel nutzt, ergibt sich aus *sphereDiaMin*. *sphereDiaMin* muss vorher festgelegt werden und beschreibt das Verhältnis zwischen dem Durchmesser der Kugel und der Höhe des Barcodes (bzw. des Codefensters). Auch hier sollten wieder Toleranzen, besonders für die perspektivische Verkürzung des Barcodes, berücksichtigt werden.

Andere Verfahren, wie die Kreiserkennung auf der Basis von Hough-Transformationen wurden zwar getestet, aber entweder wegen zu großem Zeitbedarf oder Problemen bei der Erkennung von sehr kleinen Kreisen (Durchmesser <20 Pixel) nicht weiter genutzt. Das genutzte Verfahren liefert eine Liste mit allen im Kugelfenster gefundenen Formen, die sich als Achteck darstellen lassen, sowie deren Mittelpunkt als subpixelgenaue 2D Koordinaten. Es ist nicht zu erwarten, dass hierbei immer nur die zum Zylinder gehörende Kugel gefunden wird, weshalb die potentiellen Kreise noch weiteren Prüfungen unterzogen werden:

- **Formfaktor:** Da nicht zu erwarten ist, dass ausschließlich perfekte Achtecke gefunden werden, muss ein Intervall [*minFormFactor*, *maxFormfactor*] definiert werden, in dem z.B. auch leichte Ellipsen enthalten sind. Somit lassen sich effektiv Formen aus der Ergebnisliste herausfiltern, die nicht durch Abbildung der Kugel entstanden sein können. Als Beispiel einer solchen Form sei der unterste Balken des Barcodes genannt, welcher sich bei nicht gedrehten Marken oftmals im Kugelfenster befindet.
- **y-Koordinate:** Die y-Koordinate des Zentrums des gefundenen Kreises darf nicht höher liegen, als der größte y-Wert (also der unteren Kante) des Codefensters. Dies ist zumindest bei den in dieser Arbeit verwendeten Maßen selbst bei Rotation und der festgelegten maximalen Neigung der Marken nicht möglich. außerdem wird für die y-Koordinate des Zentrums des gefundenen Kreises geprüft, dass sie nicht größer als der y-Wert der unteren Kante des Codefensters addiert zu dem Produkt aus *sphereCenterMaxY* und der Barcodehöhe (bzw. der Codefensterhöhe) *height_{cw}* ist.
- **Maße:** Um zu kleine oder zu große Kreise ausfiltern zu können wird eine Überprüfung des Durchmessers des Kreises durchgeführt. Hierbei werden 2 verschiedene Werte gebildet, um auch Ellipsen berücksichtigen zu können. Diese Werte werden mit einem Intervall verglichen, das sich aus den Werten *circleDiaMin* und *circleDiaMax* multipliziert mit der Barcodehöhe (bzw. der Codefensterhöhe) *height_{cw}* zusammensetzt. Des Weiteren ist es sinnvoll eine minimal zulässige Größe *circleDiaMinPx* des Kreises in Pixel festzusetzen, um eine möglichst genaue Mittelpunktbestimmung zu erhalten.
- **Eindeutigkeit des Mittelpunkts:** Der gefundene Mittelpunkt wird mit den Mittelpunkten der schon gefundenen Marken verglichen, um eine Eindeutigkeit festzustellen. Sollte der

gefundene Mittelpunkt schon bei einer anderen Marke gefunden worden sein und sonst alle genannten Prüfungen bestanden haben, sind beide Marken ungültig, da nicht sicher ist, zu welcher er wirklich gehört. Man könnte versuchen durch weitere Bedingungen solche Fälle zu lösen, was aber zu dem Problem führen würde, dass es für diese Lösungen meistens Ausnahmen gibt, bei denen es dann nicht funktioniert.

Die genannten Prüfungen stellen nur einen Bruchteil der möglichen Tests dar, die durchgeführt werden könnten. Sie haben sich während den Tests zu dieser Arbeit aber als ausreichend erwiesen.

Sollte nach diesen Prüfungen mehr als ein gültiger Kreis übrigbleiben wird die Marke als nicht korrekt erkannt und es geht mit einer neuen Basis wieder bei Schritt 6.6.4.1 weiter. D.h. also, dass wenn sich z.B. zwei Marken ungefähr gleicher Größe im Bild direkt nebeneinander befinden, diese u.U. nicht erkannt werden. U.u. wäre es auch möglich bei mehreren korrekt erkannten Kreisen eine Marke zu identifizieren, was aber meist voraussetzen würde, dass die Orientierung der Marken bestimmt wird oder spezielle Regeln für einzelne Fälle eingebaut werden, um relativ exakt feststellen zu können welcher Kreis/Kugel zu welcher Marke gehört. Auf diese Prüfungen wurde bei diesem Vorgehen aus Gründen der Einfachheit bewusst verzichtet. Eine andere Möglichkeit besteht darin, die Entfernung der Kreismitte zur Mitte des Codefensters zu messen, um so den Kreis mit der kürzesten Distanz, als zur Marke gehörende Kugel zu identifizieren. Dies wird auch in den meisten Fällen funktionieren, es sind aber auch Fälle denkbar, in denen es zu einer Fehlidentifikation kommt. Eine Lösung für diese Problematik könnte darin bestehen, die Kugel so zu vergrößern, dass sie von der Diagonalen des Zylinder-Rechtecks geschnitten wird, um eine Verifikation über den Schnitt mit einem/jedem Barcodesegment zu ermöglichen. Diese Lösung würde aber sehr große Kugeln benötigen, die wiederum die Markengröße erhöhen würden, weshalb sie innerhalb dieser Arbeit nicht genutzt wird.

Gleiches wie für das Problem der eindeutigen Zuordnung von Barcodes und Kugeln gilt auch für das Überlappen von zwei Barcodes, die jeweils zu einer korrekten Marke führen, auch hier werden in dieser Arbeit beide Marken als nicht korrekt erkannt. Auch dieses Problem könnte durch Spezialregeln für einzelne Fälle gelöst werden, was aus Gründen der Einfachheit und den Erfahrungen während der Tests erstmal nicht für nötig erachtet wurde. Am Einfachsten ließe sich dieses Problem durch die Erweiterung des Barcodes um Start- Stop- und Mittelzeichen oder ein Prüfzeichen lösen, was aber wie schon erwähnt zu einer Vergrößerung der Marke oder zu einer Verringerung der maximalen Erkennungsentfernung führen würde.

Des Weiteren findet am Ende auch noch ein Test daraufhin statt, dass kein Barcode einer Marke durch die Nutzung der Kanten einer erkannten Kugel/eines Barcodes einer anderen Marke entstanden ist, denn auch dies ist natürlich theoretisch denkbar.

Zusammenfassend kann also festgehalten werden, dass bei der Anbringung der Marken und der Festlegung der Parameter der Erkennung darauf zu achten ist, dass eine Kollision von zwei Marken verhindert wird, die dazu führt, dass zwei (oder mehr) korrekte Kreise erkannt

werden oder die Barcodes sich überlappen. Dies kann z.B. erreicht werden, indem der Abstand zwischen zwei Marken maximiert wird.

Die 3 beschriebenen Schritte der Markenextraktion werden nun für jede Liste identischer Barcode-segmente durchgeführt, wobei am Ende aus den Daten die gefundenen Marken mit ID und der Position der Kugelmitte übrigbleiben.

Der Ablauf der Markenextraktion ist auch übersichtlich dem folgenden stark vereinfachten Pseudocode zu entnehmen:

TARGETEXTRACTION(B, T)

- ▷ B Datenstruktur (Map) für die gefundenen Barcodes
- ▷ T Datenstruktur (Liste) für die gefundenen Marken

```

1  forall  $c \in \text{keys}[B]$  do
2       $valid \leftarrow \text{FALSE}$ 
3      forall  $base \in B[c]$  do
4          if  $valid = \text{TRUE}$  then
5              break
6          endif
7          ▷ Berechnung von  $[\text{minXCode}, \text{maxXCode}]$ ,
8          ▷  $[\text{minYCode}, \text{maxYCode}]$  u.  $[\text{minCodeHeight}, \text{maxCodeHeight}]$ 
9           $searchWin \leftarrow \text{CalcSearchWin}(base)$ 
10          $found \leftarrow 0$ 
11          $codeWin \leftarrow \text{InitCodeWin}(base, codeWin)$ 
12         forall  $segment \in B[c] \neq base$  do
13             if  $(\text{overlaps}[searchWin, segment] = \text{TRUE})$  then
14                  $codeWin \leftarrow \text{ReCalcCodeWin}(segment, codeWin)$ 
15                  $found \leftarrow found + 1$ 
16                  $delete[segment, B[c]]$ 
17             endif
18         endforall
19         if  $(found \geq \text{MINCODESEGMENTS})$  then
20              $sphereWin \leftarrow \text{CalcSphereWin}(codeWin)$ 
21              $circleList \leftarrow \text{FindCircles}(sphereWin)$ 
22              $validCircles \leftarrow \text{TestCircles}(circleList)$ 
23         endif
24         if  $(\text{length}[validCircles] = 1)$  then
25              $valid \leftarrow \text{TRUE}$ 
26              $\text{CalcID}(id, c)$ 
27              $T \leftarrow T + \text{new Container}(codewin, validCircles, id)$ 
28         endif
29     endforall
30 endforall
31     ▷ Mehrfache Kreise, sich überlappende Barcodes und
32     ▷ Barcodes, die aus Kugeln entstanden sind aussortieren
33  $T \leftarrow \text{FinalTests}(T)$ 
34 return

```

6.7. Impresario-Macro

Wie schon erwähnt findet die softwaretechnische Umsetzung der in Abschnitt 6.6 beschriebenen Verfahren auch bei dieser Marke in Form eines Impresario-Macros mit dem Namen CylinderTargets statt. Die Eingabe findet auch hier wieder anhand eines Grauwertbildes statt, während die Ausgabe wieder in einer von der Posenbestimmung vorgegebenen Datenstruktur erfolgt.

6.7.1. Umsetzung

Zur Umsetzung werden teilweise Bibliotheksfunktionen der LTI-Lib genutzt, wobei sich dies hauptsächlich auf die Methoden zum Bildzugriff beschränkt, da sich die anderen benötigten Algorithmen effizienter selbst umsetzen lassen. Dies ist darin begründet, dass für das vorgestellte Verfahren der Kantenfilterung ausschließlich in y-Richtung erfolgen muss, was die Bibliotheksfunktionen nicht anbieten. Weiterhin wird, wie schon erwähnt, den von Matthias Wippich (siehe Kapitel 5.1.3.2) entwickelten Kreiserkennungsalgorithmus genutzt.

6.7.2. Parameter

Aus Abschnitt 6.6 ergeben sich für das Macro die in Tabelle 6.2 aufgelisteten Parameter, zu denen noch die in Tabelle 4.1 aufgeführten Parameter des Trackers hinzukommen. Welcher Barcode verwendet wird und ebenso die verwendete Methode zur Bestimmung der Balken dieses Barcodes sind nur über Präprozessordirektiven zum Zeitpunkt der Compilierung einstellbar, da dies per Macroparamter nur uneffizient umzusetzen ist.

Empfohlene Werte für die in Abschnitt 6.3 vorgestellten Beispielmarken mit 2-aus-8-Code sind in Tabelle 6.3 zu finden. Diese sind auf die verwendete Kamera (siehe Tabelle 2.1) in Verbindung mit den vorgestellten Infrarotscheinwerfer ausgerichtet. Einige der Werte sind gänzlich experimentell gefunden worden, während andere anhand von Grenzwerten geschätzt und experimentell verifiziert wurden. Als Grundlage dienten einige Testvideos, die z.B. bei Kamerafahrten mit Rollwagen oder Testroboter aufgenommen wurden, um Toleranzen für diese Situationen zu ermitteln. Diese Werte sind bewusst sehr allgemein gehalten, um einen generellen Überblick zu erhalten, es ist natürlich sinnvoll diese Werte an die Extremsituationen des jeweiligen Einsatzszenario bezüglich maximal möglicher Neigung u.ä. anzupassen. Für andere Markengrößen und Codes sind dementsprechend u.U. auch abgewandelte Werte zu nutzen.

6.8. Verfolgung

Wie schon bei der Doppelverhältnismarke kommt auch bei dieser Marke zur Erhöhung der Erkennungsgeschwindigkeit der in Kapitel 4 vorgestellte Tracker zum Einsatz. Als zu verfol-

Parameter	Bedeutung
<i>stepSize</i>	Schrittweite des Scanlinienverfahrens.
<i>derivThreshold</i>	Schwellwert für die Ableitungen mit Hilfe der Sobelmaske in y-Richtung.
<i>incLikelihoodPx</i>	Inkrement für die Suche bei Nutzung des Maximum-Likelihood-Verfahrens in Pixel.
<i>narrowMin</i> u. <i>narrowMax</i>	Intervallgrenzen für die schmalen Balken in Abhängigkeit des geschätzten Werts.
<i>wideMin</i> u. <i>wideMax</i>	Intervallgrenzen für die breiten Balken in Abhängigkeit des geschätzten Werts für die schmalen Balken.
<i>spaceMin</i> u. <i>spaceMax</i>	Intervallgrenzen für die Zwischenräume in Abhängigkeit des geschätzten Werts für die schmalen Balken.
<i>minCodeSegments</i>	Mindestanzahl der gefundenen Barcodesegmente.
<i>codeSizeTolerance</i>	Toleranz für die Größe der Barcodesegmente in Abhängigkeit zur Basis der Codefensterbestimmung.
<i>codeYTolerance</i>	Toleranz für die y-Position der Barcodesegmente in Abhängigkeit zur Basis der Codefensterbestimmung.
<i>codeRatio</i>	Verhältnis aus der Breite der Marke zur Höhe der Marke (des Barcodes) inkl. Toleranzen.
<i>codeSphereRatioX</i>	Verhältnis aus Höhe des Barcodes und dem Durchmesser der Kugel inkl. Toleranzen.
<i>codeSphereRatioY</i>	Verhältnis der Höhe des Barcodes und dem Abstand zwischen Barcode und der Unterkante der Kugel inkl. Toleranzen.
<i>sphereCenterMaxY</i>	Maximale y-Position (von der unteren Kante des Codefensters gesehen) der Kugelmitte in Abhängigkeit zur Höhe des Barcodes.
<i>sphereYTolerance</i>	Toleranz zur Ausdehnung des Kugelfensters in das Codefenster in Abhängigkeit zur Höhe des Barcodes.
<i>minFormFactor</i> u. <i>maxFormFactor</i>	Intervallgrenzen für den Formfaktor der gefundenen Kreise.
<i>sphereDiaMin</i> u. <i>sphereDiaMax</i>	Intervallgrenzen für den Durchmesser der gefundenen Kreise in Abhängigkeit zur Höhe des Barcodes.
<i>sphereDiaMinPx</i>	Minimaler Durchmesser der Kugel in Pixel.
<i>reflectiveSheetGrayScaleValueMin</i>	Parameter für die Kreiserkennung von Matthias Wippich. (siehe Tab. 5.2)
<i>edgeGoodRatio</i>	Parameter für die Kreiserkennung von Matthias Wippich. (siehe Tab. 5.2)
<i>edgeDifferenceMin</i>	Parameter für die Kreiserkennung von Matthias Wippich. (siehe Tab. 5.2)

Tabelle 6.2. – Parameter des Macros CylinderTargets

Parameter	Wert	Begründung
<i>stepSize</i>	1	Erhöhung für schnellere Verarbeitungsgeschwindigkeit möglich, wobei dann auch eine Anpassung von <i>minCodeSegments</i> sinnvoll ist.
<i>derivThreshold</i>	80	Hat sich im Bezug auf eine möglichst große Erkennungsentfernung als guter Wert erwiesen.
<i>incLikelihoodPx</i>	0,1	Experimentell ermittelt.
<i>narrowMin</i> u. <i>narrowMax</i>	0,76-1,45	Experimentell ermittelte Extremwerte für narrowAvg.
<i>wideMin</i> u. <i>wideMax</i>	1,46-2,30	Experimentell ermittelte Extremwerte für narrowAvg.
<i>spaceMin</i> u. <i>spaceMax</i>	0,50-1,65	Experimentell ermittelte Extremwerte für narrowAvg.
<i>minCodeSegments</i>	3	Scheint auch auf große Entfernung ein guter Wert zu sein.
<i>codeSizeTolerance</i>	0,2	Experimentell ermittelt.
<i>codeYTolerance</i>	0,3	Geschätzt über die Dreiecksbeziehungen bei einer maximal rotierten Marken mit der Eigenschaft die unter <i>codeRatio</i> beschrieben ist.
<i>codeRatio</i>	0,6	Ausgangswert war eine um $\beta = 50^\circ$ geneigte Marke (siehe Abb. 6.12) unter Parallelprojektion zu der Toleranzen hinzugezählt wurden.
<i>codeSphereRatioX</i>	1,5	Geschätzt auf Basis der Werte von <i>codeRatio</i> und dem Zylinderdurchmesser als Kugeldurchmesser inkl. Toleranzen bei Rotation.
<i>codeSphereRatioY</i>	0,7	Geschätzt auf Basis der Werte von <i>codeRatio</i> .
<i>sphereCenterMaxY</i>	0,5	Geschätzt aus den Extremwerten für die Kugelgröße/-position und die Barcodehöhe.
<i>sphereYTolerance</i>	0,15	Geschätzt über Dreiecksbeziehungen einer maximal gedrehten Marke mit der Eigenschaft die unter <i>codeRatio</i> beschrieben ist.
<i>minFormFactor</i> u. <i>maxFormFactor</i>	1,0-1,3	Experimentell ermittelte Grenzwerte.
<i>sphereDiaMin</i> u. <i>sphereDiaMax</i>	0,15-0,4	Experimentell und geschätzt auf Basis der Werte von <i>codeRatio</i> und dem Kugeldurchmesser.
<i>sphereDiaMinPx</i>	5	-
<i>reflectiveGraySheet – ScaleValueMin</i>	50	Experimentell ermittelt.
<i>edgeGoodRatio</i>	0,8	Experimentell ermittelt.
<i>edgeDifferenceMin</i>	40	In [Wippich, 2008a] empfohlener Wert.

Tabelle 6.3. – Beispielwerte für die in Kapitel 6.3 vorgestellten Beispielmarken mit 2-aus-8 Barcode.

gende Punkte der Marke dient nicht der Messpunkt der Kugel, da sich über dessen Position und Größe kein direkter Zusammenhang mit der Position des zu erkennenden Barcode ergibt. Somit müssten noch weitere Umrechnungen erfolgen, weshalb zur Verfolgung schon verfügbare Informationen genutzt werden. Dies sind zwei Punkte, die ein Rechteck um den Barcode aufspannen. Diese beiden Punkte ergeben sich aus der minimalen x-Koordinate des Kugelfensters und der minimalen y-Koordinate des Codefensters einerseits und der maximalen x-Koordinate des Kugelfensters und der maximalen y-Koordinate des Codefensters andererseits. Problematisch hierbei ist, dass diese Werte je nach Erkennung der Marke sehr stark schwanken können und es sich auch um keine echten Messwerte handelt, sondern um aus anderen Messungen und Schätzungen berechnete Werte. Dies führte aber bei den durchgeführten Tests zu keinen negativen Auswirkungen auf die Erkennung auf Basis der Vorhersagen des Trackers, was sicherlich auch auf die großzügig gewählten Toleranzen für die Suchfenster zurückzuführen ist. Zur Optimierung könnten hier u.U. noch die Werte für das Messrauschen des Kalman Filter angepasst werden. Eine direkte Nutzung der Ecken des Codefensters hat sich als wenig praktikabel herausgestellt, da dieses bedingt durch Störungen bei der Erkennung teilweise sehr viel kleiner wird als der eigentliche Barcode und somit auch die vorhergesagten Werte des Trackers viel zu kleine Bereiche für die Suche ergeben, bzw. diese u.U. durch die erkannte Geschwindigkeit immer kleiner werden. Für andere Markenkongfigurationen mit breiteren Zylindern (die dann z.B. breiter als das Kugelfenster sind. . .) müssen ggf. andere Lösungen gefunden werden.

Weiterhin steht natürlich auch bei dieser Marke die in Abschnitt 5.3.2 beschriebene Möglichkeit der Zerteilung des Bildes durch eine horizontale Trennlinie (*cut*), um die Geschwindigkeit weiter zu erhöhen, zur Verfügung.

6.9. Tests

Zu der Zylindermarke wurden einige Tests durchgeführt, die manche der in diesem Kapitel beschriebenen Eigenschaften und die generelle Funktionsfähigkeit dieses Ansatzes zeigen sollen.

Die Tests wurden, wenn nicht anders angegeben, mit Marken, die den Ausmaßen der in Abbildung 6.1 abgebildeten Marke durchgeführt. Als Software kam das beschriebene Impresario-Macro mit den in Tabelle 6.3 angegebenen Beispielwerten zum Einsatz. Die genutzte Kamera ist in Tabelle 2.1 beschrieben und die nötige Beleuchtung wurde durch die in Kapitel 2 vorgestellten Infrarotscheinwerfer (inkl. Infrarotfilter für die Kamera) hergestellt. Als Einstellung für die Kamera diente ein Gain (Helligkeits- und Kontrastanpassung) von 460 und eine Belichtungszeit von $\frac{1}{120}s$ bei einer Blendenöffnung von $f/2$ und gleichbleibender Einstellung des Fokusrings. Die (roten) Kreuze zur Visualisierung des Mittelpunktes der Kugel auf den folgenden Bildern entsprechen nicht den realen gemessenen Werten mit Subpixelgenauigkeit. Zur Auswertung wurde der in Tabelle 2.3 angegebene PC genutzt. Als Schätzverfahren für den Ba-

siswert der Balkenklassifikation wird das narrowAvg-Verfahren genutzt, wenn nichts anderes angegeben ist.

6.9.1. Barcodes

In diesem Abschnitt sind für jeden genutzten Code ein Beispielbild aus 3 m Entfernung und eines aus der ungefähren maximalen Erkennungsentfernung, bei der die Marke ohne oder mit nur geringer Drehung (siehe 6.5.1) noch robust zu erkennen ist. Die Bilder sind jeweils mit direkt auf die Marken gerichtetem Infrarotscheinwerfer und mit nur geringer Neigung/Verkipfung (ca. 5°) zwischen Bildebene und der Wand, aufgenommen worden. Als Ergebnis kann festgehalten werden, dass mit den unter 6.4 angegebenen Maßen die maximale Erkennungsentfernung bei der Marke mit 2-aus-8 Code am höchsten ist und die beiden anderen Codes ungefähr gleichauf liegen.

6.9.1.1. 2-aus-8 Code

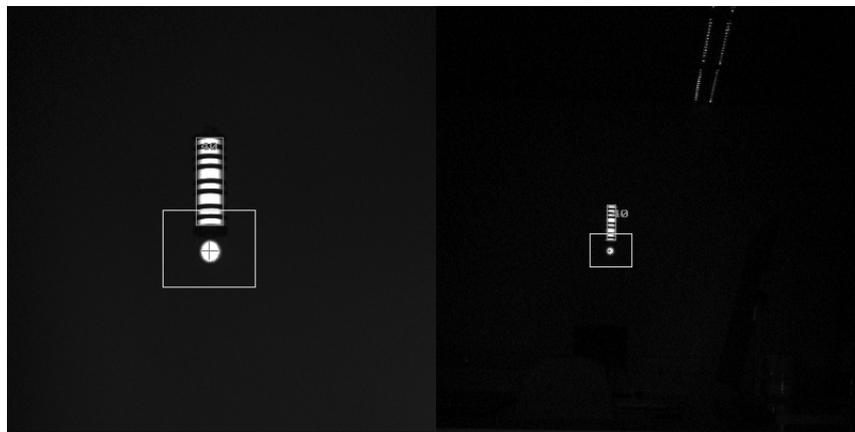


Abbildung 6.23. – Links: Marke mit 2-aus-8 Code in 3 m Entfernung. Rechts: Ungefähre maximale Erkennungsentfernung ca. 9,50 m. In Kapitel 6.9.9 konnte diese Marke sogar aus über 10 m noch sicher erkannt werden, was wohl an der etwas schwachen Beleuchtung und damit geringerer Überstrahlung lag. Bei der Beleuchtung gibt es also noch genügend Spielraum für Veränderungen.

6.9.1.2. 2-aus-10 Code

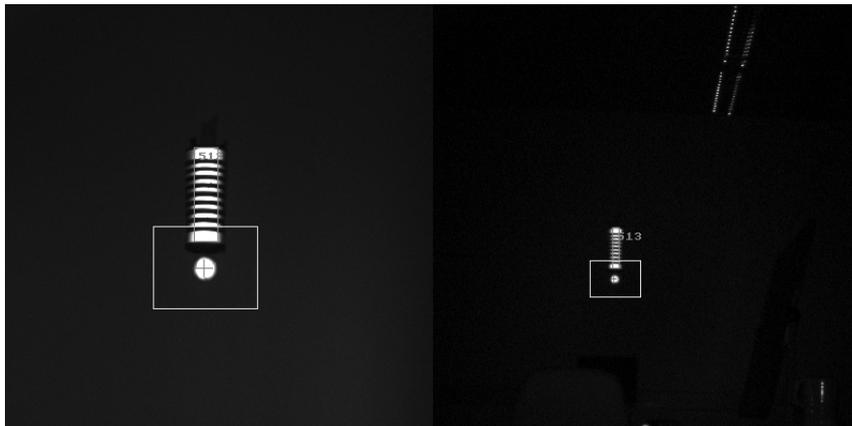


Abbildung 6.24. – Links: Marke mit 2-aus-10 Code in 3 m Entfernung. Rechts: Ungefähre maximale Erkennungsentfernung ca. 7,90m

6.9.1.3. 2-aus-5 Code

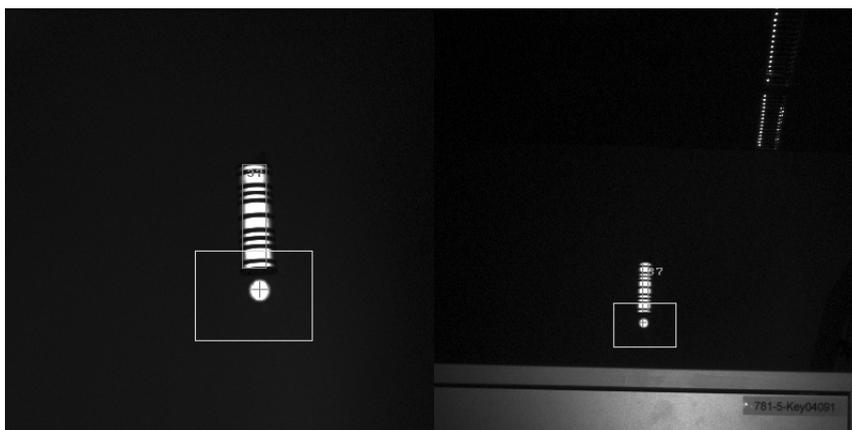


Abbildung 6.25. – Links: Marke mit 2-aus-5 Code in 3 m Entfernung. Rechts: Ungefähre maximale Erkennungsentfernung ca. 7,60m

6.9.2. Gedrehte Abbildung des Zylinders

In Abbildung 6.26 ist eine aus 3 m Entfernung aufgenommene Marke mit 2-aus-8 Code zu sehen, die um den maximal möglichen Drehwinkel (siehe Kapitel 6.5.1), rotiert im Bild abgebildet

wird, bei dem noch eine Erkennung möglich ist. Das Bild ist ebenfalls unter optimaler Beleuchtung und mit nur geringem Winkel/Verkipfung (ca. 5°) zwischen Bildebene und der Wand, an der die Marke befestigt war, aufgenommen worden. Bei der Messung in einem Bildbearbeitungsprogramm wurde ein Winkel Rotationswinkel von ca. 20° ermittelt, was ungefähr dem Erwartungswert entspricht, wenn man bedenkt, dass die zu erkennende Codefläche mindestens 3 Pixel breit sein muss und durch die Nähe zur Marke auch die Krümmung des Zylinders eine geringe Rolle spielt. Abbildung 6.27 zeigt eine Marke, die im Bild rotiert erscheint, weil die Bildfläche und der Längsschnitt des Zylinders der 4 m entfernten Marke nicht parallel waren. Der Winkel der Kameraneigung³⁵ betrug ca. 40° . Gleichzeitig ist aber auch die Kamera zum Ausgleich etwas nach rechts gekippt.

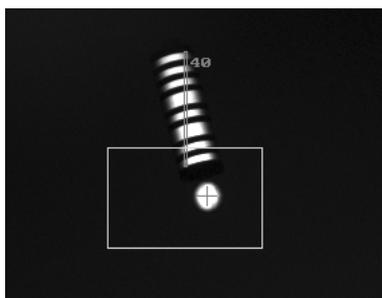


Abbildung 6.26. – Marke mit 2-aus-8 Code in 3 m Entfernung mit maximaler Rotation der Codefläche.

6.9.3. Neigung der Kamera

Abbildung 6.28 zeigt die in Abbildung 6.12 illustrierte Situation der Neigung der Kamera, um den Winkel $\beta = 50^\circ$ und dem Fuss des Zylinders ungefähr auf der optischen Achse, aus einer Entfernung von 2 m aufgenommen und der damit einhergehenden perspektivischen Verkürzung des Zylinders. Es ist zu sehen, dass die Annahmen, die für die Wahl der Parameter des Erkennungsalgorithmus getroffen wurde soweit korrekt sind, da die Kugel vollständig im Kugelfenster abgebildet wird. Des Weiteren ist zu sehen, dass die durch die perspektivische Verzerrung hervorgerufenen unterschiedlichen Balkenbreiten (siehe u.a. 6.5.2) in der Praxis kaum eine Rolle spielen, da es wie erwartet nur einen geringen Unterschied zwischen dem obersten (ca. 5 Pixel) und dem untersten schmalen (ca. 6 Pixel) Balken des Barcodes gibt.

6.9.4. Klassifikation der Barcodebalken

In Abbildung 6.29 ist ein Vergleich der beiden in Abschnitt 6.6.3 Verfahren zur Bestimmung einer Basis-Balkenbreite des Barcodes, anhand derer die Klassifikation der gemessenen Bal-

³⁵Winkel zwischen optischer Achse und einer senkrechten Geraden zur Ebene des Zylinderlängsschnitts.

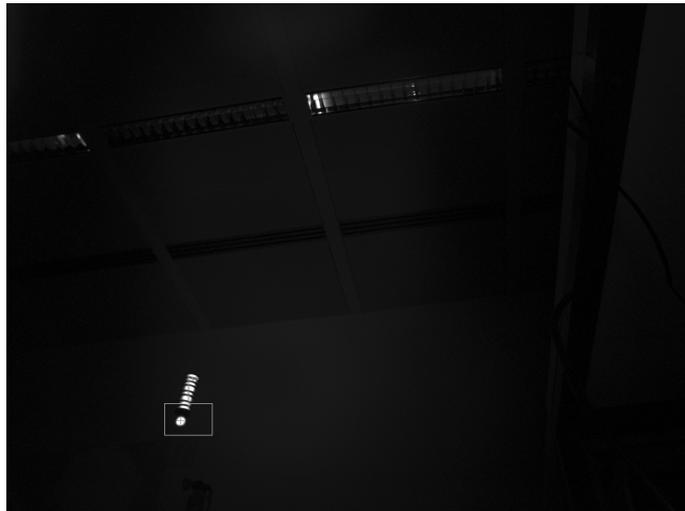


Abbildung 6.27. – Marke mit 2-aus-8 Code in 4 m Entfernung und maximaler Rotation der Codefläche durch Neigung der Kamera um ca. 40° und zum Ausgleich etwas nach rechts gekippter Kamera.

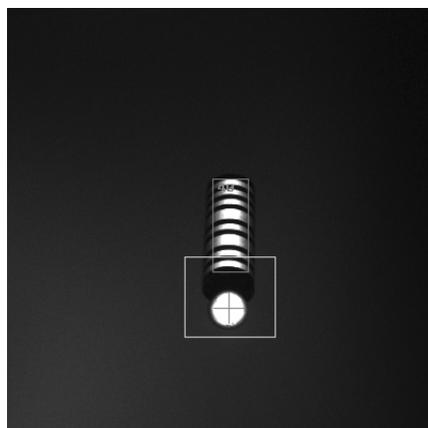


Abbildung 6.28. – Marke mit 2-aus-8 Code, deren Codefläche ungefähr auf der optischen Achse steht, an 2 m entfernter Wand mit Neigung der Kamera um $\beta = 50^\circ$ (zum Versuchsaufbau siehe auch Abb. 6.12).

ken in breit und schmal geschehen kann. Zu diesem Zwecke wurde eine Messung an einem Testvideo durchgeführt, wobei die Kamera auf einem Rollwagen befestigt aus einer weiten Entfernung (ca. 12 m) startend bis zur Position direkt unterhalb der an der Decke befindlichen Marke mit 2-aus-8 Barcode zufährt. Also ein ganz normales Anwendungsszenario der Marke, wobei auf eine möglichst gute Ausleuchtung und Fokussierung (leichte Unschärfe ist aber für die subpixelgenauen Messungen hilfreich) geachtet wurde. Es wurden jeweils einzelne erkannte Barcode-segmente betrachtet und pro Segment der Mittelwert der gemessenen Balkenbreite jeweils getrennt nach Typ (schmal,breit) berechnet. außerdem ist in Abbildung 6.29 der durch die beiden Verfahren (Maximum-Likelihood mit einem Inkrement von 0,1) bestimmte Basiswert für die Klassifikation der Balken mit abgebildet, um einen Vergleich ziehen zu können. Es ist zu sehen, dass das Maximum-Likelihood-Verfahren ungefähr den gemessenen Mittelwerten für schmale Balken (narrowAvg) der jeweiligen Codesegmente entspricht. Mit beiden Verfahren wurde in diesem Testvideo die gleiche Anzahl an Barcode-segmenten erkannt (insgesamt 2303) und Marken (insgesamt 186 bestehend aus 2298 Segmenten), was zeigt dass es hier mit dem einfacheren narrowAvg-Verfahren (sehr wahrscheinlich³⁶) zu keinen Einbußen kommt. In der Abbildung nicht sofort zu sehen, aber durch Messungen festgestellt, ergibt sich insgesamt eine vergrößerte Abbildung der schmalen Balken gegenüber den breiten Balken. Die genauen Gründe und Faktoren der Vergrößerung im Einzelnen müssten noch durch weitere und genauere Auswertungen erfolgen. In Frage kommen hierfür u.a. die perspektivische Verzerrung und die Überstrahlung. Im Durchschnitt über alle Segmente ergibt sich hierbei ein Faktor von ca. 1,86 (dieser Wert schwankte im Vergleich von Testvideos zwischen 1,80 und 1,95) statt dem eigentlich erwarteten Faktor 2,0. Dies ist einerseits durch mangelnde Genauigkeit und die perspektivische Verzerrung zu erklären, hat aber scheinbar andererseits auch mit den Reflexionseigenschaften der Reflexfolie zu tun. Es ist sinnvoll dies noch genauer zu untersuchen, um z.B. gegebenenfalls den Faktor bei dem Maximum-Likelihood-Schätzverfahren anzupassen.

Um alle gefundenen und zu einer Marke gehörenden Codesegmente korrekt klassifizieren zu können, waren für den mittels Maximum-Likelihood bestimmten Wert Toleranzfaktorintervalle von $[0,82;1,44]$ für die schmalen Balken und $[1,60;2,13]$ für die breiten Balken notwendig. Für das andere einfachere Verfahren (narrowAvg) waren Toleranzfaktorintervalle von $[0,79;1,32]$ für die schmalen Balken und $[1,57;2,08]$ für die breiten Balken nötig. Die Intervalle der beiden Verfahren unterscheiden sich also kaum. Die Intervalle für den 2-aus-8 Barcode könnte man z.B. noch dadurch verkleinern, indem man sich entschließt Codesegmente, bei denen es extreme Ausreißer einzelner Balken gibt, nicht mehr zu berücksichtigen.

6.9.5. Beleuchtung der Kugel

Die gute Beleuchtung einer Kugel stellt besonders im Nahbereich eine Herausforderung dar. Hier können leicht kleine Schatten entstehen, die die Form der Kugel im Bild im Extremfall so verfälschen können, dass die Marke entweder nicht erkannt wird oder der gemessene

³⁶Es wurde nur die Anzahl von Barcode-segmenten gezählt und nicht deren Identität festgestellt.

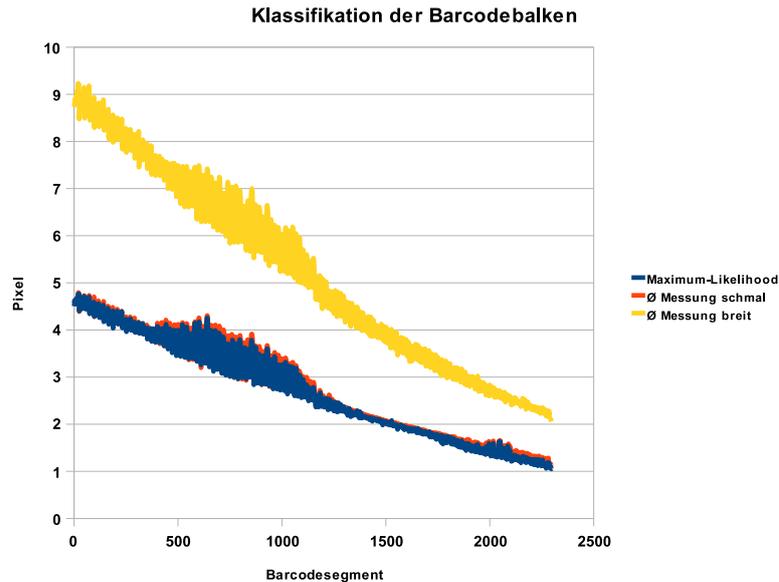


Abbildung 6.29. – Vergleich zwischen den Verfahren zur Ermittlung eines Basiswerts zur Klassifikation der Balken des Barcodes.

Kugelmittelpunkt nicht dem tatsächlichen entspricht. Hierbei kommt es aufgrund von großen Beobachtungswinkeln und der damit einhergehenden geringeren Überlappung der Bereiche, die die Strahlen des Scheinwerfers und die Abbildungsstrahlen der Kamera erreichen, dazu, dass sich im vom Scheinwerfer abgewandten Bereich der Kugel ein leichter Schatten bildet, der im Extremfall im Bild zu sehen ist. Befinden sich Kamera und Scheinwerfer weiter von der Kugel entfernt spielt dieses Problem eine geringere Rolle, weil die Beobachtungswinkel kleiner sind. Abbildung 6.30 zeigt ein Extrembeispiel, bei dem sich die Lichtquelle 25 cm links neben der Kamera (optische Achse) auf gleicher Höhe befand. In diesem Fall ist bei der ca. 2 m entfernten Kugel im rechten Bereich ein kleiner Schatten zu erkennen. Gleichzeitig findet hier durch den Verzicht auf einen Infrarotfilter auf der Schattenfläche noch eine Reflexion einer anderen Lichtquelle statt, was den Schatten noch mehr hervorhebt. In anderen Tests, bei denen die Lichtquelle maximal ca. 10 cm von Kamera entfernt war, waren die Schatten durch die kleineren Beobachtungswinkel auch auf dieser kurze Distanz nicht mehr sichtbar, bzw. nicht von einem u.a. durch den maximalen Einfallswinkel der Folie verursachten, annähernd gleichmäßig um die Kugel verteilten, nur gering reflektierenden Rand zu unterscheiden, womit sich auch die Abweichung des Messpunktes in einem tolerablen Bereich³⁷ bewegen sollte. Als bessere Lösung ist aber dennoch zu empfehlen die Kugel von an zwei gegenüberliegenden Seiten der Kamera befindlichen Scheinwerfern beleuchten zu lassen, um eine einseitige Schattenbildung im Nahbereich zu verhindern, da hier der sichtbare Schatten der einen Lichtquelle durch die andere ausgeleuchtet wird. Eine noch gleichmäßigere und großflächigere Beleuchtung kann durch ein Ringlicht erreicht werden. Gleiches gilt auch für den Zylinder, wobei sich hier keine

³⁷Genauereres müsste noch untersucht werden.

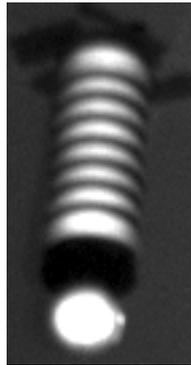


Abbildung 6.30. – Extrembeispiel für die Schattenbildung durch einseitige Beleuchtung (hier von links) der Kugel.

wirklichen Probleme ergeben, da jede Scanlinie bezüglich der Kanten und Grauwerte getrennt behandelt wird. Diese Problematik sollte aber auf jeden Fall vor einem produktiven Einsatz ohne symmetrische Beleuchtung weiter untersucht werden.

6.9.6. Identifikation eines anderen Elements im Bild als Marke

Um die Identifikation eines nicht durch eine Marke ausgelösten Musters im Bild als Marke auszuschließen, muss die Marke in der abgebildeten Umgebung einzigartig sein. Dies ist durch den Einsatz von Infrarotfilter und Infrarotbeleuchtung schon sehr gut gelungen, da nun nur noch stark reflektierende Objekte überhaupt als Quelle dieses Problems in Frage kommen. In Abbildung 6.31 ist der einzige Ansatz einer Identifikation eines anderen Elements im Bild als Marke, der während der Tests zu dieser Arbeit aufgetreten ist zu sehen. Hier wurden die durch die Infrarotscheinwerfer angestrahlten Lamellen der Deckenbeleuchtung als 2-aus-8 Barcode interpretiert, aber keine entsprechend passende Kugel gefunden. Selbst die Identifikation des Barcodes konnte durch die aufgrund der Testvideos angepassten Parameter für die Balkenerkennung verhindert werden. Dennoch ist natürlich eine Identifikation von anderen Elementen als Marke in der Praxis nicht gänzlich auszuschließen, da diese Lamellen, Lüftungsgitter oder andere Objekte, die mehrfach untereinander durch Kanten in y-Richtung abgebildet werden, potentiell die Gefahr bergen als Barcode erkannt zu werden. Mit Pech ist dann auch noch eine passende kreisförmige Reflexion im Kugelfenster abgebildet, was dann zu einer Identifikation dieses Musters als Marke führen würde. Während der Tests kam dies aber bisher nie vor und die Gefahr kann auch durch die Verwendung von Barcodes mit noch mehr Balken u.ä. weiter verringert werden.

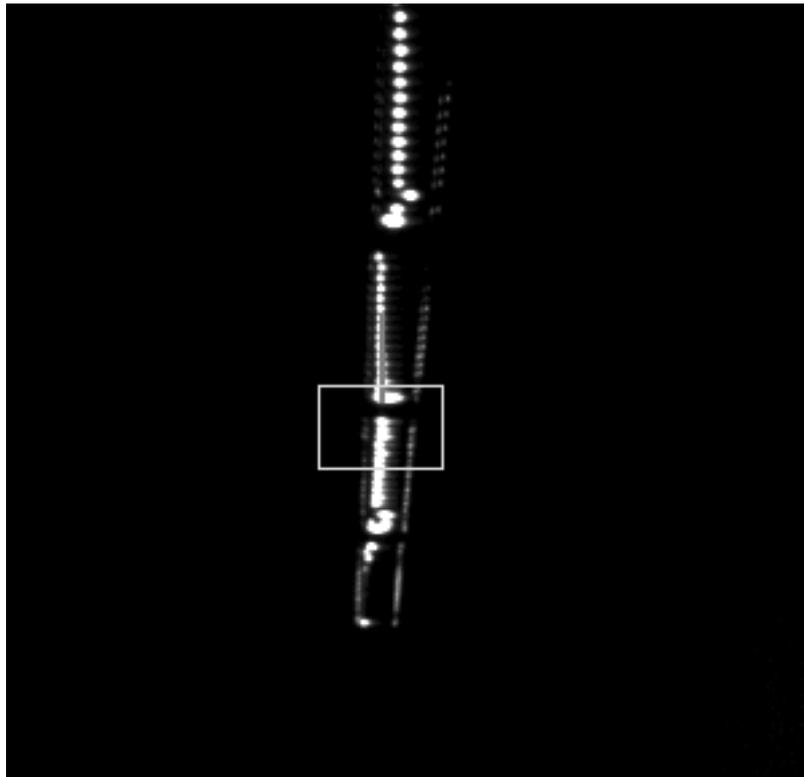


Abbildung 6.31. – Identifikation von Lamellen der Deckenbeleuchtung als korrekten 2-aus-8 Barcode.

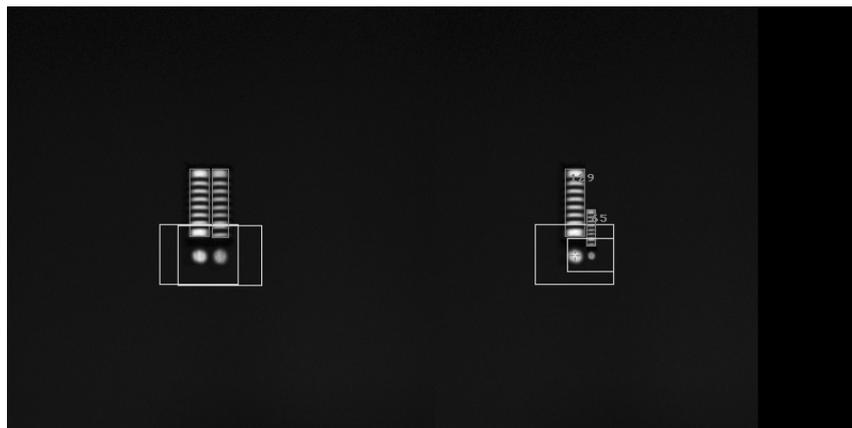


Abbildung 6.32. – Links: Marken werden nicht identifiziert, da sich jeweils zwei passende Kugeln im Kugelfenster befinden. Rechts: Kugeln passen aufgrund der Größe nur zu jeweils einem Barcode, weshalb hier beide Marken erkannt werden.

6.9.7. Fehlidentifikationen

Bei der Wahl der Regeln innerhalb des Erkennungsalgorithmus und der dazugehörigen Werte für die Parameter wurde, wie schon beschrieben, darauf geachtet, dass es möglichst zu keiner Fehlidentifikationen von Marken kommt. Besonders kritisch hierbei sind nahe beieinanderliegende oder sich überlappende³⁸ Marken. Wie schon erwähnt, wird bei Situationen, in denen eine Fehlidentifikation nicht sehr einfach ausgeschlossen werden kann (2 passende Kugeln, 2 Barcodes eine passende Kugel, . . .), die Marke als nicht erkannt gewertet (siehe hierzu auch Kapitel 6.6.4). Einige Beispiele sind in den Abbildungen 6.32 bis 6.35³⁹ zu sehen, die einen Teil dieser speziellen Situationen zeigen, deren Problematik meist auch durch spezielle Regeln nicht generell gelöst werden kann. Vorkommen sollten diese Situationen aber schon aufgrund von aufmerksamer Platzierung der Marken nur selten.

Trotz der genannten Vorsichtsmaßnahmen sind Fehlidentifikationen einer Marke aber auf keinen Fall ausgeschlossen, da es durch die besondere Form und Sichtbarkeit der Marke u.a. keine Begrenzungsbalken wie z.B. bei der Doppelverhältnismarke gibt, sodass die Zugehörigkeit einer Kugel zu einer Marke immer einer höheren Unsicherheit unterliegt, als dies beispielsweise bei den Kreisen der Doppelverhältnismarke der Fall ist.

³⁸Zwei Barcodes liegen so übereinander, dass aus diesen zwei ein anderer Barcode entsteht.

³⁹Die Bilder, sind mit einem Bildbearbeitungsprogramm passend zusammengeschnitten worden.

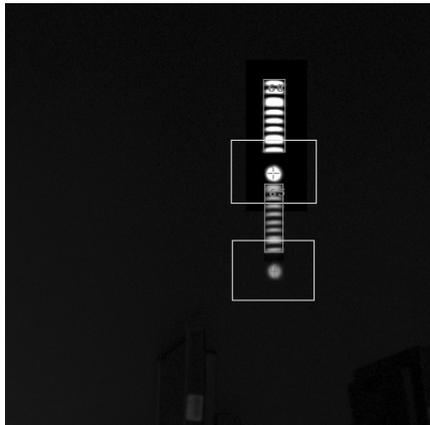


Abbildung 6.33. – Beide Marken werden erkannt, weil die Balken der Barcodes der beiden Marken zu weit voneinander entfernt sind und auch die Kugel der oberen Marke als Balken des Barcodes der unteren Marke ausscheidet, da dies überprüft wird.

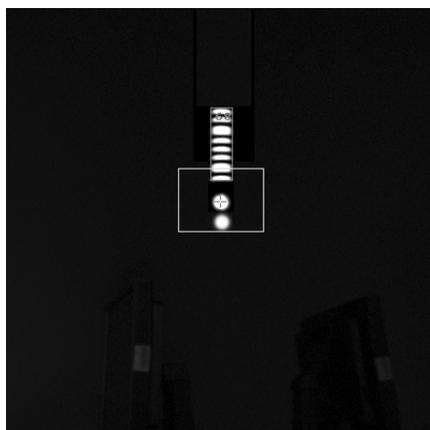


Abbildung 6.34. – Die Marke wird erkannt, obwohl sich zwei gleichgroße Kugeln im Kugelfenster befinden. Die untere Kugel kommt aber nicht in Frage, da ihr Mittelpunkt zu weit vom untersten Barcode-Balken entfernt ist (siehe Parameter *sphereCenterMaxY*).

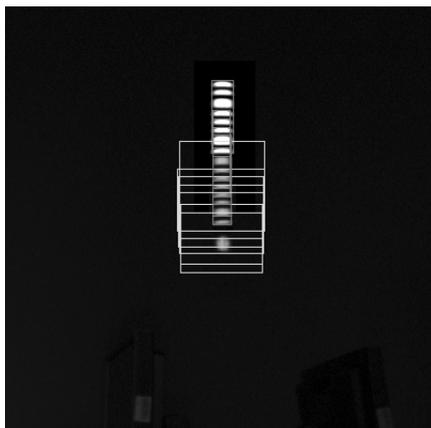


Abbildung 6.35. – Die Marke wird nicht erkannt, da sich hier drei Marken von oben nach unten überlappen und zwei korrekte Barcodes entstehen, die jeweils die gleiche Kugel finden.

6.9.8. Verteilung der Laufzeit

Dieser Abschnitt zeigt anhand von 3 Beispielen die Verteilung der Gesamtlaufzeit auf die einzelnen Verarbeitungsschritte der Erkennung. Zu diesem Zwecke wurden die Einzelbilder⁴⁰ ca. 100 mal untersucht und daraus die durchschnittliche prozentuale Verteilung errechnet. Die Angaben zur Laufzeit sind ebenfalls Durchschnittswerte und ohne die Erzeugung eines Ausgabebildes des Macros ermittelt worden.

In Abbildung 6.37 ist die Verteilung für Abbildung 6.36 mit den Parametern aus Tabelle 6.3 zu sehen. Als Schätzverfahren für den Basiswert zur Klassifikation der Balken des Barcodes wurde das einfache narrowAvg-Verfahren genutzt, wobei sich für das Maximum Likelihood-Verfahren nur unwesentlich andere Ergebnisse ergeben und die Anzahl der gültigen Barcode-segmente und die der Marken exakt gleich sind. Diese Unterschiede waren aber so klein, dass die Ergebnisse des Maximum Likelihood-Verfahrens hier nicht extra aufgeführt werden. Die 3 Marken sind bei einer Bildgröße von 1280 x 960 Pixel jeweils ca. 9 Pixel breit und ca. 26 Pixel hoch. Der Kreisdurchmesser beträgt ca. 5 Pixel. Es ist zu sehen, dass über 90% der Gesamtlaufzeit von ca. 18 ms für die Kantenfilterung verbraucht werden, wobei die gesamte Filterung des Bildes mit dem Sobel-Operator am längsten dauert. Die Zeit, die Non-Maxima-Suppression und die subpixelgenaue Bestimmung der Kantenposition⁴¹ verbrauchen ist so gering, da nur ein kleiner Teil der Pixel überhaupt durch den Schwellwertprüfung gelangt. Die Filterung des Barcodes, die Ermittlung der Fenster und auch die eigentliche Erkennung der Kugel sind hier zu vernachlässigen. Sonstige Dinge wie die Verwaltung der Datenstrukturen u.ä. haben ebenso nur einen geringen Einfluss.

⁴⁰ Auch diese Bilder sind mit einem Bildbearbeitungsprogramm passend zusammengeschnitten worden.

⁴¹ Diese beiden Schritte sind zusammengefasst, weil die Zeit, die die Kantenbestimmung selbst einnimmt so gering ist. Bei allen Tests war dieser Wert nie größer als 0,2%

Abbildung 6.38 zeigt die Verteilung für das selbe Bild unter Nutzung des Trackers mit den Parametern aus Tabelle 4.1. Für das Schätzverfahren gilt das gleiche, wie bei Abbildung 6.37. Die Laufzeit sinkt hierdurch auf ca. 6 ms. Die Verteilung verändert sich dahingehend, dass der Sobel-Operator wieder die meiste Zeit in Anspruch nimmt und Non-Maxima-Suppression mehr an Gewicht gewinnt, da in dem kleinen Trackerfenster eine höhere Dichte an potentiellen Kantenpunkten besteht, die oberhalb des Schwellwerts liegen. Der Tracker mit Kalman Filter benötigt hier mit 10% schon relativ viel Zeit, wenn man bedenkt, dass nur 6 Punkte verfolgt werden. Ein Vergleich, mit dem einfachen Algorithmus ergab, dass bei einer etwas geringeren Gesamtlaufzeit (<1 ms) dort nur ca. 1% der Zeit auf den Tracker entfielen.

In Abbildung 6.39 ist fast das komplette Bild (1280 x 960 Pixel) mit Zylindermarken ausgefüllt. Hierzu waren 4 Marken (jeweils ca. 150 Pixel breit und ca. 400 Pixel hoch; Kreisdurchmesser ca. 70 Pixel) notwendig, deren Erkennung mit dem narrowAvg-Verfahren zur Schätzung des Basiswerts ca. 44 ms dauerte (siehe Abb. 6.40 links). Hierbei zeigt sich, dass bei dieser großen Anzahl von möglichen Kantenpunkten Non-Maxima-Suppression und subpixelgenaue Kantenbestimmung die meiste Zeit verbrauchen. Auch die Werte für Barcodefilterung und Kugelerkennung sind prozentual klar höher. Die Erhöhung bei der Barcodefilterung hängt nicht mit der Höhe der Marken, sondern hauptsächlich mit der Breite dieser zusammen. Der Mehrbedarf bei der Kugelerkennung ist durch die Größe der Kugeln zu erklären. Hierbei hilft auch nicht die adaptive Bestimmung der Schrittweite, da die meiste Zeit für die Kantenbestimmung verbraucht wird, bei der mehr Pixel untersucht werden müssen, je größer die Kugel im Bild ist. Verstärkt wird dies hier zusätzlich noch dadurch, dass manche Kugeln in andere Kugelfenster ragen, so dass auch hierdurch weitere Rechenzeit verbraucht wird. Die Zeit für Sobel-Operator und Thresholding ist erwartungsgemäß konstant gegenüber der Erkennung von Abbildung 6.36. Für das gleiche Testbild sind in Abbildung 6.40 auch die Werte der Erkennung mit dem Maximum-Likelihood-Verfahren zur Schätzung des Basiswerts zu finden. Hierbei ergibt sich eine Laufzeit von ca. 53 ms, was eine Steigerung von fast 10 ms gegenüber dem anderen Verfahren (narrowAvg) darstellt. Für die Erhöhung der Laufzeit ist die Barcodefilterung verantwortlich, was nicht überrascht.

Als Fazit kann festgehalten werden, dass die beiden ersten Schritte der Erkennung, Sobel-Operator und Thresholding, zeitlich konstant sind, weil diese immer auf das komplette Bild angewendet werden müssen⁴². Bei den anderen Schritten hängt es stark von der Anzahl und Größe der verwendeten Marken ab. Der Anteil an der Gesamtlaufzeit der Non-Maxima-Suppression steigt an, wenn die Markenfläche und damit die Anzahl von möglichen Kantenpunkten steigt. Dies muss nicht unbedingt mit sehr großen Marken zusammenhängen, sondern kann auch durch sehr viele kleine Marken, mit einer ähnlichen Gesamtfläche an zum Barcode gehörenden Balken, ausgelöst werden.

Die Zeit für die Barcodefilterung ist bei der Schätzung mit dem narrowAvg-Verfahren nicht abhängig von der Höhe der Marken, sondern von deren Breite, also der Anzahl von Barco-

⁴²Gilt natürlich nur, wenn *stepSize* = 1 ist.

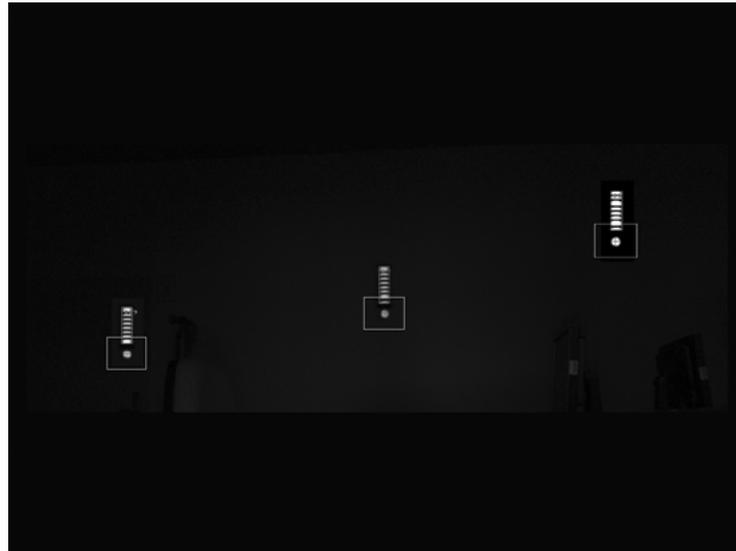


Abbildung 6.36. – Testbild für Laufzeittest mit 3 Zylindermarken.

desegmenten. D.h. eine sehr große Anzahl von kleinen Marken im Bild benötigt bei der Barcodefilterung u.U. mehr Zeit, als einige wenige sehr große Marken. Bei der Schätzung mit dem Maximum-Likelihood-Verfahren ist die Laufzeit auch von der Höhe der Balken abhängig, da hier mit einem konstanten Inkrement von 0,1 gearbeitet wurde. Der Laufzeitunterschied ist aber bei dem Testbild mit den kleinen Marken nicht wirklich messbar. Daraus kann gefolgert werden, dass sich für den Praxiseinsatz mit entfernten Marken auch das Maximum-Likelihood-Verfahren ohne wirklich messbare Laufzeiterhöhungen einsetzbar ist.

Die Dauer der Kugelerkennung ist hauptsächlich abhängig von der Größe der Kugelfläche und von sonst im Kugelfenster befindlichen hellen Objekten. Es kann also keine generell gültige Laufzeit des Erkennungsalgorithmus angegeben werden, da es auf die jeweilige Anzahl und Größe der Marken ankommt. Auch spielt die Umgebung eine entscheidende Rolle, da auch andere helle Objekte die Laufzeit erhöhen.

6.9.9. Vergleichstest

Als abschließender Test wurde eine ähnliche Situation wie in Kapitel 5.4 hergestellt (siehe Abb. 6.42), um erstens einen Vergleich der beiden Markentypen durchführen zu können und zweitens die Praxistauglichkeit der Zylindermarke zu prüfen. Die Einstellungen von Kamera und Objektiv entsprechen denen aus Kapitel 5.4 mit der Ausnahme, dass das Video mit 15 fps aufgenommen wurde, wodurch ein Video⁴³ mit insgesamt 729 Einzelbildern entstanden ist. Der Versuchsaufbau entspricht dem aus Abbildung 5.17 mit den folgenden Unterschieden:

⁴³Das Video befindet sich auf der beiliegenden DVD-ROM (cylinderfinaltest.avi).

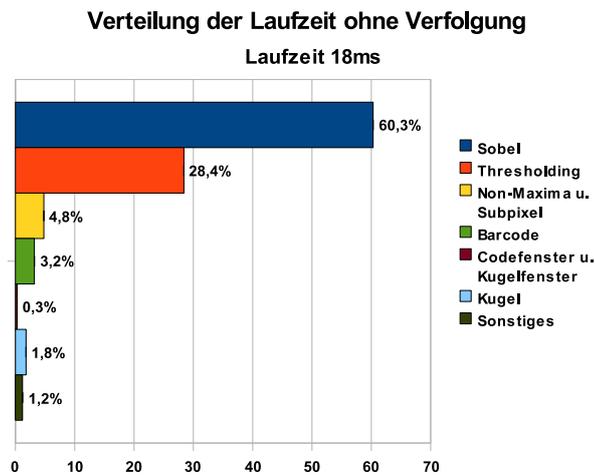


Abbildung 6.37. – Verteilung der Gesamtlaufzeit der Erkennung auf die einzelnen Erkennungsschritte für Abbildung 6.36 ohne Verfolgung.

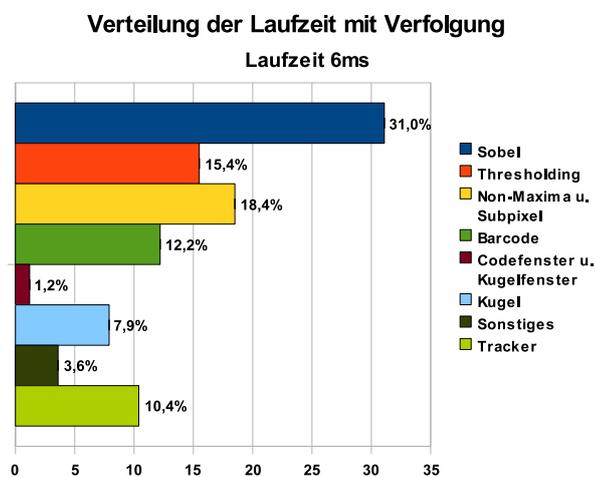


Abbildung 6.38. – Verteilung der Gesamtlaufzeit der Erkennung auf die einzelnen Erkennungsschritte für Abbildung 6.36 mit Verfolgung.

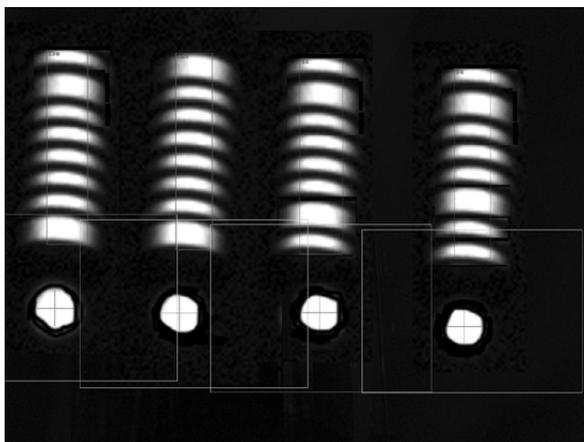


Abbildung 6.39. – Testbild für Laufzeittest mit 4 Zylindermarken, die fast das komplette Bild ausfüllen.

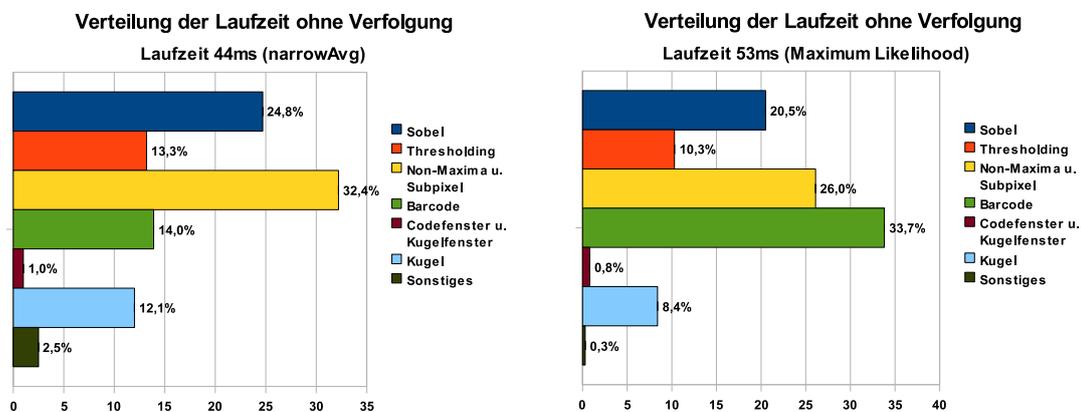


Abbildung 6.40. – Verteilung der Gesamtlaufzeit der Erkennung auf die einzelnen Erkennungsschritte für Abbildung 6.39 ohne Verfolgung.

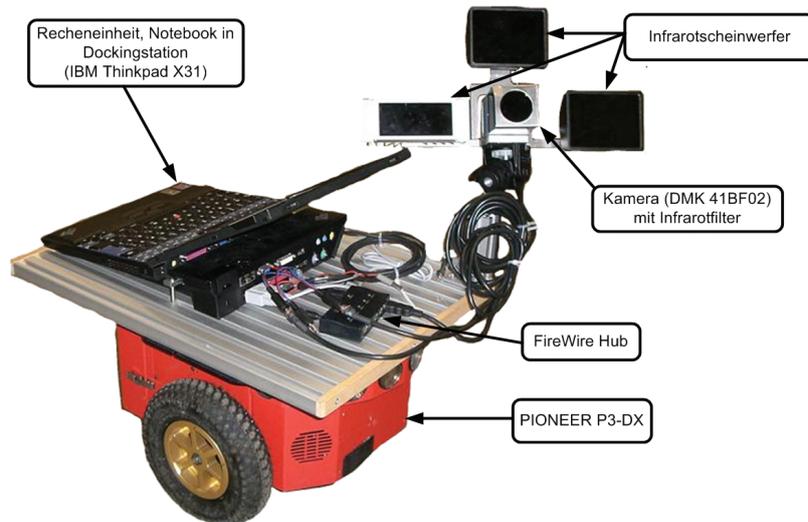


Abbildung 6.41. – Pioneer Roboter mit Vorrichtung für Kamera und Infrarotscheinwerfer, der zur Aufnahme eines Testvideos für die Zylindermarke genutzt wurde. Infrarotscheinwerfer (v.l.n.r): IR-30/80-880, IR-56/(60-40)-880, IR-56/20-880. Abbildung aus [Manske, 2008a]

- Es wurde kein Rollwagen eingesetzt, sondern ein Pioneer 3-DX Roboter, den auch Nico Manske für die Tests zu seiner Arbeit (siehe [Manske, 2008a]) genutzt hat, der von Hand im Schritttempo gefahren wurde. Dadurch ergibt sich eine Kamerahöhe von ca. 50 cm über dem Boden. Gleichzeitig ist extra für das Projekt eine Vorrichtung für die Ausrichtung der Kamera und die Anbringung der Infrarotscheinwerfer von der Werkstatt der HAW angefertigt worden, die die Scheinwerfer in einem Abstand von ca. 10 cm von der optischen Achse in die selbe Richtung wie die Kamera ausrichtet. Die Kamera und damit auch die Scheinwerfer hatten eine Neigung von ca. 28° (bezogen auf die optische Achse und den Boden) in Richtung Flurdecke. (siehe Abb. 6.41)
- Es wurden aus Gründen der Einfachheit nur 3 Marken (2-aus-8 Code) genutzt, die aber wie in Kapitel 5.4 jeweils im Abstand von 360 cm entlang des Flures aufgehängt wurden. Die erste Marke (ID 5) rechts versetzt, die zweite Marke (ID 40) in der Mitte und die dritte Marke (ID 68) links versetzt, um den Abstand zwischen ihnen zu maximieren.
- Auf die Nutzung der Funktion *cut* des Macros wurde verzichtet, da sich dies durch die Neigung der Kamera nicht lohnte, was aber zur Folge hat, dass (ohne Tracker) das gesamte Bild durchsucht werden muss.

Als Ergebnis ist eine durchschnittliche Laufzeit⁴⁴ von ca. 18 ms bei 3 erkannten Marken pro Einzelbild für das Macro auf dem verwendeten Testrechner (siehe Tab. 2.3), mit den in Tabelle 6.3 aufgeführten Werten für die Parameter, festzustellen. Die Laufzeit verändert sich kaum

⁴⁴Laufzeittests wurden durchgeführt ohne ein Ausgabebild zu erzeugen, da dies ca. 4 ms Zeit kostet.



Abbildung 6.42. – Einzelbild aus dem abschließenden Vergleichstest der Zylindermarke, in dem alle 3 Marken erkannt wurden.

verändert, wenn am Ende nur noch eine Marke erkannt wird (Laufzeit ca. 16ms). Die Größe der Marken im Bild hat auf die Laufzeit, da diese einen größeren Teil des Bildes einnehmen und somit breiter sind. Die reine Höhe der Marke spielt für die Erkennungsgeschwindigkeit kaum eine Rolle, jedenfalls bei der Nutzung des narrowAvg-Verfahrens. Das Hinzuschalten des Trackers ergab eine durchschnittliche Laufzeit von ca. 6ms bei der Verfolgung aller 3 Marken, wobei sich das Wegfallen der ersten Marke durch eine Zeitersparnis von ca. 2ms ausgewirkt hat, woran wieder zu sehen ist, wie wenig Zeit die Erkennung einer Marke selbst bei flächenmäßig größtmöglicher Abbildung in diesem Anwendungsszenario einnimmt. Das Tracking hat hier also zu einer Verringerung der Laufzeit um ca. $\frac{2}{3}$ geführt. Ähnliche Ausführungszeiten ergeben sich aber auch bei einer Abbildung 6.23 ähnlichen Situation mit 3 gleichgroß im Bild abgebildeten Marken. Das Vorhersagen mit dem Tracker selbst hat zuverlässig funktioniert, wenn auch mit einigen, durch die Aufnahme bedingten Rucklern im Video, kleineren Aussetzern.

Bei der Erkennungsentfernung waren die genutzten Zylindermarken nicht gleichwertig zu den in 5.4 genutzten Doppelverhältnismarken, wo teilweise bis zu 4 Marken gleichzeitig erkannt wurden, was einer maximalen Entfernung von ca. 13m-14m entspricht. Diese Entfernung ist mit den Marken in dieser Größe nicht möglich. Dennoch konnten, solange sie zu sehen waren, immer 3 Marken erkannt werden. Wobei die hinterste Marke am maximalen Limit war und somit auch bei noch mehr Marken nicht die Reichweite der getesteten Doppelverhältnismarken möglich ist. Hierfür müsste dann bei gleichem Barcode eine Erweiterung in der Höhe erfolgen, was sie aber noch weiter in den Raum ragen lassen würde als die verwendeten Doppelverhältnismarken. Die beiden größten bei diesem Test festgestellten Probleme betreffen aber den Durchmesser des Zylinders und die Beleuchtungssituation. Durch den nur sehr geringen Durchmesser des Zylinder von 11cm ergibt sich schnell ein Problem bei Situationen (siehe 6.5.1 und 6.9.2), die zu einem gedrehten Zylinder im Bild führen und es so schnell zu Problemen kommen kann. Aus diesem Grunde wurde auch auf die Nutzung eines anderen Wertes als 1 für den Parameter *stepSize* verzichtet, da dies die Situation verschlechtert hätte. Es ist deshalb zu empfehlen den Durchmesser des Zylinders ungefähr zu verdoppeln. Dies würde mit Blick auf die Erfahrungen aus 6.9.8 bezgl. der Laufzeit des Macros bedeuten, dass es sich maximal in den Regionen befinden würde, in denen sich auch die Erkennung der Doppelverhältnismarken bewegt. Wenn der Durchmesser des Zylinders noch weiter erhöht werden würde, oder man sich mit einer geringeren Erkennungsentfernung zufriedengibt, kann hier auch durch den Einsatz der Parameters *stepSize* Zeit gespart werden. Genauso wäre auch noch die dynamische Bestimmung dieses Parameters für die Suchfenster, die der Tracker bildet möglich, da die ungefähre Breite der Marke geschätzt wurde.

Als weitere Problematik stellte sich die Beleuchtung heraus, die mit der genutzten Vorrichtung, z.B. durch fehlende Gelenke für die Scheinwerfer, nicht flexibel genug einstellbar ist. Durch die Neigung der Kamera und der Scheinwerfer ergeben sich z.B. im Nahbereich unter der Decke sehr große Einfallswinkel, was, wie schon bei der Doppelverhältnismarke, zu sichtbar geringerer Reflexion des Lichts führt. Insgesamt sollte auf jeden Fall eine Erweiterung der Scheinwerfer erfolgen, die dazu führt, dass der ganze Bereich, den die Kamera erfassen kann ausgeleuchtet ist. Wie diese Ausleuchtung aber tatsächlich aussehen muss, sollte noch

durch weitere Tests geklärt werden, da sich, wie in Kapitel 6.9.1.1 beschrieben, eine größere Erkennungsentfernung durch eine eher schwache Beleuchtung und dadurch geringere Überstrahlung der Reflexfolie ergab. In diesem Zusammenhang sollten auch noch weitere Tests mit verschiedenen Blendeneinstellungen und Belichtungszeiten erfolgen.

Als Fazit für diesen abschließenden Vergleichstest kann also mitgenommen werden, dass das Konzept der Zylindermarke in der hier vorgestellten Form generell funktioniert, es aber noch einiger kleinerer Verbesserungen bedarf, um sie genauso praxistauglich, wie die Doppelverhältnismarke zu machen. Dies betrifft vor allem die Ausmaße des Zylinders, gerade dann, wenn ein anderer als der 2-aus-8 Barcode genutzt werden soll, aber auch die Beleuchtungssituation. Wobei die gemachten Angaben zur Beleuchtung auch die Doppelverhältnismarke betreffen. In Sachen der Laufzeit scheint das Paket aus Erkennungsalgorithmus und Marke der Zylindermarke insgesamt effizienter und schneller zu sein, als das der Doppelverhältnismarke. Dies ist u.a. dadurch bemerkbar, dass selbst bei einer Analyse des kompletten Bildes immer schnellere Ergebnisse geliefert werden, wobei sowohl die im Vergleich geringe abgebildete als auch die geringe reale Markengröße und damit auch der Verzicht auf eine (bei der Doppelverhältnismarke eingesetzten) größere Schrittweite zwischen den Scanlinien, noch Potential für Optimierungen der Laufzeit auf Kosten der Markengröße offen lassen.

6.10. Bewertung

Mit der Zylindermarke wird in dieser Arbeit ein speziell an die Anforderungen der Indoor-Navigation von autonomen Fahrzeugen mit bodennaher Kamera angepasster Markentyp inkl. Erkennungsalgorithmus vorgestellt, bei dem das zur Codierung der ID genutzte Muster auf einem Zylinder aufgebracht ist und der Messpunkt durch eine unter dem Zylinder befestigte Kugel dargestellt wird. Dieser Markentyp ist auf die Anbringung unter der Decke und die Beobachtung aus Richtung des Bodens ausgelegt und soll die Möglichkeit zur Einsparung von IDs im Vergleich zu anderen Marken geben. In diesem Fall ergibt sich durch die Zylinderform und die Kugel als Messpunkt eine Erkennung von nahezu jedem Standpunkt im Raum, wenn er sich nicht zu weit von der Marke entfernt oder unterhalb der Marke befindet. Ebenfalls ist zu beachten, dass wegen der Verwendung eines Barcodes z.B. die durch Neigung der Kamera in Richtung Decke entstehende perspektivische Verzerrung beachtet werden muss. Es wird aber gezeigt, dass der Verzicht auf eine perspektivisch invariante Codierung der ID, wie dem Doppelverhältnis, in der Praxis kaum Probleme bereitet. Wie auch schon bei der Doppelverhältnismarke kommt zur Signalisierung des Barcodes und der Kugel retroreflektierende Folie zum Einsatz, was in Kombination mit Infrarotscheinwerfern und eines Infrarotfilters vor der Kameralinse zu einer sehr effektiven Vereinfachung der Analyse des Bildes führt.

Generell haben sich alle 3 verwendeten Barcodes als funktionsfähig erwiesen, wenngleich mit dem 2-aus-8 Code die meisten Tests gemacht wurden. Dieser bietet zwar nur 28 eindeutige IDs, ist aber auch bei ähnlichen Ausmaßen des Zylinders am weitesten zu erkennen. Mit den beiden anderen Barcodes stehen noch andere Codes mit mehr Möglichkeiten zur Verfügung

und es scheint sinnvoll in dieses Thema noch mehr Arbeit zu stecken. Ziel kann es z.B. sein einen interleaved Barcode zu verwenden, der platzsparender ist. Generell sind alle in dieser Arbeit verwendeten Maße als Beispielmaße anzusehen, da der Erkennungsalgorithmus ausschließlich auf Verhältnissen arbeitet, die entsprechend angepasst werden können. Es können auch beliebige andere Barcodes verwendet werden. Hierbei ist nur zu beachten, dass alle codierten IDs eine gleiche Größe auf der Marke haben müssen und dass der entsprechende Abschnitt in der Erkennung angepasst werden muss.

Dieser spezielle Markentyp ist, in Verbindung mit dem genutzten Erkennungsalgorithmus, nicht invariant gegenüber allen affinen Transformationen, da u.a. die Invarianz gegenüber der, normalerweise bei codierten Messmarken geforderten, Rotation fehlt. Wobei dies neben dem sehr einfachen Erkennungsalgorithmus auch mit der freihängenden Kugel unterhalb des Zylinders begründet werden kann.

Durch die Tests konnte gezeigt werden, dass das Konzept dieses Markentyps funktionsfähig ist, es aber zur praxistauglichkeit im Vergleich zu den schon erfolgreich in der Arbeit von Nico Manske eingesetzten Doppelverhältnismarken noch einige Veränderungen, bezüglich der Ausmaße des Zylinders, vorgenommen werden sollten.

Der entwickelte Erkennungsalgorithmus ist sehr effizient und funktionierte während der Tests ohne Probleme. Wobei hier noch zu evaluieren ist, ob es nicht wegen seiner Einfachheit Probleme bei der Verwendung von anderen Zylindermaßen (besonders Zylinder mit größerem Durchmesser im Verhältnis zur Höhe) kommt, da dadurch mit der genutzten Methode z.B. sehr große Kugelfenster entstehen können.

Die gewählten Werte für die Parameter haben sich während der Tests als richtig erwiesen. Die Laufzeiten bewegen sich bei der genutzten Markengröße deutlich⁴⁵ unter denen der Doppelverhältnismarke, obwohl immer das ganze Bild untersucht wurde. Auch der Tracker hat in diesem Fall wieder sehr gut funktioniert. Beim Erkennungsalgorithmus sind aber noch Erweiterungen bezüglich der gezeigten Spezialfälle möglich, wobei dies auch teilweise durch Anpassungen an der Marke selbst passieren kann. Eine weitere mögliche Anpassung, die einerseits die Sicherheit, andererseits aber die Laufzeit erhöhen würde, betrifft den verwendeten Sobel-Operator. Hier wird zur Zeit nur die y-Maske genutzt, dies könnte auch auf die x-Maske ausgeweitet werden, um nur noch Balken zu berücksichtigen, die ungefähr die gleiche Orientierung besitzen. Dies wäre zur Zeit bei einem Vergleich der Werte der y-Maske nur ansatzweise möglich, da sich z.B. durch geneigte Marken extreme Unterschiede in Abhängigkeit von der Position zur Lichtquelle ergeben. In den bisherigen Tests hat die einseitige Berücksichtigung der y-Maske keine Probleme ergeben, weshalb diese Erweiterung nicht besonders wichtig wäre. Weitere Möglichkeiten zur Verbesserung bieten die genutzten Intervalle zur Klassifikation der Balken des Barcodes.

Zur Konfiguration der Marke und der Ermittlung der Parameterwerte für die Erkennung ist

⁴⁵nur ca. $\frac{2}{3}$ der Laufzeit ohne Tracking bei beiden

es sicherlich nützlich ein Konfigurationsprogramm zu entwickeln. Dieses Programm sollte dem Benutzer ermöglichen Marken mit beliebigen Maßen zu konfigurieren, bzw. ihn bei der Wahl der richtigen Maße zu unterstützen. Ein weiterer wichtiger Bestandteil des Programms sollte eine Unterstützung bzw. automatische Berechnung/Schätzung von sinnvollen Werten für die Parameter des Erkennungsalgorithmus sein.

7. Kreisförmige Barcodemarke

7.1. Motivation

Während der Tests im Rahmen der Masterarbeit von Nico Manske (siehe [Manske, 2008a]) hat sich ergeben, dass es nützlich wäre einen Markentyp zu haben, deren Marken direkt mit ihrer größten Fläche unter der Decke angebracht werden können⁴⁶, um sehr einfach einen Testparcours aufbauen zu können. Für dieses Anwendungsszenario sind aber Marken notwendig, die rotationsinvariant sind, da sich durch eine Drehung des Fahrzeugs auch die Marke im Bild dreht. Manske hat hierfür die zu diesem Zeitpunkt schon fertiggestellte Doppelverhältnismarke genutzt (siehe Abb. 7.1), die zwar durch den fünften Kreis rotationsinvariant ist, aber deren Erkennungssoftware, wie schon erwähnt (siehe Kapitel 5.5), nur eingeschränkt und mit Einbußen bezüglich der Erkennungsgeschwindigkeit, rotierte Marken erkennt. Aus diesem Grunde entstand die Idee, aufbauend auf den während dieser Arbeit gewonnen Erkenntnissen, noch ein einfaches Konzept für eine konzentrische Kreismarke zu entwerfen, dessen prototypische Umsetzung aber zeitlich nicht mehr möglich war.

Die Eigenschaften und Vorteile einer kreisförmigen Marke sind in Kapitel 3.2.1 schon hinreichend erklärt. Bei einer parallelen Ausrichtung der Bildebene zur Decke kommt vorteilhaft noch hinzu, dass keine nennenswerte perspektivische Verzerrung der Marke zu befürchten ist, was bezüglich der genutzten Codes mehr Möglichkeiten offen lässt. Außerdem sollte sich durch die konstante Entfernung von der Decke auch die korrekte Ausleuchtung der Marken/Reflexfolie einfacher gestalten, als in anderen Szenarien.

7.2. Konzept

7.2.1. Aufbau

Eine einfache Marke, die auf den bei der Entwicklung und den Tests der Zylindermarke gemachten Erfahrungen und den somit schon vorhandenen Verfahren aufbaut, ist in Abbildung 7.2 zu sehen. Hierbei ist der in Kapitel 6.4.3 beschriebene 2-aus-5 Barcode, zur Codierung einer zweistelligen Dezimalzahl, in konzentrischen Ringen von außen nach innen aufgetragen. Den Messpunkt bildet der Mittelpunkt eines Kreises (Messkreis) in der Mitte⁴⁷ der Ringe,

⁴⁶Die Kamera/Bildebene wird in diesem Fall parallel zur Decke ausgerichtet.

⁴⁷gleicher Mittelpunkt



Abbildung 7.1. – Testumgebung aus [Manske, 2008a] mit Doppelverhältnismarken und parallel zur Decke ausgerichteter Kamera. Abbildung aus [Manske, 2008a]

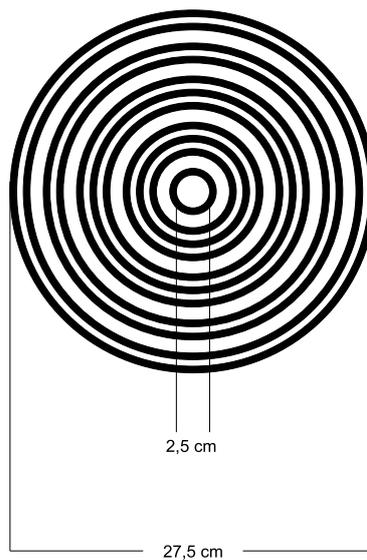


Abbildung 7.2. – Kreisförmige Barcodemarke mit 2-aus-5 Code und beispielhaften Maßen, um eine zweistellige Dezimalzahl zu codieren. Die Abmessungen der schwarzen Zwischenräume und der schmalen Balken betragen 0,5 cm. Die breiten Balken sind mit 1,5 cm doppelt so groß.

welcher, wie auch alle anderen hellen Elemente, aus Reflexfolie gefertigt werden sollte. Dieses Muster ist durch ein einfaches Scanlinienverfahren effizient erkennbar und bietet durch die Wiederholung des Barcodes auch eine sehr hohe Sicherheit beim Auslesen der codierten ID, bzw. dem Auffinden der Marke im Bild. Mit konzentrischen Ringen zu arbeiten ist nichts neues, wie z.B. an der Marke "a" in Abbildung 3.2 zu sehen ist.

Wenn man, wie in den Räumen der HAW üblich, von einer Raumhöhe von 3 m ausgeht, sind die angegebenen Abmessungen der Marke ausreichend, um mit der in dieser Arbeit verwendeten Kamera eine Abbildung von ca. 2 Pixel für die schmalen Balken und ca. 9 Pixel für den Messkreis auf der Bildebene zu erzeugen, was für eine robuste Erkennung genügen sollte. U.u. kann es sinnvoll sein, zur Abgrenzung den schwarzen äußeren Rand zu vergrößern. Mit einer Vergrößerung der Balken, und der damit noch besseren Erkennbarkeit der Marke, um den Faktor 0,5 hat die Marke einen Durchmesser von 40 cm, was immernoch unterhalb der Breite der in [Manske, 2008a] von Manske genutzten Kombination aus zwei Doppelverhältnismarken liegen würde. Da durch den verwendeten Barcode 100 Dezimalzahlen als eindeutige IDs zur Verfügung stehen und Manske bis jetzt 18 Doppelverhältnismarken für ca. 4 m² verwendet hat, kann diese Menge an möglichen Marken, bei geschickter Platzierung, für ca. 4 m² ausreichen. Auch diese Maße der Marke sind wieder nur als Beispiele zu verstehen, es ist selbstverständlich auch möglich andere Größen zu verwenden.

7.2.2. Erkennung

Generell kann hier nach dem gleichen Ablaufmuster (siehe Abb. 6.16) verfahren werden, wie bei der Zylindermarke in Kapitel 6.6. Die Verfahren stehen durch die Zylindermarke auch alle schon weitestgehend zur Verfügung und müssen nur an einigen Stellen angepasst werden. Ziel ist es die Marke mit einer Scanlinie so zu treffen, dass diese durch den Messkreis der Marke geht (siehe Abb. 7.3). Als Schrittweite (*stepSize*) zwischen den Scanlinien empfiehlt sich hier ein Wert, der ungefähr der Hälfte der erwarteten Größe der Abbildung des Messkreises auf dem Kamerasensor entspricht. Der Ablauf ergibt sich dann vereinfacht wie folgt:

1. **Kantenfilterung:** Entlang vertikaler Scanlinien wird eine subpixelgenaue Kantenfilterung, bestehend aus der Anwendung der Sobelmaske in y-Richtung, Non-Maxima-Suppression und der Interpolation eines Polynoms vorgenommen. Die Speicherung der gefilterten Informationen kann in der gleichen Datenstruktur, wie auch bei der Zylindermarke gespeichert werden. U.u. kann es hier bei den/dem innersten Kreis/den innersten Ringen wegen der starken Krümmung zu Problemen wegen zu schwachen Werten der Sobelmaske kommen. In diesem Falle ist am einfachsten entweder der Kreisdurchmesser zu erhöhen oder die Schrittweite (*stepSize*) zu verringern.
2. **Barcodefilterung:** Statt, wie bei der Zylindermarke bei einem doppelten 2-aus-5 Code, nach 10 schwarz-weiß-, weiß-schwarz Übergängen entlang einer Scanlinie zu suchen, werden in diesem Fall 20 dieser Paare, plus einem weiteren Paar für den Messkreis gesucht. Das vorhandene Verfahren kann hier also fast unverändert übernommen werden.

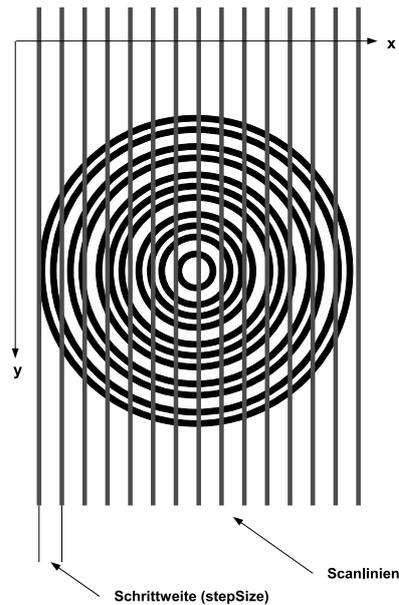


Abbildung 7.3. – Kreisförmige Barcode-Marke mit 2-aus-5 Code, die von Scanlinien getroffen wird. Allerdings wird nur die mittlere zu einer erfolgreichen Identifikation der Marke führen.

Die Bestimmung eines Basiswerts für die Klassifikation der Codebalken sollte über den gesamten Code durchgeführt werden, um die Zusammengehörigkeit aufgrund der Größe testen zu können. Die benötigten Intervalle für die Klassifikation sollten aufgrund der fehlenden perspektivischen Verzerrung wesentlich kleiner sein und lassen sich durch einige Testvideos ermitteln. Als korrekt wird ein Code befunden, wenn er den Maßen entspricht, die die Intervalle festlegen und zusätzlich die Barcodes auf beiden Seiten des Messkreises gleich sind. Die Suche wird auch hier solange vertikal in einer Spalte des Bildes fortgesetzt, bis keine Balken für mögliche Barcodes mehr übrig sind (siehe Kapitel 6.6.3). Die Speicherung der gewonnenen Informationen kann ebenfalls in der vorhandenen Datenstruktur vorgenommen werden.

3. **Markenextraktion:** Durch die Redundanz des Barcodes ist es bei der Markenextraktion vertretbar, anders als bei der Zylindermarke, eine Marke auch schon bei dem auffinden eines einzelnen Barcode-segments eine korrekte Marke zu vermuten. Dies kann dann durch eine Suche nach einem passenden Messkreis mit dem vorhandenen Algorithmus aus Kapitel 5.1.3.2) geschehen. Als Startwert dient hierbei die Mitte des mittleren gefunden schwarz-weiß-, weiß-schwarz Übergangspaares. Wenn dort kein passenden Kreiselement gefunden wurde, was z.B. durch den Formfaktor, den Durchmesser und die Lage des Mittelpunktes geprüft werden kann, ist keine korrekte Marke gefunden worden. Ansonsten stehen damit die subpixelgenauen 2D Koordinaten der gefundenen Marke im Bild fest.



Abbildung 7.4. – Konzentrische Ringmarken aus [Thomas u. a., 1997]

Fehlidentifikationen sind aufgrund der fehlenden starken perspektivischen Verzerrung sehr unwahrscheinlich und auch das Problem, dass aus mehreren benachbarten Barcodes ein neuer korrekter Code entsteht ist durch die Redundanz des Codes stark gemindert. Auch die Problematik der u.U. nicht eindeutigen Zuordbarkeit von Barcode und Kugel ist durch die sowohl durch den Barcode als auch durch den Messkreis führende Scanlinie hier nicht gegeben. Die Identifizierung eines anderen Musters als Marke ist einerseits ebenfalls durch die Redundanz noch unwahrscheinlicher geworden, aber andererseits trägt auch die wahrscheinlich unter Decke geringere Anzahl von Störobjekten dazu bei.

Diese Marke muss auch nicht unbedingt nur auf eine Deckenmontage, oder eine Montage parallel zur Bildebene der Kamera festgelegt sein. Durch perspektivische Verzerrung wird sich hier aber eine Ellipsenbildung einstellen und auch der Barcode müsste dann wieder genauer beachtet werden.

7.2.3. Weitere Ideen

Wie schon bei der Zylindermarke erwähnt, ist es auch hier sinnvoll einen interleaved Barcode auszuprobieren, um den zur Verfügung stehenden Platz noch effektiver zu nutzen.

Ein ähnlicher Ansatz ist in [Thomas u. a., 1997] beschrieben. Dort geht es um ein System zur Ermittlung der Position einer Kamera in einem Fernsehstudio. Hierfür werden die Marken ebenfalls parallel zur Decke befestigt und mit einer ebenfalls parallelen Kamera aufgenommen. Zwei Dinge sind hierbei interessant:

- Die Marken werden teilweise an Stangen mit unterschiedlichen Abständen unter der Decke angebracht, um so die Messgenauigkeit zu erhöhen, da sich so nicht mehr alle Marken/Punkte in der gleichen Ebene befinden.
- Zur Codierung der ID scheinbar ein sehr einfacher Code genutzt (siehe Abb. 7.4). Hierbei gibt es einen (unterschiedlich großen) mittleren Messkreis, der selbst auch als Codeelement genutzt wird und um den konzentrische Ringe angebracht sind. Zum verwendeten Code selbst ist nichts angegeben, es ist aber anzunehmen, dass jeder dieser Ringe aus Reflexfolie für ein Bit (bzw. je nach Breite für mehrere) steht. Die Wertigkeit ergibt sich hierbei wahrscheinlich aus der Position vom Mittelpunkt der Marke aus gesehen. Der äußerste Ring scheint hierbei immer vorhanden zu sein und die Größe der Marke konstant zu definieren. Hierdurch sind mit einfachen Mitteln sehr viele eindeutige IDs möglich. Dieser sehr einfache Ansatz scheint durch die Symmetrie der Marke und die (fast) fehlende perspektivische Verzerrung begünstigt, da sonst doch davon auszugehen ist, dass die Bestimmung der Bitwertigkeit Probleme bereitet.

8. Fazit und Ausblick

8.1. Fazit

Das Hauptziel dieser Arbeit war die prototypische Entwicklung von codierten Messmarken und Verfahren zu deren Erkennung in Einzelbildern eines digitalen Videosignals. Diese Marken sollen die kamerabasierte Posenbestimmung und damit später auch die autonome Navigation eines mobilen Systems im Indoor-Bereich ermöglichen. Als Szenario für diese Arbeit wurde ein auf dem Boden fahrendes Fahrzeug angenommen, wobei sich die Konzepte auch auf andere Szenarien übertragen lassen.

Generell ist das Hauptziel als erfüllt anzusehen, da durch die Weiterentwicklung der Doppelverhältnismarke von Matthias Wippich (siehe [Wippich, 2008a]) eine Marke geschaffen wurde, die schon praktisch bei Messungen mit einer Genauigkeit von 2 – 3 cm eingesetzt werden konnte (siehe Kapitel 5). Als sinnvollste Anbringungsform, im Bezug auf einen möglichst sparsamen Umgang mit den zur Verfügung stehenden IDs, wurde die Deckenmontage analysiert. Dabei sind die Marken so angebracht, dass sie in den Raum hängen. Da bei dieser Anbringungsform in großen Räumen aber trotzdem sehr viele Doppelverhältnismarken nötig sind, um möglichst häufig die für die Posenbestimmung nötigen 3 Marken im Kamerabild identifizieren zu können, wurde eine Marke entwickelt, die von fast allen Standpunkten im Raum sichtbar ist (siehe Kapitel 6). Um dies zu ermöglichen wurde ein Zylinder mit einem Barcode ausgestattet und zur Realisierung der Messpunktes eine Kugel untergehängt. Da auch diese Marke nicht für alle Situationen geeignet ist, wurde außerdem ein Konzept für eine kreisförmige Barcodemarke entworfen (siehe Kapitel 7). Die Anzahl der eindeutigen IDs, die codiert werden können, hängt bei allen Markentypen hauptsächlich von der Größe der Marke und der beabsichtigten Erkennungsentfernung in Kombination mit der eingesetzten Kamera ab. Bei allen Markentypen stehen aber in den meisten Fällen mindestens 28 eindeutig identifizierbare Marken zur Verfügung.

Sowohl der schon größtenteils vorhandene Erkennungsalgorithmus der Doppelverhältnismarke als auch das komplett innerhalb dieser Arbeit entwickelte Verfahren zur Erkennung der Zylindermarke sind als Modul für die Testumgebung Impresario umgesetzt worden. Beide Verfahren sind sehr robust und effizient und ließen auf dem genutzten Testrechner eine Erkennung im unteren bis mittleren zweistelligen Millisekundenbereich zu. Diese Eigenschaft zeigt, dass sie für den Einsatz in einem Gesamtsystem geeignet, welches sich eine Abtastrate von 10 Hz als Ziel gesetzt hat (siehe [Manske, 2008b]). Die Laufzeit der Module konnte sowohl durch den Einsatz eines Verfolgungsansatzes (siehe Kapitel 4) als auch durch die Verwendung

von Infrarotbeleuchtung (in Kombination mit retroreflektierender Folie) weiter gesenkt werden. Zur Verfolgung der Marken, mit dem Ziel für jede Marke ein Suchfenster für das nächste Bild vorhersagen zu können, wurde ein einfaches Verfahren, welches auf der Errechnung der Geschwindigkeit einer Marke aus ihren letzten beiden Positionen basiert, und das Kalman Filter analysiert. Das Kalman Filter hat sich hierbei als etwas genauer gezeigt, was die Schätzungen der Position betrifft, verbraucht aber auch mehr Laufzeit.

Die Funktionsfähigkeit der Konzepte und Erkennungsmodule für Doppelverhältnis- und Zylindermarke konnte in Tests nachgewiesen werden. Gleichzeitig wurden im Rahmen dieser Tests auch sinnvolle Maße der jeweiligen Marke diskutiert und für den Praxiseinsatz notwendige Verbesserungen aufgezeigt.

8.2. Ausblick

Für die Zylindermarke und die noch praktisch umzusetzende kreisförmige Barcodemarke sollte nach anderen u.U. effizienteren Barcodes gesucht werden.

Eigentlich war geplant eine Marke, die nur aus einer Kugel besteht zu entwickeln, dies ist leider nicht gelungen. Der Ansatz sollte aber weiter verfolgt werden, da diese Marke in fast allen Situationen einsetzbar wäre.

Als ungelöstes Problem bleibt die z.Z. nicht optimale Beleuchtungssituation. Diese ist mit den zur Verfügung stehenden Infrarotscheinwerfern wahrscheinlich auch nicht wirklich besser zu lösen, weshalb, auch gerade im Hinblick auf die Problematik der korrekten Beleuchtung der Kugel der Zylindermarke, eine symmetrisch um die Kamera verteilte Beleuchtung angestrebt werden sollte, die das komplette Blickfeld der Kamera möglichst auf allen Erkennungsentfernungen optimal⁴⁸ ausleuchtet. Auch die Analyse anderer Typen von Reflexfolie, die einen höheren Rückstrahlwert bei großen Einfallswinkeln besitzen sollte in Betracht gezogen werden.

Weiterhin gibt es bis jetzt noch keine Tests zu Genauigkeit des genutzten Verfahrens zur Bestimmung der Kugel- bzw. Kreismitte der Messmarken. Hier sollte eine genauere Analyse erfolgen, ob diese Algorithmen überhaupt in der Lage sind millimetergenaue Messungen zu ermöglichen.

Eine weitere noch offene Aufgabe bezieht sich auf die optimale Positionierung der Marken im Raum. Hier sollte ein Verfahren gefunden werden, mit dem es möglich ist die beste Verteilung der Marken zu bestimmen, sodass aus möglichst vielen Standpunkten im Raum eine Erkennung von 3 Marken möglich ist.

Auch im Bezug auf das verwendete Kalman Filter sind noch eine Menge Optimierungen möglich. Auch hier könnten noch andere Konzepte, wie das des Partikelfilters, überprüft werden,

⁴⁸Was optimal wirklich bedeutet, müsste durch weitere Tests bestimmt werden (siehe auch Kapitel 6.9.1.1).

um festzustellen, ob noch bessere Ergebnisse für die Schätzungen möglich sind. Zu besseren Schätzungen kann auch der Einsatz von Daten aus der Odometrie (o.ä.) des Fahrzeugs führen, die dann ebenfalls in das Kalman Filter aufgenommen werden könnten.

Eine Identifikation von 3 Marken von jedem Standort im Raum aus, um zu jeder Zeit eine Posenbestimmung durchführen zu können, lässt sich mit den von der Decke hängenden Zylinder- oder Doppelverhältnismarken nicht realisieren, da z.B. eine Drehung des Fahrzeugs im getesteten Flurszenario dazu führen würde, dass sich gar keine Marke mehr im Bild befindet, da an der Wand keine Marken befestigt waren. Bei den mit ihrer größten Fläche direkt unter der Decke angebrachten kreisförmigen Barcodemarken ist dieses Problem zwar weniger stark vertreten, es werden hierfür aber auch wesentlich mehr Marken benötigt. Und in Wandnähe oder bei Verdeckung einiger Marken, z.B. durch neben der Kamera stehende Personen, ergeben sich auch hier Probleme, die zu einer Anbringung von noch mehr und dichter befestigten Marken führen müssten. Dies setzt ausserdem die Nutzung eines anderen Codes voraus, der mehr Möglichkeiten der Abbildung bietet. Eine Posenbestimmung zu jedem Zeitpunkt sollte für eine Navigation aber auch nicht erforderlich sein, da ein Fahrzeug nicht zu jedem Zeitpunkt in der Lage sein muss seine Position und Orientierung per Messung zu verifizieren. Da, um die Pose des Fahrzeugs trotzdem zu jedem Zeitpunkt ungefähr schätzen zu können, der Einsatz eines Filters zur Zustandsschätzung sinnvoll ist, könnte dieser u.U. auch dazu genutzt werden, um für die Lokalisierung der Marken sinnvolle Hinweise zu geben. Durch die geschätzte Pose und die bekannten Positionen der Marken im Raum, könnte der in dieser Arbeit vorgestellte Ansatz zur Markenverfolgung noch verbessert werden.

A. Anhang

Die beiliegende DVD-ROM ist in folgende Verzeichnisse unterteilt:

Verzeichnis	Inhalt
/thesis/	Dieses Dokument als PDF-Datei, in der es möglich ist die meisten Diagramme und Grafiken stufenlos zu vergrößern.
/impresario/ /impresario/imp/	Impresario Enthält sowohl die ausführbare Impresario.exe für Windows als auch die XML-Dateien zum Test der Doppelverhältnismarken.
/impresario/imp/imageinput/ /impresario/imp/documents/	Testvideos und Bilder zu dieser Arbeit. Vorgefertigte Graphen zum einfachen ausprobieren der entwickelten Macros.
/impresario/imp/macrodev/projects/codedtargets/	Enthält den Quellcode der erstellten Macros.
/DVkonfiguration/	Konfigurationsprogramm für die Doppelverhältnismarken.

Literaturverzeichnis

- [3M 2004a] 3M: Engineer Grade Work Zone Sheeting Series 3210. URL <http://www.3m.com>, 2004. – Product Bulletin. Zuletzt abgerufen am 23.11.2008
- [3M 2004b] 3M: Reflectivity / Traffic Safety Systems Division. URL <http://www.3m.com>, 2004. – Forschungsbericht. Zuletzt abgerufen am 23.11.2008
- [3M 2007] 3M: 3M Scotchlite Reflective Material - SOLAS Grade Products. URL <http://www.3m.com>, 2007. – Technical Data Sheet. Zuletzt abgerufen am 23.11.2008
- [3M 2008] 3M: *Was ist Retroreflexion?*, Webseite, 2008. – URL http://solutions.3m.com/wps/portal/3M/de_DE/Scotch-Lite/Home/ProductInfo/InAction/. – Zuletzt abgerufen am 23.11.2008
- [Ahn und Oberdorfer 1998] AHN, S. J. ; OBERDORFER, B.: Optoelektronisch erfaßbares Identifizierungs- oder Zielelement sowie Verfahren zu seiner Erfassung. 1998 (DE 19632058 C1). – Deutsches Patent
- [Ahn u. a. 1999] AHN, S. J. ; RAUH, W. ; RECKNAGEL, M.: Circular coded landmark for optical 3D-measurement and robot vision. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS '99* Bd. 2, 1999, S. 1128–1133 vol.2
- [Arıcan 2004] ARICAN, Z.: *Vision-Based Robot Localization Using Artificial And Natural Landmarks*, Middle East Technical University, Masterarbeit, 2004. – URL <http://etd.lib.metu.edu.tr/upload/12605333/index.pdf>. – Zuletzt abgerufen am 23.11.2008
- [ARTag 2008] ARTAG: *Offizielle Webseite*, Webseite, 2008. – URL <http://www.artag.net>. – Zuletzt abgerufen am 23.11.2008
- [Blanche und Lemarquand 1992] BLANCHE, C. ; LEMARQUAND, G.: New structures for linear displacement sensor with high magnetic field gradient. In: *Proc. International Magnetism Conference Digests of Intermag '92*, 1992, S. 44–44
- [Bradley 2005] BRADLEY, G.: Natural Interaction with Virtual Objects Using Vision-Based Six DOF Sphere Tracking / National Research Council of Canada. URL <http://iit-iti.nrc-cnrc.gc.ca/iit-publications-iti/docs/NRC-48212.pdf>, 2005. – Forschungsbericht. Zuletzt abgerufen am 23.11.2008

- [Braun 2005] BRAUN, T.: *Detektion von Landmarken zur Navigation von Laufmaschinen*, AG Robotersysteme Fachbereich Informatik an der Universität Kaiserslautern, Seminararbeit, 2005. – URL agrosy.informatik.uni-kl.de/fileadmin/Literatur/image_processing/Braun05d.pdf. – Zuletzt abgerufen am 23.11.2008
- [Bregler u. a. 2005] BREGLER, C. ; CASTIGLIA, C. ; DEVINCEZO, J. ; DUBOIS, R. L. ; FEELEY, K. ; IGOE, T. ; MEYER, J. ; NAIMARK, M. ; POSTELNICU, R. ; RABINOVICH, M. ; ROSENTHAL, S. ; SALEN, K. ; SUDOL, J. ; WRIGHT, B.: *Squidball: An Experiment in Large-Scale Motion Capture and Game Design*. URL <http://www.cs.nyu.edu/csweb/Research/TechReports/TR2005-869/TR2005-869.pdf>, 2005. – Forschungsbericht. Zuletzt abgerufen am 23.11.2008
- [Briggs u. a. 2000] BRIGGS, A.J. ; SCHARSTEIN, D. ; BRAZIUNAS, D. ; DIMA, C. ; WALL, P.: *Mobile robot navigation using self-similar landmarks*. In: *Proc. IEEE International Conference on Robotics and Automation ICRA '00* Bd. 2, 2000, S. 1428–1434 vol.2
- [Bundesministerium für Verkehr, Bau und Stadtentwicklung 1998] BUNDESMINISTERIUM FÜR VERKEHR, BAU UND STADTENTWICKLUNG: *ECE-Regelung Nr. 104: Retroreflektierende Markierungen an schweren und langen Fahrzeugen (Konturmarkierung)*, ECE-Regelung, 1998
- [Chen u. a. 2007] CHEN, Z. ; CHAN, D. T. W. ; YE, Z. ; PENG, G.: *Target Recognition Based on Mathematical Morphology*. In: *Proc. 10th IEEE International Conference on Computer-Aided Design and Computer Graphics*, 2007, S. 457–460
- [Coughlan u. a. 2006] COUGHLAN, J. ; MANDUCHI, R. ; SHEN, H.: *Cell Phone-based Wayfinding for the Visually Impaired* / Smith-Kettlewell Eye Research Institute, San Francisco University of California, Santa Cruz. URL <http://www.soe.ucsc.edu/~manduchi/papers/WayfindingFinal.pdf>, 2006. – Forschungsbericht. Zuletzt abgerufen am 23.11.2008
- [Cuevas u. a. 2005] CUEVAS, E. ; ZALDIVAR, D. ; ROJAS, R.: *Kalman filter for vision tracking* / Freie Universität Berlin, Institut für Informatik. URL <http://www.inf.fu-berlin.de/~rojas/2005/tr-b-05-12.pdf>, 2005. – Forschungsbericht. Zuletzt abgerufen am 23.11.2008
- [Davies 2005] DAVIES, E.: *Machine Vision*. Third Edition. Amsterdam : Elsevier, 2005
- [Devernay 1995] DEVERNAY, F.: *A non-maxima suppression method for edge detection with sub-pixel accuracy* / INRIA Research Rep. 2724, SophiaAntipolis. 1995. – Forschungsbericht
- [Federal Highway Administration 2001] FEDERAL HIGHWAY ADMINISTRATION: *Guidelines and Recommendations to Accommodate Older Drivers and Pedestrians* / U.S. Department of Transportation. URL <http://www.tfhrc.gov/humanfac/01105/01-051.pdf>, 2001. – Forschungsbericht. Zuletzt abgerufen am 23.11.2008

- [Fiala 2005a] FIALA, M.: ARTag, a fiducial marker system using digital techniques. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2005* Bd. 2, 2005, S. 590–596 vol. 2
- [Fiala 2005b] FIALA, M.: Comparing ARTag and ARToolkit Plus fiducial marker systems. In: *Proc. IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, 2005, S. 6 pp.–
- [Fiala und Shu 2005] FIALA, M. ; SHU, C.: Fully Automatic Camera Calibration Using Self-Identifying Calibration Targets / National Research Council Canada. URL <http://www.iit-iti.nrc-cnrc.gc.ca/iit-publications-iti/docs/NRC-48306.pdf>, 2005. – Forschungsbericht. Zuletzt abgerufen am 23.11.2008
- [Garcia Garcia u. a. 2002] GARCIA GARCIA, J.C. ; MARRON ROMERA, M. ; MAZO QUINTAS, M. ; URENA URENA, J.: Positioning and localization system for autonomous wheelchairs. In: *Proc. IEEE 2002 28th Annual Conference of the IECON 02 [Industrial Electronics Society]* Bd. 2, 2002, S. 1555–1560 vol.2
- [Gonzales und Woods 2002] GONZALES, R. C. ; WOODS, R. E.: *Digital Image Processing*. Second Edition. Prentice Hall, 2002
- [Hartley und Zisserman 2003] HARTLEY, R. ; ZISSERMAN, A.: *Multiple View Geometry in Computer Vision*. Second Edition. Cambridge University Press, 2003
- [Hornberg 2006] HORNBERG, A.: *Handbook of Machine Vision*. Weinheim : Wiley-VCH, 2006
- [Jähne 2005] JÄHNE, B.: *Digitale Bildverarbeitung*. 6., überarb. u. erw. Aufl. Springer, Berlin, 5 2005
- [Lin und Chang 2005] LIN, Huei-Yung ; CHANG, Chia-Hong: Automatic speed measurements of spherical objects using an off-the-shelf digital camera. In: *Proc. IEEE International Conference on Mechatronics ICM '05*, 2005, S. 66–71
- [Luhmann 2000] LUHMANN, T.: *Nahbereichsphotogrammetrie: Grundlagen, Methoden und Anwendungen*. Heidelberg : Herbert Wichmann Verlag, 2000
- [Manske 2008a] MANSKE, N.: *Kamerabasierte Präzisionsnavigation mobiler Systeme im Indoor-Bereich*, Hochschule für Angewandte Wissenschaften Hamburg, Masterarbeit, 2008
- [Manske 2008b] MANSKE, N.: *Verfahren zur Positionsbestimmung im Indoor-Bereich*, Hochschule für Angewandte Wissenschaften Hamburg, Seminararbeit, 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/manske/bericht.pdf>. – Zuletzt abgerufen am 23.11.2008
- [Manske und Jost 2008] MANSKE, N. ; JOST, T.: *Posenbestimmung in Räumen mit einem 3D-Kameramodell*, Hochschule für Angewandte Wissenschaften Hamburg, Projektarbeit, 2008. – Zuletzt abgerufen am 23.11.2008

- [MSN 2008] MSN: *2007 Toyota Prius Safety and Crash Tests*, Webseite, 2008. – URL <http://autos.msn.com/research/vip/safety.aspx?year=2007&make=Toyota&model=Prius>. – Zuletzt abgerufen am 23.11.2008
- [Negre u. a. 2007] NEGRE, A. ; PRADALIER, C. ; DUNBABIN, M.: Robust Vision-based Underwater Target Identification and Homing Using Self-Similar Landmarks. In: *Proc. of the Int. Conf. on Field and Service Robotics (2007)*, 07 2007
- [Niederöst und Maas 1997] NIEDERÖST, M. ; MAAS, H.-G.: Entwurf und Erkennung von codierten Zielmarken. URL <http://e-collection.ethbib.ethz.ch/show?type=inkonf&nr=71>, 1997. – Forschungsbericht. Zuletzt abgerufen am 23.11.2008
- [Nischwitz und Haberäcker 2004] NISCHWITZ, A. ; HABERÄCKER, P.: *Masterkurs Computergrafik und Bildverarbeitung*. 1. Vieweg Friedr. + Sohn Ver, 8 2004
- [Owen u. a. 2002] OWEN, C.B. ; XIAO, F. ; MIDDLEIN, P.: What is the best fiducial? In: *Proc. First IEEE International Workshop Augmented Reality Toolkit*, 2002, S. 8 pp.–
- [Pavlidis u. a. 1990] PAVLIDIS, T. ; SWARTZ, J. ; WANG, Y.P.: Fundamentals of bar code information theory. In: *Computer* 23 (1990), Nr. 4, S. 74–86
- [Perret Optic 2008] PERRET OPTIC: *OZON - SONNE - AUGEN*, Webseite, 2008. – URL http://www.perret-optic.ch/Soleil_ozone_yeux/soleil_ozone_yeux_d.htm. – Zuletzt abgerufen am 23.11.2008
- [Saito u. a. 2007] SAITO, S. ; HIYAMA, A. ; TANIKAWA, T. ; HIROSE, M.: Indoor Marker-based Localization Using Coded Seamless Pattern for Interior Decoration. In: *Proc. IEEE Virtual Reality Conference VR '07*, 2007, S. 67–74
- [Sattar u. a. 2007] SATTAR, J. ; BOURQUE, E. ; GIGUERE, P. ; DUDEK, G.: Fourier tags: Smoothly degradable fiducial markers for use in human-robot interaction. In: *Proc. Fourth Canadian Conference on Computer and Robot Vision CRV '07*, 2007, S. 165–174
- [Shakey 2008] SHAKEY: *Offizielle Webseite*, SRI International's Artificial Intelligence Center, Webseite, 2008. – URL <http://www.ai.sri.com/shakey/>. – Zuletzt abgerufen am 23.11.2008
- [Stöcker 1995] STÖCKER, Horst: *Taschenbuch mathematischer Formeln und moderner Verfahren*. 3., überarb. u. erw. A. Harri Deutsch, 1995
- [Thomas u. a. 1997] THOMAS, G.A. ; JIN, J. ; NIBLETT, T. ; URQUHART, C.: A versatile camera position measurement system for virtual reality TV production. In: *Proc. International Broadcasting Convention*, 1997, S. 284–289
- [U.S. National Communications System Technology & Standards Division 1996] U.S. NATIONAL COMMUNICATIONS SYSTEM TECHNOLOGY & STANDARDS DIVISION: *Federal Standard 1037C*, Federal Standard, August 1996. – URL <http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm>

- [Volpe u. a. 1995] VOLPE, R. ; LITWIN, T. ; MATTHIES, L.: Mobile robot localization by remote viewing of a colored cylinder. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots'* Bd. 1, 1995, S. 257–263 vol.1
- [Welch und Bishop 2006] WELCH, G. ; BISHOP, G.: An Introduction to the Kalman Filter / Department of Computer Science University of North Carolina at Chapel Hill. URL <http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html>, 2006. – Forschungsbericht. Zuletzt abgerufen am 23.11.2008
- [Wippich 2008a] WIPPICH, M.: *Bildverarbeitungs-basierte Identifikation von Positionsmarken für die Präzisionsnavigation*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2008
- [Wippich 2008b] WIPPICH, M.: *Kolloquiumsvortrag zur Bachelorarbeit Bildverarbeitungs-basierte Identifikation von Positionsmarken für die Präzisionsnavigation*, Hochschule für Angewandte Wissenschaften Hamburg, Kolloquiumsvortrag, 2008
- [Yeh und Kriegman 1995] YEH, E. ; KRIEGMAN, D.J.: Toward selecting and recognizing natural landmarks. In: KRIEGMAN, D.J. (Hrsg.): *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots'* Bd. 1, 1995, S. 47–53 vol.1

Glossar

Augmented Reality (AR) Kombiniert die reale und virtuelle Welt miteinander z.B. durch 3D-Projektionen von Objekten in den Raum, mit denen in Echtzeit interagiert werden kann.

Barcode Balkencode mit dem Informationen in Form von parallelen Balken und Zwischenräumen unterschiedlicher Breite abgebildet werden können. Der Code lässt sich mit optischen Lesegeräten auslesen.

Binarisierung Transformation eines Grauwertbildes in ein Binärbild, welches nur noch zwei Grauwerte (meist 0 und 255 oder 0 und 1) enthält.

Closing Morphologische Operation, die aus einer Dilation und anschließender Erosion besteht.

Dilation Morphologische Operation, die einen Pixel auf dem sich der Bezugspunkt des Strukturelements befindet, im gefilterten Bild auf 1 setzt, wenn irgendein Element des Strukturelements mit denen des unterliegenden Bildes übereinstimmt. Diese Operation führt zu einer Vergrößerung von hellen Bildbereichen.

Diskrete Fouriertransformation In der Bildverarbeitung eine Methode um ein Bild in Sinus- und Kosinusfunktionen zu zerlegen und es so in den n den Frequenzbereich zu transformieren.

Diskrete Kosinustransformation Ähnlich der diskreten Fouriertransformation eine Methode zur Transformation eines Bildes in den Frequenzbereich, wobei hierzu ausschließlich Kosinuskfunktionen verwendet werden.

Erosion Morphologische Operation, die einen Pixel auf dem sich der Bezugspunkt des Strukturelements befindet, im gefilterten Bild auf 1 setzt, wenn alle Elemente des Strukturelements mit denen des unterliegenden Bildes übereinstimmen. Diese Operation führt zu einer Verkleinerung von hellen Bildbereichen.

GPS Das Global Positioning System, ist ein weltweit nutzbares System zur satellitengestützten Positionsbestimmung.

Hough-Transformation Ist ein Verfahren zur Suche nach geometrischen Figuren, z.B. Geraden, in einem Kantenbild.

ID In dieser Arbeit wird die Zeichenfolge als Abkürzung für die (eindeutige) Identifikationsnummer einer Marke verwendet.

Morphologischen Operatoren Klasse von Verfahren in der Bildverarbeitung, die eine Filterung mit Strukturelementen vornehmen, die den hervorzuhebenden oder zu unterdrückenden Formen entsprechen. Hierzu wird das Strukturelement pixelweise über das geschoben.

Opening Morphologische Operation, die aus einer Erosion und anschließender Dilation besteht.

Parity-Bit (deutsch: Paritätsbit) Dient zur Kontrolle der Übertragung einer Anzahl von Bits. Es wird entweder gesetzt, wenn die Anzahl übertragener Bits gerade oder ungerade ist, dies ist Definitionssache.

Partikelfilter Ist der Name von Methoden zur Zustandsschätzung von dynamischen Prozessen auf Basis von Monte-Carlo Simulationen. Anders als durch das einfache Kalman Filter können hiermit auch nicht-lineare Prozesse geschätzt werden.

Photogrammetrie Befasst sich mit der Aufnahme und Auswertung von Messbildern zur Messung im 3D-Raum. Mit ihren Methoden lassen sich z.B. berührungslose Messungen von Objekten durchführen.

Subpixel Ist die Untereinheit von Pixel. Die enthaltene Information muss per Interpolation ermittelt werden, da das Auflösungsvermögen von Kameras auf Pixel beschränkt ist.

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) bzw. §24(4) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 12. Dezember 2008 Matthias Wilms