UNIVERSITÄT
KOBLENZ · LANDAU
Fachbereich 4: Informatik

Erweiterung des SURF-Algorithmus um Farbmerkmale für die Objekterkennung

Diplomarbeit
zur Erlangung des Grades
Diplom-Informatiker
im Studiengang Computervisualistik

vorgelegt von

David Gossow

Betreuer: Dipl.-Inform. Peter Decker, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau
Erstgutachter: Dipl.-Inform. Peter Decker, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau
Zweitgutachter: Prof. Dr.-Ing. Dietrich Paulus, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau

Koblenz, im Dezember 2009

Kurzfassung

Diese Diplomarbeit befasst sich damit, den SURF-Algorithmus zur performanten Extraktion von lokalen Bildmerkmalen aus Graustufenbildern auf Farbbilder zu erweitern.

Dazu werden zuerst verschiedene quelloffene Implementationen mit der Originalimplementation verglichen. Die Implementation mit der größten Ähnlichkeit zum Original wird als Ausgangsbasis genutzt, um verschiedene Erweiterungen zu testen. Dabei werden Verfahren adaptiert, die den SIFT-Algorithmus auf Farbbilder erweitern.

Zur Evaluation der Ergebnisse wird zum Einen die Unterscheidungskraft der Merkmale sowie deren Invarianz gegenüber verschiedenen Bildtransformationen gemessen. Hier werden verschiedene Verfahren einander gegenüber gestellt. Zum Anderen wird auf Basis des entwickelten Algorithmus ein Framework zur Objekterkennung auf einem autonomen Robotersystem entwickelt und dieses evaluiert.

Abstract

In this diploma thesis, an extension of the SURF algorithm, which extracts local features from grayscale images, to color images is presented.

First, various open source implementations are compared to the original implementation. The implementation which shows the largest similarity to the original is used to test various extensions to color images. These extensions are adapted from already existing methods of including color information in the SIFT algorithm.

For evaluation, the distinctiveness of the features as well as the invariance against different image tranformations is measured. Here, different algorithms are compared. In addition, an object recognition framework for an autonomous mobile system is constructed based on the novel algorithm and evaluated.

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Vereinbarung der Arbeitsgruppe für Studien- und Abschlussarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.	ja ☐	nein ☐
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	ja ☐	nein ☐

Koblenz, den 7. Dezember 2009

Danksagung

An dieser Stelle möchte ich mich bei meiner Familie sowie Erika und Christian Janosch bedanken, die mich auf dem Weg zu dieser Diplomarbeit unterstützt haben.

Prof. Paulus danke ich für die Möglichkeit, diese Diplomarbeit in seiner Arbeitsgruppe anzufertigen. Mein besonderer Dank gilt auch meinem Betreuer Peter Decker für die konstante Unterstützung und die vielen Gespräche und Anregungen.

Inhaltsverzeichnis

1 Einleitung **15**

2 Lokale Bildmerkmale **17**
2.1 Motivation 17
2.2 Invarianzeigenschaften von Lokalen Merkmalen 19
2.3 Vorläufer von SURF 20
2.3.1 Die Hessematrix 20
2.3.2 Der Punktdetektor von Harris 21
2.3.3 Erweiterung um Skaleninvarianz 22
2.3.4 Harris-Laplace 25
2.3.5 SIFT 26
2.4 SURF (Speeded Up Robust Features) 28
2.4.1 Integralbilder 29
2.4.2 Detektionsschritt 29
2.4.3 Zuweisung einer normalisierten Orientierung 34
2.4.4 Merkmalsdeskriptor 36
2.5 Objekterkennung durch lokale Merkmale 37
2.5.1 Erstellung der Objektbeschreibung 37
2.5.2 Nearest Neighbour Ratio Matching 38
2.5.3 Hough Clustering 38
2.5.4 Bestimmung der Homographie 39
2.5.5 Kriterium für die Detektion eines Objekts 39

3 Erweiterung des SURF-Algorithmus **41**
3.1 Zugrunde liegendes physikalisches Modell 41
3.2 Untersuchte Farbräume 43
3.2.1 Gaußsches Farbmodell 43
3.2.2 YCrCb 44
3.2.3 IRG 44
3.2.4 RGB-Gegenfarbraum 44
3.3 Photometrische Invarianten 45

3.3.1 Die W-Invariante . . . 46
3.3.2 Die C-Invariante . . . 46
3.3.3 Approximation durch Rechteckfilter . . . 46
3.4 Erweiterung des Detektors . . . 49
3.4.1 Photometrische Invarianz . . . 49
3.4.2 Kombinierte Detektion . . . 50
3.4.3 Kanalweise Detektion . . . 52
3.4.4 Farbverstärkung . . . 52
3.5 Orientierungszuweisung und Deskriptor . . . 54

4 Implementation **55**
4.1 Detektor und Deskriptor . . . 55
4.1.1 Klassenhierarchie . . . 56
4.2 Objekterkennungssystem . . . 58
4.2.1 Wrapper-Klassen für die Merkmalsextraktion . . . 58
4.2.2 Parallelisierung des Deskriptors . . . 59
4.2.3 Verwaltung von Objektdaten . . . 60
4.2.4 Detektion von Objekten in Kamerabildern . . . 62
4.2.5 Module . . . 62
4.2.6 Benutzerschnittstelle . . . 64
4.3 Evaluationsframework . . . 66
4.3.1 Evaluation der Merkmale . . . 66
4.3.2 Evaluation der Objekterkennung . . . 67

5 Evaluation **69**
5.1 Vergleich verschiedener SURF-Implementationen . . . 69
5.1.1 Ähnlichkeit mit der Originalimplementation . . . 70
5.1.2 Datenbasis für die Evaluation . . . 71
5.1.3 Detektor . . . 74
5.1.4 Deskriptor . . . 75
5.1.5 Besonderheiten der verwendeten Implementation . . . 77
5.2 Evaluation der Farbmerkmale . . . 80
5.2.1 Vorgehensweise . . . 85
5.2.2 Dimensionalität des Deskriptors . . . 85
5.2.3 Invarianten im Deskriptor . . . 86
5.2.4 Invarianten im Detektionsschritt . . . 87
5.2.5 Zuweisung der Orientierung . . . 91
5.2.6 Wahl des Farbraums . . . 92
5.2.7 Vergleich mit SURF . . . 93
5.3 Evaluation der Objekterkennung . . . 93
5.4 Laufzeit und Speicherbedarf des Algorithmus . . . 97

6 Zusammenfassung **99**
6.1 Ausblick 102

Tabellenverzeichnis

5.1 Vergleich der Implementationen des Fast Hessian-Detektors 70
5.2 Analyse der ersten beiden Komponenten des OrigSURF-Deskriptors 79
5.3 Laufzeit der Algorithmen zur Merkmalsextraktion 97

Abbildungsverzeichnis

2.1 Schnitte durch den Skalenraum für verschiedene Werte von σ 22
2.2 Schnitt durch den Skalenraum in der x-σ-Richtung 23
2.3 Faltungskerne für gaußsche Ableitungen 1. bis 4. Grades 24
2.4 Skalensignatur von (spur $\boldsymbol{H}_{\text{norm}})^2$ 24
2.5 Maxima von (spur $\boldsymbol{H}_{\text{norm}})^2$ 25
2.6 Berechnung des SIFT-Deskriptors 28
2.7 Verwendung von Integralbildern 29
2.8 Faltungskerne des SURF-Algorithmus 30
2.9 Schnitte durch den approximierten Skalenraum 30
2.10 Schnitt durch den approximierten Skalenraum in x-σ-Richtung . . . 31
2.11 Wert von $det(\overline{\boldsymbol{H}}(x, y, \sigma))$ für verschiedene Werte von σ 31
2.12 Wert von $det(\overline{\boldsymbol{H}}(x, y, \sigma))$ in x- und σ-Richtung 32
2.13 B_{yy} und B_{yy} für $\sigma = 1.2$ und $\sigma = 2.0$ 33
2.14 Filtergrößen bei SURF 34
2.15 Haar Wavelet-Filter in x- und y-Richtung 34
2.16 Zuweisung einer Orientierung bei SURF 35
2.17 Beispiel für Orientierungen von SURF-Schlüsselpunkten 35
2.18 SURF-Deskriptor für verschiedene Bildmuster 36

3.1 Sensitivitätskurven des gaußschen Farbmodells 43
3.2 Rechteckfilter für Ableitungen ersten und zweiten Grades 47
3.3 Photometrisch invariante Blob-Detektionsfunktionen Ω_W und Ω_C . 49
3.4 Merkmalsdetektion mit Farbverstärkung 53

4.1 Klassen zur Berechnung der Farbmerkmale 57
4.2 Wrapper-Klassen für die Merkmalsextraktion 59
4.3 Klassen zur Verwaltung von Objekteigenschaften 60
4.4 Klassen zur Detektion von Objekten in Kamerabildern 61
4.5 Module des Objekterkennungssystems 63
4.6 Sequenzdiagramm Objekterkennung 64
4.7 Grafische Oberfläche des Objekterkennungssystems 65

4.8 Grafische Oberfläche zur Erstellung der Objektdatensätze 66

5.1 Orientierungsfehler bei verschiedenen SURF-Implementationen . . . 71
5.2 Bildserien zur Evaluation der SURF-Implementationen (Teil 1) . . . 72
5.3 Bildserien zur Evaluation der SURF-Implementationen (Teil 2) . . . 73
5.4 Überlappungsbereich zwischen zwei Bildern 74
5.5 Überlappungsfehler zweier Schlüsselpunkte 74
5.6 Vergleich der Implementationen bei Rotation & Zoom 75
5.7 Vergleich der Implementationen bei Unschärfe 76
5.8 Vergleich der Implementationen bei Änderung des Blickwinkels . . . 76
5.9 Vergleich der Implementationen bei abnehmender Helligkeit 77
5.10 Genauigkeit und Trefferquote der SURF-Implementationen 78
5.11 Deskriptorfenster mit und ohne Interpolation 78
5.12 Evaluation des Deskriptors mit und ohne Interpolation 79
5.13 Verwendete Objekte aus der ALOI-Datenbank (Teil 1) 81
5.14 Verwendete Objekte aus der ALOI-Datenbank (Teil 2). 82
5.15 Beispiel für eine Bildserie aus der ALOI-Datenbank 83
5.16 Bildserie “Fields” 83
5.17 Evaluation des Farbdeskriptors mit unterschiedlicher Anzahl von Teilfenstern 86
5.18 Evaluation des Deskriptors mit und ohne Farbinformationen 87
5.19 Reproduzierbarkeit des Farbdetektors 88
5.20 Evaluation des Farbdetektors bei Rotation & Zoom 89
5.21 Evaluation des Farbdetektors bei Unschärfe 89
5.22 Evaluation des Farbdetektors bei Änderung des Blickwinkels 90
5.23 Evaluation des Farbdetektors bei abnehmender Helligkeit 90
5.24 Evaluation des Farbdeskriptors für verschiedene Detektionsverfahren 91
5.25 Fehler in der Orientierungszuweisung auf Farbbildern 92
5.26 Evaluation des Farbdeskriptors für verschiedene Verfahren zur Zuweisung einer Orientierung 93
5.27 Evaluation des Farbdeskriptors bei Verwendung unterschiedlicher Farbräume 94
5.28 Performanz des Farbdeskriptors im Vergleich zu SURF 94
5.29 Synthetisches Testbild mit inhomogenem Hintergrund 95
5.30 Mittlere Genauigkeit und Trefferquote des Objekterkennungssystems auf den Testbildern der ALOI-Datenbank 96

Kapitel 1

Einleitung

Lokale Merkmale stellen ein Werkzeug zur Lösung von Problemstellungen des künstlichen Sehens dar. Das umfasst beispielsweise Objekterkennung, Tiefenrekonstruktion aus Stereobildern, Kamerakalibrierung oder Selbstlokalisation. Lokale Merkmale erlauben es, markante Regionen in Bilddaten zu finden und ihre Eigenschaften in kompakter Form zu beschreiben. Anhand dieser Eigenschaften können Korrespondenzen zwischen verschiedenen Bildern der selben Szene oder des selben Objekts bestimmt werden.

In den vergangenen Jahren haben Verfahren an Bedeutung gewonnen, die invariant gegenüber geometrischen Transformationen sind [TM08]. Dies betrifft sowohl die Lokalisierung der Merkmale im Bild als auch die Eigenschaften, die zu ihrer Beschreibung verwendet werden.

Einige davon [MS02, MCUP02]) sind invariant gegen affine Transformationen. Diese umfassen alle Kombinationen aus Verschiebung, Drehung, Skalierung und Scherung. Damit werden alle Verzerrungen abgedeckt, die durch die orthografische Projektion von planaren Flächen entstehen. Perspektivische Verzerrung und nicht planare Oberflächen werden außen vor gelassen. Andere Verfahren, beispielsweise SIFT [Low04] und SURF [BTVG06, BETG08], sind nur gegen Translation, Skalierung und Rotation in der Bildebene invariant, erzielen jedoch vergleichbare Ergebnisse in der Objekterkennung [MP07].

Der Anwendungsfokus dieser Arbeit liegt auf einer Applikation für robuste Objekterkennung in der Robotik. In diesem Zusammenhang spielt die Laufzeit eine wichtige Rolle bei der Bewertung eines Algorithmus, da der Roboter in der Lage sein soll, auf Veränderungen seiner Umwelt zu reagieren. Einen speziell auf diese Eigenschaft optimierten Algorithmus stellt SURF dar. Durch Verwendung einer speziellen Repräsentation von Bildern ist er in der Lage, ähnliche Ergebnisse wie SIFT zu erzielen, während seine Laufzeit um mehr als 60% geringer ist [BTVG06, BETG08].

Alle bisher genannten Verfahren arbeiten auf Graustufenbildern. Verschiedene Verfahren zur Hinzunahme von Farbinformationen zum SIFT-Algorithmus wurden vorgeschlagen [BG09, AHF06, BZM06, WS06]. In Vergleichsstudien [BG09, SGS08b] wurde gezeigt, dass dadurch eine Verbesserung der Erkennungsleistung in der Objekterkennung und -klassifikation erzielt werden kann. Eine wichtige Rolle spielen dabei zusätzliche Invarianzen gegenüber photometrischen Transformationen [BG09]. Das sind Änderungen der Intensitätswerte des Bildes, welche durch die Beleuchtungssituation, z.B. durch Änderung der Positionen der Lichtquelle oder des abgebildeten Objekts, hervorgerufen werden.

Ziel dieser Arbeit ist, verschiedene Erweiterungen des SIFT-Algorithmus aus der Literatur auf SURF zu übertragen. Da die Originalimplementation nicht quelloffen ist, sollen zunächst eine Reihe quelloffener Implementationen evaluiert werden. Die am besten geeignete soll schließlich als Ausgangsbasis für die Implementation der neuen Verfahren dienen. Anschließend soll ein Framework zur Evaluation der Verfahren implementiert werden. Dieses soll sich an den etablierten Testumgebungen aus der Literatur orientieren, um die Vergleichbarkeit der Ergebnisse zu gewährleisten. Zudem soll eine Applikation zur Objekterkennung entwickelt und in eine Softwarearchitektur für Robotik-Anwendungen integriert werden. Anhand dieser sollen die Verfahren ebenfalls evaluiert werden.

Die Arbeit ist wie folgt aufgebaut: In Kapitel 2 werden die theoretischen Grundlagen lokaler Merkmale beschrieben. Dies beinhaltet eine Darstellung des SURF-Algorithmus und seiner Vorläufer sowie die Beschreibung eines Verfahrens zur Objekterkennung, welches auf lokalen Merkmalen basiert. In Kapitel 3 werden die verschiedenen Erweiterung des SURF-Algorithmus durch Farbmerkmale erarbeitet.

Kapitel 4 liefert eine Übersicht über die Implementation des Algorithmus zur Berechnung von lokalen Merkmalen, der darauf basierten Objekterkennungsapplikation, sowie des Frameworks zur Evaluation. Kapitel 5 enthält eine Evaluation von bisherigen Implementationen von SURF, aus denen die Ausgangsbasis für die Implementation der Farbmerkmale ausgewählt wird. Zudem wird der neu entwickelte Algorithmus anhand verschiedener Kriterien evaluiert und mit SURF verglichen. In Kapitel 6 werden die erzielten Ergebnisse zusammengefasst und ein Ausblick auf sich daraus ergebende weiterführende Fragestellungen gegeben.

Kapitel 2

Lokale Bildmerkmale

Lokale Bildmerkmale bezeichnen Bildmuster, die sich von ihrer direkten Umgebung unterscheiden und üblicherweise mit einer Änderung von einer oder mehreren Bildeigenschaften assoziiert sind [TM08]. Sie können beispielsweise durch Punkte, Kanten oder kleine Bildregionen definiert sein. Das Auffinden solcher Merkmale wird als Detektion bezeichnet. Aus um die Merkmale zentrierten Regionen werden üblicherweise anschließend Bildeigenschaften berechnet, die den sogenannten Deskriptor bilden. Dieser dient dazu, das gleiche Merkmal in unterschiedlichen Bildern zu identifizieren. Die Kombination aus Lokalisation und Deskriptor wird im Folgenden auch als Schlüsselpunkt bezeichnet.

Das folgende Kapitel ist wie folgt aufgebaut: Zuerst wird ein Überblick über die Eigenschaften von lokalen Merkmalen im Vergleich zu anderen Bildmerkmalen gegeben. Anschließend werden die Vorläufer des SURF-Algorithmus sowie die Grundlagen der Theorie des Skalenraums erläutert. Schließlich wird der SURF-Algorithmus selbst beschrieben. Dies umfasst die verschiedenen Teilschritte des Algorithmus sowie die zugrunde liegende Datenstruktur der Integralbilder, welche eine beschleunigte Berechnung der verwendeten Bildmerkmale erlauben. Schließlich wird beschrieben, wie die Erkennung von Objekten aufgrund von lokalen Merkmalen möglich ist.

2.1 Motivation

In [TM08] wird eine Vergleichstudie über aktuelle Verfahren zur Extraktion von Bildmerkmalen angestellt. Der folgende Abschnitt folgt in Teilen der dort entwickelten Argumentationskette.

Lokale Bildmerkmale können je nach Kontext eine eigene Semantik besitzen. Beispielsweise können Kanten in der Aufnahme einer Netzhaut mit Blutgefäßen korrespondieren. Im Kontext der Objekterkennung stellen sie jedoch ein Werkzeug

dar, um Bilder durch eine begrenzte Anzahl von Regionen und deren Eigenschaften zu repräsentieren, welche leichter zu vergleichen sind als die zugrunde liegenden Bilddaten.

Will man das Bild eines Objekts in einem zweiten Bild wiederfinden, ist ein Ansatz eine vollständige Suche über alle möglichen Transformationen des Objekts. Unter der Annahme, dass die beiden Bilder nur durch eine Translation, Rotation und Skalierung aufeinander abgebildet werden können, besitzt der resultierende Suchraum jedoch bereits vier Dimensionen. In [VJ01] wird eine besonders effiziente Implementation der elementaren Vergleichsoperation für zwei Bilder vorgeschlagen. Mit dieser ist eine vollständige Suche in Echtzeit möglich, jedoch ohne Berücksichtigung von Rotation oder partieller Verdeckung.

Eine Alternative stellen globale Bildmerkmale, z.B. Farbhistogramme, dar, die statistische Eigenschaften des gesamten Bildes beschreiben. Diese unterscheiden jedoch nicht zwischen dem eigentlichen Objekt und dem Hintergrund, vor dem es sich befindet. Sie eignen sich daher vor Allem dann, wenn das Objekt einen großen Teil des Bildes bedeckt, bzw. die Region, in der sich das Objekt befindet, bereits manuell isoliert wurde.

Dies gilt nicht für Verfahren zur Bildsegmentierung. Hierbei wird das Bild, meist anhand von Farb- oder Texturmerkmalen, in Regionen unterteilt, wobei eine Region idealerweise ein Objekt oder einen Teil davon repräsentiert. Unter unkontrollierten Bedingungen ist die Segmentierung an sich jedoch ein Problem, welches im Allgemeinen nicht ohne a-priori-Wissen über den Bildinhalt gelöst werden kann.

Die Notwendigkeit einer Segmentierung kann durch die Auswahl von lokalen Merkmalen umgangen werden, die nur eine sehr begrenzte Region des Bildes beschreiben. Was die Merkmale repräsentieren, ist dabei zweitrangig. Wichtig ist, dass deren Position im Bild unter einer möglichst großen Klasse von möglichen Transformationen des Eingabebildes stabil bleibt. Für die Regionen um die lokalen Merkmale werden wiederum globale Eigenschaften, die Deskriptoren, berechnet. Diese müssen genug Unterscheidungskraft besitzen, um das lokale Merkmal in einer möglichst großen Menge von Merkmalen aus transformierten Bildern identifizieren zu können. Zudem sollte der Vergleich zweier Deskriptoren möglichst effizient möglich sein, was von deren Dimensionalität sowie dem Vergleichskriterium abhängt.

In [TM08] werden folgende Eigenschaften eines idealen lokalen Merkmals herausgestellt:

- **Reproduzierbarkeit (Repeatability):** In zwei Bildern der gleichen Szene oder des gleichen Objekts, die unter verschiedenen Bedingungen aufgenommen wurden, sollten möglichst viele lokale Bildmerkmale in der gemeinsamen Bildregion in beiden Bildern vorhanden sein.

- **Unterscheidungskraft (Distinctiveness):** Die Bildregionen um die lokalen Merkmale sollten genug Variationen enthalten, um sie voneinander zu unterscheiden.

- **Lokalität (Locality):** Die Merkmale sollten lokal begrenzt sein, um die Wahrscheinlichkeit von Verdeckung zu verringern und einfaches approximiertes Modell der geometrischen und photometrischen Deformationen zwischen zwei Bildern zu erlauben, die unter unterschiedlichen Bedingungen aufgenommen wurden, z.B. basierend auf der Annahme von lokaler Planarität der abgebildeten Oberfläche.

- **Quantität (Quantity):** Die Anzahl der gefundenen Merkmale sollte ausreichend groß sein, so dass auch auf kleinen Objekten genügend Merkmale gefunden werden. Die Dichte der Merkmale sollte den Informationsgehalt des Bildes entsprechen, um eine kompakte Repräsentation dessen Inhalts zu ermöglichen.

- **Genauigkeit (Accuracy):** Gefundene Merkmale sollten exakt lokalisiert sein in Bezug auf Position, Skala und möglicherweise Form.

- **Effizienz (Efficiency):** Die Detektion der Merkmale sollte für zeitkritische Anwendungen geeignet sein.

2.2 Invarianzeigenschaften von Lokalen Merkmalen

Eine Eigenschaft von lokalen Bildmerkmalen ist ihre Invarianz gegenüber verschiedenen Transformationen des Eingabebildes, z.B. Translation, Rotation und Skalierung. Im betrachteten Kontext der Objekterkennung kommt dieser eine besondere Bedeutug zu, da die relative Position von Kamera und Objekt nicht im Voraus bekannt ist. Eine Funktion $f(x)$ ist invariant gegenüber einer bestimmten Transformation $t(x)$, wenn t das Ergebnis nicht beeinflusst [TM08]:

$$f(\boldsymbol{x}) = f(t(\boldsymbol{x}))$$

Um diese Invarianz zu erreichen, können Bildeigenschaften, die mit den Transformationen kovariant sind, zur Normalisierung der lokalen Bildregion verwendet werden. Eine Funktion $f_k(x)$ ist kovariant mit einer bestimmten Transformation $t(x)$, wenn die Reihenfolge der Operationen austauschbar ist:

$$f(t(\boldsymbol{x})) = t(f(\boldsymbol{x}))$$

Beispielsweise ist die Gradientenrichtung in einem Bild kovariant mit Rotationen, was zur Konstruktion einer entsprechend rotierten Bildregion genutzt werden kann. Aus der normalisierten Bildregion können dann Merkmale extrahiert werden, die ansonsten gegen die jeweilige Transformation nicht invariant sind.

2.3 Vorläufer von SURF

In [TM08] wird die Entwicklung des Konzepts von lokalen Bildmerkmalen bis zu einer Veröffentlichung von Fred Attneave aus dem Jahr 1954 [Att54] zurückverfolgt. Aus psychologischen Experimenten mit Strichzeichnungen schließt Atneave, dass die Information über die Form eines Gegenstandes an Punkten konzentriert ist, an denen die Krümmung einer Linie ein lokales Maximum erreicht.

Diese Idee mündete später Verfahren zur Analyse von Konturbildern, beispielsweise von technischen Zeichnungen. Diese Verfahren weisen jedoch eine zu geringe Stabilität, Robustheit und Distinktivität auf [TM08] und sind deshalb zur Objekterkennung unter unkontrollierten Bedingungen nicht geeignet. Stattdessen haben Detektoren an Bedeutung gewonnen, die direkt auf den Bildintensitäten arbeiten.

Translations- und Rotationskovarianz wurde bereits mit frühen intensitätsbasierten Detektoren erreicht. Hierbei werden Maxima und Minima von Funktionen, die auf dem Bildsignal definiert sind, betrachtet. Prominente Beispiele sind die Spur und Determinante der Hesse-Matrix sowie die Bewertungsfunktion von Harris [HS88].

2.3.1 Die Hessematrix

Die Verwendung der Hessematrix wurde bereits in [Bea78] vorgeschlagen. Diese enthält die zweiten partiellen Ableitungen der Bildintensitäten in x- und y-Richtung und beschreibt somit deren Krümmung:

$$\boldsymbol{H} = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \tag{2.1}$$

Die partiellen Ableitungen können durch Faltung des Bildes mit geeigneten Masken berechnet werden. Von $\boldsymbol{H}$ werden folgende (rotationsinvariante) Kennwerte betrachtet:

$$\text{spur}\, \boldsymbol{H} = I_{xx} + I_{yy} \tag{2.2}$$

$$\det \boldsymbol{H} = I_{xx} I_{yy} + I_{xy}^2 \tag{2.3}$$

Die Maxima von spur $\boldsymbol{H}(x, y)$ und $\det \boldsymbol{H}(x, y)$ entsprechen Blob-ähnlichen Strukturen im Bild, deren Ausdehnung der Größe des Filters für die partielle Ableitung entspricht. Als Blob ("Klecks") wird eine Bildregion bezeichnet, deren Intensität sich von ihrer Umgebung unterscheidet, z.B. einem hellen Fleck auf dunklem Grund.

Die Positionen der so gefundenen Maxima sind kovariant mit Translationen und Rotationen des Eingabebildes. Ein Nachteil von spur $\boldsymbol{H}$ ist allerdings, dass diese auch Maxima auf länglichen Bildstrukturen wie Konturen und Kanten besitzen, bei denen sich das Signal nur senkrecht zur Richtung der Struktur ändert. Diese Maxima besitzen weniger Stabilität, da eine geringe Änderung der Bildintensitäten entlang der Richtung der Struktur die Position des Maximums stark verändern kann [Mik02].

2.3.2 Der Punktdetektor von Harris

Beim Verfahren von Harris [HS88] wird eine Funktion auf den Bildintensitäten definiert, mit der sich Kanten, Ecken und homogene Flächen unterscheiden lassen. Er stellt eine Weiterentwicklung des Ansatzes von Moravec [Mor80] dar. Dazu wird zunächst auf Basis der Momente zweiten Grades die Matrix $\boldsymbol{A}$ bestimmt:

$$\boldsymbol{A} = w \circledast \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{2.4}$$

I_x und I_y bezeichnen diskrete Ableitungen des Bildsignals in x- und y-Richtung. Sie werden durch Faltung des Bildes mit den Masken $(-1, 0, 1)$ und $(-1, 0, 1)^{\mathrm{T}}$ berechnet. Als Gewichtungsfunktion $w(x, y)$ wird eine isotropische Gaußfunktion $G(x, y, \sigma)$ vorgeschlagen:

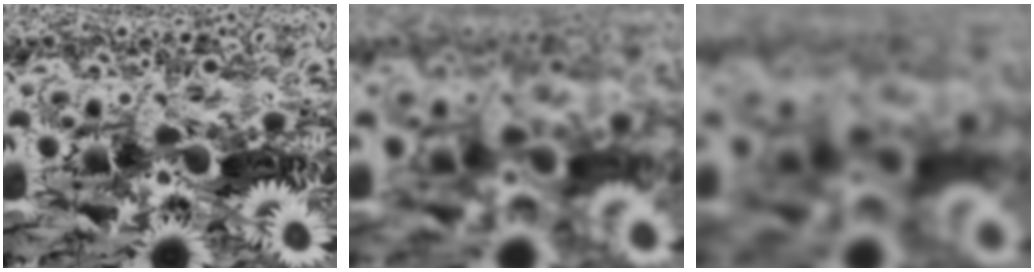
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)} \tag{2.5}$$

$\boldsymbol{A}(x, y)$ enthält dann die Gauß-gewichteten Produkte von Ableitungen in einer Nachbarschaft von (x, y). Auf $\boldsymbol{A}$ wird nun die Bewertungsfunktion μ definiert, die für Ecken positive und für Kanten negative Werte liefert:

$$\mu(x, y) = \det(\boldsymbol{A}(x, y)) - \alpha \,\mathrm{spur}^2(\boldsymbol{A}(x, y))$$

α legt die Sensitivtät des Detektors gegenüber Ecken fest und muss experimentell ermittelt werden. $\det(A(x, y))$ und $\mathrm{spur}(A(x, y))$ sind invariant gegenüber Rotationen. Durch Auswahl von lokalen Maxima dieser Funktion werden damit Schlüsselpunkte erzeugt, deren Positionen kovariant mit Translationen und Rotationen des Eingabebildes sind.

(a) Originalbild ($\sigma = 0.0$) **(b)** $\sigma = 1.2$ **(c)** $\sigma = 2.0$



(d) $\sigma = 3.6$ **(e)** $\sigma = 6.8$ **(f)** $\sigma = 13.2$

Abbildung 2.1: Schnitte durch den Skalenraum für verschiedene Werte von σ. Zu erkennen ist, wie feinere Bildstrukturen (weiter entfernte Sonnenblumen) mit zunehmendem σ verschwinden und die längerwelligen Anteile des Bildsignals dominieren. Die Bildauflösung beträgt 637×502 Pixel. Quelle des Originalbildes: `http://upload.wikimedia.org/wikipedia/commons/e/e3/Field_of_sunflowers.JPG`

2.3.3 Erweiterung um Skaleninvarianz

Um Skalenkovarianz zu erreichen, muss auf die Theorie des Skalenraums [Lin94] zurückgegriffen werden. Der Skalenraum ist definiert als die Erweiterung des Bildraums um eine Dimension $\sigma \in \mathbb{R}$. Diese definiert die Standardabweichung eines gaußschen Glättungskerns G, mit dem das Bild gefaltet wird (vgl. Gleichung 2.5):

$$L(\sigma) = G(\sigma) \circledast I$$

wobei $\circledast$ die Faltung in x- und y-Richtung bezeichnet. In Abbildung 2.1 ist der Aufbau des Skalenraums anhand eines Beispiels illustriert.

Lindeberg hat diese zuerst 1998 zur automatischen Bestimmung einer charakteristischen Skala für lokale Merkmale eingesetzt, die kovariant mit Skalierungen des Eingabebildes ist [Lin98]. Lindeberg formuliert dort das Prinzip der Skalenauswahl wie folgt:

"Falls keine anderweitigen Indizien dagegensprechen, ist anzunehmen, dass eine Skala, bei der eine (möglicherweise nicht-lineare) Kombination von normalisierten

Abbildung 2.2: Schnitt durch den Skalenraum in x- und σ-Richtung für die zentrale Bildzeile in Abbildung 2.1

Ableitungen ein lokales Maximum im Skalenraum besitzt, eine charakteristische Länge von einer korrespondierenden Struktur in den Eingabedaten reflektiert."

Lindeberg untersucht in diesem Kontext auch Maße, die auf der Hessematrix $\boldsymbol{H}(x, y, \sigma)$ basieren (vgl. Gleichung 2.1):

$$\boldsymbol{H}(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix} \tag{2.6}$$

L_{xx} bezeichnet hier die Faltung des Bildes mit der zweiten partiellen Ableitung der Gaußfunktion G_{xx} (L_{yy} und L_{xy} analog, siehe Abbildung 2.3):

$$L_{xx}(x, y, \sigma) = G_{xx}(\sigma) \circledast I(x, y)$$
$$G_{xx} = \frac{\delta^2 G}{\delta x^2}$$

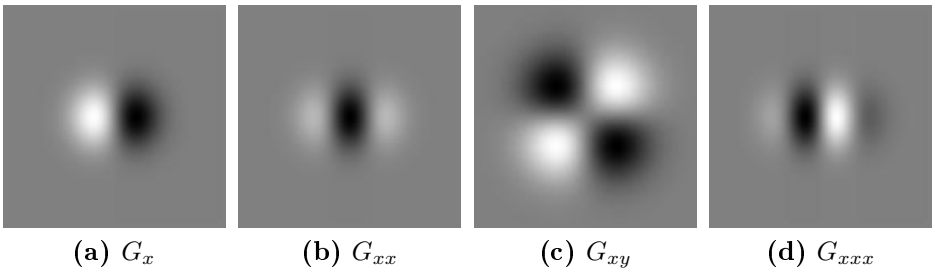
Lindeberg experimentiert unter anderem mit der Skalen-normalisierten Spur und Determinante der Hesse-Matrix, um Blobs zu detektieren (vgl. Gleichung 2.3 und 2.2):

$$\text{spur}\, \boldsymbol{H}_{\text{norm}}(x, y, \sigma) = \sigma^2 (L_{xx}(x, y, \sigma) + L_{yy}(x, y, \sigma)) \tag{2.7}$$

$$\det \boldsymbol{H}_{\text{norm}}(x, y, \sigma) = \sigma^4 (L_{xx}(x, y, \sigma) L_{yy}(x, y, \sigma) + L_{xy}(x, y, \sigma)^2) \tag{2.8}$$

Die Normalisierung mit dem Faktor σ^2 bzw. σ^4 ist nötig, da der Betrag der beiden Werte sonst mit zunehmender Skala abnehmen würde.

Abbildung 2.4 zeigt den Verlauf von $(\text{spur}\, \boldsymbol{H}_{\text{norm}})^2$ entlang der σ-Achse in der Mitte von zwei typischen blobartigen Bildstrukturen mit unterschiedlicher Ausdehnung. Zu erkennen ist, dass sich das Maximum von $(\text{spur}\, \boldsymbol{H}_{\text{norm}})^2$ kovariant mit einer Skalierung des Eingabebildes verhält.



(a) G_x **(b)** G_{xx} **(c)** G_{xy} **(d)** G_{xxx}

(e) G_{xxy} **(f)** G_{xxxx} **(g)** G_{xxxy} **(h)** G_{xxyy}

Abbildung 2.3: Faltungskerne für gaußsche Ableitungen 1. bis 4. Grades. G_{yy}, G_{yyy}, G_{yyyx}, G_{yyyy} und G_{yyyx} sind analog zu G_{xx}, G_{xxx}, G_{xxy}, G_{xxxx} und G_{xxxy}. Quelle: [Mik02]

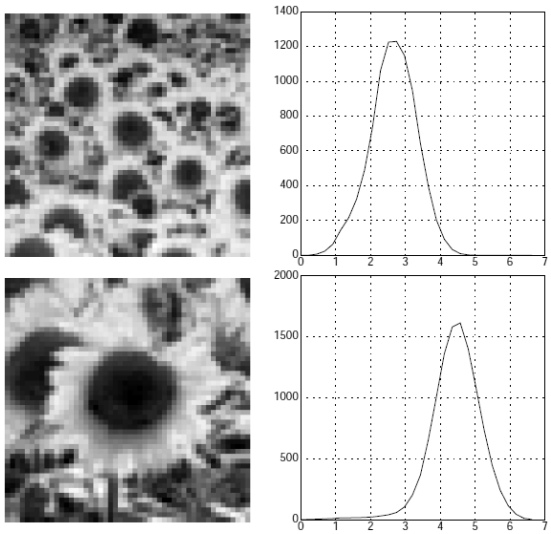
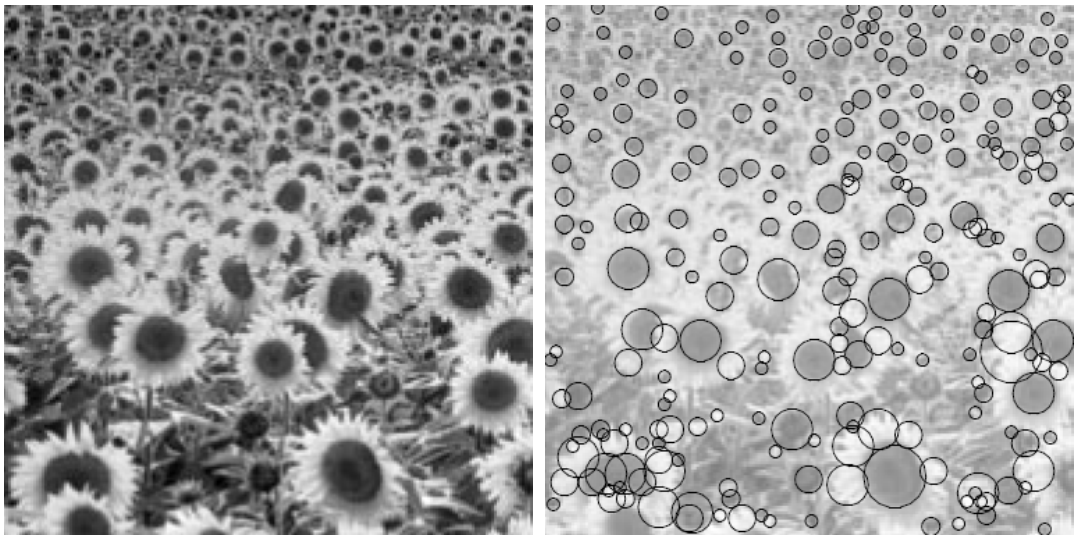


Abbildung 2.4: Ausschnitt aus einem Grauwertbild von Sonnenblumen und dazugehörige Signaturen von $(\text{spur}\,\boldsymbol{H}_{\text{norm}})^2$ entlang der σ-Achse. Die Signaturen wurden in der Mitte des Bildes berechnet. Die x-Achse der Graphen zeigt die effektive Skala $\tau \approx log(\sigma^2)$, die y-Achse ist linear skaliert. Quelle: [Lin98]



(a) Originalbild **(b)** Maxima von (spur $\boldsymbol{H}_{\text{norm}})^2$

Abbildung 2.5: Bild von Sonnenblumen und die 250 Betragsgrößten Maxima (x, y_i, σ_i) von (spur $\boldsymbol{H}_{\text{norm}})^2$. Die Kreise repräsentieren die Maxima, ihr Radius entspricht dabei σ_i. Quelle: [Lin98]

2.3.4 Harris-Laplace

Die Spur der Hesse-Matrix entspricht dem zweidimensionalen Laplace-Operator und wird daher in dieser Variante auch als *Laplacian of Gaussian (LoG)* bezeichnet. In [Mik02, MS04] wird eine Kombination aus Laplace-Operator und einer adaptierten Version des Harris-Maßes verwendet (genannt Harris-Laplace). Die Skalen-normalisierte Definition von $\boldsymbol{A}$ lautet:

$$\boldsymbol{A}(x, y, \sigma_I, \sigma_D) = \sigma_D^2 G(\sigma_I) \circledast \begin{bmatrix} L_x^2(x, y, \sigma_D) & L_x I_y(x, y, \sigma_D) \\ L_x I_y(x, y, \sigma_D) & L_y^2(x, y, \sigma_D) \end{bmatrix}$$

Da sowohl für die Gewichtung als auch zur Glättung eine Gaußfunktion verwendet wird, muss zwischen der Integrations-Skala σ_I und der Ableitungs-Skala σ_D unterschieden werden. Das Skalen-adaptierte Harrismaß ist dann wie folgt definiert:

$$\mu(x, y, \sigma_I, \sigma_D) = \det(\boldsymbol{A}(x, y, \sigma_I, \sigma_D)) - \alpha \,\text{spur}^2(\boldsymbol{A}(x, y, \sigma_I, \sigma_D))$$

Die Detektion von Maxima in x- und y-Richtung erfolgt zunächst durch μ für einige wenige Skalen $\sigma_n = \xi^n \cdot \xi_0$ mit $\xi = 1.4$. Anschließend wird überprüft, ob die gefundenen Stellen auch ein Maximum von spur $\boldsymbol{H}$ in σ-Richtung sind und im negativen Fall verworfen.

Ein alternativer Algorithmus wird ebenfalls angegeben. Er besteht darin, für alle im ersten Schritt gefundenen Maxima iterativ die nächstgelegene Stelle $(x, y, \sigma)^{\mathrm{T}}$ im Skalenraum zu suchen, bei denen μ ein Maximum in x- und y-Richtung und spur $\boldsymbol{H}$ ein Maximum in σ-Richtung besitzt. Um die Genauigkeit der Lokalisierung im Skalenraum zu erhöhen, wird im iterativen Teil $\xi = 1.1$ verwendet. Bei diesem Verfahren kann das selbe Maximum mehrmals gefunden werden, weshalb die so generierten Schlüsselpunkte auf Duplikate geprüft werden müssen.

In der selben Veröffentlichung entwickeln die Autoren einen Detektor (Harris-Affine), der invariant gegenüber affinen Transformationen ist. Dies geschieht, indem nach der Lokalisierung eines Maximums im Skalenraums die affinen Parameter anhand von $\boldsymbol{A}$ bestimmt werden. Die Lokalisation mit diesem Parametern wird anschließend wiederholt und die affinen Parameter erneut bestimmt. Dies wird so lange wiederholt, bis das Verfahren konvergiert.

Aus der normalisierten Bildregion eines Schlüsselpunkts an der Stelle $\boldsymbol{x}_0 = (x_0, y_0, \sigma_0)^{\mathrm{T}}$ werden Ableitungen bis zum 4. Grad. (siehe Abbildung 2.3) berechnet. Daraus resultiert ein 12-dimensionaler Deskriptor. Die Ableitungen werden anschließend mit dem Verfahren aus [FA91] anhand der mit einer Gaußfunktion gewichteten gemittelten Gradientenrichtung θ_G normalisiert, um Rotationsinvarianz zu erreichen [Mik02]:

$$\theta_G = \theta(\boldsymbol{x}_0) - \frac{\sum_{x',y'} G(x', y', \sigma_0/3)(\theta(\boldsymbol{x}_0) - \theta(x_0 + x', y_0 + y', \sigma_0))}{\sum_{x',y'} G(x', y', \sigma_0/3)}$$

$$\theta(x, y, \sigma) = \arctan\left(\frac{\log_y(x, y_0)}{\log_y(x, y_0)}\right)$$

2.3.5 SIFT

Der Laplace-Operator wird auch vom SIFT-Algorithmus [Low04], einem direkten Vorläufer von SURF, verwendet. Dort wird er allerdings durch die Differenz zweier Gauß-geglätteter Bilder approximiert, wobei sich deren Skalen um einen konstanten Faktor k unterscheiden:

$$\text{spur } \boldsymbol{H}_{\text{norm}} = \sigma^2 \nabla^2 G \approx D = \frac{L(k\sigma) - L(\sigma)}{k - 1}$$

$D(x, y, \sigma)$ wird analog zum LoG als Difference of Gaussian (DoG) bezeichnet. Lowe entwirft hierfür eine spezielle Datenstruktur, genannt Bildpyramide, mit der sich D besonders effektiv berechnen lässt.

k ist frei wählbar und entscheidet, wie dicht der Skalenraum abgetastet wird. Je größer k ist, desto mehr Extrema können nicht detektiert werden. Dies ist allerdings nur für nahe beieinander liegende Extrema der Fall. Da diese jedoch eine geringere

Robustheit gegenüber Bildstörungen haben, wird der durch diese Diskretisierung eingeführte Fehler begrenzt.

Die Lokalisierung der Extrema erfolgt auf Subskalen- und Subpixelebene. Dafür wird die Taylor-Reihe zweiten Grades von D mit dem gefundenen Extremum $\boldsymbol{x}_0 = (x_0, y_0, \sigma_0)^{\mathrm{T}}$ als Entwicklungspunkt berechnet:

$$T(\boldsymbol{x}) = D(\boldsymbol{x}_0) + (\boldsymbol{x} - \boldsymbol{x}_0)^{\mathrm{T}} \cdot \nabla D(\boldsymbol{x}_0) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^{\mathrm{T}} \cdot \boldsymbol{H}_D(\boldsymbol{x}_0) \cdot (\boldsymbol{x} - \boldsymbol{x}_0)$$

∇D und $\boldsymbol{H}_D$ bezeichnen hier Gradienten und Hessematrix von D:

$$\nabla D = \begin{pmatrix} D_x \\ D_y \\ D_\sigma \end{pmatrix}$$

$$\boldsymbol{H}_D = \begin{bmatrix} D_{xx} & D_{xy} & D_{x\sigma} \\ D_{xy} & D_{yy} & D_{y\sigma} \\ D_{x\sigma} & D_{y\sigma} & D_{\sigma\sigma} \end{bmatrix}$$

Zu beachten ist hier der Unterschied zur von Lindeberg verwendeten Hessematrix $\boldsymbol{H}(x, y, \sigma)$, die die partiellen Ableitungen von L in x- und y-Richtung enthält.

Die Ableitungen von D werden durch Differenzen der bereits berechneten Werte von D in der Umgebung von $\boldsymbol{x}_0$ approximiert. Das Extremum $\hat{\boldsymbol{x}}_0$ der so konstruierten quadratischen Funktion kann analytisch bestimmt werden, indem man ihre Ableitung mit Null gleichsetzt. Dadurch erhält man:

$$\hat{\boldsymbol{x}}_0 = -\boldsymbol{H}_D^{-1}(\boldsymbol{x}_0) \cdot \nabla D(\boldsymbol{x}_0)$$

In [LB02] wurde dieses Verfahren bereits eingesetzt und gezeigt, dass die Reproduzierbarkeit der so gefundenen Extrema höher ist als ohne Interpolation.

Anschließend werden Extrema verworfen, die sich an Stellen mit geringem Bildkontrast befinden und damit instabil sind. Das Kriterium hierfür ist ein geringer Wert von T am gefundenen Extremum ($T(\hat{\boldsymbol{x}}_0) < 0.03$). Zuletzt werden Extrema verworfen, die entlang von Kanten lokalisiert und damit ebenfalls instabil sind (vgl. 2.3.1). Diese zeichnen sich dadurch aus, dass D an dem gefundenen Extremum in eine Richtung eine große Krümmung aufweist, während senkrecht dazu die Krümmung sehr gering ist. Dies wird überprüft, indem das Verhältnis der Eigenwerte von $\boldsymbol{H}_D(\sigma)$ für ein festes σ berechnet wird. Unterscheiden sich die Eigenwerte sehr stark, wird das Extremum verworfen.

Die Zuweisung einer Orientierung θ erfolgt durch ein Histogrammverfahren. Dabei werden in einem Fenster um den Schlüsselpunkt die Gradienten von L berechnet. Deren Orientierungen werden mit ihrer Länge und einer um den Schlüsselpunkt zentrierten Gaußfunktion gewichtet und in ein Histogramm eingetragen.

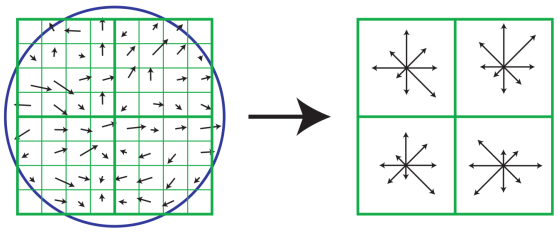


Abbildung 2.6: Berechnung des SIFT-Deskriptors. Für jedes der Teilfenster (links) wird ein Orientierungshistogramm (rechts) berechnet. Der eigentliche SIFT-Deskriptor betrachtet 4×4 Teilfenster, während hier nur 2×2 Teilfenster dargestellt sind. Quelle: [Low04]

In diesem Histogramm wird das Maximum bestimmt und anschließend durch eine Parabel interpoliert, die an die benachbarten Histogrammwerte angepasst wird. Enthält das Histogramm mehrere ähnlich starke Maxima, werden mehrere Schlüsselpunkte mit den jeweiligen Orientierungen erzeugt.

Der Deskriptor wird aus einem um θ gedrehten Fenster um den Schlüsselpunkt extrahiert. Dafür werden die gaußgewichteten, rotationsnormalisierten Gradienten in ein Gitter aus 4×4 Teilregionen eingeordnet (vgl. Abbildung 2.6). Für jedes dieser Teilfenster wird nun ein Histogramm der gewichteten Gradientenorientierungen mit 8 Einträgen erstellt, was einen Deskriptor der Größe $4 \times 4 \times 8 = 128$ ergibt. Um Randeffekte zu vermeiden, wird zwischen den Teilregionen des Deskriptorfensters und Einträgen der Gradientenhistogramme trilinear interpoliert.

2.4 SURF (Speeded Up Robust Features)

SURF ist analog zu Harris-Laplace und SIFT ein Verfahren zur skalen- und rotationsinvarianten Detektion und Deskription von lokalen Merkmalen und baut auf diesen auf. Die Besonderheit des SURF-Algorithmus [BTVG06, BETG08] ist, dass hierfür Integralbilder verwendet werden.

Diese Art der Repräsentation von Bilddaten wurde bereits 1984 im Kontext der Computergrafik als effiziente Methode zum Zeichnen von Texturen vorgeschlagen [Cro84]. In [VJ01] wird es zur Gesichtsdetektion eingesetzt, wobei die benötigte Rechenzeit gegenüber anderen Verfahren signifikant geringer ist. In [GGB06] wird eine Variante von SIFT vorgestellt, die mit Integralbildern arbeitet, jedoch eine geringere Genauigkeit gegenüber der Originalimplementation aufweist. Ziel von SURF ist es, eine Geschwindigkeitsverbesserung gegenüber vorangegangenen Verfahren bei gleichbleibender Genauigkeit zu erzielen.

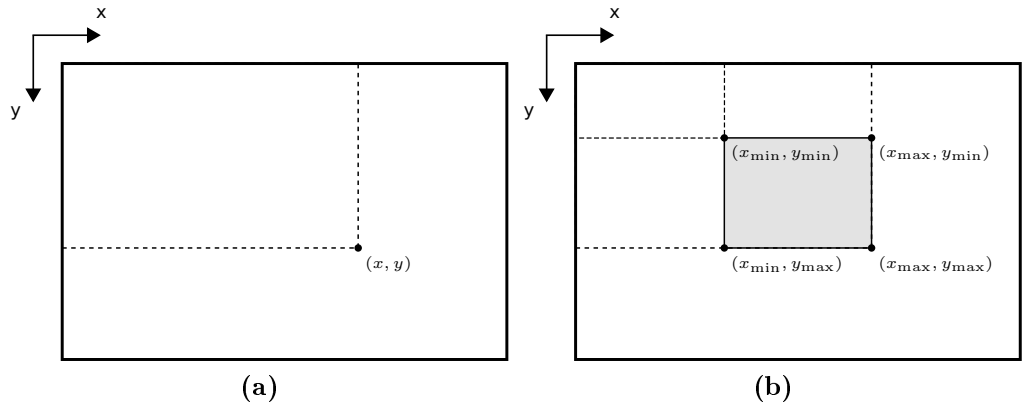


Abbildung 2.7: Verwendung von Integralbildern zur Berechnung von Mittelwerten.

2.4.1 Integralbilder

Der Wert eines Integralbildes $I_\Sigma(x, y)$ lässt sich aus dem konventionellen Bild I berechnen als die Summe aller Pixel links oberhalb von (x, y):

$$I_\Sigma(x, y) := \sum_{\substack{0 \le x' \le x, \\ 0 \le y' \le y}} I(x', y')$$

Um die Summe aller Pixel in einem rechteckigen Bildausschnitt zu berechnen, werden anschließend nur 3 Additionen benötigt (vgl. Abbildung 2.7):

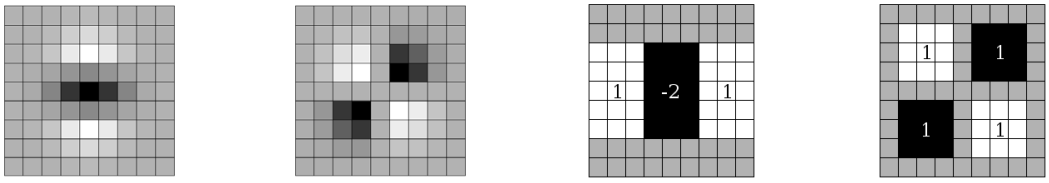
$$\sum_{\substack{x_{min} < x \le x_{max}, \\ y_{min} < y \le y_{max}}} I(x, y) = I_\Sigma(x_{max}, y_{max}) - I_\Sigma(x_{max}, y_{min}) - I_\Sigma(x_{min}, y_{max}) + I_\Sigma(x_{min}, y_{min})$$

In SURF wird diese Möglichkeit genutzt, um gaußsche Filterkerne zu approximieren. In Abbildung 2.8 ist dies für die zweiten partiellen gaußschen Ableitungen illustriert, welche im Detektionsschritt des Algorithmus verwendet werden.

Integralbilder können auf mehr als zwei Bilddimensionen generalisiert werden, z.B. zur Analyse von Videosequenzen [KSH05]. In [DLS07] wird das Konzept von Integralbildern sogar soweit generalisiert, dass damit die Verwendung von beliebigen B-Spline-Basisfunktionen als Filterkerne möglich wird.

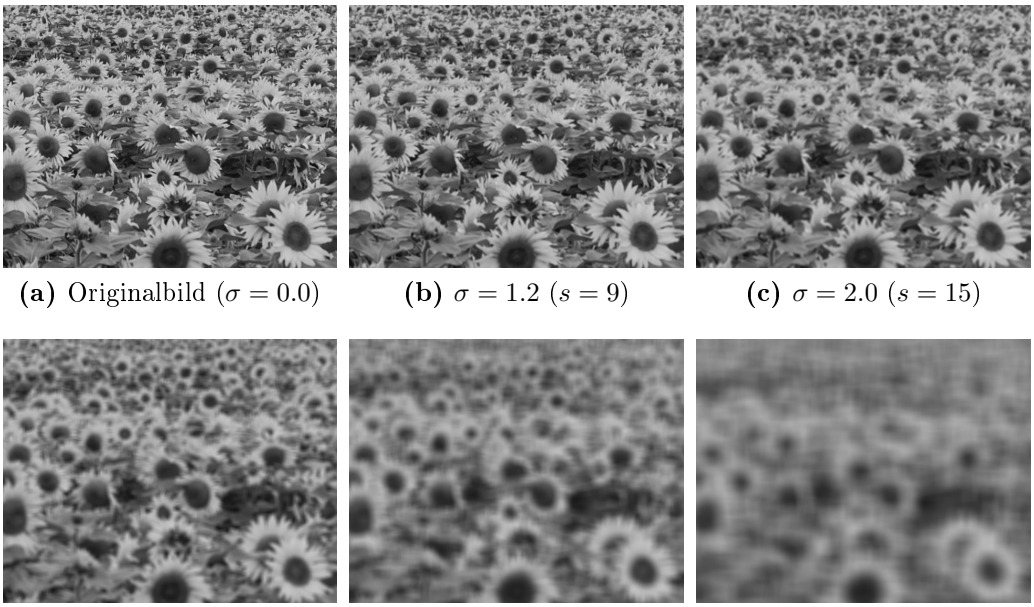
2.4.2 Detektionsschritt

Die Auswahl der lokalen Merkmale erfolgt anhand einer Bewertungsfunktion Ω, deren Konstruktion an die Determinante der Hessematrix $\overline{\boldsymbol{H}}(x, y, \sigma)$ angelehnt ist (vgl. Gleichungen 2.6, 2.7, 2.8). Die Skalenrepräsentation des Bildes und seine Ableitungen werden dabei durch Rechteckfilter (vgl. Abbildung 2.8) approximiert:



(a) G_{yy} **(b)** G_{xy} **(c)** B_{yy} **(d)** B_{xy}

Abbildung 2.8: a), b): Faltungskerne für die gaußschen Ableitungen G_{yy} und G_{xy} mit $\sigma = 1.2$. c), d): Approximation durch Rechteckfilter. Graue Bereiche entsprechen dem Wert 0. Quelle: [BETG08]

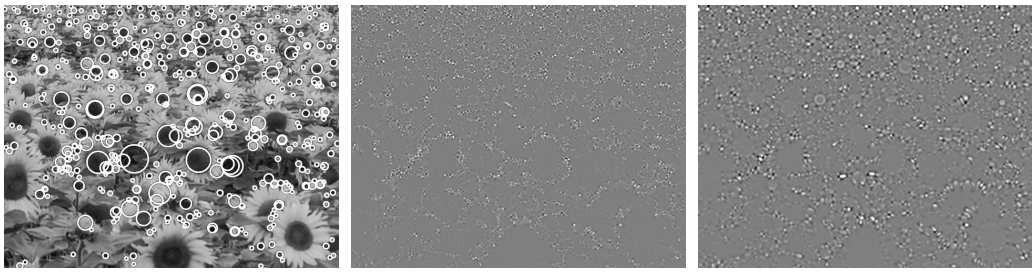


(a) Originalbild ($\sigma = 0.0$) **(b)** $\sigma = 1.2$ ($s = 9$) **(c)** $\sigma = 2.0$ ($s = 15$)

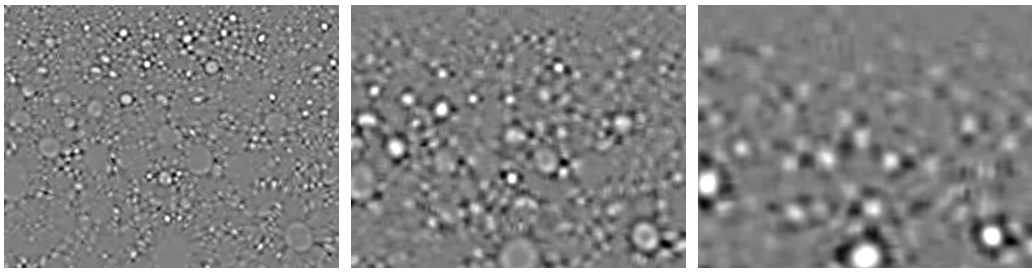
(d) $\sigma = 3.6$ ($s = 27$) **(e)** $\sigma = 6.8$ ($s = 51$) **(f)** $\sigma = 13.2$ ($s = 99$)

Abbildung 2.9: Schnitte durch den approximierten Skalenraum $\overline{L}(x, y, \sigma)$ für verschiedene Filtergrößen s, wobei der Filterkern eine quadratische Region mit Seitenlänge $\frac{5}{9} \cdot s$ enthält. Zu erkennen ist das Entstehen von horizontalen und vertikalen Strukturen im vergleich zu L (vgl. Abbildung 2.1).

Abbildung 2.10: Schnitt durch den approximierten Skalenraum in x- und σ-Richtung für die zentrale Bildzeile in Abbildung 2.9 (vgl. Abbildung 2.2)



(a) SURF-Schlüsselpunkte **(b)** $s = 9$ ($\sigma = 1.2$) **(c)** $s = 15$ ($\sigma = 2.0$)



(d) $s = 27$ ($\sigma = 3.6$) **(e)** $s = 51$ ($\sigma = 6.8$) **(f)** $s = 99$ ($\sigma = 13.2$)

Abbildung 2.11: Wert von $det(\overline{\boldsymbol{H}}(x, y, \sigma))$ für verschiedene Werte von σ. 50% Grau entspricht $det(\overline{\boldsymbol{H}}) = 0$, hellere Graustufen $det(\overline{\boldsymbol{H}}) > 0$ und dunklere $det(\overline{\boldsymbol{H}}) < 0$

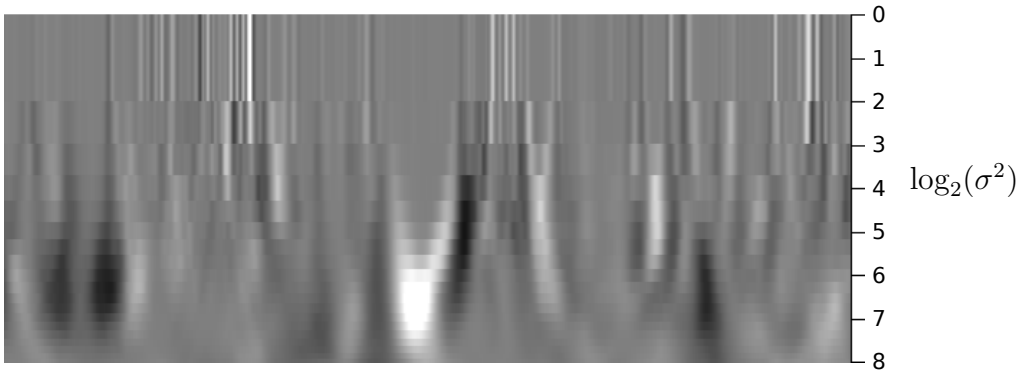


Abbildung 2.12: Signatur von $det(\overline{\boldsymbol{H}}(x, y, \sigma))$ in x- und σ-Richtung für die zentrale Bildzeile in Abbildung 2.11

$$\overline{\boldsymbol{H}}(x, y, \sigma) = \begin{bmatrix} \overline{L}_{xx}(x, y, \sigma) & \overline{L}_{xy}(x, y, \sigma) \\ \overline{L}_{xy}(x, y, \sigma) & \overline{L}_{yy}(x, y, \sigma) \end{bmatrix} \tag{2.9}$$

$$\overline{L}_{xx}(x, y, \sigma) = B_{xx}(\sigma) \circledast I(x, y) \tag{2.10}$$

$$\overline{L}_{yy}(x, y, \sigma) = B_{y}(\sigma) \circledast I(x, y) \tag{2.11}$$

$$\overline{L}_{xy}(x, y, \sigma) = B_{xy}(\sigma) \circledast I(x, y) \tag{2.12}$$

$$\det \boldsymbol{H}_{\text{norm}}(x, y, \sigma) \approx \Omega(x, y, \sigma) \tag{2.13}$$

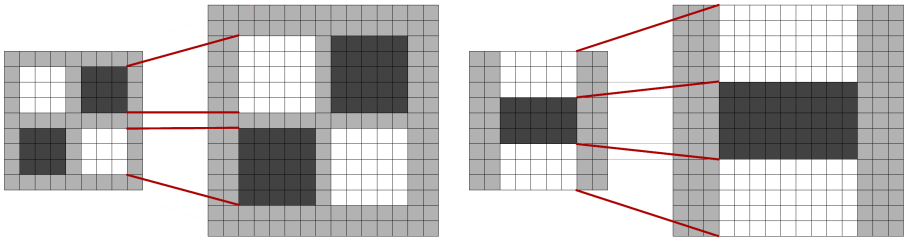
Die Rechteckfilter summieren die Pixelwerte. Um einen Mittelwertfilter zu erhalten, dessen Wertebereich über die Skalen konstant ist, muss man die Ableitungen des Bildsignals deshalb mit $\frac{1}{\sigma^2}$ normieren. Damit ergibt sich für Ω:

$$\Omega(x, y, \sigma) = \frac{\overline{L}_{xx}(x,y,\sigma)}{\sigma^2} \frac{\overline{L}_{yy}(x,y,\sigma)}{\sigma^2} - \left(\frac{w\overline{L}_{xy}(x,y,\sigma)}{\sigma^2}\right)^2 \tag{2.14}$$

$$= \frac{\overline{L}_{xx}(x,y,\sigma)\overline{L}_{yy}(x,y,\sigma) - (w\overline{L}_{xy}(x,y,\sigma))^2}{\sigma^4} \tag{2.15}$$

Über einen Schwellenwert für Ω kann die Anzahl der detektierten Schlüsselpunkte gesteuert werden. Der Faktor w wird mit 0,9 angegeben.

Durch die rechteckige Form der verwendeten Filterkerne wird die Reproduzierbarkeit bei Rotationen um Vielfache von 45° reduziert. In einer experimentellen Überprüfung anhand eines synthetisch rotierten Bildes stellen die Autoren von [BETG08] jedoch fest, dass die Reproduzierbarkeit trotzdem höher ist, als bei der Verwendung eines Gaußfilters. Dies erklären sie damit, dass die Größe der Gaußkerne in der Praxis aus Performanzgründen begrenzt ist, und so deren Rotationsinvarianz ebenfalls eingeschränkt ist.



(a) B_{xy} **(b)** B_{yy}

Abbildung 2.13: B_{yy} und B_{yy} für $\sigma = 1.2$ und $\sigma = 2.0$. Die Filtergröße muss um minimal 6 Pixel vergrößert werden, damit die Proportionen möglichst wenig verzerrt werden und weiterhin ein zentraler Pixel vorhanden ist. Quelle: [BETG08]

Da die Berechnungszeit für Rechteckfilter unabhängig von ihrer Größe ist, kann $\overline{L}$ anders als bei SIFT für jedes σ direkt berechnet werden. Durch ihre diskrete Natur sind jedoch nur eine begrenzte Anzahl von Filtergrößen zulässig. Der kleinste Filterkern hat die Seitenlänge $s = 9$ Pixel ($\sigma = 1.2$). In den nachfolgenden Skalen muss diese je um mindestens 6 Pixel in Breite und Höhe vergrößert werden, was einer Vergrößerung von σ um 0,8 entspricht. Dies liegt daran, dass auf der kleinsten Skala B_{xx} einen 3 Pixel breiten mittleren Teil hat (vgl. Abbildung 2.13). Damit ein zentraler Pixel erhalten bleibt, muss dieser Bereich um zwei Pixel vergrößert werden, was eine Vergrößerung der gesamten Maske um 6 Pixel bedeutet.

Eine geringe Verzerrung der Proportionen tritt durch die diskreten Größen der rechteckigen Teilregionen der Kerne dennoch auf. Beispielsweise beträgt die Breite der Regionen in B_{yy} 5 Pixel für $s = 9$. Im nächstgrößeren Filter mit $s = 15$ müsste diese also $15/9 * 5 = 8\frac{1}{3}$ betragen, was nicht möglich ist. Eine Methode zur Diskretisierung dieses Werts wird nicht angegeben.

Je größer σ, desto gröber kann der Skalenraum bei der Suche nach Maxima abgetastet werden. Daher werden die σ-Ebenen in Oktaven gruppiert. Innerhalb einer Oktave ist der Abstand der Skalen konstant. Bei jeder Oktave halbiert sich die Genauigkeit der Abtastung in x-,y- und σ-Richtung im Vergleich zur nächstniedrigeren Oktave.

Maxima werden dadurch detektiert, dass jeder abgetastete Wert von $\overline{L}$ mit den Werten in seiner $3 \times 3 \times 3$-Nachbarschaft verglichen wird. Daher können auf der niedrigsten und höchsten Skala einer Oktave keine Maxima gefunden werden. Um dennoch Maxima im gesamten Skalenraum finden zu können, müssen sich die Oktaven überschneiden. Daraus ergeben sich für die erste Oktave die Filtergrößen 9, 15, 21, und 27, für die zweite Oktave 15, 27, 39, und 51 usw. (vgl. Abbildung 2.14).

Wie auch bei SIFT (vgl. Abschnitt 2.3.5) wird anschließend das Interpolationsverfahren aus [LB02] eingesetzt, um die Maxima genauer zu lokalisieren. Dafür wird der Gradient und die Hessematrix von Ω durch Differenzbildung von benachbarten Abtastwerten berechnet. Daraus wird die Tailorentwicklung von Ω bis zu den quadratischen Termen bestimmt und deren Maximum zur Erzeugung eines Schlüsselpunktes verwendet.

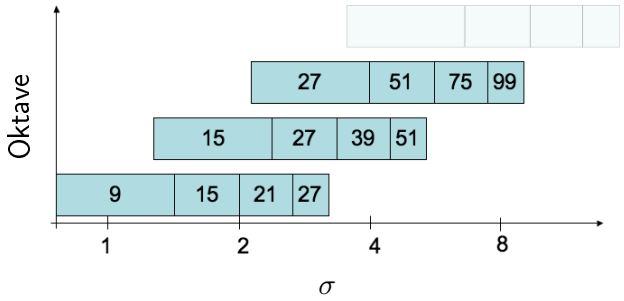
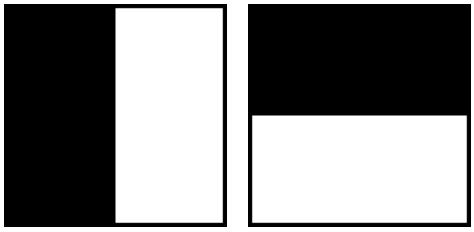


Abbildung 2.14: Gruppierung der Filtergrößen bzw. Skalen in Oktaven. Die Oktaven überschneiden sich, um auch an den Grenzen zwischen zwei Oktaven Maxima finden zu können. Quelle: [BETG08]



(a) B_x **(b)** B_y

Abbildung 2.15: Haar Wavelet-Filter in x- (links) und y-Richtung (rechts). Schwarze Bereiche entsprechen dem Wert -1, weiße dem Wert 1. Quelle: [BETG08]

Liegt das Maximum in einer Richtung näher am nächsten Abtastwert, also beispielsweise mehr als 0,5 Pixel in x- oder y-Richtung oder mehr als 0.4 in σ-Richtung in der ersten Oktave, wird es verworfen. So können in der ersten Oktave Maxima zwischen $\sigma = 1.6$ und $\sigma = 3.2$ detektiert werden.

Bei diesem Vorgehen werden die Skalen, besonders in der kleinsten Oktave, sehr grob abgetastet. Eine alternative Strategie besteht daher darin, die Größe des Eingabebildes durch lineare Interpolation zu verdoppeln und die Suche im Skalenraum mit einer Filtergröße von 15 und einer Schrittweite von 6 zu starten. So wird die erste Oktave mit den Filtergrößen 15, 21, 27 und 33 abgetastet, was Filtergrößen von 7,5, 10,5, 13,5 und 16,5 im Originalbild entspräche. Die kleinste mögliche Skala, auf der ein Maximum gefunden werden kann, reduziert sich dadurch zudem auf $\sigma = 1.2$.

2.4.3 Zuweisung einer normalisierten Orientierung

Die Zuweisung einer Orientierung, die sich kovariant mit einer Rotation des Eingabebildes verhält, ist nötig, damit der von SURF berechnete Deskriptor rotationsinvariant ist. Die Autoren weisen jedoch darauf hin, dass dies in manchen Anwendungsgebieten, z. B. Roboternavigation, nicht nötig ist. Daher schlagen sie eine Variante names U-SURF vor,

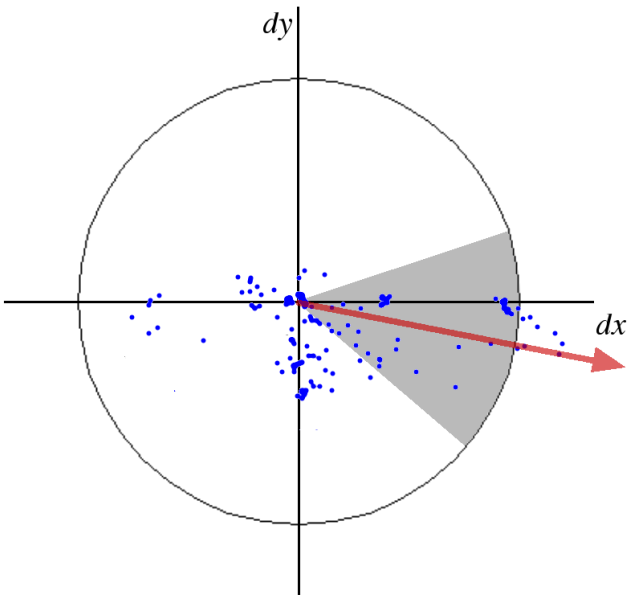


Abbildung 2.16: Zuweisung einer Orientierung anhand eines gleitenden Fensters. Quelle: [BETG08]

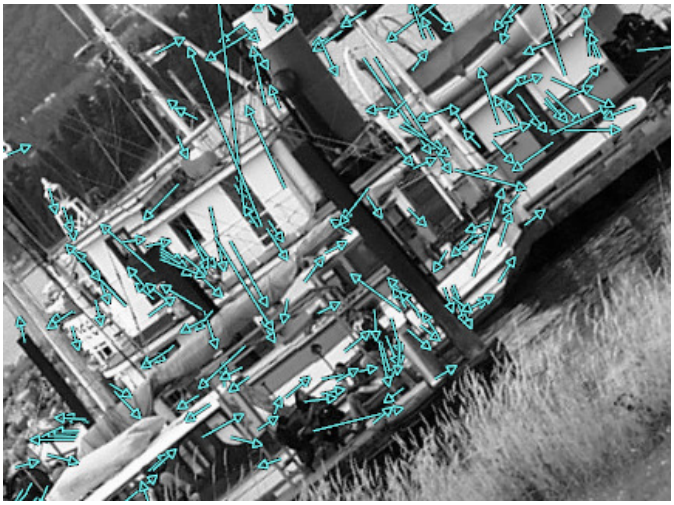


Abbildung 2.17: Beispiel für Orientierungen von SURF-Schlüsselpunkten. Ein Schlüsselpunkt und seine Richtung werden hier durch einen Pfeil dargestellt. Zu erkennen ist, wie sich die Orientierungen an den Bildstrukturen ausrichten. Quelle des Originalbildes: [Mik09]

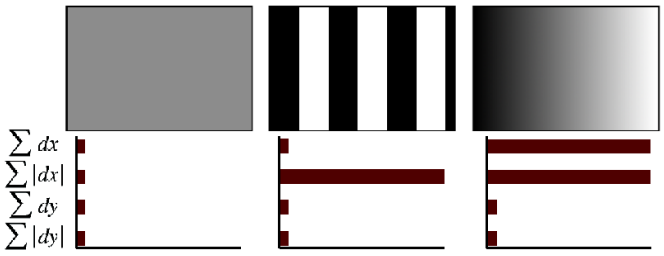


Abbildung 2.18: SURF-Deskriptor für eine Teilregion des Deskriptorfensters bei verschiedenen zugrundeliegenden Bildmustern. Quelle: [BETG08]

bei der allen Schlüsselpunkten eine feste Orientierung zugewiesen wird. Da der Normalisierungsschritt wegfällt und der Deskriptor über ein achsenparalleles Fenster berechnet wird, wird dadurch dieser Teil des Algorithmus beschleunigt und die Unterscheidungskraft der Deskriptoren erhöht.

In der Standardvariante von SURF werden zur Berechnung der Orientierung eines Schlüsselpunkts $(x_{\boldsymbol{k}}, y_{\boldsymbol{k}}, \sigma_{\boldsymbol{k}})$ die Gradienten in einer kreisförmigen Region mit Radius $6\sigma_{\boldsymbol{k}}$ um den Schlüsselpunkt berechnet. Diese werden durch eine Faltung mit Haar Wavelets in x- und y-Richtung approximiert (siehe Abbildung 2.15). Die Abtastung geschieht dabei mit einer Schrittweite von $\sigma_{\boldsymbol{k}}$, während die Wavelet-Kern eine Seitenlänge von $4\sigma_{\boldsymbol{k}}$ besitzen. Die konkrete Umsetzung der Wavelet-Kern als Rechteckfilter wird nicht angegeben.

Die so gewonnenen Wavelet-Antworten $d_i = (d_x(i), d_y(i))$ werden mit einer Gaußfunktion mit $\sigma = 2\sigma_{\boldsymbol{k}}$ gewichtet und in einer nach Winkeln sortierten Liste gespeichert. Über dieses wird nun ein gleitendes Fenster der Größe $\frac{\pi}{3}$ geschoben (vgl. Abbildung 2.16). In diesem Fenster wird an jeder Stelle die Summe aller d_i gebildet, woraus sich ein Orientierungsvektor für dieses Fenster ergibt. Die Richtung des längsten so gefundenen Vektors wird dem Schlüsselpunkt als Orientierung $(\theta_{\boldsymbol{k}})$ zugewiesen.

2.4.4 Merkmalsdeskriptor

Zur Berechnung des Merkmalsdesriptors wird eine um $\theta_{\boldsymbol{k}}$ rotierte quadratische Region mit Seitenlänge $20\sigma_{\boldsymbol{k}}$ betrachtet. Diese wird in 4×4 Teilregionen unterteilt. Für jede dieser Teilregionen $r_i, i \in [1..16]$ wird an 5×5 Messpunkten die Antwort $(d_x(i,j), d_y(i,j)), j \in [1..25]$ von Waveletfiltern mit Seitenlänge $2\sigma_{\boldsymbol{k}}$ berechnet und mit einer um den Schlüsselpunkt zentrierten Gaußfunktion mit $\sigma = 3,3\sigma_{\boldsymbol{k}}$ gewichtet.

Für jede der Teilregionen wird daraus ein 4-dimensionaler Deskriptor $\boldsymbol{v}_i$ berechnet. Dieser enthält die Summen der Wavelet-Antworten und ihrer Absolutwerte:

$$\boldsymbol{v}_i = \sum_{j=1}^{25} (d_x(i,j), d_y(i,j), |d_x(i,j)|, |d_y(i,j)|)$$

Dieser Deskriptor erlaubt es, zwischen verschiedenen Typen von Intensitätsmustern zu unterscheiden. Einige davon sind in Abbildung 2.18 gezeigt. Insgesamt ergibt sich aus

der Aneinanderreihung der einzelnen Deskriptoren für den Schlüsselpunkt ein Deskriptor der Länge $4 \cdot 4 \cdot 4 = 64$. Die Autoren schlagen vor, in zeitkritischen Anwendungen die Region in 3×3 Teilfenster aufzuteilen, was einen Deskriptor der Länge 36 ergibt. Dieser besitzt eine geringere Unterscheidungskraft, kann aber schneller mit anderen Deskriptoren verglichen werden.

Zudem wird ein erweiterter Deskriptor der Länge 128 vorgeschlagen. Dabei wird das Deskriptorfenster wie in der Standardvariante in 4×4 Teilfenster unterteilt. Die Werte für $d_x(i,j)$ und $|d_x(i,j)|$ werden jedoch abhängig vom Vorzeichen von $d_y(i,j)$ in unterschiedlichen Komponenten des Deskriptors gespeichert. Dasselbe gilt analog für $d_y(i,j)$ und $|d_y(i,j)|$. Dadurch verdoppelt sich die Länge des Deskriptors.

2.5 Objekterkennung durch lokale Merkmale

In [LB02] wird ein Verfahren beschrieben, mit dem Bilder von Objekten anhand ihrer lokalen Merkmale in einem anderen Bild (im Folgenden *Szenebild*) wiedergefunden werden können. Dazu werden die Schlüsselpunkte in beiden Bildern anhand ihrer Merkmalsvektoren einander zugeordnet. Da dadurch auch eine Reihe falscher Korrespondenzen entstehen, werden anschließend Gruppen von Korrespondenzen bestimmt, die eine konsistente Aussage über die Transformation des Objektbildes treffen.

Jedes Paar von einander zugeordneten Schlüsselpunkten macht dabei eine Aussage über Translation, Rotation und Skalierung des Objektbildes. Anhand dessen werden die Korrespondenzen mit einem Histogrammverfahren, genannt *Hough Clustering*, grob in konsistente Gruppen unterteilt. Anschließend wird für jede Gruppe mit der RANSAC-Methode eine affine Transformation bestimmt, bestehend aus Rotation, Translation, Skalierung und Scherung. Für die verbleibenden Korrespondenzen wird die Fundamentalmatrix zwischen den Kamerapositionen in beiden Bildern bestimmt und alle Korrespondenzen anhand der epipolaren Bedingung geprüft.

In dem für diese Arbeit entwickelten Objekterkennungssystem wurde die Bestimmung der affinen Transformation und Fundamentalmatrix durch die Bestimmung einer Homographie ersetzt.

2.5.1 Erstellung der Objektbeschreibung

Ein Objekt wird durch eine Reihe von Einzelbildern repräsentiert, die aus unterschiedlichen Blickwinkeln aufgenommen werden. Damit der Hintergrund, vor dem das Objekt aufgenommen wurde, das Resultat nicht beeinflusst, wird er aus dem Objektbild entfernt. Dafür wird jeweils ein Bild mit und ohne Objekt aufgenommen. Durch ein Differenzbildverfahren wird daraufhin eine Maske berechnet, die Objekt und Hintergrund separiert. Um Rauschen zu unterdrücken, wird eine Reihe von morphologischen Operationen auf sie angewandt und schließlich das Segment mit der größten Fläche isoliert. Maskierte Bildbereiche werden durch Schwarz ersetzt.

Aus dem so modifizierten Bild werden schließlich Merkmale extrahiert. Schlüsselpunkte, die zu einem gewissen Grad den maskierten Bildbereich überschneiden, werden

verworfen. Dafür wird für jeden Schlüsselpunkt $\boldsymbol{k}$ mit der Skala $\sigma_{\boldsymbol{k}}$ überprüft, ob in einem Radius von $1,875\sigma_{\boldsymbol{k}}$ Pixeln maskierte Pixel vorhanden sind.

2.5.2 Nearest Neighbour Ratio Matching

Für die initiale Zuordnung der Schlüsselpunkte zwischen dem Objektbild und dem Szenebild werden diese anhand ihrer Merkmalsvektoren verglichen. Zu einem Schlüsselpunkt im Szenebild mit dem Merkmalsvektor $\boldsymbol{v}^1$ werden die zwei Schlüsselpunkte im Objektbild gesucht, deren Merkmalsvektoren $\boldsymbol{v}_1^2$ und $\boldsymbol{v}_2^2$ den kleinsten bzw. zweitkleinsten euklidischen Abstand zu $\boldsymbol{v}^1$ haben.

Der Quotient aus diesen beiden Abständen wird als *Nearest Neighbour Ratio (NNR)* Φ bezeichnet. Liegt Φ unter einem Schwellenwert t_Φ, so wird $\boldsymbol{v}^1$ $\boldsymbol{v}_1^2$ zugeordnet. Damit liefert Φ ein Maß, welches relativ zur geschätzten lokalen Dichte im Merkmalsraum definiert ist. In [MS05] wurde gezeigt, dass es im Vergleich zu anderen Verfahren, zum Beispiel dem absoluten Abstand zwischen einem Merkmal und seinem nächsten Nachbarn, bessere Resultate liefert.

2.5.3 Hough Clustering

Die durch das NNR-Matching ermittelten Korrespondenzen können einen großen Anteil falscher Zuordnungen enthalten [LB02]. Daher kann man nicht direkt mit RANSAC eine Homographie schätzen. Als Zwischenschritt wird daher Hough Clustering eingesetzt.

Da den Schlüsselpunkten eine Position, Skala und Orientierung zugeordnet ist, stellt jede Korrespondenz eine Hypothese über die relative Translation, Skalierung und Rotation des Objektbildes im Szenebild dar. Die Hypothesen aller Korrespondenzen werden in ein Histogramm eingetragen.

Für zwei einander zugeordnete Schlüsselpunkte $\boldsymbol{k}_o$ und $\boldsymbol{k}_s$ mit den Bildpositionen $(x_o, y_o)^\mathrm{T}$ und $(x_s, y_s)^\mathrm{T}$, den Orientierungen θ_o und θ_s und den Skalen σ_o und σ_s wird die relative Orientierung definiert als der minimale Drehwinkel α, um θ_o in θ_s zu überführen [Thi09]. α hat damit einen Wertebereich von $-180°$ bis $180°$. Die relative Skala ist definiert als $\log_2 \frac{\sigma_s}{\sigma_o}$.

Die Position $(\overline{x}_p, \overline{y}_p)^\mathrm{T}$ des Objektbildes im Kamerabild wird für den geometrischen Schwerpunkts $(\overline{x}_o, \overline{y}_o)^\mathrm{T}$ der Objektpixel definiert:

$$\begin{pmatrix} \overline{x}_p \\ \overline{y}_p \end{pmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \cdot \left(\begin{pmatrix} \overline{x}_o \\ \overline{y}_o \end{pmatrix} - \begin{pmatrix} x_o \\ y_o \end{pmatrix} \right) \cdot \frac{\sigma_s}{\sigma_o} + \begin{pmatrix} x_s \\ y_s \end{pmatrix}$$

Das entstehende Histogramm hat vier Dimensionen $(x_p, y_p, \alpha, \log_2 \frac{\sigma_s}{\sigma_o})$, die in Intervalle wählbarer Breite unterteilt werden. Der betrachtete Raum wird in vierdimensionale Quader quantisiert, wovon jedes durch einen Eintrag im Histogramm repräsentiert wird. Um Randeffekte zu vermeiden, erhöht jede Hypothese in allen Dimensionen den Wert der beiden Einträge, deren Intervallzentren den kleinsten Abstand zum gemessenen Wert haben, also insgesamt $2^4 = 16$ Einträge.

Nachdem das Histogramm gefüllt wurde, werden alle Einträge gesucht, deren Wert einen Schwellenwert $t_n \geq 5$ übersteigen. Für jeden dieser Einträge werden alle Korrespondenzen ausgegeben, die dessen Wert erhöht haben, also die durch ihn repräsentierte Menge von Hypothesen stützen.

2.5.4 Bestimmung der Homographie

Jede der mit dem Hough Clustering bestimmten Gruppen von Korrespondenzen wird genutzt, um eine Homographie zu berechnen, die korrespondierende Punkte aufeinander abbildet. Diese wird rein auf der Basis der Bildpositionen der Schlüsselpunkte bestimmt. Orientierung und Skala werden nicht berücksichtigt.

Die berechnete Homographie $\boldsymbol{H}$ ist eine homogene 3×3-Matrix, für die gilt:

$$w_p \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = \boldsymbol{H} \begin{pmatrix} x_o \\ y_o \\ 1 \end{pmatrix}$$

Durch die Verwendung von homogenen Koordinaten kann die Berechnung der Abbildung als lineares Problem dargestellt werden. Dafür werden eine Reihe zufälliger Untergruppen von jeweils 5 Korrespondenzen ausgewählt und für diese jeweils eine Homographie bestimmt. Die Untergruppe, deren Homographie zu den meisten ursprünglichen Korrespondenzen passt, wird ausgewählt. Anhand dieser passenden Korrespondenzen wird schließlich die endgültige Homographie berechnet.

Da nach dem Hough Clustering für jede Gruppe von Korrespondenzen eine Homographie berechnet wurde, wird hieraus wiederum diejenige ausgewählt, die mit dem meisten Korrespondenzen verträglich ist.

2.5.5 Kriterium für die Detektion eines Objekts

Um zu entscheiden, ob ein Objekt gefunden wurde, wird die Anzahl der verbleibenden Korrespondenzen nach Bestimmung der Homographie mit der Anzahl der vorhandenen Schlüsselpunkte verglichen:

$$p = \frac{N_c}{\min(N_s, N_o)}$$

Dabei ist N_o die Anzahl der Schlüsselpunkte im Objektbild. N_s ist die Anzahl der Schlüsselpunkte in dem Ausschnitt des Szenebildes, der von dem Objektbild bedeckt wird und wird anhand der Homographie bestimmt. N_c ist die Anzahl der Korrespondenzen, die zu der berechneten Homographie passen. Ist p über einem gewissen Schwellenwert, gilt das Objekt als erkannt.

Kapitel 3

Erweiterung des SURF-Algorithmus

Der SURF-Algorithmus arbeitet auf Graustufenbildern. In vorangegangenen Evaluationen [BG09, SGS08b] wurde allerdings gezeigt, dass durch Berücksichtigung von Farbe eine Erhöhung der Invarianz bzw. Robustheit gegenüber Änderungen der Beleuchtung und eine Vergrößerung der Unterscheidungskraft der daraus gewonnenen Bildmerkmale erzielt werden kann.

Vergleichbare Verfahren zu dem im Folgenden beschriebenen stellen in vielen Fällen eine Erweiterung des SIFT-Algorithmus dar [BG09, AHF06, BZM06, WS06]. Dieser besitzt aufgrund seiner Laufzeiteigenschaften eine geringe Eignung für Robotik-Applikationen. Daher werden die in der Literatur entwickelten und untersuchten Erweiterungen von SIFT auf den SURF-Algorithmus adaptiert.

In den folgenden Abschnitten werden zunächst die physikalischen Grundlagen der Bildentstehung und mögliche Räume zur Repräsentation von Farbinformationen beschrieben. Daraus werden Bildeigenschaften definiert, die unter den Annahmen des beschriebenen physikalischen Modells invariant gegen bestimmte Klassen von photometrischen Transformationen sind. Dies umfasst Transformationen, welche durch eine Änderung der Beleuchtung oder Betrachterposition hervorgerufen werden, jedoch nicht die Bildgeometrie betreffen. Schließlich wird beschrieben, wie diese Bildeigenschaften in die verschiedenen Phasen des SURF-Algorithmus integriert werden können.

3.1 Zugrunde liegendes physikalisches Modell

Um die Eigenschaften eines Farbbildes herzuleiten, die invariant gegenüber Änderungen in der Beleuchtungssituation sind, müssen zuerst die physikalischen Prozesse analysiert werden, die der Bildentstehung zugrunde liegen.

Eine gute Näherung hierfür stellt das dichromatische Reflexionsmodell dar [Sha92]. Dieses beruht auf einer Beschreibung der Reflexion von Licht an Materialien, bei denen das Licht zum Einen an der Grenzfläche zwischen dem Material und dem umgebenden Medium reflektiert wird (*Interface Reflection*) und zum Anderen teilweise in das Medium eindringt (*Body Reflection*), wobei es wiederum teilweise absorbiert und teilweise erneut

an der Eintrittsfläche abgegeben wird. Dadurch können viele Materialen wie z.B. Farbe, Papier und Plastik hinreichend beschrieben werden, jedoch z.B. keine durchscheinenden Materialien, Metalle und Kristalle.

Die Reflexion an der Grenzfläche ist lokal eine perfekte Spiegelung des eintreffenden Lichts entlang der Oberflächennormale. Da die Oberflächennormalen bei den betrachteten Materialien jedoch auf kleinen Skalen variieren, ist auf makroskopischer Ebene auch dieser Anteil des gespiegelten Lichts in gewissem Maß gestreut. Das Licht, welches nicht an der Grenzfläche reflektiert wird, dringt in das Material ein, wo es abhängig von seiner Wellenlänge mit unterschiedlicher Wahrscheinlichkeit absorbiert wird oder gestreut wieder austritt.

In diesem Modell kann die spektrale Zusammensetzung des in eine bestimmte Richtung reflektierten Lichts abhängig von den geometrischen Eigenschaften $\boldsymbol{g}$ der betrachteten Oberfläche und Lichtquelle also durch zwei getrennte Terme R_i und R_b dargestellt werden. Diese beschreiben jeweils die Reflexion an der Grenzfläche sowie innerhalb des Mediums:

$$R(\lambda, \boldsymbol{g}) = R_b(\lambda, \boldsymbol{g}) + R_i(\lambda, \boldsymbol{g})$$

$$R_b(\lambda, \boldsymbol{g}) = m_b(\boldsymbol{g}) b(\lambda, \boldsymbol{g}) e(\lambda)$$

$$R_i(\lambda, \boldsymbol{g}) = m_i(\boldsymbol{g}) i(\lambda) e(\lambda)$$

Die Grenzflächenreflexion $i(\lambda)$ ist abhängig von der Wellenlänge des eintreffenden Lichts. Da dieser Effekt jedoch im Allgemeinen sehr gering ist, kann i als Konstante angenommen werden. In [Sha92] wird zudem ein Term eingeführt, welcher diffuse Umgebungsbeleuchtung (beispielsweise gestreutes Licht bei bedecktem Himmel) approximiert und unabhängig von der Geometrie ist:

$$R(\lambda, \boldsymbol{g}) = R_b(\lambda, \boldsymbol{g}) + R_i(\lambda, \boldsymbol{g}) + a(\lambda) \tag{3.1}$$

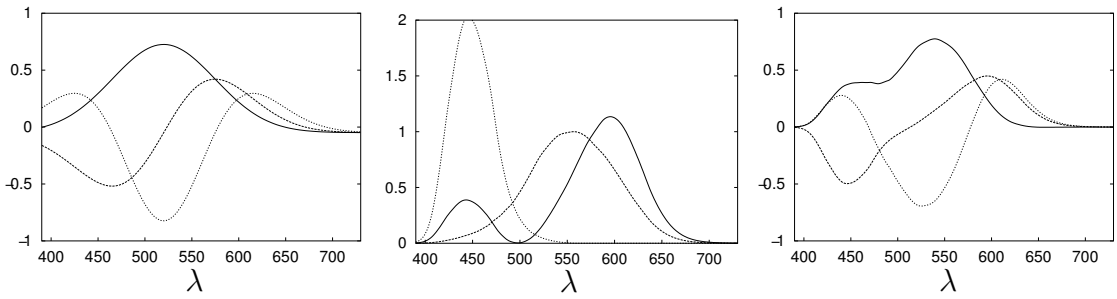
Fällt Licht von einer Oberfläche $\boldsymbol{g}$ mit der spektralen Energiedichte R in eine (idealisierte) Kamera an der Bildposition x, y, so wird diese linear auf einen Farbwert abgebildet:

$$I^k(x, y) = \int_\omega R(\lambda, \boldsymbol{g}) \rho_k(\lambda) d\lambda$$

wobei $\rho_k(\lambda)$ die Sensitivitätskurve der Kamera für den Bildkanal k (z.B. R,G oder B) und ω das Intervall der Wellenlängen von sichtbarem Licht bezeichnen. Für das dichromatische Modell mit diffusem Umgebungslicht ergibt sich damit:

$$I^k(x, y) = m_b(\boldsymbol{g}) \int_\omega b(\lambda, \boldsymbol{g}) e(\lambda) \rho_k(\lambda) d\lambda + m_i(\boldsymbol{g}) \int_\omega i(\lambda) e(\lambda) \rho_k(\lambda) d\lambda + \int_\omega a(\lambda) \rho_k(\lambda) d\lambda \tag{3.2}$$

Eine Änderung des diffusen Lichts resultiert demnach in der Addition einer Konstanten zu $I^k(x, y)$, wodurch die Ableitungen I^k_x und I^k_y dagegen invariant sind.



(a) G, G_λ, $G_{\lambda\lambda}$ **(b)** CIE X, Y, Z **(c)** $\hat{G}$, $\hat{G}_\lambda$, $\hat{G}_{\lambda\lambda}$

Abbildung 3.1: Sensitivitätskurven G, G_λ, $G_{\lambda\lambda}$ des gaußschen Farbmodells, von CIE XYZ (1964) und resultierende Kurven $\hat{G}$, $\hat{G}_\lambda$, $\hat{G}_{\lambda\lambda}$ bei Approximation durch eine Linearkombination der XYZ-Kurven. Quelle: [GBSD00]

3.2 Untersuchte Farbräume

Die Farbräume, welche im Folgenden auf ihre Eignung zur Repräsentation der Bilder und ihrer Merkmale verwendet werden, lassen sich durch lineare Transformationen des RGB-Farbraums darstellen. Als RGB-Farbraum wird im Folgenden der im HDTV-Standard BT.709 [BT702] festgelegte bezeichnet, jedoch ohne Gamma-Korrektur. Dessen Primärvalenzen werden auch im sRGB-Farbraum verwendet, welches ein Standard für die digitale Darstellung von Farben ist. Weitere Gemeinsamkeit der betrachteten Farbräume ist, dass sie einen Kanal enthalten, der die Intensität des Bildsignals kodiert sowie zwei Kanäle, die sowohl Intensitäts- als auch Farbinformationen enthalten.

3.2.1 Gaußsches Farbmodell

In [GBSD00] wird ein Farbmodell entwickelt, dessen Sensitivitätskurven eine um λ_0 zentrierte Gaußfunktion $G(\lambda, \lambda_0, \sigma_\lambda)$ mit Standardabweichung σ_λ und deren Ableitungen approximieren. Daher kann der Helligkeitskanal E als Resultat einer Faltung (bzw. Glättung) der spektralen Energiedichte $R(\lambda)$ mit der Gaußfunktion G an der Stelle λ_o interpretiert werden, und die beiden Farbkanäle als ihre erste und zweite gaußsche Ableitung:

$$E \approx \int R(\lambda) G(\lambda, \lambda_0, \sigma_\lambda) d\lambda = (R \circledast G(\sigma_\lambda))(\lambda_0)$$

$$E_\lambda \approx \int R(\lambda) G_\lambda(\lambda, \lambda_0, \sigma_\lambda) d\lambda = (R \circledast G_\lambda(\sigma_\lambda))(\lambda_0)$$

$$E_{\lambda\lambda} \approx \int R(\lambda) G_{\lambda\lambda}(\lambda, \lambda_0, \sigma_\lambda) d\lambda = (R \circledast G_{\lambda\lambda}(\sigma_\lambda))(\lambda_0)$$

Damit kann aus den Farbwerten E, E_λ und $E_{\lambda\lambda}$ die Taylor-Entwicklung zweiten Grades von R an der Stelle λ_0 bestimmt werden:

$$R(\lambda) \approx E + \lambda E_\lambda + \frac{1}{2}\lambda^2 E_{\lambda\lambda}$$

Die Parameter λ_0 und σ_λ sind so gewählt, dass der größte Hauptwinkel zwischen den dreidimensionalen Unterräumen des unendlichdimensionalen Raumes der Lichtspektren, welche durch die CIE XYZ-Sensitivitätskurven und die gaußschen Kurven G, G_λ und $G_{\lambda\lambda}$ definiert werden, minimiert wird. Die lineare Transformation M_E vom XYZ-Farbraum in den gaußschen Farbraum wird durch Minimierung des quadratischen Fehlers bestimmt.

Sei M_{RGB} die Matrix zur Transformation von XYZ nach RGB, so ergibt sich die Transformation von RGB in den gaußschen Farbraum als $M_E \cdot M_{RGB}^{-1}$:

$$\begin{bmatrix} E \\ E_\lambda \\ E_{\lambda\lambda} \end{bmatrix} = \begin{bmatrix} 0.06 & 0.63 & 0.27 \\ 0.30 & 0.04 & 0.35 \\ 0.34 & 0.60 & 0.17 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

3.2.2 YCrCb

Der YCrCb-Farbraum ist für die Übertragung von Videosignalen konzipiert. Dieser wird in den ITU-Standards [BT607] und [BT702] leicht unterschiedlich spezifiziert. Im Folgenden wird die Definition aus [BT607] verwendet, welche auch für die Kodierung von JPEG-Bildern eingesetzt wird. Vorteil dieses Farbraumes ist, dass digitale Videokameras ihre Bilddaten häufig im YCrCb- oder JPEG-Format liefern und daher keine zusätzliche Konvertierung nötig ist. Die Abbildung von RGB nach YCrCb lautet:

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0,2989 & 0,5866 & 0,1145 \\ 0,5000 & -0,4183 & 0,0817 \\ -0,1688 & -0,3312 & 0,5000 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

3.2.3 IRG

In [SGS08b] wird vorgeschlagen, die Farbinformation durch den R- und G-Kanal und die Helligkeitsinformation als Summe der Kanäle zu repräsentieren. Daraus ergibt sich der IRG-Farbraum:

$$\begin{bmatrix} I \\ R \\ G \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

3.2.4 RGB-Gegenfarbraum

Der RGB-Gegenfarbraum wird in [WGB06] eingeführt. Die Kanäle $O1$ und $O2$ sind so definiert, dass sie invariant gegenüber der Addition einer Konstante auf die RGB-Kanäle sind, sofern diese für jeden Kanal gleich ist. Somit sind diese invariant gegenüber der Grenzflächen-Reflexion R_i, wenn diese als weiß angenommen wird, sowie gegenüber

weißem Umgebungslicht a (vgl. Gleichung 3.1). $O3$ bezeichnet in diesem Farbraum die Intensität.

$$\begin{bmatrix} O3 \\ O1 \\ O2 \end{bmatrix} = \begin{bmatrix} 0.5777 & 0.5777 & 0.5777 \\ 0.7071 & -0.7071 & 0.0000 \\ 0.4082 & 0.4082 & -0.8165 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

3.3 Photometrische Invarianten

Um Robustheit bzw. Invarianz von Bildmerkmalen gegenüber einer Änderung der Beleuchtungssituation zu erreichen, wurde eine Vielzahl von photometrisch invarianten Eigenschaften von Farb- und Graustufenbildern vorgeschlagen [GBSG01, GS97, ZMKB08, WGG03, WGB06]. Verschiedene Kombinationen dieser Invarianten wurden benutzt, um den SIFT-Deskriptor zu erweitern. Die meisten Verfahren haben gemeinsam, dass an den SIFT-Deskriptor für den Intensitätskanal ein weiterer Deskriptor angehängt wird, der die Farbinformationen enthält.

Zur Beschreibung der Farbinformation werden Farbhistogramme [WS06, AB07, SGS08b, WGB06], Farbmomente [MTGM04] oder die zusätzliche Berechnung des SIFT-Deskriptors auf den Farbkanälen [BG09, SGS08b, SGS08c, AHF06, BZM06] vorgeschlagen.

Die Farbdeskriptoren wurden in unterschiedlichen Kontexten evaluiert. In [BG09] wird der SIFT-Deskriptor auf den invarianten Farbkanälen des gaußschen Farbraums berechnet und mit zwei Varianten verglichen, die auf dem HSV-Farbraum basieren [AHF06, BZM06]. Als Datenbasis wird die ALOI-Datenbank verwendet, auf der die Robustheit der Verfahren gegenüber einer Änderung der Beleuchtungsrichtung, Beleuchtungsfarbe, Betrachterposition, Unschärfe und JPEG-Komprimierung gemessen wird. Die besten Ergebnisse werden mit den Deskriptoren erzielt, die auf den W- und C-Invarianten des gaußschen Farbraums basieren, sowie mit dem Verfahren aus [BZM06].

In [SGS08b] werden verschiedene Farbhistogramme, Farbmomente, SIFT, der um ein Farbhistogramm erweiterte SIFT-Deskriptor aus [WS06], und SIFT-Deskriptoren, die auf mehreren Kanälen berechnet werden, verglichen. Als Kriterium für die Evaluation dient die Klassifikationsleistung auf einer Datenbank mit Objektbildern sowie eine Datenbank mit Objekt- und Szenebildern. Die beste Leistung erzielen die Invarianten des gaußschen Farbraums, RGB-Gegenfarbraums und des IRG-Farbraums.

Im Folgenden werden zwei Invarianten betrachtet, die sich in den erwähnten Evaluationen als besonders geeignet heraus gestellt haben. Diese können analog auf den gaußschen Farbraum, IRG, YCrCb und den RGB-Gegenfarbraum angewandt werden. In [GBSG01] werden sie für den gaußschen Farbraum definiert und in [GBSD00] auf die Theorie des Skalenraums angewandt.

Da zudem die Verfahren die beste Leistung erzielt haben, die den SIFT-Deskriptor zusätzlich auf den Farbkanälen berechnen und dieser ähnliche Bildeigenschaften beschreibt wie der SURF-Deskriptor, wird dieses Vorgehen auch zur Erweiterung des SURF-Deskriptors gewählt.

3.3.1 Die W-Invariante

Für die Skalenrepräsentation $L(x, y, \sigma)$ eines Farbbildes, bei dem der erste Kanal L_1 die Intensität des Bildsignals beschreibt, ist die W-Invariante für $k \in [1, 2, 3]$ definiert als die intensitätsnormierte Ableitung des Bildsignals

$$W_{k,x} = \frac{L_{k,x}}{L_1} \tag{3.3}$$

$$W_{k,xx} = \frac{L_{k,xx}}{L_1} \tag{3.4}$$

$W_{k,y}$, $W_{k,yy}$ und $W_{k,xy}$ sind analog definiert. Unter der Annahme, dass weder uniformes Umgebungslicht noch Grenzflächen-Reflexion vorhanden sind, beschreibt W den Verlauf der Intensität unabhängig von Variationen in der lokalen Helligkeit [GBSG01]. Sie ist damit nicht invariant gegenüber Helligkeitsverläufen, wie sie bei dreidimensionalen Objekten entstehen. Trotzdem sollte sie auch hier eine größere Robustheit gegenüber Beleuchtungsänderungen aufweisen als die nicht-normierten Ableitungen.

3.3.2 Die C-Invariante

Die C-Invariante ist für die Farbkanäle definiert als die Ableitung des intensitätsnormierten Bildsignals $\frac{L_{k,x}}{L_1}, k \in 2, 3$:

$$C_{k,x} = \frac{L_{k,x}L_1 - L_k L_{1,x}}{L_1^2} \tag{3.5}$$

$$C_{k,xx} = \frac{L_{k,xx}L_1^2 - L_k L_{1,xx} L_1 - 2L_{k,x}L_{1,x}L_1 + 2L_k L_{1,x}^2}{L_1^3} \tag{3.6}$$

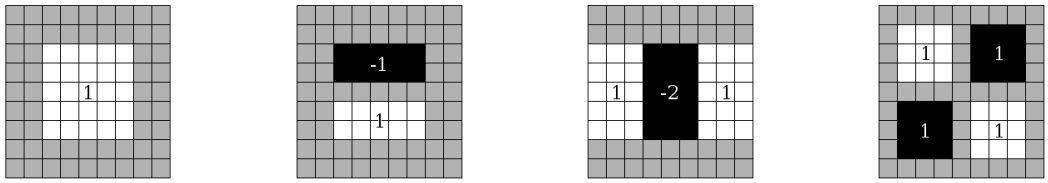
$$C_{k,xy} = \frac{L_{k,xy}L_1^2 + L_{k,x}L_{1,y}L_1 - L_{k,y}L_{1,x}L_1 - L_k L_{k,xy} L_1 - 2L_{k,x}L_{1,y}L_1 + 2L_k L_{1,x}L_{1,y}}{L_1^3} \tag{3.7}$$

$C_{k,y}$ und $C_{k,yy}$ sind analog zu $C_{k,x}$ und $C_{k,xx}$ definiert. Damit ist sie unter den gleichen Annahmen wie für W invariant gegenüber Schatten- und Schattierungseffekten [GBSG01]. Für den Intensitätskanal ist C nicht definiert.

Sowohl die W- als auch die C-Invariante sind im Gegensatz zu den nicht-normalisierten Ableitungen nicht invariant gegenüber diffusem Umgebungslicht, da sie abhängig von L sind. Dies gilt auch für den RGB-Gegenfarbraum.

3.3.3 Approximation durch Rechteckfilter

Die Invarianten W und C werden durch Kombinationen von verschiedenen partiellen Ableitungen des Bildsignals konstruiert und sind für die Verwendung mit gaußschen Ableitungen ausgelegt. Da die Rechteckfilter jedoch andere Eigenschaften als gaußsche Filterkerne haben, muss ein zusätzlicher Normalisierungsschritt eingeführt werden.



(a) B **(b)** B_y **(c)** B_{yy} **(d)** B_{xy}

Abbildung 3.2: Rechteckfilter für Ableitungen ersten und zweiten Grades.

Für eine Funktion $f : \mathbb{R}^k \mapsto \mathbb{R}$ sei $|f|_{\circledast}$ definiert als das Volumen zwischen der Funktion und der x_1-y_2-Ebene:

$$|f(x_1, x_2, \ldots, x_k)|_{\circledast} = \iint_{-\infty}^{\infty} |f(x_1, x_2, \ldots, x_k)| \, dx_1 \, dx_2$$

Damit ist $|f|_1^2$ ein Maß für die Verstärkung eines Signal durch die Faltung mit f. Für ein Bildsignal $I(x, y) : \mathbb{R}^2 \mapsto [0, 1]$ und einen Faltungskern f gibt $|f|_{\circledast}$ die Intervallbreite des Bildbereichs von $f \circledast I$ an.

Die Gaußfunktion G (Gleichung 2.5) ist bereits skaleninvariant, da $|G(x, y, \sigma)|_1^2$ unabhängig von σ ist:

$$|G(x, y, \sigma)|_1^2 = 1 \quad \forall \sigma \in \mathbb{R} \setminus \{0\}$$

Die Ableitungen vom Grad n müssen jeweils mit dem Faktor σ^n normalisiert werden, um Skaleninvarianz zu erzielen [Lin98]. Dadurch ergibt sich $\forall \sigma \in \mathbb{R} \setminus \{0\}$:

$$|\sigma G_x(x, y, \sigma)|_{\circledast} = |\sigma G_y(x, y, \sigma)|_1^2 \approx 0,7976 = \alpha_{G_x}$$
$$|\sigma^2 G_{xx}(x, y, \sigma)|_{\circledast} = |\sigma^2 G_{yy}(x, y, \sigma)|_1^2 \approx 0,9664 = \alpha_{G_{xx}}$$
$$|\sigma^2 G_{xy}(x, y, \sigma)|_{\circledast} = |\sigma^2 G_{yx}(x, y, \sigma)|_1^2 \approx 0,6362 = \alpha_{G_{xy}}$$

Die Rechteckfilter, welche bei SURF zur Berechnung der Ableitungen verwendet werden, sind für $\sigma = 1.2$ in Abbildung 3.2 abgebildet. Lässt man, wie in [BETG08] vorgeschlagen, die Diskretisierungseffekte außer Acht, durch die die Proportionen der Rechteckfilter auf verschiedenen Skalen verzerrt werden, so erhält man:

$$|B(x, y, \sigma)|_1^2 \approx 17,36\sigma^2 = \alpha_B$$
$$|B_x(x, y, \sigma)|_{\circledast} = |B_y(x, y, \sigma)|_1^2 \approx 13,88\sigma^2 = \alpha_{B_x}\sigma^2$$
$$|B_{xx}(x, y, \sigma)|_{\circledast} = |B_{yy}(x, y, \sigma)|_1^2 \approx 41,66\sigma^2 = \alpha_{B_{xx}}\sigma^2$$
$$|B_{xy}(x, y, \sigma)|_{\circledast} \approx 25\sigma^2 = \alpha_{B_{xy}}\sigma^2$$

Damit die Rechteckfilter den gleichen Wertebereich besitzen wie die gaußschen Filterkerne, müssen sie also jeweils mit einem konstanten Faktor und $\frac{1}{\sigma^2}$ gewichtet werden.

Der Faktor $\frac{1}{\sigma^2}$ fällt jedoch bei den C- und $W-$Invarianten durch die Division durch den Intensitätskanal weg.

Für W ergeben sich die approximierten Invarianten wie folgt:

$$\overline{W}_{k,x} = \frac{\alpha_B \alpha_{G_x} \overline{L}_{k,x}}{\alpha_{B_x} \overline{L}_1} \tag{3.8}$$

$$\overline{W}_{k,xx} = \frac{\alpha_B \alpha_{G_{xx}} \overline{L}_{k,x}}{\alpha_{B_{xx}} \overline{L}_1} \tag{3.9}$$

Analog lassen sich auch die approximierten Invarianten $\overline{C}_x$, $\overline{C}_{xx}$ usw. zu C durch Einsetzen in Gleichungen 3.5 - 3.7 konstruieren. Dies führt jedoch nicht zu den gewünschten Ergebnissen, wie in Abbildung 3.3d zu erkennen ist. Dargestellt ist die auf den zweiten partiellen Ableitungen von $\overline{C}_2$ definierte Bewertungsfunktion $\Omega_{C_0,2}$ (vgl. Abschnitt 3.4.1). An den Rändern des dunklen Schriftzugs ist zu erkennen, dass die Funktion wie auch die für $\overline{W}$ definierte Bewertungsfunktion Maxima an Hell-Dunkel-Grenzen besitzt.

Wie in Abbildung 3.2 zu erkennen, sind die Filterkerne aller verwendeten Rechteckfilter aus mehreren Teilregionen zusammengesetzt, denen jeweils ein Gewichtungsfaktor zugeordnet ist. Beispielsweise lässt sich B_y darstellen als $B_y = -1 \cdot B_{y1} + 1 \cdot B_{y2}$, wobei B_{y1} die obere Teilregion und B_{y2} die untere Teilregion bezeichnet.

Allgemeiner formuliert: jeder Rechteckfilter $R : \mathbb{R}^2 \mapsto \mathbb{R}$ mit n Teilregionen lässt sich darstellen als

$$R = \sum_{i=1}^{n} w_i R_i \quad , w_i \in \mathbb{R} \forall i \in [1, 2, \ldots, n].$$

Dabei sind alle R_i ungewichtete Rechteckfilter, d.h. es gilt $R_i(m, n) \in \{0, 1\}\ \forall\, m, n \in \mathbb{R}$. Zur C-Invariante kann man nun einen speziellen Faltungsoperator $\diamond$ definieren, bei dem jede Region einzeln mit dem Intensitätskanal I_1 normalisiert wird:

$$(R \diamond I_k)(x, y, \sigma) = \frac{1}{\sum_{i=1}^{n} |w_i|} \sum_{i=1}^{n} w_i \frac{R(\sigma) \circledast I_k(x, y)}{B(\sigma) \circledast I_1(x, y)}$$

Gilt $I_k(x, y) \leq I_1(x, y) \quad \forall x, y \in \mathbb{R}$, so ist der Wertebereich von $R \diamond I_k$ zudem auf das Intervall $[0, 1]$ beschränkt.

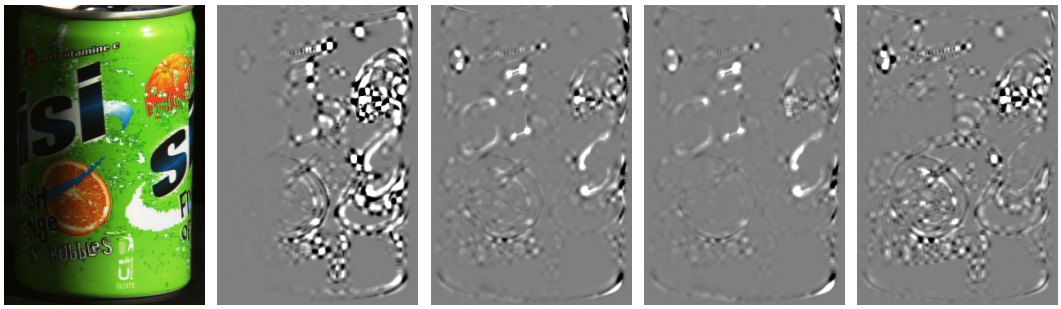
Damit lassen sich die approximierten Invarianten $\overline{C}_{k,x}$, $\overline{C}_{k,xx}$ und $\overline{C}_{k,xy}$, $k \in \{2, 3\}$ wie folgt darstellen ($\overline{C}_{k,y}$ und $\overline{C}_{k,y}$ analog):

$$\overline{C}_{k,x} = \alpha_{G_x} B_x \diamond I_k \tag{3.10}$$

$$\overline{C}_{k,xx} = \alpha_{G_{xx}} B_{xx} \diamond I_k \tag{3.11}$$

$$\overline{C}_{k,xy} = \alpha_{G_{xy}} B_{xy} \diamond I_k \tag{3.12}$$

Das Ergebnis dieser Operation ist in Abbildung 3.3e dargestellt. Die auf den so definierten Invarianten basierte Bewertungsfunktion $\Omega_{\overline{C},3}$ zeigt einen deutlich geringeren Ausschlag an Übergängen zwischen Bereichen verschiedener Intensität, dafür aber an Farbkanten.



(a) Eingabebild **(b)** Ω_2 **(c)** $\Omega_{\overline{W},3}$ **(d)** $\Omega_{\overline{C},3}$ **(e)** $\Omega_{\overline{C},3}$

Abbildung 3.3: Dreidimensionales Objekt mit Schattierungseffekt und die verschiedenen Bewertungsfunktionen zur Detektion von blobartigen Strukturen. Gezeigt ist jeweils der Wert für den zweiten Bildkanal des gaußschen Farbraums und $\sigma = 2$. Die Stärke der Bewertungsfunktion Ω_2 hängt von der Intensität ab. d) wurde nach Gleichungen 3.5-3.7 berechnet, e) nach nach Gleichungen 3.10-3.12

3.4 Erweiterung des Detektors

Zur Erweiterung des Detektors auf Farbbilder werden zwei unterschiedliche Varianten implementiert. Bei der ersten wird auf den Bildkanälen L_i, $i \in [1, 2, 3]$ eine gemeinsame Bewertungsfunktion definiert und nach deren Maxima im Skalenraum gesucht. Die zweite Methode betrachtet die Kanäle getrennt. Beide Varianten werden jeweils auf dem nicht normierten Bildsignal L sowie für die Invarianten W und C betrachtet.

3.4.1 Photometrische Invarianz

Da der Detektor von SURF auf den Ableitungen des Bildsignals arbeitet, ist er invariant gegenüber der Addition einer Konstanten auf alle Bildkanäle, sofern diese für alle Kanäle und alle Pixel des Bildes identisch ist. Damit ist er invariant gegenüber weißem Umgebungslicht a (vgl. Gleichung 3.1).

Gegen eine gleichmäßige Skalierung der Intensität des Eingabesignals, z.B. durch Änderung der Lichtintensität (Skalierung von $e(\lambda)$), besitzt er keine Invarianz, wenn zur Detektion ein fester Schwellenwert auf die Bewertungsfunktion angewandt wird. Dies liegt daran, dass sich durch die Skalierung der Intensitätswerte die Amplitude der Bewertungsfunktion ebenfalls ändert und damit auch die Anzahl der gefundenen Maxima. Dieser Effekt kann z.B. durch eine Kontrastspreizung des Eingabebildes behoben werden, falls kein Umgebungslicht vorhanden ist und keine Überbelichtung stattfindet. In den Experimenten in Kapitel 5 wird eine feste Anzahl von Schlüsselpunkten aus allen Bildern extrahiert. In diesem Fall ist der Detektor invariant gegen eine Skalierung der Intensitäten.

Bei Änderungen der Beleuchtungsrichtung bei dreidimensionalen Objekten trifft die Annahme einer gleichmäßigen Skalierung der Intensitäten durch Schattierung und Ver-

schattung jedoch nicht zu. Dies führt bei dem von SURF verwendeten Schema dazu, dass hauptsächlich Maxima in den stärker beleuchteten Bildbereichen gefunden werden. Dieses ist bei den W- und C-Invarianten unter den in Abschnitt 3.3.1 genannten Annahmen nicht der Fall.

Wird die Detektion direkt auf dem Bildsignal ausgeführt oder die W-Invariante verwendet, kann zudem eine Änderung der Helligkeitsverläufe die Lokalisation der Maxima der Bewertungsfunktion beeinflussen. Dieses trifft unter der Annahmen aus Abschnitt 3.3.2 für C nicht zu.

Die Erweiterung des Detektors erfolgt durch Einsetzen der Invarianten zweiter Ordnung in die Hessematrix. Da $\overline{C}$ nur auf den Farbkanälen definiert ist, werden $\overline{C}_2$ und $\overline{C}_3$ bei der Detektion mit $\overline{W}_1$ kombiniert. Dieses wurde in [BG09] bereits für den Deskriptorschritt von SIFT vorgeschlagen. Die getesteten Kombinationen von Farbkanälen sind sind also $(\overline{L}_1, \overline{L}_2, \overline{L}_3)$, $(\overline{W}_1, \overline{W}_2, \overline{W}_3)$ und $(\overline{W}_1, \overline{C}_2, \overline{C}_3)$.

Für $\overline{W}_k$ ist die Hessematrix $\overline{\boldsymbol{H}}_{W,k}$ demnach folgendermaßen definiert:

$$\overline{\boldsymbol{H}}_{W,k} = \begin{bmatrix} \overline{W}_{k,xx} & \overline{W}_{k,xy} \\ \overline{W}_{k,xy} & \overline{W}_{k,yy} \end{bmatrix}$$

und die Bewertungsfunktion für $\overline{W}$ lässt sich effizient berechnen als:

$$\Omega_{W,k}(x, y, \sigma) = \frac{\overline{L}_{k,xx}\overline{L}_{k,yy} - (w\overline{L}_{k,xy})^2}{\overline{L}_1^2}$$

Da bei kleinen Intensitätswerten der Einfluss von Rauschen auf den Wert der Invarianten steigt, werden Maxima, bei denen die Intensität einen Schwellenwert $i_{\min} = 0,05$ unterschreitet, verworfen. Dabei gilt die Voraussetzung, dass die Intensitätswerte im Intervall $[0, 1]$ liegen.

3.4.2 Kombinierte Detektion

Zur Berechnung von Merkmalen, die auf den Ableitungen von Mehrkanalbildern definiert sind, existieren unterschiedliche Verfahren. Viele davon bauen jedoch auf dem Farbtensor auf, welcher eine Erweiterung der Harrismatrix auf Mehrkanalbilder darstellt [WGS04, Diz86, SGS08c] und sind deshalb nicht direkt auf den Detektionsalgorithmus von SURF anwendbar.

Das Verfahren in [MM07] erweitert einen auf der Hessematrix basierten Blob-Detektor auf Farbbilder, indem die Ableitungen in der Hessematrix durch eine gewichtete Summe der Ableitungen der drei Bildkanäle ersetzt werden:

$$\boldsymbol{H}_\Sigma = \begin{bmatrix} I_{\Sigma,xx} & I_{\Sigma,xy} \\ I_{\Sigma,xy} & I_{\Sigma,yy} \end{bmatrix}$$

$$I_{\Sigma xx} = \omega_1 I_{1_{xx}} + \omega_2 I_{2_{xx}} + \omega_3 I_{3_{xx}}$$

$I_{\Sigma yy}$ und $I_{\Sigma xy}$ sind analog zu $I_{\Sigma xx}$ definiert. Die Gewichte ω_i ergeben sich aus dem Anteil des jeweiligen Kanals an der Signalintensität:

$$\omega_k = \frac{I_k}{I_1 + I_2 + I_3}$$

Bei diesem Vorgehen ist es allerdings möglich, dass sich durch entgegengesetzte Vorzeichen der Ableitungen die Beiträge der einzelnen Kanäle gegenseitig aufheben. Dies wäre beispielsweise der Fall, wenn eine Blobstruktur in zwei Kanälen eine unterschiedliche Polarität aufweist, also in einem Bildkanal einen größeren und in einem anderen Bildkanal einen kleineren Wert als seine Umgebung hat. Um dieses Problem zu beheben, wird in [SFH08] vorgeschlagen, die Farbwerte und damit auch Einträge der Hessematrix als reine Quaternionen (mit Realteil 0) zu repräsentieren:

$$\boldsymbol{H}_Q = \begin{bmatrix} I_{Q,xx} & I_{Q,xy} \\ I_{Q,xy} & I_{Q,yy} \end{bmatrix}$$

$$I_{Qxx} = iI_{1,xx} + jI_{2,xx} + kI_{3,xx}$$

$I_{Q,yy}$ und $I_{Q,xy}$ sind analog zu $I_{Q,xx}$ definiert. Anschließend müssen bei dem vorgeschlagenen Verfahren allerdings die Eigenwerte von H_Q explizit durch eine Singulärwertzerlegung berechnet werden. Dies wird bei SURF durch Berechnung der Determinante vermieden, da diese das Produkt der Eigenwerte ist und damit eine indirekte Aussage über den Krümmungsverlauf liefert. Insbesondere für Matrizen von Quaternionen ist die Berechnung von Eigenwerten ein rechenintensives Verfahren. Im Falle der 2×2-Hessematrix von Quaternionen muss ihre adjunkte komplexe 4×4-Matrix berechnet und auf diese die Singulärwertzerlegung angewandt werden [LBS03]. Da der SURF-Algorithmus auf eine besonders effiziente Berechnung der Merkmale optimiert ist und dies auch für seine Erweiterung auf Farbbilder gelten soll, ist dieses Verfahren ebenfalls nicht geeignet.

Die verwendete Alternative beruht auf einem wesentlich einfacheren Prinzip: die Bewertungsfunktion Ω^{Σ} wird als die Summe der Bewertungsfunktionen $\Omega_k, k \in [1, 2, 3]$ für die einzelnen Kanäle des durch Rechteckfilter approximierten Skalenraums definiert (vgl. Gleichungen 2.9 - 2.15). Ist in einem Kanal k eine Blobstruktur präsent, so besitzt Ω_k an dieser Stelle ein lokales Maximum. Blobstrukturen, die in mehreren Kanälen vorhanden sind, verstärken sich somit unabhängig von ihrer Polarität gegenseitig.

$$\overline{\boldsymbol{H}}_k = \begin{bmatrix} \overline{L}_{k,xx} & \overline{L}_{k,xy} \\ \overline{L}_{k,xy} & \overline{L}_{k,yy} \end{bmatrix}$$

$$\det \boldsymbol{H}_{k,\text{norm}} \approx \Omega_k = \frac{\overline{L}_{k,xx}\overline{L}_{k,yy} - (w\overline{L}_{k,xy})^2}{\sigma^4}$$

$$\Omega_{\Sigma} = \sum_{k=1}^{3} \Omega_k$$

3.4.3 Kanalweise Detektion

Das zweite Detektionsverfahren besteht aus einer separaten Suche von Maxima in allen Bildkanälen. Die Suche auf einem Kanal ist dabei identisch zum Detektionsschritt der Graustufenvariante von SURF. Zusätzlich wird jedoch die (nicht interpolierte) Lokalisierung von bereits gefundenen Maxima in einer Karte gespeichert. Wird an der selben Stelle im Skalenraum ein Maximum in mehreren Kanälen gefunden, wird nur ein einziger Schlüsselpunkt erzeugt.

Jedem Schlüsselpunkt wird in SURF eine Stärke zugewiesen, welche dem Wert der interpolierten Bewertungsfunktion an der Stelle des gefundenen Maxmimums entspricht. In dieser Variante wird diese als das Maximum der interpolierten Bewertung über alle gefundenen kanalweisen Maxima definiert. Zusätzlich wird das Vorzeichen der Spur der Hessematrix ausgegeben, welche beim Vergleich der Schlüsselpunkte als Kriterium zur Vorauswahl für mögliche Zuordnungen dienen kann. Dieses wird ebenfalls für den Kanal berechnet, in dem das Maximum gefunden wurde.

3.4.4 Farbverstärkung

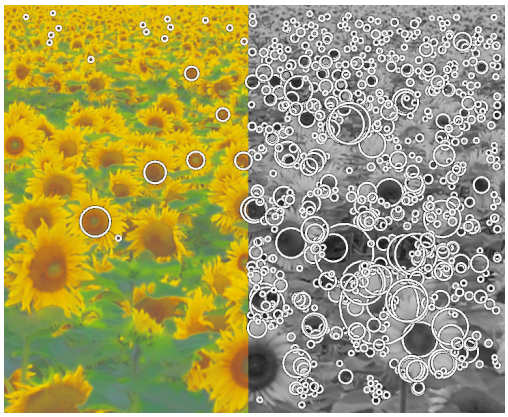
In [WGB06] werden zwei der Kriterien für die Güte eines Merkmalsdetektors aus [SMB00] auf Farbbilder angewandt: Reproduzierbarkeit und Unterscheidungskraft. Reproduzierbarkeit bedeutet, dass die Lokalisierung der Merkmale sich auch bei einer Änderung der Aufnahmebedingungen möglichst wenig ändert. Unterscheidungskraft bedeutet, dass bevorzugt Merkmale an Stellen gefunden werden, die eine besonders hohe Informationsdichte besitzen. Ein Merkmalsdetektor, der auf Graustufenbildern arbeitet, lässt die Informationsdichte der Farbkanäle außer Acht. Wird dieser auf Farbbilder erweitert, sollte die Gewichtung der Kanäle entsprechend ihrer Informationsdichte ausgeglichen sein.

Durch statistische Auswertung einer Bilddatenbank stellen die Autoren fest, dass die Farbkanäle im RGB-Gegenfarbraum relativ zu ihrer Informationsdichte eine geringere Skalierung im Vergleich mit dem Intensitätskanal aufweisen. Als Ausgleich werden die Farbwerte so skaliert, dass der Informationsgehalt für alle Kanäle gleich ist. Gleichzeitig stellen die Autoren allerdings fest, dass diese Transformation die Reproduzierbarkeit des Detektors bei Präsenz von Rauschen negativ beeinflusst, da sich dadurch das Signal-Rausch-Verhältnis des Eingabesignals verschlechtert.

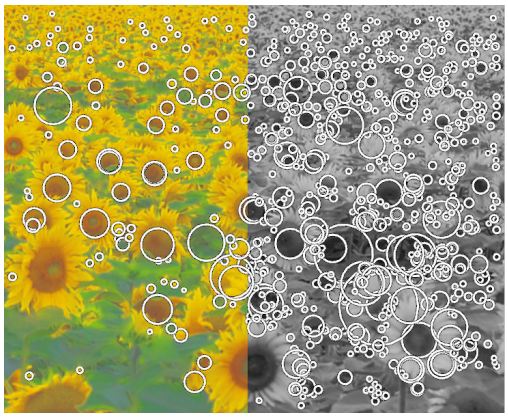
In [SGS08a] wird gezeigt, dass die Kombination von verschiedenen um Farbinformationen erweiterten SIFT-Varianten mit einem Detektor, der die Farbverstärkung verwendet, die Klassifikationsleistung für eine Objektdatenbank um bis zu 30% gegenüber der Verwendung des rein intensitätsbasierten SIFT-Deskriptors verbessert. Eine getrennte Betrachtung der Effekte von Detektor und Deskriptor wird hier allerdings nicht vorgenommen.

In dem hier vorgeschlagenen Verfahren wird eine vereinfachte Variante der Farbverstärkung verwendet, wobei der Intensitätskanal in allen Farbräumen mit dem Faktor $0,5$ skaliert wird. Dies bedeutet eine geringere Verstärkung des Farbsignals als durch die in [WGB06] vorgeschlagenen Werte und stellt somit einen Kompromiss zwischen Rauschabstand und Informationsdichte dar.

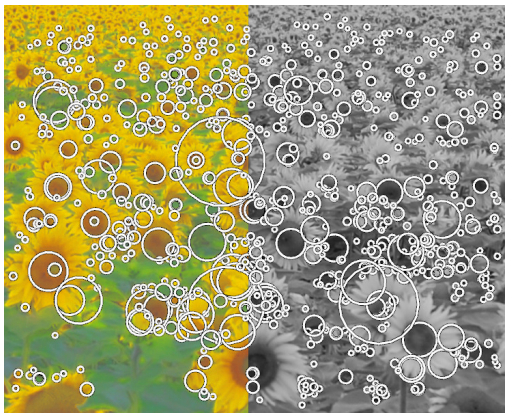
(a) Eingabebild



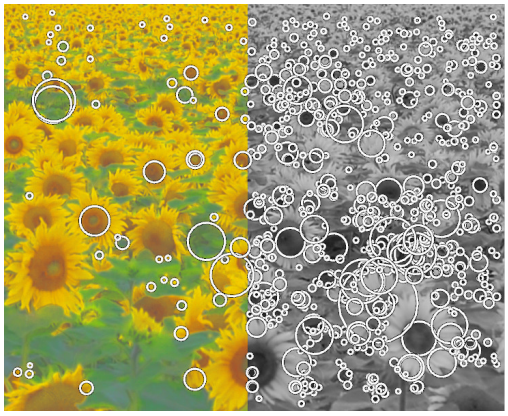
(b) Detektierte Schlüsselpunkte im Graustufenbild



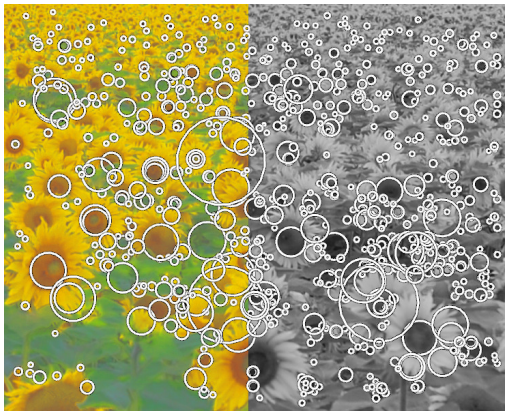
(c) Detektierte Schlüsselpunkte mit dem Summenverfahren



(d) Detektierte Schlüsselpunkte mit dem Summenverfahren und Farbverstärkung



(e) Kanalweise detektierte Schlüsselpunkte



(f) Kanalweise detektierte Schlüsselpunkte mit Farbverstärkung

Abbildung 3.4: Ergebnisse von unterschiedlichen Verfahren zur Detektion von Schlüsselpunkten im gaußschen Farbraum. In der linken Bildhälfte wurde der Helligkeitskontrast reduziert, während in der rechten Bildhälfte der Farbkontrast auf Null gesetzt wurde.

In Abbildung 3.4 sind die Ergebnisse der Farbdetektoren mit und ohne Farbverstärkung im Vergleich zum SURF-Detektor beispielhaft dargestellt.

3.5 Orientierungszuweisung und Deskriptor

Zur Zuweisung einer Orientierung an die gefundenen Schlüsselpunkte werden ebenfalls zwei Verfahren verglichen. Beim ersten Verfahren werden die Gradienten $\nabla L_k = (L_{k,x}, L_{k,y})^{\mathrm{T}}$ zuerst summiert:

$$\nabla L_k^{\Sigma} = \sum_{k=1}^{3} \begin{pmatrix} L_{k,x} \\ L_{k,y} \end{pmatrix}$$

Anschließend wird mit den Gradienten ∇L_k^{Σ} das gleiche Verfahren wie zur Bestimmung der Orientierung in Grauwertbildern verwendet (vgl. Abschnitt 2.4.3). Für die Invarianten W und C ist der summierte Gradient analog zu ∇L_k^{Σ} definiert, indem L durch die jeweilige Invariante ersetzt wird.

Das zweite Verfahren berechnet die Gradienten kanalweise, speichert sie jedoch in einer gemeinsamen Liste. Auf diese Liste kann anschließend ebenfalls der Originalalgorithmus angewandt werden.

Das Summenverfahren bietet den Vorteil, dass die sortierte Gradientenliste im Vergleich zum Grauwertverfahren nicht größer wird und dadurch die Rechenzeit gering gehalten wird. Es besteht jedoch die Gefahr, dass sich die Gradienten der drei Kanäle in der Summe gegenseitig aufheben, was zu einer erhöhten Instabilität führen kann.

Der Deskriptor wird in seiner Form beibehalten. An den Merkmalsvektor für den Intensitätskanal werden gleichartige Deskriptoren angehängt, welche auf den beiden anderen Bildkanälen berechnet werden. Die Anzahl der Teilregionen, in die das Deskriptorfenster eingeteilt wird, und damit die Länge des resultierenden Merkmalsvektor, kann für die drei Bildkanäle unabhängig gewählt werden.

Sowohl die Orientierungszuweisung als auch die Berechnung des Deskriptors beruhen auf den partiellen Ableitungen ersten Grades des Bildsignals. Daher werden in beiden Schritten neben den ursprünglichen Bildkanälen $(\overline{L}_1, \overline{L}_2, \overline{L}_3)$ alternativ die Kombination der invarianten Bildkanäle $(\overline{W}_1, \overline{W}_2, \overline{W}_3)$ oder $(\overline{W}_1, \overline{C}_2, \overline{C}_3)$ zur Berechnung verwendet.

Bei SURF wird die Länge des Merkmalsvektors normiert, wodurch er invariant gegenüber Änderungen der Beleuchtungsintensität ist, sofern diese das gesamte Deskriptorfenster gleichmäßig betrifft. Da nur Ableitungen des Bildsignals zur Berechnung verwendet werden, ist er zudem invariant gegenüber der Addition einer Konstanten. Dies ist auch anwendbar auf den zusammengesetzten Merkmalsvektor, der die Deskriptoren der drei Kanäle von L enthält. Bei Verwendung der W- und C-Invarianten wird dieser Schritt jedoch ausgelassen, da diese bereits normiert sind.

Wie bereits im Detektionsschritt muss die Instabilität der Invarianten bei kleinen Intensitätswerten berücksichtig werden. Daher sind die Ableitungen bei der Berechnung des Deskriptors und bei der Orientierungszuweisung als Null definiert, falls die Intensität den Schwellenwert $i_{\min}$ unterschreitet.

Kapitel 4

Implementation

Die Implementation des Objekterkennungssystems erfolgte in C++ und baut auf der Softwarearchitektur auf, die für das Robotersystem Robbie entwickelt wurde [PGP09]. Die Applikation ist dabei in eine Reihe von Modulen unterteilt, die über Nachrichten miteinander kommunizieren. Welche Module geladen werden, wird zur Laufzeit bestimmt und lässt sich für unterschiedliche Profile konfigurieren. Dies erlaubt eine flexible Konfiguration und ermöglicht zudem einen vereinfachten Austausch von Einzelkomponenten.

Die Module stellen dabei hauptsächlich die Schnittstelle zu den anderen Softwarekomponenten sowie einen Verknüpfungspunkt für die verschiedenen Algorithmen dar. Die eigentliche Funktionalität ist in eine Reihe von Klassen verkapselt, die als *Worker* bezeichnet werden. Besteht die Hauptaufgabe einer Klasse darin, mit Hardwarekomponenten zu kommunizieren, wird sie *Device* genannt. Die im Rahmen dieser Diplomarbeit implementierten Algorithmen zur Extraktion von Farbmerkmalen und zur Objekterkennung sind in diesem Schema als Worker zu bezeichnen.

Zur Evaluation der Objekterkennung wurde eine spezielle Softwarekomponente in dieses System eingebunden. Die isolierte Evaluation der Merkmale wurde in Matlab implementiert.

4.1 Detektor und Deskriptor

Als Ausgangsbasis für die Implementation des Detektors und Deskriptors für Farbmerkmale wurde eine quelloffene Implementation von SURF gewählt. Grund für diese Entscheidung ist, dass der Fokus dieser Arbeit nicht auf dem SURF-Algorithmus selbst, sondern seinen möglichen Erweiterungen liegt. Zudem werden in den Veröffentlichungen zum SURF-Algorithmus [BTVG06, BETG08] Teile des Algorithmus nur unzureichend beschrieben. In Kapitel 5 wird an mehreren Beispielen gezeigt, dass die Ergebnisse der Originalimplementation durch Umsetzen der Angaben in der Veröffentlichung nicht reproduziert werden können.

Die verwendete Implementation ist Teil der in C++ geschriebenen Software Pan-o-matic [Orl09]. Sie wurde ausgewählt, da sie die Ergebnisse der Originalimplementation in

einer Evaluation (siehe Kapitel 5) am genauesten reproduzieren konnte. Zudem besitzt sie einen modularen Aufbau und eignet sich daher, um verschiedene Erweiterungen in einem Rahmenwerk zu implementieren und testen. Obwohl sie Teil eines Anwenderprogrammes ist, besitzt sie keine Abhängigkeiten zum Rest der Software und lässt sich dadurch ohne zusätzlichen Aufwand aus dieser herauslösen.

4.1.1 Klassenhierarchie

Abbildung 4.1 zeigt eine schematische Abbildung der Klassen, die die Kernfunktionen des Algorithmus zur Extraktion von lokalen Merkmalen bereitstellen. Der Übersichtlichkeit halber sind nur die wichtigsten Methoden und Eigenschaften dargestellt. Eigenschaften, die als `private` markiert sind, werden durch den Konstruktor und entsprechende `get`- und `set`-Funktionen abgefragt und geändert. Diese Funktionen sind ebenfalls in der Darstellung nicht enthalten. Ausnahme bildet die Klasse `ColorVec`, auf deren Eigenschaften direkt zugegriffen werden kann. Soweit nicht anders angegeben, sind Funktionsparameter Eingabewerte. Typenbezeichner wurden weggelassen, wenn sie aus dem Kontext und den Variablennamen ersichtlich sind.

Bilder werden durch die Klasse Image repräsentiert. Sie besitzt eine Funktion `filter`, der als Template-Parameter eine weitere Funktion übergeben wird, welche dadurch auf jeden Pixel des Bildes angewandt wird. Diese Funktionalität wird zur Konvertierung von Bildern zwischen verschiedenen Farbräumen verwendet. Dazu sind im Namensraum `colorspace` mehrere Funktionen definiert, die jeweils einen Farbwert mittels einer Matrixmultiplikation konvertieren (`linRgbToGauss`, `linRgbToYCrCb` etc.). Da der Algorithmus auf linearen Farbwerten arbeitet, wird eine Funktion `linearizeSRgb` angeboten, mit der sich Gamma-korrigierte Bilder nach dem sRGB-Standard [BT702] linearisieren lassen.

Die Klasse `IntegralImage` ist von `Image` abgeleitet. Sie besitzt keine zusätzlichen Eigenschaften, übernimmt jedoch die Berechnung des Bildintegrals aus einem normalen Bild der Klasse `Image`.

Die Detektion der Schlüsselpunkte geschieht in der Klasse `KeyPointDetector`. Die Eigenschaft `minIntensity` gibt den Schwellenwert für die Bildintensität an, unter dem bei Verwendung der W- und C-Invarianten gefundene Schlüsselpunkte verworfen werden. Die Mehrkanal-Detektion mit dem Summenverfahren wird durch die Funktion `detectKeyPointsSum` realisiert, die Kanalweise Detektion über `detectKeyPointsSeparate`.

Die abstrakte Klasse `KeyPointInsertor` übernimmt die Speicherung der gefundenen Schlüsselpunkte. Sie ist eine reine Interfaceklasse. Vom Benutzer muss eine abgeleitete Klasse definiert werden, die den Klammer-Operator mit einem `KeyPoint`-Objekt als Parameter erhält. `KeyPoint` ist außerhalb von `colorsurf` spezifiziert und ist die Klasse, die im Objekterkennungssystem Schlüsselpunkte repräsentiert (siehe Abschnitt 4.2.1).

Die Bewertungsfunktionen zur Detektion von der lokalen Merkmale werden von der Klasse `BoxFilter` bereit gestellt. Die Methode `getDet` bzw. `getDetSum` berechnet die Bewertungsfunktion für einen Kanal (Ω_k) bzw. die Summe der Bewertungsfunktionen (Ω_Σ). Über den Parameter `normType` kann festgelegt werden, ob und welche Invariante

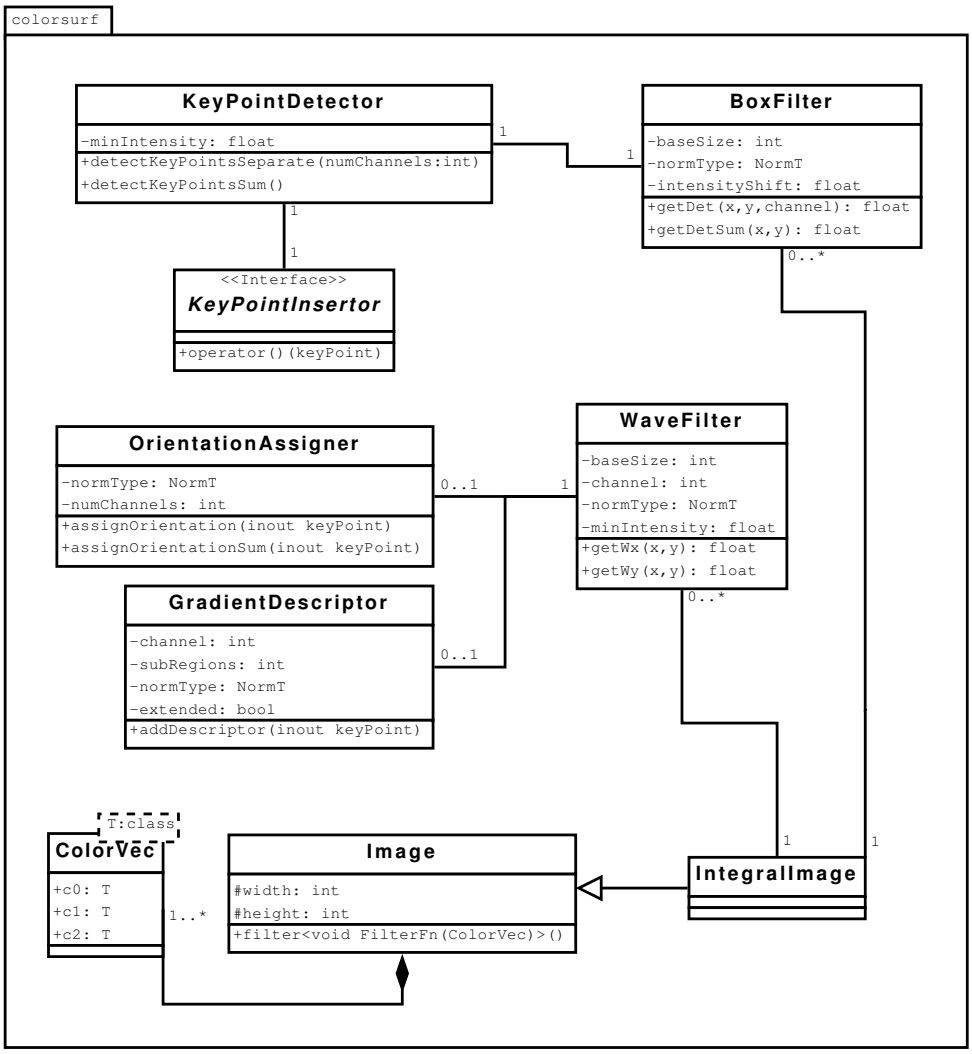


Abbildung 4.1: Klassen zur Berechnung der Farbmerkmale

zur Berechnung verwendet wird. Die Skala wird über den Parameter `baseSize` festgelegt. Für $\sigma = 1,2$ ist `baseSize`$= 3$.

Schlüsselpunkte können einzeln an die Klasse `OrientationAssigner` übergeben werden. Diese berechnet mit der durch `normType` angegebenen Invariante eine Orientierung und weist sie den Schlüsselpunkt zu. Anschließend kann der Schlüsselpunkt an `GradientDescriptor` übergeben werden, welcher den Descriptor für einen Kanal berechnet. Soll ein zusammengesetzter Deskriptor über mehrere Kanäle berechnet werden, muss die Funktion `addDescriptor` mehrmals hintereinander aufgerufen werden, jeweils mit anderen Werten für `channel`. Beide Klassen greifen zur Berechnung der horizontalen und vertikalen Ableitung des Bildsignals auf die Klasse `WaveFilter` zurück.

4.2 Objekterkennungssystem

Das Objekterkennungssystem basiert auf einer modularen Architektur. Die Funktionalität ist dabei über mehrere parallel ausgeführte Module verteilt, welche über Nachrichten kommunizieren. Die eigentlichen Extraktionsalgorithmen werden dabei durch generische Schnittstellen und Datenstrukturen angebunden, welche gleichzeitig eine Parallelisierung des Deskriptorschritts erlauben. Die Software enthält eine grafische Benutzeroberfläche, welche die Steuerung des Systems sowie die Visualisierung der generierten Daten erlaubt.

Eine Anleitung zur Bedienung und Konfiguration der Software findet sich in [Thi09].

4.2.1 Wrapper-Klassen für die Merkmalsextraktion

Da verschiedene Implementationen von SURF verglichen werden, wurde eine Reihe von Klassen implementiert, die die Funktionalität der einzelnen Verfahren abkapseln (siehe Abbildung 4.2). Die Kombination aus Detektor und Deskriptor wird Extraktor genannt. Die rein abstrakte Basisklasse für alle Extraktoren heißt `KeyPointExtractor`. Die durch sie definierte Schnittstelle beinhaltet Funktionen, mit denen das Eingabebild übergeben und die Extraktion der Merkmale ausgeführt werden kann.

Für alle Implementationen des SURF-Algorithmus ist die abstrakte Klasse `SurfExtractorBase` als Zwischenstufe in der Klassenhierarchie definiert. Sie beinhaltet eine Reihe von Parametern für den SURF-Algorithmus und die zugehörigen `get-` und `set`-Methoden. Diese werden bei der Initialisierung des Objekts aus einer zentralen Konfigurationsdatenbank gelesen.

Die eigentliche Funktionalität wird von den Klassen implementiert, die von `SurfExtractorBase` abgeleitet sind. Dies ist unter anderem `SurfExtractor` für die Originalimplementation von SURF, `PanoSURFExtractor` für die Implementation aus Pano-matic und `ColorSURFExtractor` für die um Farbmerkmale erweiterte Implementation. `ColorSURFExtractor` beinhaltetet zudem eine Reihe von Parametern, die denen aus Abschnitt 4.1.1 entsprechen, und ebenfalls aus der Konfigurationsdatenbank ausgelesen werden.

Um sicherzugehen, dass in allen Teilen des Systems der gleiche Algorithmus zur Extraktion benutzt wird, existiert die Klasse `DefaultExtractor`. Diese erzeugt auf Anfrage

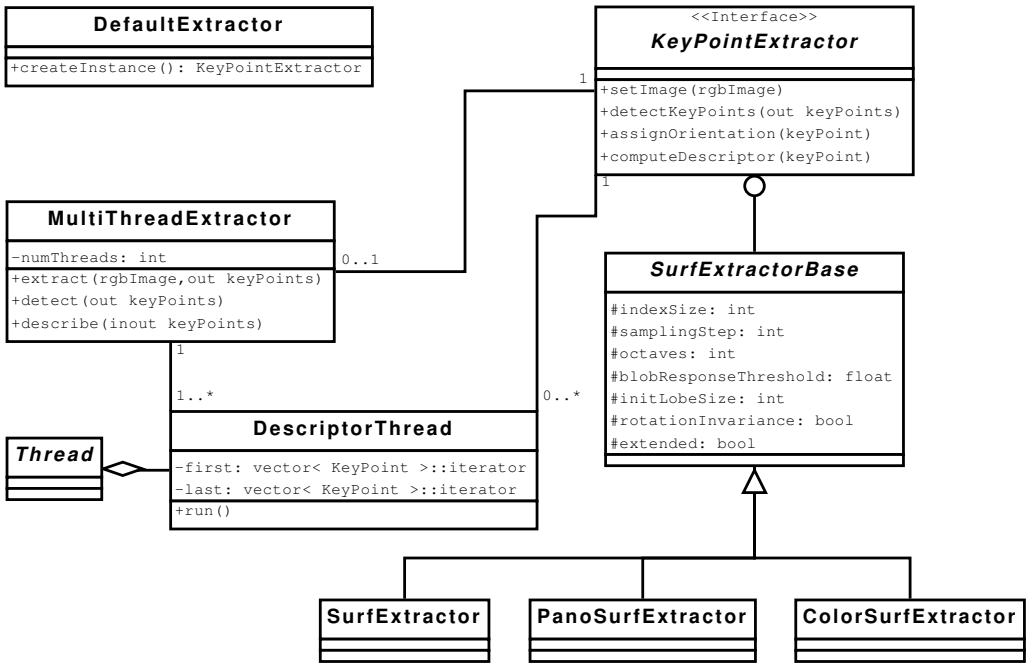


Abbildung 4.2: Wrapper-Klassen zur Kapselung von Detektoren und Deskriptoren für lokale Merkmale

eine Instanz des Extraktors, welchen den in der Konfiguration angegebenen Algorithmus implementiert. Schlüsselpunkte und ihre Merkmale werden im gesamten System durch die Klasse `KeyPoint` repräsentiert, welche in Abbildung 4.3 dargestellt und in Abschnitt 4.2.3 beschrieben wird.

4.2.2 Parallelisierung des Deskriptors

Die Extraktion von lokalen Bildmerkmalen ist rechenaufwändig (vergleiche Abschnitt 5.4) und stellt daher einen Flaschenhals in Robotik-Anwendungen dar. Daher ist eine weitgehende Parallelisierung des Algorithmus erstrebenswert. Dies wird für die Berechnung der Deskriptoren durch die Klasse `MultiThreadExtractor` (vgl. Abbildung 4.2) realisiert. Ihr wird eine Instanz eines Extraktors übergeben. Durch die Methode `describe` wird eine wählbare Anzahl an Instanzen von `DescriptorThread` erzeugt. Jede dieser Instanzen erzeugt einen eigenen Thread. Darin werden für einen Teil der übergebenen Schlüsselpunkte die Funktionen zur Berechnung der Orientierung und des Merkmalsvektors des zugewiesenen Extraktors aufgerufen. Die Ergebnisse werden in eine gemeinsame Datenstruktur geschrieben. Da die einzelnen Threads auf unterschiedliche Teile der Datenstruktur zugreifen, ist keine gesonderte Verwaltung der Zugriffsrechte nötig.

Diese Architektur hat den Vorteil, dass sie eine Parallelisierung des Deskriptors erlaubt, ohne in diesen eingreifen zu müssen. Damit ist sie auch auf nicht quelloffene Imple-

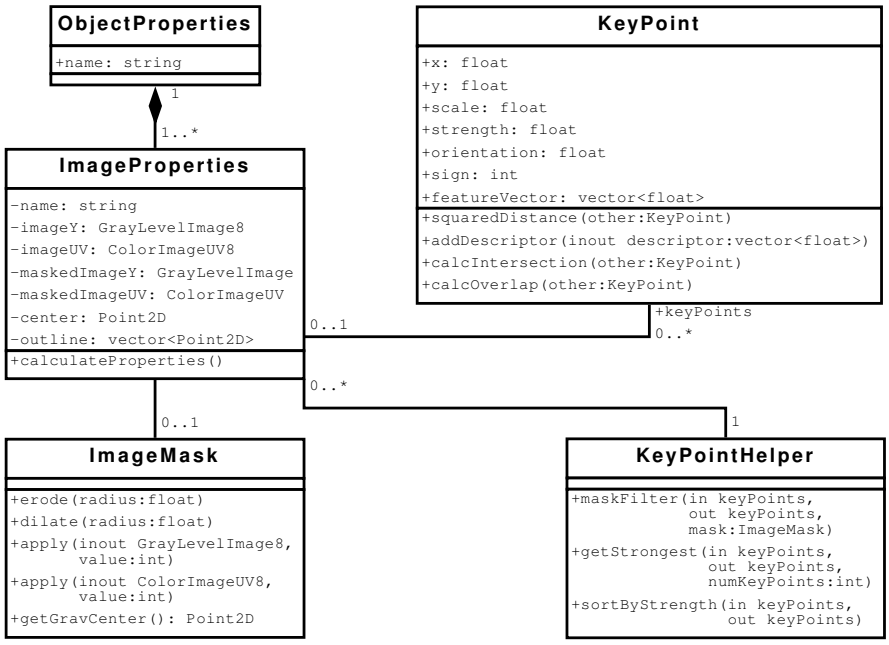


Abbildung 4.3: Klassen zur Verwaltung von Schlüsselpunkten und Objekteigenschaften

mentation anwendbar. Nachteil hierbei ist allerdings, dass der Detektorschritt auf dieser Ebene nicht parallelisiert werden kann. Bei Implementationen, die während dem parallelisierten Teil des Algorithmus schreibend auf interne Objektdaten zugreifen, müssten mehrere Instanzen des Extraktors erzeugt werden. Da dies bei keiner der getesteten Implementationen der Fall ist und dadurch zusätzliche Daten kopiert werden müssen, wurde diese Funktionalität jedoch nicht implementiert.

4.2.3 Verwaltung von Objektdaten

Objekte werden durch eine Reihe von Bildern repräsentiert, welche in der Klasse `ObjectProperties` zusammengefasst werden (Abbildung 4.3). Dies sind üblicherweise Bilder des Objekts, die unter verschiedenen Blickrichtungen aufgenommen wurden. Die Bilddaten werden dabei im $YCrCb$-Format abgelegt. Die Bildkanäle werden getrennt in Objekten der Klassen `GrayLevelImage8` (Y-Kanal) und `ColorImageUV8` abgelegt, welche Teil der Bibliothek *Puma2* sind. Zusätzlich kann eine Bildmaske (`ImageMask`) abgegeben werden, die die Unterscheidung von Objekt und Hintergrund erlaubt.

`ImageProperties` besitzt Methoden zur Serialisierung und Deserialisierung (im Diagramm nicht dargestellt), welche die Bilddaten in einen Datenstrom schreiben. Dies ermöglicht das Speichern und Einlesen des Objekts von einem Datenträger oder die Übertragung über ein Netzwerk.

Die Eigenschaft `center` gibt das geometrische Zentrum der Objektpixel an und wird für das Hough Clustering benötigt. Die Eigenschaft `outline` beinhaltet den Umriss der

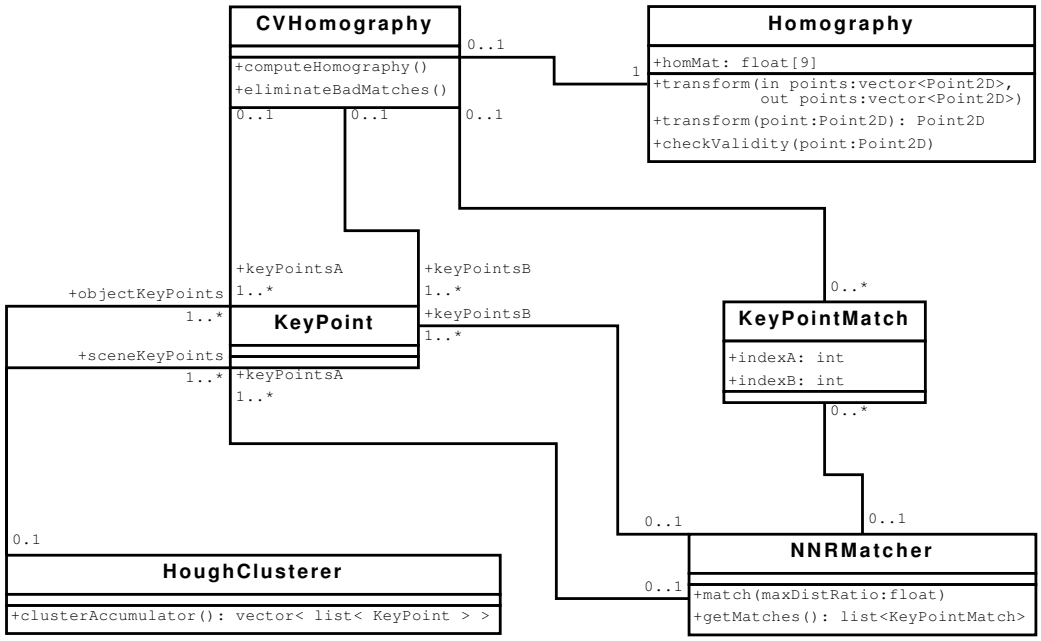


Abbildung 4.4: Klassen zur Detektion von Objekten in Kamerabildern

Bildmaske und wird in der grafischen Schnittstelle des Systems zur Visualisierung verwendet.

Die in `keyPoints` abgeleten Schlüsselpunkte werden mit dem `DefaultExtractor` berechnet (vgl. Abschnitt 4.2.1). Anschließend werden alle Schlüsselpunkte verworfen, die im maskierten Bildbereich liegen. Sie werden durch die Klasse `KeyPoint` repräsentiert. In ihr ist der Merkmalsvektor (`featureVector`), die Position im Skalenraum (`x`,`y`, `scale`) sowie die Orientierung (`orientation`) gespeichert.

Zusätzlich gibt `strength` die Stärke eines Schlüsselpunktes an, welche den Wert der Bewertungsfunktion an der Stelle des gefundenen Maximums im Skalenraum angibt. Diese Eigenschaft wird im Detektionsschritt verwendet, um Maxima mit geringem Kontrast zu verwerfen. Sie wurde in der Evaluation dazu verwendet, nachträglich einen Schwellenwert auf die Bewertungsfunktion anzuwenden, um eine feste Anzahl von Schlüsselpunkten für alle verglichenen Implementationen und Verfahren zu erhalten. Dies wird durch die Methoden `sortByStrength` und `getStrongest` von `KeyPointHelper` ermöglicht. `sortByStrength` sortiert eine Liste von Schlüsselpunkten nach deren Stärke, während `getStrongest` die stärksten n Schlüsselpunkte ausgibt.

Die Eigenschaft `sign` gibt im Falle von SURF das Vorzeichen der Spur der Hessematrix an und wird bei der Objektdetektion zur Indizierung der Schlüsselpunkte verwendet.

4.2.4 Detektion von Objekten in Kamerabildern

Zur Detektion eines Objekts geschieht durch Zuordnung der Schlüsselpunkte im Kamerabild (auch als *Szene* bezeichnet) zu den Schlüsselpunkten in dem jeweiligen Objektbild aus einer Instanz von `ObjectProperties`. Die initiale Zuordnung anhand der Merkmalsvektoren geschieht in der Klasse `NNRMatcher`. Diese erhält zwei Listen von Schlüsselpunkten, `keyPointsA` und `keyPointsB`. Durch Aufruf von `match` vergleicht sie jeden Schlüsselpunkt aus `keyPointsA` mit allen Schlüsselpunkten aus `keyPointsB`, welche das selbe Vorzeichen haben. Als Abstandsmaß dient der euklidische Abstand. Da dieser Vergleichsschritt einseitig stattfindet, werden anschließend in einem separaten Schritt Mehrfachzuordnungen verworfen.

Die so gewonnene Liste von Zuordnungen wird anschließend an die Klasse `HoughClusterer` übergeben. Diese berechnet bei Aufruf von `clusterAccumulator` alle Teilmengen der Zuordnungen, welche jeweils eine konsistente Aussage über die Pose des Objekts im Kamerabild treffen. Die Pose besteht hier aus der Position des Objektschwerpunkts im Kamerabild (vgl. Abschnitt 4.2.3) sowie der relativen Rotation und Skalierung zwischen Objekt- und Kamerabild. Dieser Teil des Objekterkennungssystems wurde im Rahmen von [Thi09] implementiert.

Jeder der berechneten Posen entspricht eine Liste von Zuordnungen. Diese Listen werden im letzten Schritt der Objekterkennung an `CvHomography` übergeben. Diese greift auf Funktionen von OpenCV zurück, um eine Homographie zwischen Kamera- und Objektbild zu bestimmen und die gefundenen Zuordnungen so zu verifizieren.

4.2.5 Module

Die Funktionalität des Objekterkennungssystems ist über mehrere Module verteilt, welche in Abbildung 4.5 dargestellt sind. Jedes Modul stellt eine Schnittstelle für die angebundenen Worker zu den anderen Programmteilen dar. Das `ORLoaderModule` übernimmt das Laden von Objektdatensätzen von der Festplatte. Das `ORControlModule` übernimmt die übergeordnete Kontrolle der Objekterkennung. Dabei ist es möglich, eine konfigurierbare Anzahl von Bildern gleichzeitig in einer Art Pipeline zu halten, so dass das ImageGrabberModule bereits neue Bilddaten aufnehmen kann, während die Objekterkennung auf dem aktuellen Kamerabild noch nicht abgeschlossen ist.

Abbildung 4.6 zeigt für einen Anwendungsfall die Interaktion der verschiedenen Module der Objekterkennung. Der Benutzer lädt eine Objektdatei über die grafische Benutzeroberfläche (*GUI*), welche eine `ORCommandM` an das `ORLoaderModule` sendet. Dieses lädt die Objektdatei und schickt sie weiter an das `ORMatchingModule`. Anschließend gibt der Nutzer den Befehl, ein Kamerabild aufzunehmen. Die entsprechende Nachricht wird vom `ORControlModule` empfangen. Dieses signalisiert dem `ORMatchingModule`, dass es auf ein Kamerabild von der durch den Nutzer ausgewählten Kamera warten soll. Gleichzeitig schickt sie eine Nachricht an das `ImageGrabberModule`, welches das Kamerabild aufnimmt und es ebenfalls an das `ORMatchingModule` schickt. Dieses sucht das zuvor geladene Objekt im Kamerabild und schickt eine Nachricht mit den Ergebnissen zurück an die GUI, wo sie visualisiert werden.

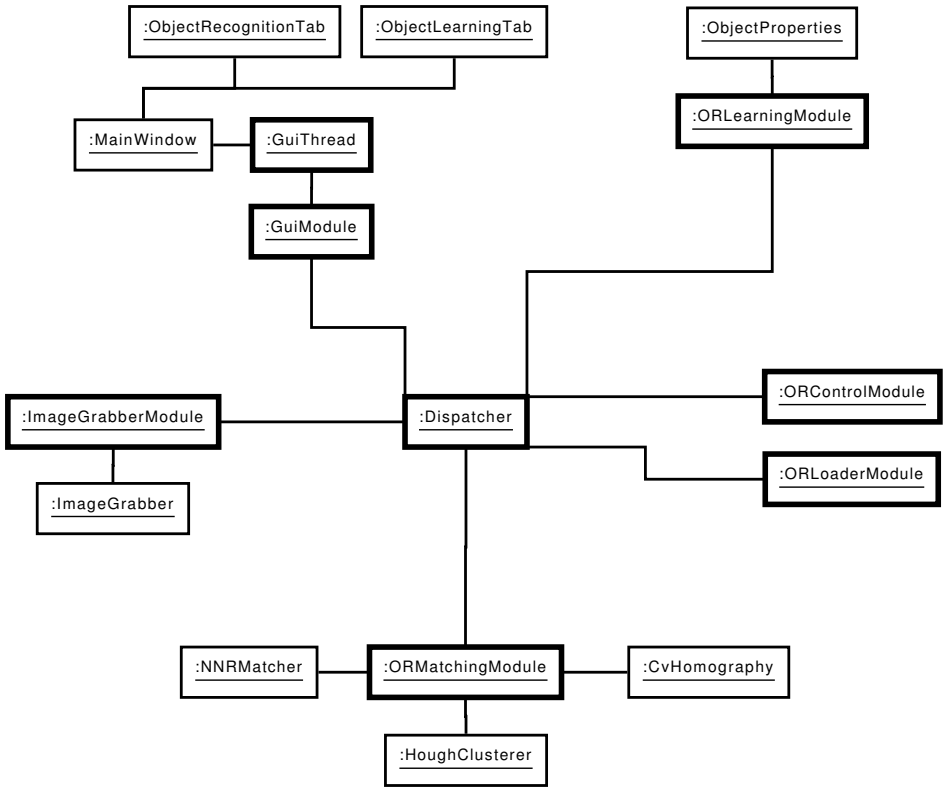


Abbildung 4.5: Relationen der Module und damit verbundenen Worker zur Laufzeit. Die Knoten entsprechen Instanzen der jeweiligen Klassen. Der Übersichtlichkeit halber wurden die Instanznamen weggelassen. Alle fett markierten Objekte sind aktiv, das heißt sie werden in einem eigenständigen Thread ausgeführt.

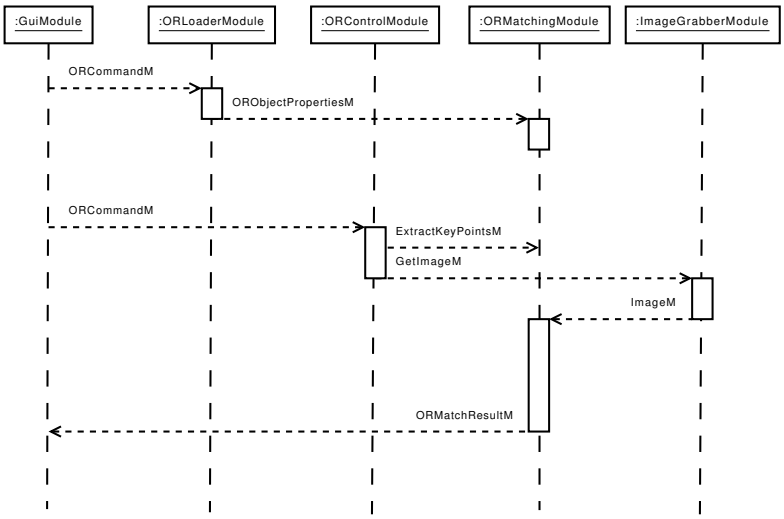


Abbildung 4.6: Interaktionsszenario, bei dem der Benutzer einen Objektdatensatz lädt und die Analyse eines Kamerabildes anfragt.

Das beschriebene Interaktionsschema trägt entscheidend dazu bei, dass die Komplexität der einzelnen Module gering gehalten wird. Beispielsweise muss das `ORMatchingModule` nicht unterscheiden, ob das zu analysierende Bild von Festplatte geladen oder von einer Kamera aufgenommen wird.

Das `ORLearningModule` verwaltet die Schritte zur Erstellung von Objektdatensätzen. Alle Benutzereinstellungen werden von der GUI an das `ORLearningModule` weitergeleitet. Dieses nimmt die nötigen Berechnungen vor und schickt ein Ergebnisbild zurück, durch das der Benutzer die Resultate kontrollieren kann. Die Auslagerung in ein Modul bietet an dieser Stelle den Vorteil, dass die GUI während der Berechnung nicht blockiert wird und weiter auf Benutzereingaben reagieren kann.

4.2.6 Benutzerschnittstelle

In Abbildung 4.7 ist die Benutzerschnittstelle dargestellt. Die untere Kontrollleiste erlaubt es, einzelne Bilder aufzunehmen oder aus einer Datei zu laden, auf denen anschließend die Objekterkennung ausgeführt wird. Außerdem kann eine Schleife gestartet werden, in der kontinuierlich Kamerabilder aufgenommen und verarbeitet werden.

Die zu verwendenden Objektdatensätze werden über die rechte Spalte geladen. Dort befinden zudem sich eine Reihe von Kontrollelementen, mit denen die Visualisierung der Zwischen- und Endergebnisse gesteuert wird. Dies erlaubt beispielsweise das Anzeigen der Deskriptorfenster, Orientierungen und Skalierungen der Schlüsselpunkte, die im Kamerabild gefunden wurden.

Die in den verschiedenen Teilschritten der Erkennung vorhandenen Zuordnungen zwischen Schlüsselpunkten im Kamera- und Objektbild können ebenfalls visualisiert werden. Die Schlüsselpunkte und der Umriß des Objektbildes werden durch die zuvor bestimm-

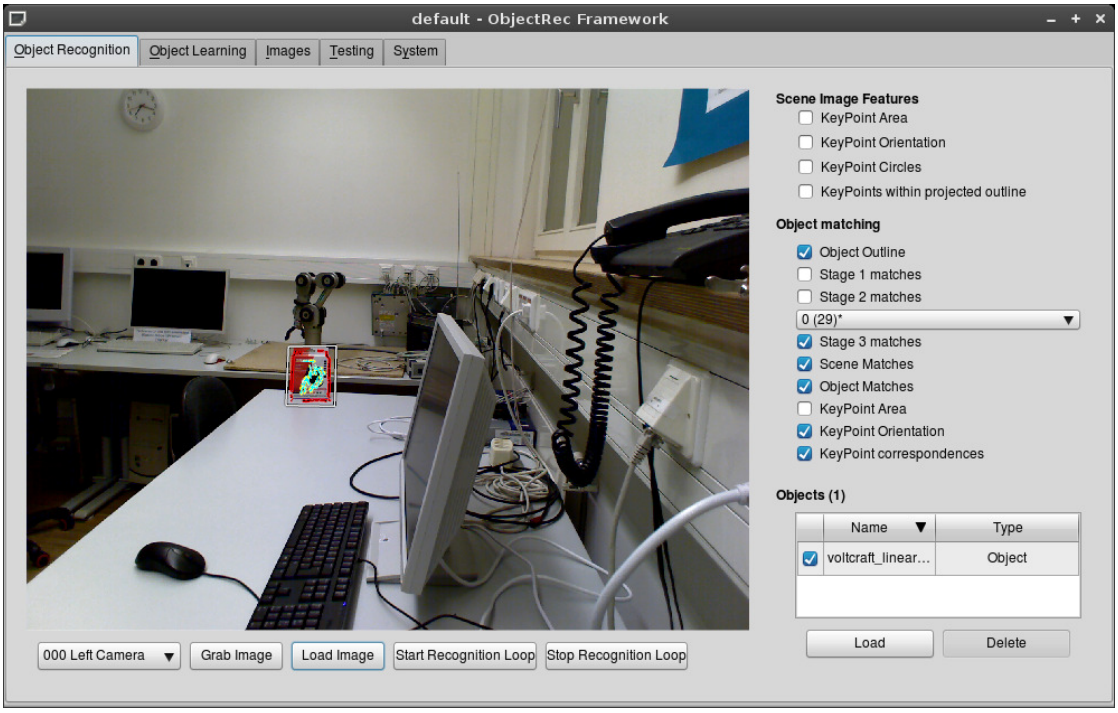


Abbildung 4.7: Grafische Oberfläche zur Steuerung der Objekterkennung und Visualisierung der Ergebnise

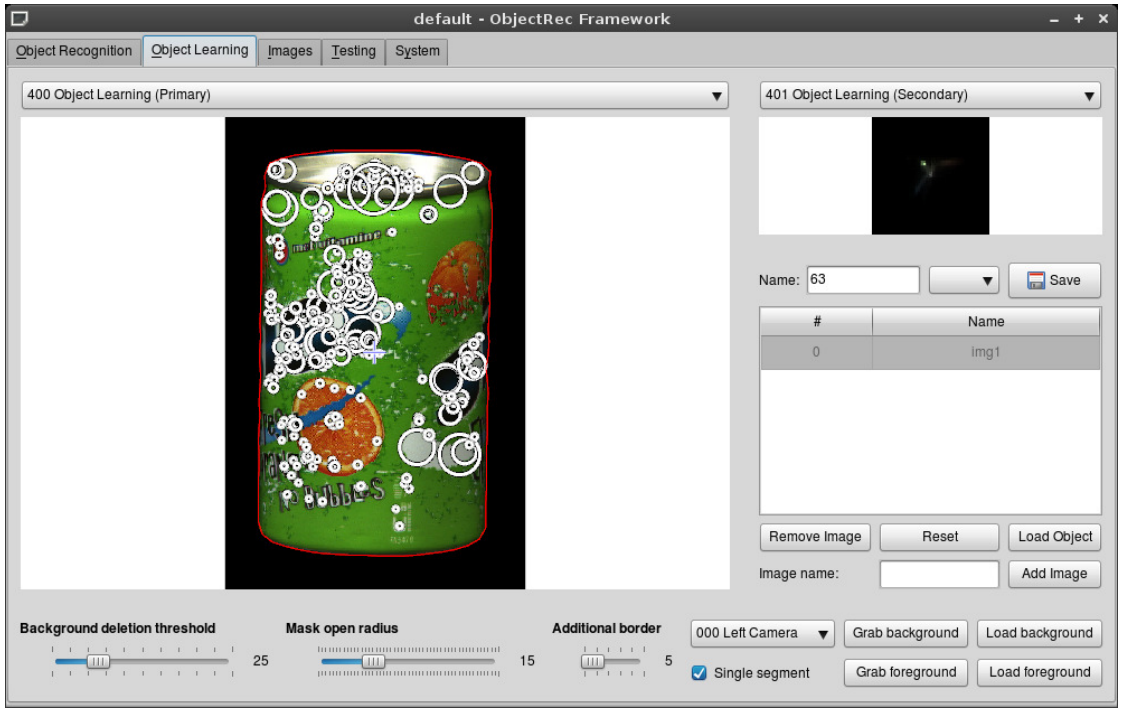


Abbildung 4.8: Grafische Oberfläche zur Erstellung der Objektdatensätze

te Homografie in das Kamerabild abgebildet. Der Umriss des erkannten Objekts ist in Abbildung 4.7 in Form einer roten Linie zu erkennen.

Abbildung 4.8 zeigt die grafische Oberfläche zur Erstellung der Objektdatensätze nach dem Verfahren aus Abschnitt 2.5.1. Sie erlaubt die Aufnahme bzw. das Laden aus einer Datei von Hintergrund- und Objektbild. Zusätzlich können die Parameter zur Erstellung der Objektmaske geändert und die Einzelbilder und weiteren Eigenschaften des Objektdatensatzes bearbeitet werden.

4.3 Evaluationsframework

Zur Evaluation der Merkmale wurden zwei Methoden verwendet, für die jeweils ein eigenes Framework entwickelt wurde. Das erste misst, wie viele Merkmale im Schnitt zwischen Bildern wiedergefunden und korrekt zugeordnet werden. Das zweite betrachtet die Erkennungsrate des Objekterkennungssystems bei Verwendung unterschiedlicher Merkmale.

4.3.1 Evaluation der Merkmale

Das Evaluationsframework basiert auf den Matlab-Skripten von Mikolajczyk [Mik09]. Dort enthalten sind die Skripte `repeatability` und `descperf`, welche die Kurven für Wiederholbarkeit bzw. Trefferquote und Genauigkeit berechnen. Sie greifen auf zwei

Funktionen zurück, die in C++ implementiert und mit der Schnittstelle *MEX* an Matlab angebunden werden. `c_eoverlap` berechnet dabei den Überlappungsfehler der Schlüsselpunkte in zwei Bildern für eine gegebene Homographie, indem sie die Regionen im zweiten Bild in das erste abbildet und auf Pixelebene mit den Regionen im ersten Bild vergleicht. `descdist` berechnet eine Matrix, die die euklidischen Abstände aller Merkmale in zwei Bildern enthält. Durch Bestimmung von Zeilen- und Spaltenminima können dadurch Zuordnungen zwischen den Merkmalen berechnet werden. Alle Skripte arbeiten auf Dateien, die im Voraus berechnete Beschreibungen der in einem Bild gefundenen Schlüsselpunkte enthalten.

Zur Evaluation wurden zusätzlich die Funktionen `repeatability_multitest` und `descriptor_multitest` implementiert, die die jeweiligen Kurven für eine Reihe von Objekten mitteln und die Ergebnisse verschiedener Verfahren in einen gemeinsamen Graphen zeichnen. Alle Graphen werden dabei in eine Ordnerstruktur abgelegt, die nach Datum und Bezeichnung des Testlaufs getrennt ist. Zusätzlich werden zu Zwecken der Nachprüfbarkeit Textdateien erstellt, in denen die verwendeten Parameter der verglichenen Verfahren angegeben sind. Die Graphen werden als Matlab-Grafik und EPS-Datei abgelegt. Die Rohdaten werden zusätzlich im nativen Matlab-Format gespeichert.

Die Skripte `repeatability_multitest_aloi`, `repeatability_multitest_miko`, `descriptor_multitest_aloi` und `descriptor_multitest_miko` greifen auf die zuvor genannten zurück und sind spezialisierte Versionen, die auf den Ordnerstrukturen der ALOI- und Mikolajczyk-Daten arbeiten. Für die in Kapitel 5 beschriebenen Experimente existiert zudem eine Reihe von einzelnen Skripten, welche die übrigen Skripte, jeweils mit spezifischen Parametern, aufrufen.

4.3.2 Evaluation der Objekterkennung

Zur Evaluierung des Objekterkennungssystem wurde ein spezielles Modul mit dem Namen `OREvaluationModule` implementiert. Dieses liest zunächst die Referenzbilder und -masken aus der ALOI-Datenbank ein, erstellt daraus Objektdatensätze und schickt diese an das `OREvaluationModule`. Anschließend werden nacheinander alle Vergleichsbilder unter geänderten Beleuchtungsbedingungen geladen und ebenfalls an das `ORMatchingModule` versendet. Dieses analysiert die Bilder und schickt eine Liste aller darin erkannten Objekte zurück, welche vom `OREvaluationModule` ausgewertet wird. Hier zeigt sich erneut ein Vorteil der Modularisierten Architektur, da an der Kernfunktionalität des Objekterkennungssystems für die Evaluation keine Anpassungen vorgenommen werden müssen. Die Ergebnisse werden wie bei der Evaluation der Merkmale in getrennten Ordnern abgelegt.

Zur Erstellung der Ergebnisgraphen wird das Hauptprogramm durch ein Shell-Skript mit unterschiedlichen Schwellenwerten für das Nearest Neighbour Ratio Matching aufgerufen. Die resultierenden Werte für Genauigkeit und Trefferquote werden für jeden Lauf ermittelt und in einer Datei abgelegt. Mit dem Matlab-Skript `plot_objectRecPrecisionRecall` wird daraus schließlich der Graph erstellt.

Kapitel 5

Evaluation

Der Evaluation der im Rahmen dieser Arbeit implementierten Farbmerkmale geht ein Vergleich von verschiedenen quelloffenen Implementationen von SURF voraus. Dies dient der Auswahl einer Ausgangsbasis für die Implementation der Farbmerkmale sowie der Klärung von Sachverhalten, die in den Veröffentlichungen zu SURF [BTVG06, BETG08] nicht hinreichend beschrieben sind.

Zur Evaluation der Farbmerkmale werden zunächst verschiedene Testverfahren aus der Literatur verglichen. Anschließend werden eine Reihe von Kombinationen der verschiedenen Verfahren zur Berechnung der Farbmerkmale verglichen und diejenige ausgewählt, die am geeignetsten erscheint. Anschließend wird geprüft, welche Auswirkungen die Verwendung der Farbmerkmale auf die Leistung des Objekterkennungssystems hat.

5.1 Vergleich verschiedener SURF-Implementationen

Die Originalimplementation, welche von Bay und Van Gool in ihrer Veröffentlichung verwendet wurde, ist nur als vorkompillierte Bibliothek verfügbar. Allerdings gibt es bereits mehrere quelloffene Implementationen in C++, welche als eigenständige Bibliotheken oder als Teil einer anderen Software veröffentlicht wurden.

1. OpenSURF [Eva09] ist eine dedizierte Bibliothek, die nur den SURF-Algorithmus und einige Hilfsfunktionen implementiert. Von ihr gibt es zwei Releases vom 22.03.2009 und 31.08.2009, die getrennt betrachtet werden. Diese werden im Folgenden als “OpenSURF” und “OpenSURF2” bezeichnet. In OpenSURF2 wird ein modifizierter Algorithmus zur Berechnung der Merkmalsvektoren verwendet, der in [AKB08] beschrieben ist.

2. dlib [Kin09] ist eine Bibliothek mit einer vielzahl von Funktionen aus Bildverarbeitung und künstlicher Intelligenz, die auch eine Implementation von SURF enthält. Diese wird im Folgenden als “DlibSURF” bezeichnet.

Detektor	Schlüsselpunkte gesamt	Anteil korrekter Schlüsselpunkte
OrigSURF	7540	
DlibSURF	4968	25.20%
OpenSURF	5781	28.04%
OpenSURF2	6857	31.43%
PanoSURF	7541	99.97%

Tabelle 5.1: Vergleich der Implementationen des Fast Hessian-Detektors

3. libmv [CRM+09] ist eine Bildverarbeitungs-Bibliothek, die eine Implementation von SURF enthält. Diese beinhaltet allerdings keine Rotationsinvarianz und wird daher in dieser Arbeit nicht berücksichtigt.

4. Panomatic [Orl09] ist eine Software zur automatischen Berechnung von korrespondierenden Punkten in Bildserien. Diese können exportiert und in entsprechenden Programmen zum Erzeugen von Panoramabildern verwendet werden. Es verwendet den SURF-Algorithmus, welcher als eigenständige Komponente implementiert ist. Diese wird im Folgenden als "PanoSURF" bezeichnet.

5. OpenCV [BDE+09] enthält in der aktuellen Version (2.0beta) eine Implementation von SURF, die jedoch unvollständig ist. Daher wird sie in dieser Arbeit nicht berücksichtigt.

5.1.1 Ähnlichkeit mit der Originalimplementation

Zunächst soll bestimmt werden, wie sehr die betrachteten quelloffenen Implementationen (DlibSURF, PanoSURF und OpenSURF) mit der Originalimplementation (OrigSURF) übereinstimmen. Da die Originalimplementation nicht quelloffen ist, kommt dafür nur ein Black-Box-Test in Frage. Dafür werden einzelne Funktionen der verschiedenen Implementationen auf ein Bild angewandt und ihre Ausgaben mit denen der Originalimplementation verglichen. Als Eingabebild dient das Referenzbild aus der "Graffiti"-Serie. Alle Implementationen wurden so konfiguriert, dass bei der Detektion der Schlüsselpunkte der vollständige Skalenraum abgetastet und kein Schwellenwert angewandt wird.

Der erste Test prüft die Ausgabe des Fast Hessian-Detektors. Dafür werden mit allen Implementationen alle Schlüsselpunkte aus dem Bild extrahiert. Schließlich wird überprüft, wie viele der Schlüsselpunkte der Originalimplementation mit den anderen Verfahren ebenfalls detektiert werden. Dabei gelten zwei Schlüsselpunkte als identisch, wenn ihr Abstand weniger als 1 Pixel und ihr Skalenunterschied weniger als 5% beträgt. Tabelle 5.1.1 zeigt die Ergebnisse dieses Tests. PanoSURF liefert hier als einzige Implementation nahezu identische Ergebnisse wie die Originalimplementation.

Der zweite Test überprüft die Zuweisung einer Orientierung an Schlüsselpunkte. Als Eingabe dienen bei allen Verfahren die Schlüsselpunkte, die mit der Originalimplementation berechnet wurden. Für diese wird mit den übrigen Implementationen die Orientierung

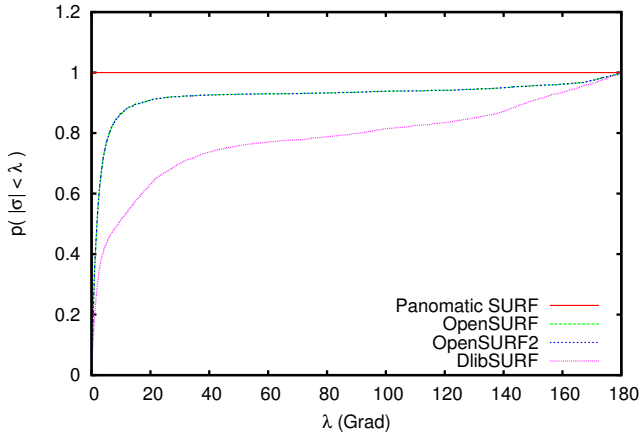


Abbildung 5.1: Ermittelte Verteilung des Fehlers in der Orientierung für verschiedene Implementationen von SURF. Die Graphen von OpenSURF und OpenSURF2 sind identisch.

errechnet. Anschließend wird jeweils die Abweichung σ von der Orientierung, die durch die Originalimplementation berechnet wurde, ermittelt. Abbildung 5.1 zeigt die ermittelte Verteilungsfunktion für den Winkelfehler der verschiedenen Implementierungen. Erneut liefert nur PanoSURF ein nahezu identisches Ergebnis wie die Originalimplementation.

5.1.2 Datenbasis für die Evaluation

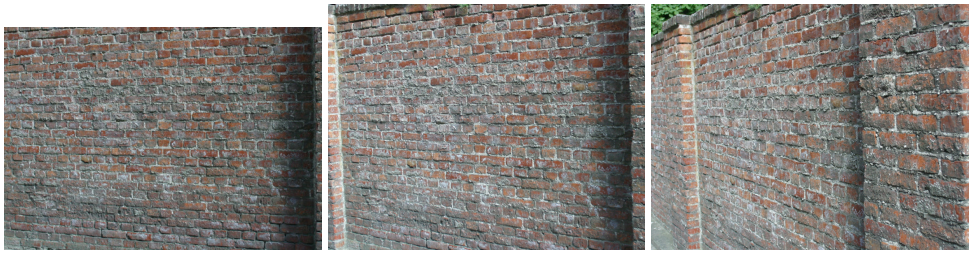
Die Evaluation der verschiedenen Implementationen erfolgt anhand der Bildsequenzen und Testsoftware von Mikolajczyk [Mik09], welche auch in [BTVG06, BETG08] sowie [MS05, MTS+05] verwendet wurde. Die Bildsequenzen bestehen aus einer Referenzaufnahme einer Szene und einer Reihe weiterer Aufnahmen, die durch eine bekannte Homographie auf die Referenzaufnahme abgebildet werden können. Dadurch kann bestimmt werden, wie viele der Schlüsselpunkte im Detektionsschritt wieder gefunden werden bzw. im Zuordnungsschritt korrekt zugeordnet werden. In Abbildung 5.2 und 5.3 sind jeweils das erste, dritte und fünfte Bild der Bildserien dargestellt.

Die Bildsequenzen beinhalten je eine spezifische Transformation in unterschiedlicher Stärke: Rotation und Zoom bis 2.8× (Boat, Bark), Blickwinkeländerung bis ca. 60° (Graffiti, Bricks), Unschärfe (Trees, Bikes), JPEG-Kompression um 60% bis 98% (UBC) und Änderung der Helligkeit (Cars). Dadurch kann der Effekt der jeweiligen Transformationen auf die Performance des Verfahrens einzeln ausgewertet werden. Die Auflösung der Bildserien ist unterschiedlich und bewegt sich zwischen 765×512 und 1000×700 Pixel. Für die Intensitäten einiger Transformationen (Unschärfe und Helligkeitsänderung) sind keine absoluten Einheiten angegeben, sie sind jedoch wie bei allen Bildserien streng monoton steigend.

(a) Bildsequenz "Boat" (Rotation und Zoom)

(b) Bildsequenz "Bark" (Rotation und Zoom)

(c) Bildsequenz "Graffiti" (Blickwinkeländerung)



(d) Bildsequenz "Bricks" (Blickwinkeländerung)

Abbildung 5.2: Bildserien zur Evaluation der SURF-Implementationen (Teil 1). Gezeigt ist jeweils das erste, dritte und fünfte Bild.

(a) Bildsequenz “Trees” (Unschärfe)

(b) Bildsequenz “Bikes” (Unschärfe)

(c) Bildsequenz “UBC” (JPEG-Kompression)

(d) Bildsequenz “Cars” (Belichtung)

Abbildung 5.3: Bildserien zur Evaluation der SURF-Implementationen (Teil 2)

(a) Bild 1 **(b)** Bild 2 **(c)** Gemeinsamer Bildbereich

Abbildung 5.4: Die ersten beiden Bilder aus der "Graffiti"-Sequenz und ihr Überlappungsbereich. Die Homographie ist in c) als graues Viereck dargestellt, die gemeinsame Bildregion ist grün markiert.

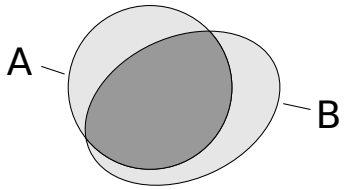


Abbildung 5.5: Der Überlappungsfehler zweier Regionen berechnet sich aus dem Flächenverhältnis von Schnittmenge und Vereinigung.

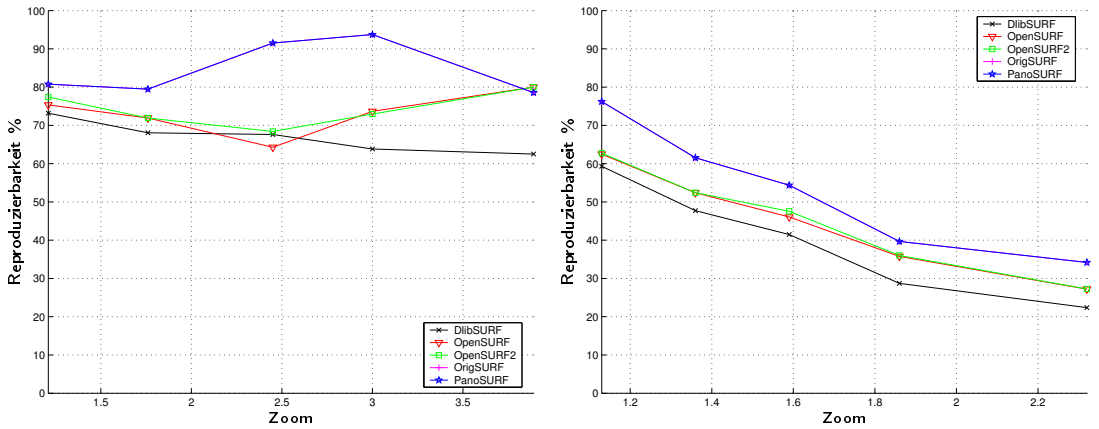
5.1.3 Detektor

Die Reproduzierbarkeit eines Detektors zwischen zwei Bildern ist definiert als der Anteil der Schlüsselpunkte, die in der gemeinsamen Bildregion liegen und deren Regionen zu einem bestimmten Grad identisch sind. Die gemeinsame Bildregion entspricht dem Teil der abgebildeten Szene, die in beiden Bildern sichtbar ist. Sie lässt sich durch die Homographie zwischen den Bildern bestimmen, wie in Abbildung 5.4 dargestellt.

Jedem gefundenen Schlüsselpunkt $\boldsymbol{k} = (x_{\boldsymbol{k}}, y_{\boldsymbol{k}}, \sigma_{\boldsymbol{k}})$ wird eine kreisförmige Region mit Radius $10 \cdot \sigma_{\boldsymbol{k}}$ zugeordnet. Diese wird anhand der bekannten Homographie H auf das Referenzbild abgebildet. Der Überlappungsfehler ϵ zweier Regionen A und B ergibt sich, wie in Abbildung 5.5 dargestellt, aus dem Größenverhältnis ihres Schnittbereiches und ihrer Vereinigung [MS05]:

$$\epsilon = 1 - \frac{A \cap B}{A \cup B}$$

Wie in [MS05] gelten zwei Regionen als übereinstimmend, wenn ihr Überlappungsfehler kleiner als 40% ist.



(a) Bildsequenz "Bark" **(b)** Bildsequenz "Boat"

Abbildung 5.6: Vergleich der Detektoren für die Bildserien "Bark" und "Boat" (Rotation & Zoom).

Die Anzahl bzw. Dichte der verwendeten Schlüsselpunkte kann das Ergebnis der Evaluation beeinflussen [MTS+05]. Daher werden im Gegensatz zum Originalverfahren bei allen Implementationen eine feste Anzahl von $n_{\boldsymbol{k}} = 500$ Schlüsselpunkten verwendet. Dies ist in diesem Kontext sinnvoll, da die Schlüsselpunkte bei allen Implementationen gleichartige Merkmale repräsentieren und somit den gleichen Informationsgehalt besitzen.

Alle Implementationen speichern den interpolierten Wert $\Omega_{\boldsymbol{k}}$ von Ω an der Stelle des gefundenen Maximums. Für die Steuerung der Anzahl der gefundenen Schlüsselpunkte wird ein Schwellenwert auf den nicht-interpolierten Wert von Ω verwendet. Die Auswahl der Schlüsselpunkte mit den $n_{\boldsymbol{k}}$ höchsten Werten von $\Omega_{\boldsymbol{k}}$ ist daher analog zu einer Anpassung des Schwellenwerts.

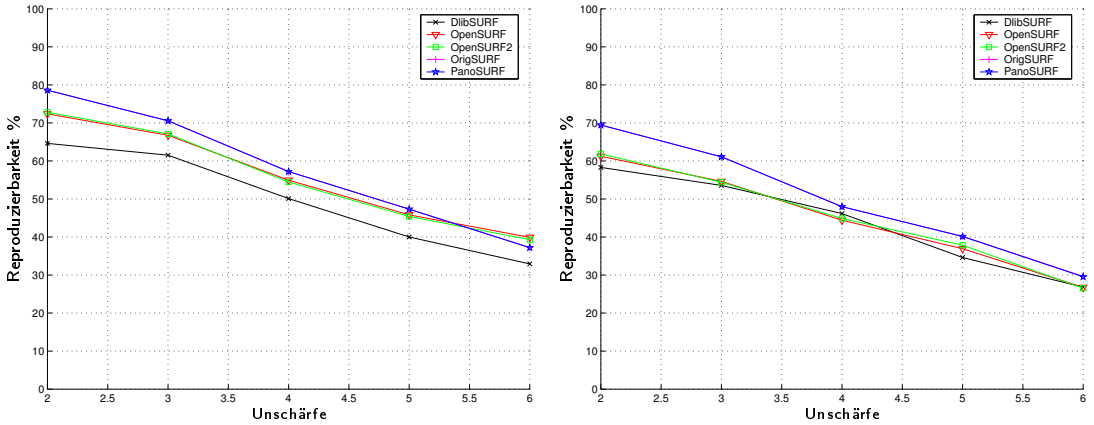
In Abbildung 5.6 - 5.9 sind die Ergebnisse dieses Vergleichs zusammengefasst. PanoSURF liefert auch hier nahezu identische Resultate wie die Originalimplementation. Die übrigen Implementationen besitzen in allen Bildserien eine geringere Reproduzierbarkeit.

5.1.4 Deskriptor

Die Schlüsselpunkte in zwei verglichenen Bildern werden einander anhand ihrer Merkmalsvektoren wie in Abschnitt 2.5.2 beschrieben zugeordnet. Korrespondenzen, deren Nearest Neighbour Ratio über einem Schwellenwert t_Φ liegt, werden verworfen. Anschließend wird analog zu dem in Abschnitt ImplDetectionEval beschriebenen Verfahren entschieden, ob die Korrespondenzen korrekt sind.

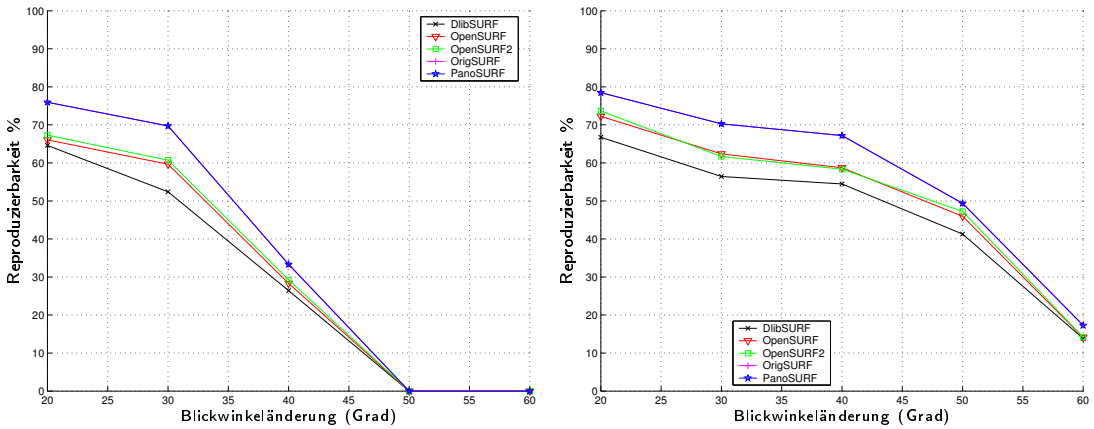
Für verschiedene Werte von t_Φ wird nun die Genauigkeit *(Precision)* und die Trefferquote *(Recall)* wie folgt berechnet [MS05]:

$$\text{Genauigkeit} = \frac{\#\text{korrekte Zuordnungen}}{\#\text{Zuordnungen gesamt}}$$



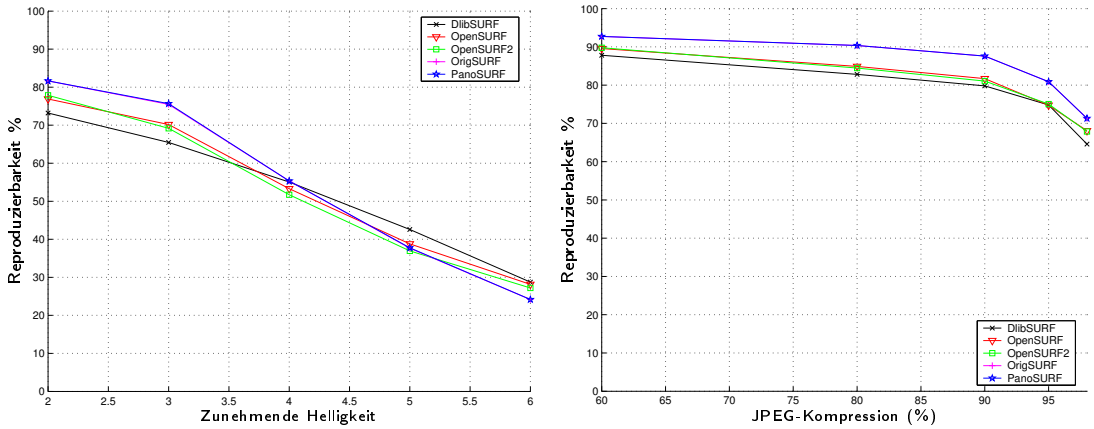
(a) Bildsequenz "Bikes" **(b)** Bildsequenz "Trees"

Abbildung 5.7: Vergleich der Detektoren für die Bildserien "Bikes" und "Trees" (Unschärfe).



(a) Bildsequenz "Graffiti" **(b)** Bildsequenz "Bricks"

Abbildung 5.8: Vergleich der Detektoren für die Bildserien "Graffiti" und "Trees" (Änderung des Blickwinkels).



(a) Bildsequenz "Cars" **(b)** Bildsequenz "UBC"

Abbildung 5.9: Vergleich der Detektoren für die Bildserien "Cars" (abnehmende Helligkeit) und "UBC" (JPEG-Kompression).

$$\text{Trefferquote} = \frac{\#\text{korrekte Zuordnungen}}{\#\text{Korrespondenzen}}$$

Abbildung 5.10 zeigt das gemittelte Resultat für alle Bildserien. Es wurde jeweils das erste mit dem dritten Bild verglichen. Wieder zeigen die Resultate, dass PanoSURF am ähnlichsten zur Originalimplementation ist. Auffällig ist zudem der große Unterschied zwischen den beiden Versionen von OpenSURF, welche durch die geänderte Berechnung des Desriptors nach [AKB08] in OpenSURF2 zu erklären ist.

5.1.5 Besonderheiten der verwendeten Implementation

Da die Testergebnisse von PanoSURF die beste Performanz sowie die größte Ähnlichkeit zur Originalimplementation zeigen, wird sie im Folgenden als Ausgangsbasis für die Implementation der Farbmerkmale genutzt. Trotz der gleichen Ergebnisse unterscheidet sich die Implementation an einigen Stellen von den Angaben in [BTVG06, BETG08].

Beispielsweise wird zur Berechnung der Bewertungsfunktion Ω eine Konstante w eingeführt, deren Wert mit 0,9 angegeben ist (vgl. Abschnitt 2.4.2). In PanoSURF beträgt deren Wert jedoch 0,6.

Die größten Unterschiede liegen in der Berechnung des Deskriptors. Zum einen wird zwischen den benachbarten Teilregionen des Deskriptors bilinear interpoliert. Der Wert des Gradienten an einer Stelle des Bildes wird also gewichtet zu den Abschnitten des Deskriptors addiert, die den 4 umliegenden Teilregionen des Deskriptorsfensters entsprechen. In Abbildung 5.11 ist dies beispielhaft für ein um 30° gedrehtes Deskriptorfenster illustriert. Zur Unterscheidung der Teilregionen sind diese abwechselnd rot und grün eingefärbt. In Abbildung 5.12 wird die Originalimplementation von PanoSURF einer

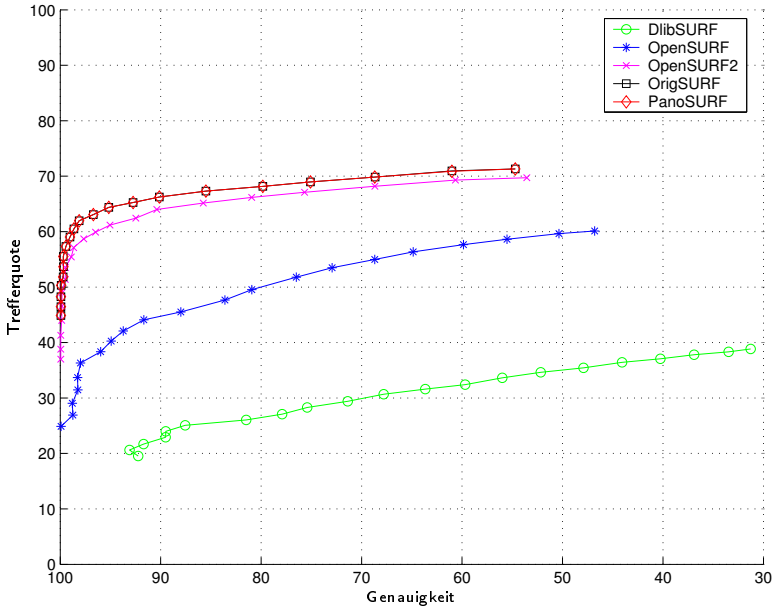
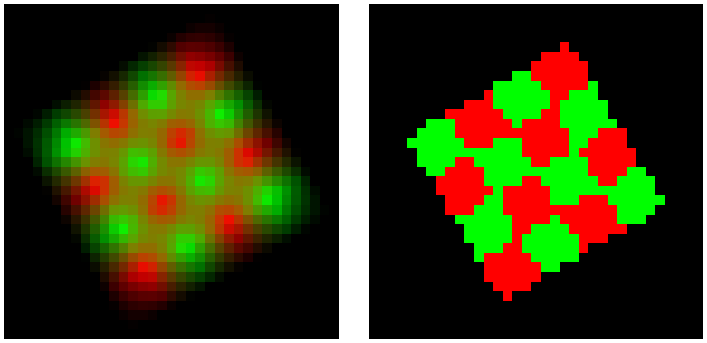


Abbildung 5.10: Mittlere Genauigkeit und Trefferquote der getesteten Implementationen von SURF.



(a) Deskriptorfenster mit Interpolation

(b) Deskriptorfenster ohne Interpolation

Abbildung 5.11: Zuteilung der Bildmerkmale zu den Teilregionen des Deskriptorfensters mit und ohne Interpolation.

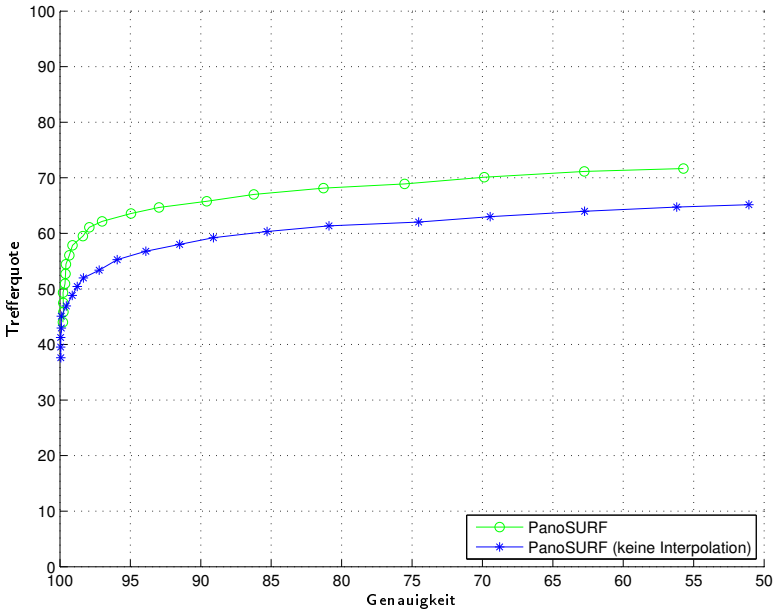


Abbildung 5.12: Mittlere Genauigkeit und Trefferquote von PanoSURF mit und ohne Interpolation.

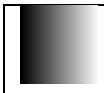
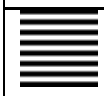
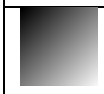
Testbild	$\boldsymbol{v}_1$	$\boldsymbol{v}_2$	$\sum d_x^-$	$\sum d_x^+$	$\sum d_x$	$\sum \lvert d_x \rvert$
	0	1	0	1	1	1
	-1	0	-1	0	-1	1
	-1	0.92	-1	1	0	1
	0	1	0	1	1	1

Tabelle 5.2: Vergleich der (normalisierten) ersten beiden Komponenten des SURF-Deskriptors mit verschiedenen analytisch bestimmten Werten. Die tatsächlichen Ausgabe $\boldsymbol{v}_1, \boldsymbol{v}_2$ lassen sich nicht mit den Angaben in [BTVG06, BETG08] erklären ($\sum d_x, \sum |d_x|$). Wahrscheinlicher ist, dass der verwendete Algorithmus mit dem in PanoSURF übereinstimmt ($\sum d_x^-, \sum d_x^+$).

modifizierten Version gegenübergestellt, die keine Interpolation bei der Berechnung des Deskriptors verwendet. Zu erkennen ist, dass die Interpolation eine deutliche Verbesserung der Ergebnisse bewirkt.

Ein weiterer Unterschied besteht darin, dass nicht für jede Teilregion die Summe der Haar-Wavelet-Antworten $(\sum d_x, \sum d_y)$ und die Summe ihrer Absolutwerte $(\sum |d_x|, \sum |d_y|)$ gespeichert wird, sondern dass diese jeweils nach positiven und negativen Werten getrennt werden $(\sum d_x^-, \sum d_x^+, \sum d_x, \sum |d_x|)$. In Tabelle 5.2 sind die ersten beiden Komponenten des von der Originalimplementation berechneten Deskriptors für verschiedene künstliche Bilder gezeigt. Die Ergebnisse lassen darauf schließen, dass die Originalimplementation dieselbe Berechnung verwendet wie PanoSURF.

Eine Anpassung der Implementation von PanoSURF an die Angaben in [BTVG06, BETG08] führt dazu, dass die erzielten Ergebnisse von denen der Originalimplementation abweichen. Daher basiert die Implementation der Farbmerkmale auf der unveränderten Version von PanoSURF.

5.2 Evaluation der Farbmerkmale

Die Evaluation findet auf der Datenbank von Mikolajczyk [Mik09] statt, wie in Abschnitt 5.1.2 beschrieben. Diese enthält nur planare Flächen und kann daher nicht zum Überprüfen der Robustheit gegenüber Änderungen der Aufnahmebedingungen bei dreidimensionalen Objekten verwendet werden. Daher wurden zusätzlich 100 Objekte aus der ALOI-Bilddatenbank [GBS05] mit Hilfe eines (Pseudo-)Zufallsgenerators ausgewählt (Abbildungen 5.13 und 5.14).

Da die Bildserie "Boat" nur in einer Graustufenversion vorliegt, wird sie durch die künstlich erzeugte Bildserie "Fields" ersetzt (Abbildung 5.16). Diese wurde erzeugt, indem ein relativ hoch aufgelöstes Bild (3794 × 2540 Pixel) mittels einer Homographie um 0 bis 50 rotiert und mit Faktoren zwischen 0,21 und 0,52 skaliert wurde. Die resultierende Bildserie hat eine Auflösung von 800 × 535 Pixel (Abbildung 5.15).

Die ALOI-Datenbank enthält Bildserien von 1000 Objekten vor schwarzem Hintergrund, die durch 3-Chip CCD-Kameras mit einer Auflösung von 768 × 576 Pixeln aufgenommen wurden. Jedes Objekt wurde von drei verschiedenen Kameras im Abstand von 125 cm aufgenommen, deren Position sich jeweils durch eine Rotation um 15° um das betrachtete Objekt unterscheidet. Als Lichtquellen dienen 5 Halogenlampen, die das Objekt in 15°-Schritten aus verschiedenen Richtungen (-30° bis 30°) beleuchten. Die Objekte befinden sich zudem auf einer drehbaren Plattform.

Für jedes Objekt existieren bei fester Beleuchtung 72 Bilder, die durch eine Rotation der Plattform in 5°-Schritten aufgenommen wurden. Zudem gibt es für jede Kamera jeweils 5 Bilder, in denen jeweils eine Lichtquelle angeschaltet ist, ein Bild, bei dem alle Lichtquellen angeschaltet sind, sowie zwei Bilder, bei denen die beiden linken bzw. rechten Lichtquellen angeschaltet sind. Durch Regulierung der Eingangsspannung der Halogenlampen wurden jeweils 12 Bilder mit Farbtemperaturen zwischen 2175 Kelvin bis 3075 Kelvin erzeugt. Für eine Auswahl von 750 Objekten sind zudem Bilder enthalten, die mit einer Kameraanordnung für die Stereobildverarbeitung aufgenommen wurden.

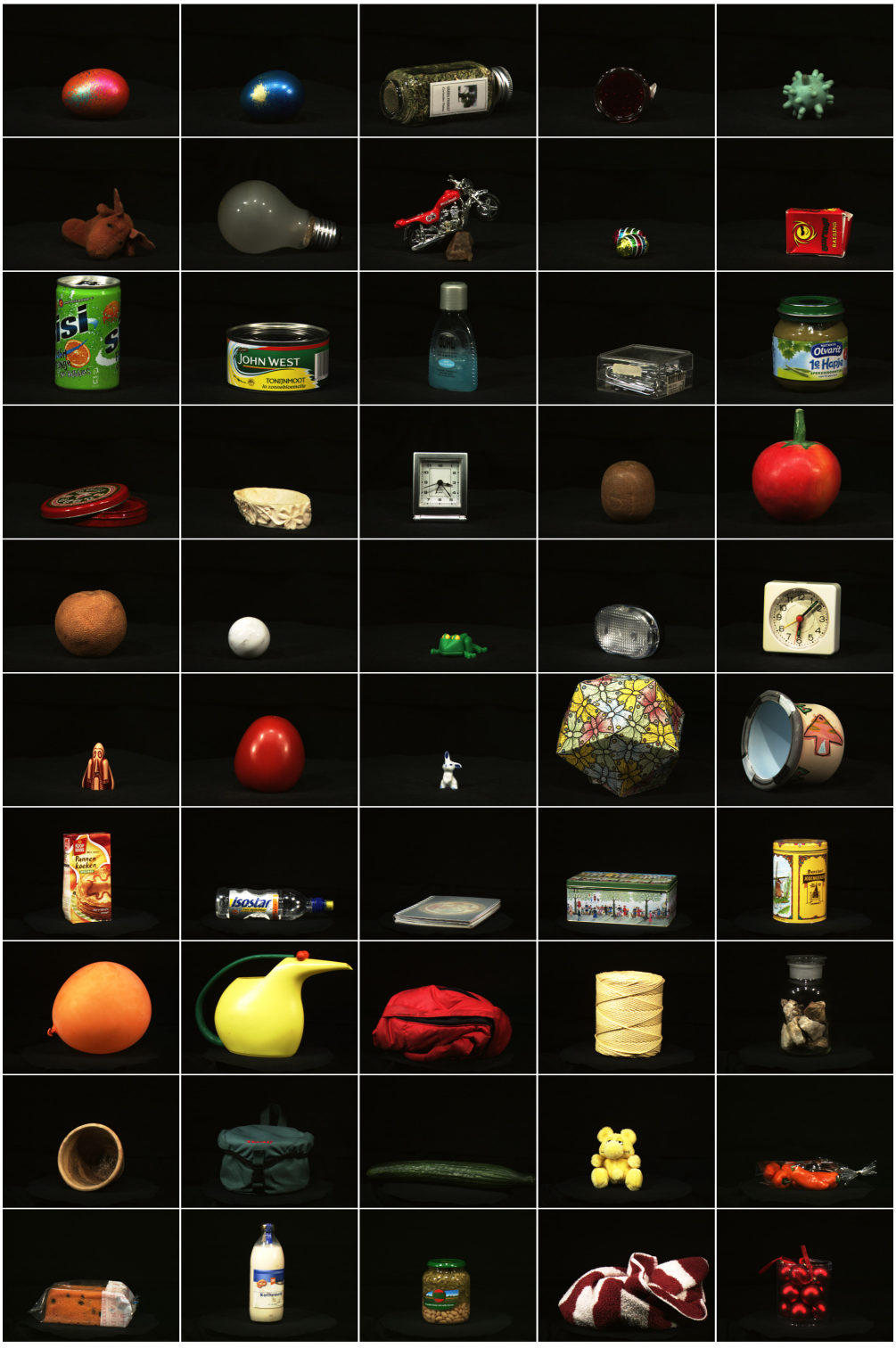
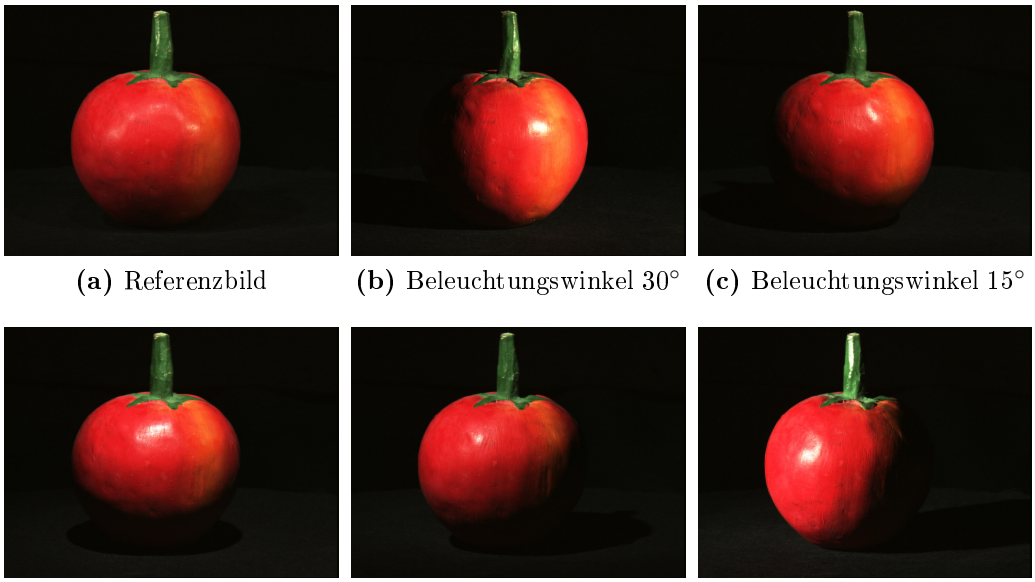


Abbildung 5.13: Verwendete Objekte aus der ALOI-Datenbank [GBS05] (Teil 1).

Abbildung 5.14: Verwendete Objekte aus der ALOI-Datenbank (Teil 2).



(a) Referenzbild **(b)** Beleuchtungswinkel 30° **(c)** Beleuchtungswinkel 15°

(d) Beleuchtungswinkel 0° **(e)** Beleuchtungswinkel -15° **(f)** Beleuchtungswinkel -30°

Abbildung 5.15: Beispiel für eine Bildserie aus der ALOI-Datenbank.

Abbildung 5.16: Bild 1, 3 und 5 der zusätzlichen Bildserie “Fields”.

In [BG09] werden aus den Invarianten des gaußschen Farbmodells SIFT-Deskriptoren berechnet und evaluiert. Dafür wird aus jedem Bild manuell eine einzige Region ausgewählt, die zwischen Bildern aus verschiedenen Betrachtungswinkeln des selben Objekts am konsistentesten erscheint. Jede dieser Regionen wird anhand ihres Deskriptors mit 100 bis 500 Regionen aus anderen Bildern verglichen.

Diese Methode kann allerdings zu einer zu optimistischen Bewertung der Verfahren führen, da die Merkmale aufgrund ihrer Stabilität vorselektiert wurden. Zudem birgt die manuelle Auswahl das Risiko, dass Merkmale verwendet werden, die für das getestete Verfahren besonders geeignet sind.

Bei einer Änderung der Kameraposition (bzw. Rotation des Objekts) können die Merkmale nicht über eine Homographie zwischen den Bildern transformiert werden. In [MP07] wird ein Testverfahren für dreidimensionale Objekte vorgestellt, das diese Einschränkung umgeht. Dafür werden die Objekte aus drei verschiedenen Kamerapositionen A, B und C aufgenommen. Eine Zuordnung zweier Merkmale $\boldsymbol{f}_A$ und $\boldsymbol{f}_B$ and den Positionen $\boldsymbol{x}_A$ und $\boldsymbol{x}_B$ zwischen A und B gilt dann als korrekt, wenn $\boldsymbol{x}_B$ auf der epipolaren Linie von $\boldsymbol{x}_A$ liegt.

Zusätzlich werden nun alle Merkmale in C bestimmt, die auf der epipolaren Linie von $\boldsymbol{x}_A$ liegen. Jedem dieser Merkmale kann eine epipolare Linie in B zugeordnet werden. Von $\boldsymbol{x}_B$ wird nun zusätzlich verlangt, dass es auf einer dieser epipolaren Linien liegt. Dadurch können ca. 98% der falschen Zuordnungen erkannt werden.

In der ALOI-Datenbank sind zwar Bilder aus verschiedenen Kamerapositionen enthalten. Die drei Kamerapositionen, welche bei der Aufnahme verwendet wurden, befinden sich jedoch auf einer Achse, so dass die epipolaren Linien in allen Bildern identisch sind oder sehr nahe beieinander liegen. Dasselbe gilt für die rotierten Objekte. In [MP07] wird zudem gezeigt, dass die Fehlerrate bei der Erkennung der falschen Zuordungen wesentlich höher ist, wenn nur zwei Kameras verwendet werden. Daher eignet sich die ALOI-Datenbank nicht für diese Art der Validierung. Stattdessen wird auch hier die Evaluationsmethode aus Abschnitt 5.1.4 und 5.1.3 verwendet. Eine Evaluation des Algorithmus ist dabei nur für Bildsequenzen möglich, in denen sich die relative Position und Rotation von Kamera und Objekt nicht ändert.

In [BG09] wird auch die Robustheit bzw. Invarianz gegenüber Änderungen der Beleuchtungsfarbe getestet. Da alle Farbmerkmale jedoch eine Genauigkeit und Trefferquote nahe 100% zeigen, erscheint dieser Test weniger aussagekräftig. Im Gegensatz dazu liegt die höchste Trefferquote bei einer Änderung der Beleuchtungsrichtung in [BG09] bei unter 50%. Die folgende Evaluation wird daher auf die Robustheit gegenüber einer Änderung der Beleuchtungsrichtung beschränkt.

Um die Robustheit gegenüber einer Änderung der Beleuchtungsrichtungsrichtung zu testen, wird als Referenzbild jeweils dasjenige, bei dem alle Lichtquellen eingeschaltet sind, ausgewählt. Als Vergleichsbilder für die Evaluation der Wiederholbarkeit dienen die 5 Bilder, bei denen jeweils eine Lichtquelle eingeschaltet ist.

Die Kameraposition wird auf Position 1 festgelegt, also senkrecht zur Ebene, auf der sich die Lampen befinden. Aus jedem Bild werden 250 Merkmale wie in Abschnitt 5.1.3 beschrieben ausgewählt. Da sich die Kameraposition nicht ändert, ist die Homographie-

matrix zwischen den Bildern die Identität. Zur Berechnung von Trefferquote und Wiederholbarkeit wird das Referenzbild mit dem Bild bei der Lichtposition von 15° verglichen. Dieses Szenario stellt die Situation nach, in der ein unter kontrollierten Bedingungen gelerntes Objektbild (das Referenzbild) in einem unter unkontrollierten Bedingungen erstellten Kamerabild gefunden werden soll.

5.2.1 Vorgehensweise

Die getesteten Verfahren sind zu vielfältig, um alle Kombinationsmöglichkeiten zu evaluieren. Um die optimale Kombination von Verfahren zu finden, wird daher folgende Strategie verfolgt: Ausgangspunkt der Evaluation ist der unmodifizierte SURF-Algorithmus. Zuerst wird getestet, wie der Merkmalsdeskriptor beschaffen sein muss, um möglichst große Unterscheidungskraft zu besitzen und gleichzeitig robust gegenüber geometrischen, photometrischen und sonstigen Transformationen des Eingabebildes zu sein.

Für diesen Deskriptor wird anschließend das Detetionsverfahren gesucht, welches eine möglichst große Reprodzierbarkeit aufweist. Da das Detektionsverfahren Einfluss auf die Leistung des Deskriptors hat, wird dies ebenfalls betrachtet. Danach wird untersucht, welches Verfahren in Kombination mit dem gewählten Detektor die größte Stabilität bei der Zuweisung einer Orientierung an Schlüsselpunkte besitzt. Schließlich wird überprüft, welchen Einfluss die Wahl des verwendeten Farbraums auf die Gesamtleistung des Verfahrens hat. Alle anderen Evaluationen beruhen auf dem gaußschen Farbraum.

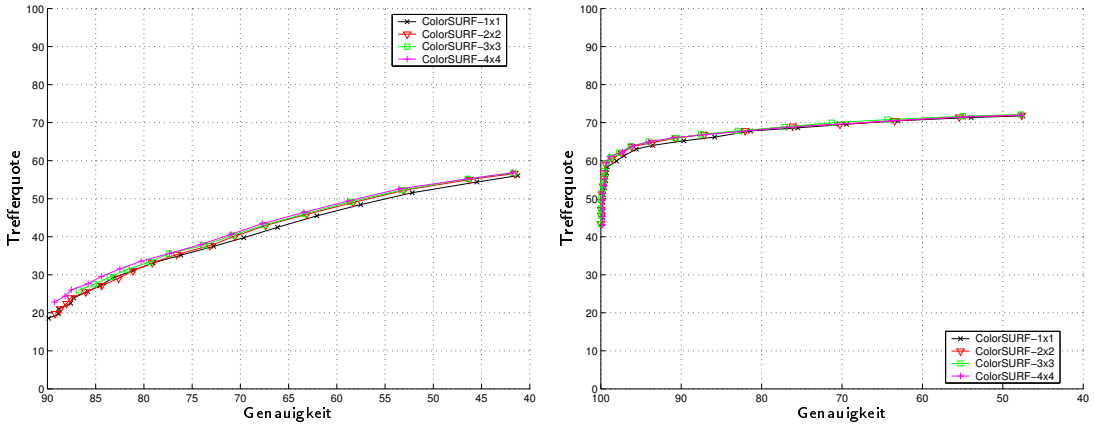
Die Kriterien zur Evaluation von Detektor und Deskriptor sind die gleichen wie in Abschnitt 5.1.3 und 5.1.4.

5.2.2 Dimensionalität des Deskriptors

Der Deskriptor für den Intensitätskanal teilt die Region um einen Schlüsselpunkt in 4×4 Teilfenster ein. Für jedes werden vier Merkmale berechnet, was zu einem Merkmalsvektor der Länge 64 führt. Dies wurde in [BETG08] als bester Kompromiss zwischen Unterscheidungskraft und Robustheit gegenüber Fehlern in Lokalisation und Orientierung des Deskriptorfensters bzw. geometrischen Transformationen, die nicht durch die Skalen- und Rotationsinvarianz abgedeckt werden, herausgestellt. Zudem ist die Länge des Merkmalsvektors entscheidend dafür, wie schnell Merkmale verglichen werden können, und sollte daher möglichst gering gehalten werden.

Um zu klären, welche Anzahl von Teilregionen optimal für die Berechnung der Deskriptoren für die Farbkanäle ist, wurde Genauigkeit und Trefferquote bei Verwendung von 1, 2×2, 3×3 oder 4×4 Teilregionen für die Farbkanäle ermittelt. Für den Intensitätkanal wurde der Deskriptor auf 4×4 Teilregionen berechnet. Der Detektionsschritt und die Zuweisung einer Orientierung wurden auf dem Intensitätskanal ausgeführt. Alle Deskriptoren wurden somit für die gleichen Schlüsselpunkte berechnet.

In Abbildung 5.17 sind die Ergebnisse für die Bildserien von Mikolajczyk für die ALOI-Datenbank dargestellt. Es zeigt sich, dass die Wahl der Anzahl der Teilregionen für die Farbkanäle einen geringen Einfluss auf das Ergebnis hat, wobei die Trefferquote



(a) ALOI **(b)** Mikolajczyk

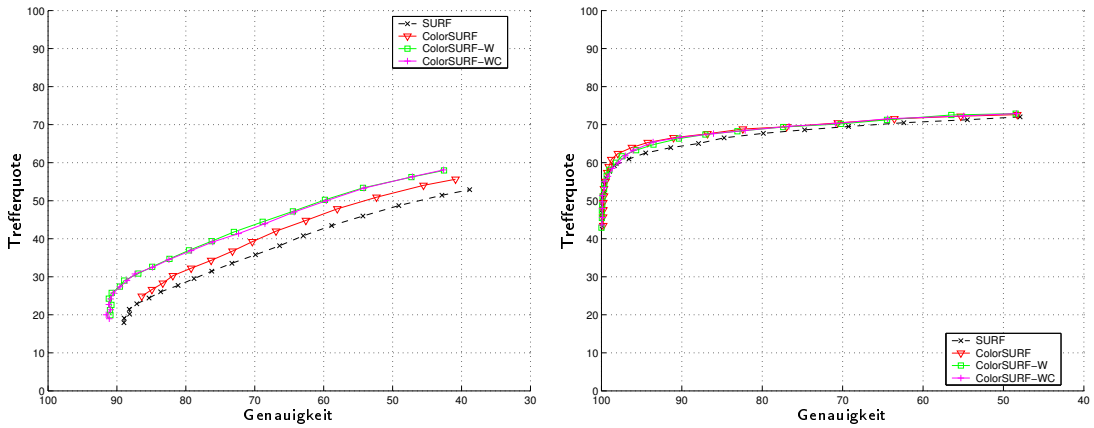
Abbildung 5.17: Performanz des Deskriptors bei unterschiedlicher Anzahl von Teilfenstern für die Farbkanäle.

am deutlichsten abfällt, wenn nur eine Teilregion betrachtet wird. Bei jeweils 2×2 Teilregionen für die Farbdeskriptoren ergibt sich insgesamt ein Merkmalsvektor der Länge 96, was einen guten Kompromiss aus Unterscheidungskraft und Deskriptorlänge darstellt. Diese Einstellung wird daher in den folgenden Tests verwendet.

5.2.3 Invarianten im Deskriptor

Es ist zu erwarten, dass das Hinzufügen von Farbinformationen die Unterscheidungskraft des Deskriptors vergrößert. Da die ALOI-Datenbank starke photometrische Transformationen durch Änderung der Beleuchtungsrichtung enthält, ist dort zu erwarten, dass die photometrischen Invarianten W und C besser zur Berechnung eines Merkmalsdeskriptors geeignet sind. Die Bildserien von Mikolajczyk enthalten nur planare Flächen und kaum photometrische Transformationen. Daher ist zu erwarten, dass der Informationsverlust durch Verwendung der Invarianten zu einer verminderten Unterscheidungskraft des Deskriptors führt.

Diese Vermutungen werden durch die Ergebnisse in Abbildung 5.18 bestätigt. Verglichen wird der reine Intensitätsdeskriptor (“SURF”) mit dem kombinierten Intensitäts- und Farbdeskriptor ohne zusätzliche Invarianz (“ColorSURF”), sowie basierend auf den W- (“ColorSURF-W”) und C-Invarianten (“ColorSURF-C”). In beiden Datenbanken wird das Ergebnis durch Hinzunahme der Farbinformationen verbessert, bei den Bildserien von Mikolajczyk fallen die Unterschiede jedoch sehr gering aus. Die Verwendung der photometrischen Invarianten auf ALOI-Daten bewirkt die signifikanteste Verbesserung der Resultate, wobei die maximale Genauigkeit des W-Deskriptors größer ist als für C. Daher wird dieser in den folgenden Experimenten verwendet.



(a) ALOI **(b)** Mikolajczyk

Abbildung 5.18: Performanz der Deskriptors mit und ohne Farbinformationen sowie für die W- und C-Invarianten.

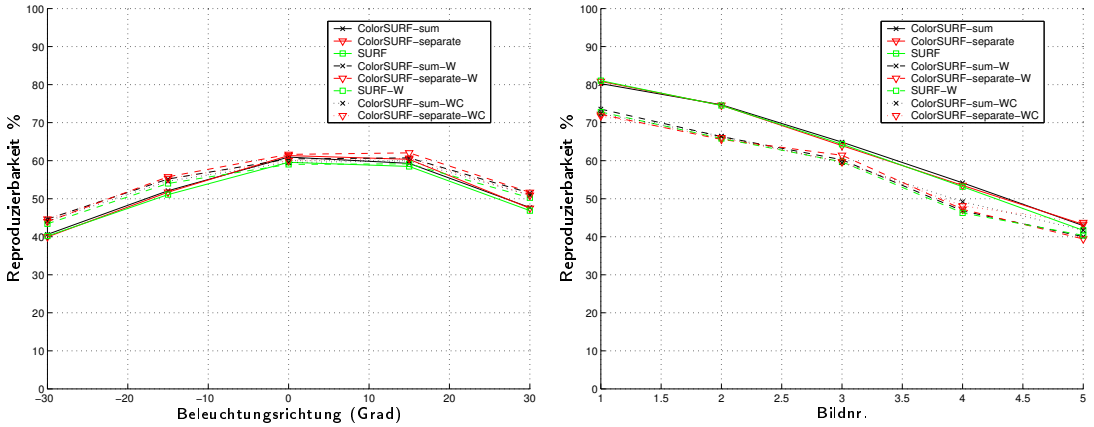
5.2.4 Invarianten im Detektionsschritt

Bevor die Merkmale einander zugeordnet werden können, müssen diese im Detektionsschritt an korrespondierenden Stellen wieder gefunden werden. Die Auswahl der Merkmale sollte zudem so gestaltet sein, dass die entsprechenden Bildregionen an sich möglichst viele Informationen enthalten. Dies ist entscheidend für die Unterscheidungskraft der Deskriptoren.

Verglichen werden in diesem Experiment die Erweiterungen des Detektors mit den verschiedenen photometrischen Invarianten und die verschiedenen Möglichkeiten zur Kombination der Bildkanäle aus Abschnitt 3.4. Es ist zu erwarten, dass durch die Verwendung der Invarianten die Reproduzierbarkeit auf den ALOI-Daten steigt. Zusätzlich wird verglichen, welche Auswirkungen die Einbeziehung von Farbinformationen (Kürzel "ColorSURF" in den Graphen) im Vergleich zu einer rein intensitätsbasierten Detektion ("SURF") hat.

Werden die Farbkanäle berücksichtigt, geschieht dies durch separate Detektion von Merkmalen auf den einzelnen Kanälen (gekennzeichnet durch das Suffix "-separate") oder durch Aufsummieren der Bewertungsfunktionen ("-sum"). Die Detektion kann auf den Bildkanälen L_1, L_2, L_3 ("-noinv") oder auf den invarianten Kanälen W_1, W_2, W_3 ("-W") bzw $W_1C_2C_3$ ("-WC") stattfinden.

Abbildung 5.19 zeigt die gemittelten Ergebnisse für beide Datenbanken. Wie erwartet, erhöhen die Invarianten die Reproduzierbarkeit bei starken Änderungen der Beleuchtungsrichtung, wobei die W-Invariante etwas bessere Resultate liefert als C. Für die Transformationen in der Mikolajczyk-Datenbank verschlechtern sich jedoch die Ergebnisse. In den Abbildungen 5.20 bis 5.23 sind die Ergebnisse für die einzelnen Bildserien von Mikolajczyk aufgeschlüsselt.



(a) ALOI **(b)** Mikolajczyk

Abbildung 5.19: Mittlere Reproduzierbarkeit der verschiedenen Detektorvarianten auf den ALOI- und Mikolajczyk-Datenbanken.

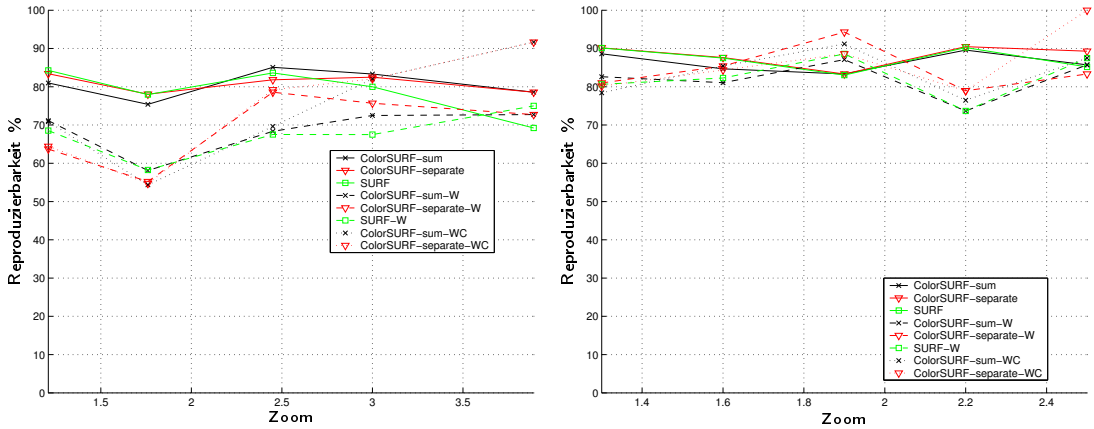
Die größte Verringerung der Reproduzierbarkeit lässt sich bei den Bildserien “Cars” und “UBC” feststellen. Beides kann mit dem verringerten Rauschabstand der Invarianten erklärt werden. In der “Cars”-Serie ist das Referenzbild stark unterbelichtet und enthält Kompressionsartefakte, wodurch die Invarianten durch das Bildrauschen dominiert werden. Die “UBC” Serie besteht aus einer Reihe von Bildern mit ansteigender JPEG-Kompression, wobei besonders bei größeren Kompressionsraten die Quantisierungsfehler zunehmen. Dadurch kommt es zu einer Verzerrung des Verhälnisses der Kanäle untereinader.

Die Reproduzierbarkeit bei kanalweiser Detektion der Schlüsselpunkte bewirkt im Vergleich zur Summierung der Kanäle keine signifikate Änderung der Ergebnisse. Auf der ALOI-Datenbank stellt die Hinzunahme der Farbkanäle im Allgemeinen eine Verbesserung dar, bei den Daten von Mikolajczyk kann darüber keine Aussage getroffen werden.

Durch die Farbverstärkung werden bevorzugt Merkmale in Regionen mit starkem Farbkontrast gefunden. Dies sollte einen positiven Effekt auf die Genauigkeit und Trefferquote der Farbdeskriptoren haben, wenn diese auf den so generierten Schlüsselpunkten berechnet werden.

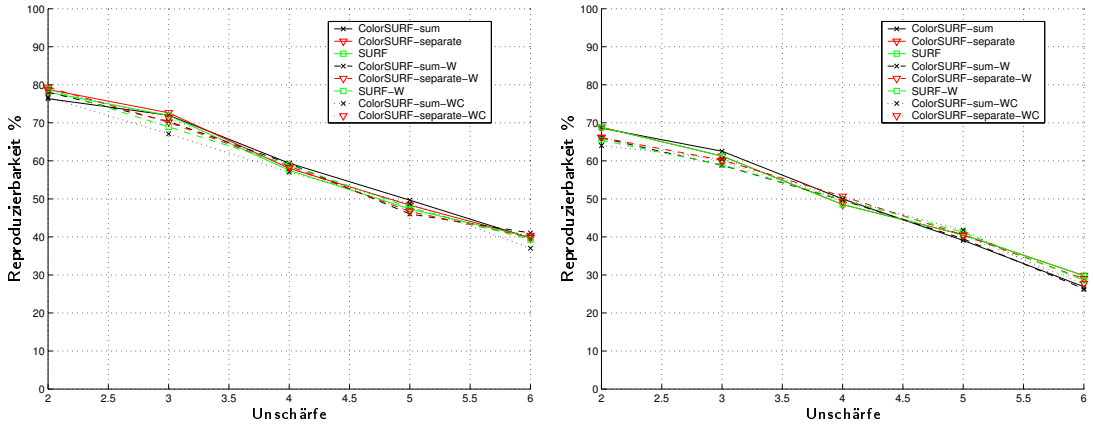
Abbildung 5.24 zeigt die Testergebnisse für den $W_1W_2W_3$-Deskriptor, wenn dieser auf die Schlüsselpunkte angewandt wird, die durch die verschiedenen Verfahren detektiert wurden. Die Ergebnisse sind analog zu den zuvor erzielten Ergebnissen zur Wiederholbarkeit. Für die ALOI-Daten verbessert sowohl die Einbeziehung der Farbkanäle als auch die Verwendung der photometrischen Invarianten die maximal erreichbare Trefferquote.

Bei Verwendung der Invarianten sinkt jedoch sowohl die Trefferquote bei hohen Genauigkeiten sowie die maximal erzielbare Genauigkeit. Daher kann hier keine eindeutige Aussage darüber getroffen werden, ob die Verwendung von Invarianten eine Verbesserung



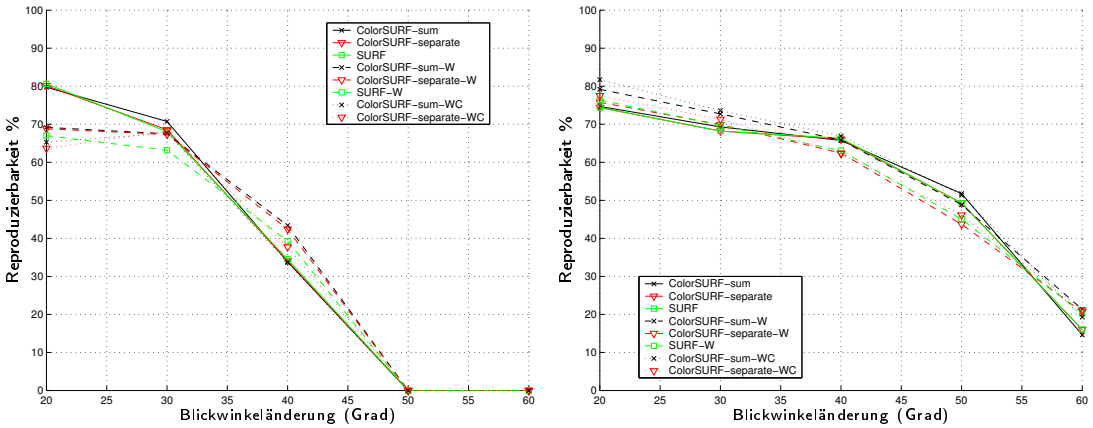
(a) Bildsequenz "Bark" **(b)** Bildsequenz "Fields"

Abbildung 5.20: Vergleich der verschiedenen Detektoren für die Bildserien "Bark" und "Fields" (Rotation & Zoom).



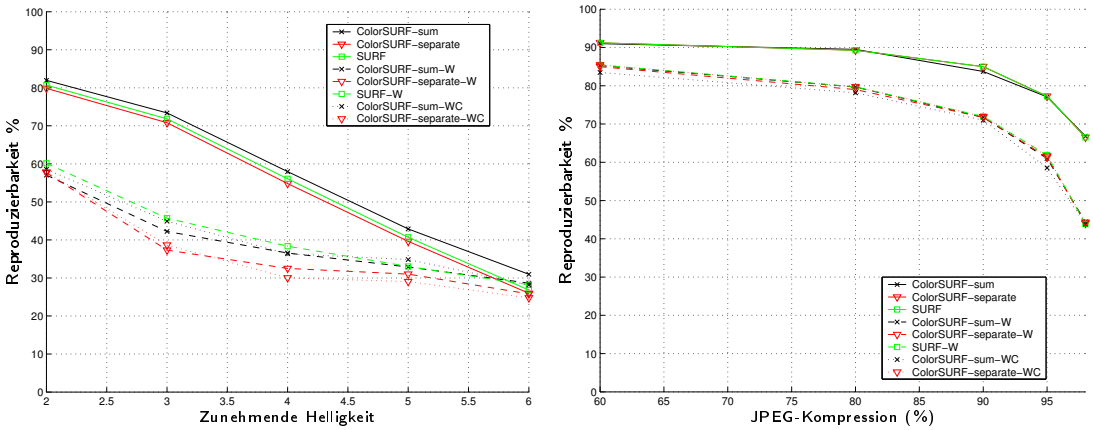
(a) Bildsequenz "Bikes" **(b)** Bildsequenz "Trees"

Abbildung 5.21: Vergleich der verschiedenen Detektoren für die Bildserien "Bikes" und "Trees" (Unschärfe).



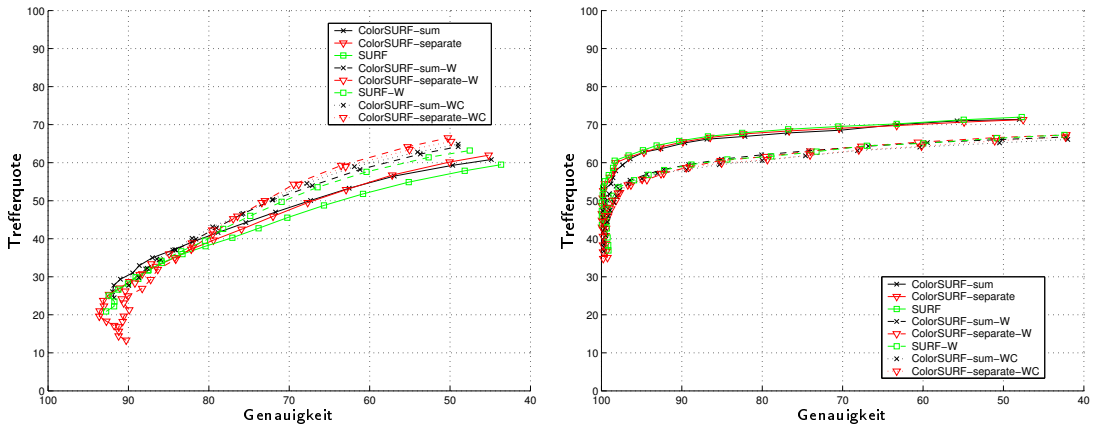
(a) Bildsequenz "Graffiti" **(b)** Bildsequenz "Bricks"

Abbildung 5.22: Vergleich der verschiedenen Detektoren für die Bildserien "Graffiti" und "Trees" (Änderung des Blickwinkels).



(a) Bildsequenz "Cars" **(b)** Bildsequenz "UBC"

Abbildung 5.23: Vergleich der verschiedenen Detektoren für die Bildserien "Cars" (abnehmende Helligkeit) und "UBC" (JPEG-Kompression).



(a) ALOI **(b)** Mikolajczyk

Abbildung 5.24: Performanz des $W_1W_2W_3$-Deskriptors in Abhängigkeit vom verwendeten Detektionsverfahren

darstellt. Zudem fällt auf den Mikoljczyk-Daten die Reproduzierbarkeit mit steigender Invarianz. Die beste Alternative stellt daher die kanalweise Detektion auf den Bildkanälen L_1, L_2, L_3 ("ColorSURF-separate-noinv") dar, da sie in allen getesteten Szenarien gute Resultate liefert.

5.2.5 Zuweisung der Orientierung

Die stabile Zuweisung einer Orientierung ist entscheidend für das Verfahren, da der Deskriptor selbst nicht rotationsinvariant ist. Da sich bei den Bildserien aus der ALOI-Datenbank die Kameraposition nicht ändert, kann hier eine eindeutige Aussage über den Fehler in der Orientierungszuweisung zwischen zwei Bildern des gleichen Objekts gemacht werden. Auf diesen Daten wurden daher absolute Winkelfehler bestimmt. Dafür wurden in allen Bildern mit dem nicht-invarianten kanalweisen Verfahren Schlüsselpunkte detektiert. Mit den verschiedenen getesteten Verfahren wurde anschließend jedem Schlüsselpunkt eine Orientierung zugewiesen.

Die Schlüsselpunkte in jeweils zwei Bildern wurden anhand des selben Kriteriums wie bei der Bestimmung der Reproduzierbarkeit einander zugeordnet. Das heißt, eine Zuordnung wurde vorgenommen, wenn der Überlappungsfehler der zugehörigen Regionen kleiner als 40% ist (vgl. Abschnitt 5.1.3). Zwischen den einander zugeordneten Schlüsselpunkten wurde anschließend der Unterschied in den zugewiesenen Winkeln bestimmt. Dieser ist definiert als der minimale benötigte Drehwinkel, um die Orientierungen ineinander zu überführen.

Um ein Aussage über die relative Genauigkeit der Verfahren unter den geometrischen Transformationen in der Mikolajczyk-Datenbank zu ermöglichen, wurden für beide Datensätze Genauigkeit und Trefferquote des Deskriptors für die W-Invariante bestimmt.

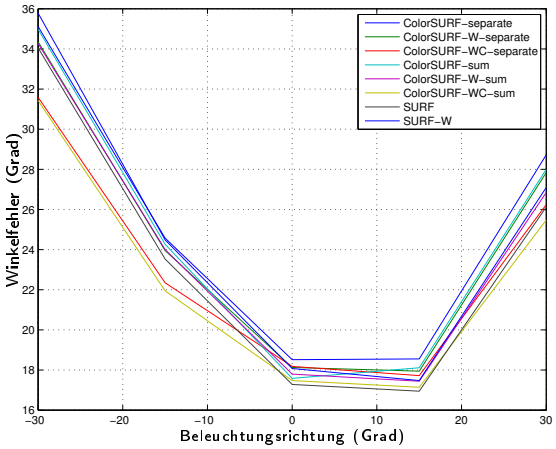


Abbildung 5.25: Mittlerer Fehler in der Orientierung für die ALOI-Datenbank bei verschiedenen Normierungsverfahren

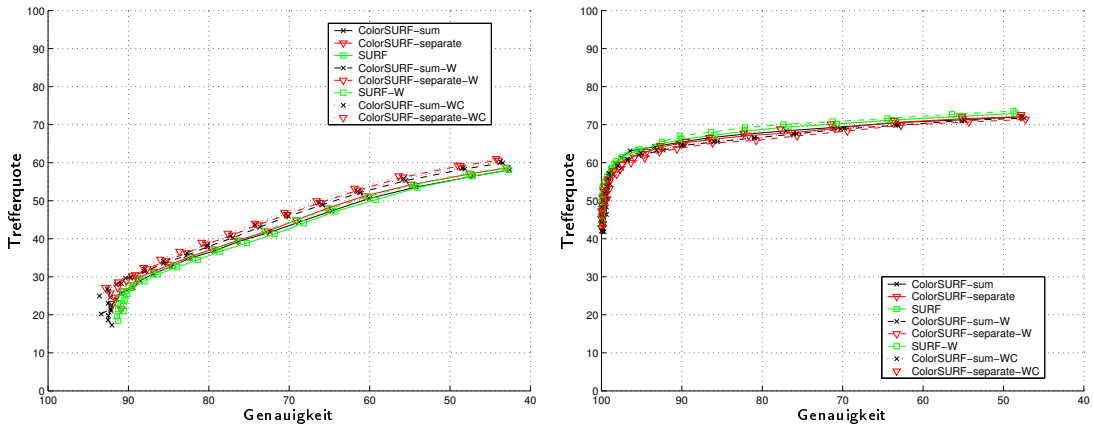
Getestet wurden die Berechnung auf dem Intensitätskanal ("SURF") und auf dem W_1-Kanal ("SURF-W"), sowie die Berechnung durch getrennte Betrachtung ("ColorSURF-separate", "ColorSURF-W-separate" und "ColorSURF-WC-separate") oder Summierung der einzelnen Kanäle ("ColorSURF-sum", "ColorSURF-W-sum" und "ColorSURF-WC-sum").

Abbildung 5.25 zeigt den mittleren absoluten Winkelfehler in Abhängigkeit von der Beleuchtungsrichtung auf den ALOI-Daten. Die Unterschiede zwischen den verschiedenen Verfahren fallen relativ gering aus. Die Kombination der Kanäle W_1, C_2 und C_3 zeigt die großen Beleuchtungswinkeln die größte Stabilität. Dies ist damit zu erklären, dass bei den C-Invarianten im Gegensatz zu den nicht-normierten Intensitäts- und Farbwerten sowie zu den W-Invarianten die Gradientenrichtung nicht vom Helligkeitsverlauf und damit von der Beleuchtungsrichtung abhängt.

Auch die Genauigkeit und Trefferquote des W-Deskriptors weist keine großen Unterschiede zwischen den Verfahren auf, wie in Abbildung 5.26 zu erkennen. Auf den Daten von Mikolajczyk liefert das Originalverfahren die besten Ergebnisse, bei den ALOI-Daten stellt sich analog zu den Ergebnissen der vorigen Evaluation die Verwendung der C-Invariante als besonders stabil heraus. Das Verfahren zur Berechnung der Orientierung durch getrennte Betrachtung der Kanäle W_1, C_2 und C_3 liefert insgesamt die ausgeglichensten Resultate und wird daher in den weiteren Tests verwendet.

5.2.6 Wahl des Farbraums

Wie in Abschnitt 3.2 erwähnt, wurde in vorangegangenen Evaluationen gezeigt, dass die Wahl des Farbraums einen Einfluss auf die Leistung von lokalen Farbmerkmalen hat. Daher wurde geprüft, welchen Einfluss die Wahl des Farbraums auf die Leistung des Deskriptors hat. Wie in Abbildung 5.27 zu erkennen, sind die Unterschiede in den Ergebnissen



(a) ALOI **(b)** Mikolajczyk

Abbildung 5.26: Evaluation des Farbdeskriptors für verschiedene Verfahren zur Zuweisung einer Orientierung

zwischen gaußschem Farbraum ("ColorSURF-gauss-WC"), $YCrCb$ ("ColorSURF-YCrCb-WC") und Gegenfarbraum ("ColorSURF-opponent-WC") sehr gering. Der IRG-Farbraum hebt sich durch geringere Genauigkeit und Trefferquote ab, was ihn in diesem Kontext als ungeeignet erscheinen lässt.

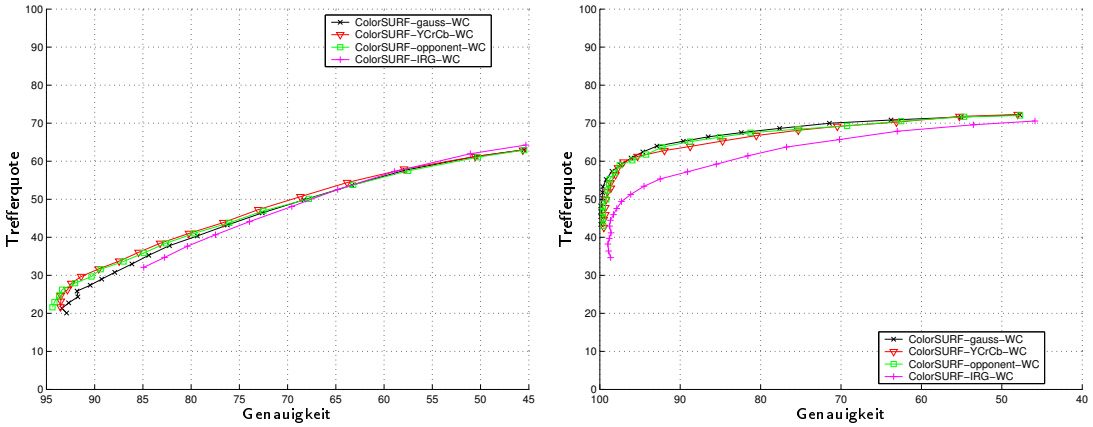
5.2.7 Vergleich mit SURF

Abbildung 5.28 zeigt die Genauigkeit und Trefferquote der Farbmerkmale im Vergleich zur Originalimplementation von SURF. Dabei wird sowohl die Standardvariante von SURF mit einem Deskriptor der Länge 64 ("SURF-64") als auch der erweiterte Deskriptor der Länge 128 ("SURF-128") berücksichtigt. Während die Ergebnisse der beiden SURF-Varianten sehr ähnlich sind, zeigt das auf Farbe basierte Verfahren eine deutlich Erhöhung der Trefferquote auf den ALOI-Daten, wobei die Verschlechterung für Mikolajczyks Bildserien vergleichsweise gering ausfällt.

Zur Berechnung der Farbmerkmale wurden Schlüsselpunkte kanalweise ohne zusätzliche Invarianz detektiert. Deren Orientierungen wurde ebenfalls kanalweise berechnet, jedoch unter Verwendung der Kanäle W_1 C_2 und C_3. Die Deskriptoren wurden mit den W-Invarianten berechnet, wobei das Deskriptorfenster im Intensitätskanal in 4×4 und in den Farbkanäle in 2×2 Teilregionen unterteilt wurde. Daraus ergibt sich für dieses Verfahren eine Deskriptorlänge von 96.

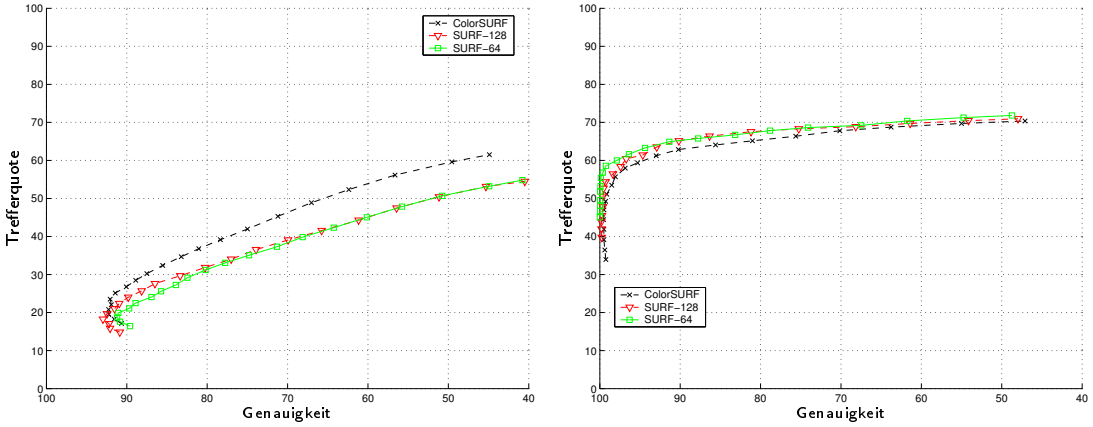
5.3 Evaluation der Objekterkennung

Um die merkmalsbasierte Objekterkennung zu evaluieren, wurden die selben 100 Objekte aus der ALOI-Datenbank genutzt wie für die übrigen Evaluationen. Als Vergleichskriteri-



(a) ALOI **(b)** Mikolajczyk

Abbildung 5.27: Performanz des Deskriptors bei Verwendung unterschiedlicher Farbräume und Normierung des Merkmalsvektors



(a) ALOI **(b)** Mikolajczyk

Abbildung 5.28: Performanz des Farbdeskriptors im Vergleich zu SURF

Abbildung 5.29: Beispiel für ein künstlich generiertes Testbild, in dem der Hintergrund ersetzt wurde.

um dient die Genauigkeit und Trefferquote, wobei diese in diesem Kontext anders definiert werden:

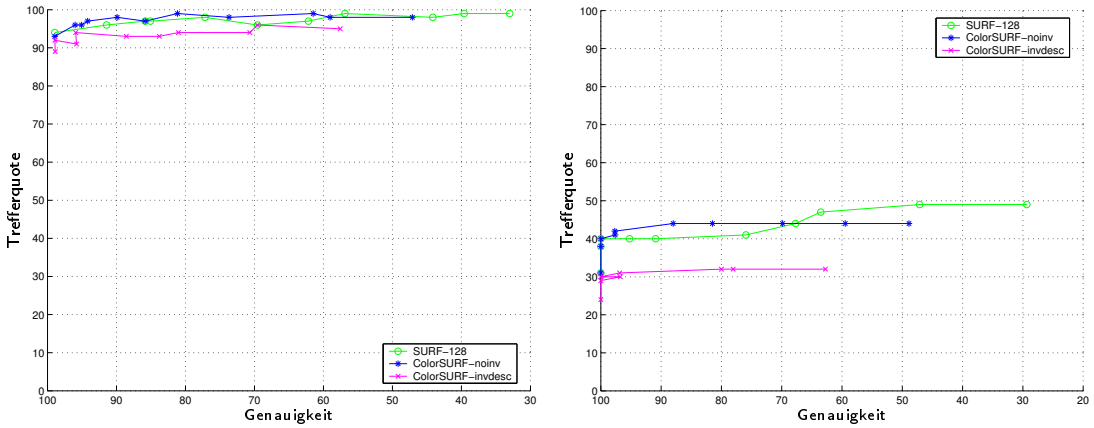
$$\text{Genauigkeit} = \frac{\#\text{korrekte erkannte Objekte}}{\#\text{erkannte Objekte gesamt}}$$

$$\text{Trefferquote} = \frac{\#\text{korrekte erkannte Objekte}}{\#\text{vorhandene Objekte}}$$

Die Anzahl der insgesamt erkannten Objekte ergibt sich aus den korrekt erkannten sowie den fälschlicherweise erkannten Objekten. Wurden beispielsweise 90 der 100 Objekte erkannt und keine falsch erkannt, ergibt sich eine Genauigkeit von 100% und Trefferquote von 90%. Wurden jedoch zu den 90 korrekt erkannten Objekten 90 fälschlicherweise erkannt, beträgt die Genauigkeit 50%.

Die Wiederholbarkeit und Trefferquote wurde mit Schwellenwerten zwischen $0,85$ und $0,95$ für das Nearest Neighbour Ratio Matching ermittelt, woraus sich die Graphen in Abbildung 5.29 ergeben. Die Quantisierung für das Hough Clustering wurde in allen Fällen auf 10 Intervalle für alle Dimensionen eingestellt.

Zusätzlich zu den in Abschnitt 5.2 genannten Objektbildern in der ALOI-Datenbank wurden künstlich generierte Bilder verwendet, bei denen der Objekthintergrund durch ein anderes Bild ersetzt wurde. Dafür wurden die 100 Objektbilder, bei denen der Beleuchtungswinkel 15° beträgt, anhand der in der Datenbank vorhandenen Bildmasken isoliert. Anschließend wurden sie in eines von 10 verschiedenen inhomogenen Hintergrundbildern eingefügt. Die Hintergrundbilder zeigen verschiedenene Innenräume von Häusern. Sie wurden aus den Suchergebnissen für den Begriff “living room” auf der Internetseite von



(a) Originalbilder **(b)** Bilder mit inhomogenem Hintergrund

Abbildung 5.30: Mittlere Genauigkeit und Trefferquote des Objekterkennungssystems auf den Testbildern der ALOI-Datenbank

Wikimedia Commons [URL09] ausgewählt. Als Referenzbild diente das in Abschnitt 5.2 verwendete. Ziel ist hierbei, eine Einschätzung der Erkennungsrate in Situationen zu ermöglichen, in denen sich das abgebildete Objekt in einer unkontrollierten Umgebung befindet.

Getestet wurde SURF mit erweitertem Deskriptor ("SURF-128"), die Farbmerkmale, welche in allen Schritten des Algorithmus auf den nicht-invarianten Bildkanälen arbeiten ("ColorSURF-noinv") sowie das in Abschnitt 5.2.7 beschriebene Verfahren zur Berechnung der Farbmerkmale unter Verwendung der Invarianten ("ColorSURF-invdesc").

Zu erwarten ist, dass analog zu den Ergebnissen aus den Abschnitten 5.2.3 und 5.2.7 eine Erhöhung von Genauigkeit und Trefferquote durch Hinzunahme von Farbinformationen entsteht, welche durch Verwendung der Invarianten zusätzlich verstärkt wird.

Zudem sollte die Trefferquote für die Bilder, bei denen das Objekt vor einem inhomogenen Objekt erkannt werden soll, geringer ausfallen. Dies hängt damit zusammen, dass zur Beschreibung des Objekts auch Schlüsselpunkte verwendet werden, deren Deskriptorfenster über den eigentlichen Objektbereich hinaus ragen. Diese können also nur mit einer geringeren Wahrscheinlichkeit korrekt zugeordnet werden, da ihre Lokalisation und Orientierung sowie ihr Deskriptor durch den Hintergrund beeinflusst werden.

Abbildung 5.30 zeigt die Ergebnisse. Wie zu erwarten, ist die Trefferquote bei den Objektbildern vor inhomogenem Hintergrund geringer. In beiden Testreihen ist eine Vergrößerung der Trefferquote durch Hinzunahme von Farbinformationen erkennbar, wobei diese geringer ausfällt als in Abschnitt 5.2.3. Die Verwendung der Invarianten resultiert entgegen der Erwartung in einer Reduktion der Trefferquote. Besonders deutlich ist dieser Effekt für die Bildserie mit inhomogenen Hintergrund.

Eine mögliche Erklärung hierfür ist, dass viele der Objekte monochrom sind bzw. nur sehr wenige Texturmerkmale in den Farbkanälen enthalten. Eine detailliertere Auswer-

Algorithmus	#Schlüsselpunkte	Laufzeit ohne Multithreading	Laufzeit mit Multithreading
SURF-128	1244	742 ms	532 ms
ColorSURF-noinv	1222	1674 ms	1275 ms
ColorSURF-invdesc	1222	2396 ms	1615 ms

Tabelle 5.3: Laufzeit der Algorithmen zur Merkmalsextraktion

tung der Ergebnisse hat ergeben, dass die Objekterkennung unabhängig vom verwendeten Algorithmus bei dieser Klasse von Objekten häufig fehl schlägt. Der SURF-Algorithmus besitzt in diesen Fällen eine geringere Stabilität, wenn die Variationen in der Helligkeit ausschließlich durch Beleuchtungseffekte verursacht werden, welche sich zwischen den verglichenen Bildern ändern. Die Invarianten haben hier zusätzlich den Nachteil, dass durch die Normalisierung ein Großteil dieser Informationen verloren geht. Dadurch können auch untexturierte Objekte nicht erkannt werden, bei denen der Einfluss der Beleuchtungseffekte geringer ist.

Die Erhöhung von Genauigkeit und Trefferquote in Abschnitt 5.2.7 ist demnach darauf zurückzuführen, dass die Anzahl der korrekten Zuordnungen in Bildern, die viele Texturinformationen enthalten, so stark erhöht wird, dass sie die Verminderung der Anzahl der korrekten Zuordnungen in wenig texturierten Bildern kompensiert.

5.4 Laufzeit und Speicherbedarf des Algorithmus

Tabelle 5.3 zeigt die gemittelten Laufzeiten der zuvor besprochenen Verfahren für die Extraktion der Merkmale. Alle Werten wurden mit der selben Implementation ermittelt, um Details der Umsetzung außen vor zu lassen. Der Speicherbedarf der Programmteile für die Merkmalsextraktion betrug in allen Fällen ca. 50 MB, da auch bei der Graustufenvariante Speicher für drei Bildkanäle allokiert wird. Als Hardware kam ein Rechner mit 2 GB Hauptspeicher und zwei Prozessorkernen mit je 2,3 GHz zum Einsatz.

Als Eingabebilder dienten die 10 Referenzbilder der Mikolajczyk-Datenbank sowie 10 Referenzbilder von verschiedenen Objekten aus der ALOI-Datenbank. Gemessen wurde die Laufzeit mit und ohne Multithreading im Deskriptorschritt, wobei zwei Threads verwendet wurden. Die Schwellenwerte wurden so eingestellt, dass die Anzahl der gefundenen Schlüsselpunkte bei allen Verfahren ähnlich ist und damit das Ergebnis nur geringfügig beeinflusst.

Zu erkennen ist, dass sich die Laufzeit durch Verwendung von Multithreading um 28% bis 33% verringert. Durch Berücksichtigung der Farbinformationen erhöht sich im Vergleich zu SURF die Laufzeit um 139% bei Verwendung von Multithreading. Dies kann darauf zurück geführt werden, dass die zu verarbeitende Datenmenge drei mal größer ist. Die Berechnung der Invarianten benötigt zusätzliche Rechenzeit, wodurch der Anstieg hier 223% beträgt.

Angesichts der signifikaten Erhöhung der Laufzeit und unter Berücksichtigung der Ergebnisse aus Abschnitt 5.3 weist die Erweiterung des SURF-Algorithmus um Farbmerkmale im Kontext der Objekterkennung in der Robotik demnach eine geringe Eignung auf.

Kapitel 6

Zusammenfassung

SURF ist ein Verfahren zur Extraktion von lokalen Bildmerkmalen aus Graustufenbildern. Es zeichnet sich im Vergleich zu ähnlichen Algorithmen durch eine besonders geringe Laufzeit aus. Ziel dieser Arbeit war die Erweiterung von SURF auf Farbbilder. Dafür sollten verschiedene Verfahren aus der Literatur in den SURF-Algorithmus integriert werden. Auf der Grundlage des Algorithmus sollte eine Software entwickelt werden, die die Erkennung von Objekten im Kontext der Robotik erlaubt. Die entwickelten Verfahren sollten anschließend in einer ausführlichen Evaluation unter verschiedenen Gesichtspunkten verglichen werden.

Zunächst wurde basierend auf einer Studie zu dem Thema ein Überblick über die Eigenschaften von lokalen Bildmerkmalen gegeben. Ihre Eigenschaften wurden mit anderen Verfahren verglichen, die als Grundlage zur Objekterkennung dienen können. Die Vorteile der lokalen Merkmale sind dabei die Möglichkeit, Objekte zu detektieren, die nur einen kleinen Ausschnitt des analysierten Bildes bedecken, Robustheit gegenüber teilweiser Verdeckung sowie eine Reduktion der Datenmenge und damit Beschleunigung der Objektsuche.

Es wurde ein Überblick über den SURF-Algorithmus und seine Vorläuferverfahren gegeben. Merkmale auf Basis der Hessematrix sowie das Harris-Maß wurden vorgestellt, welche die Detektion von Blob-artigen Strukturen erlauben. Das sind lokal begrenzte Bildregionen, deren Intensitätswert sich von ihrer direkten Nachbarschaft unterscheidet. Anschließend wurde die Theorie des Skalenraums vorgestellt, welche zusätzlich die Kovarianz der Merkmale mit Skalierungen des Bildsignals ermöglicht. Aufbauend darauf wurden zwei Verfahren vorgestellt, die die Berechnung von Deskriptoren erlauben, welche invariant gegenüber einer Translation, Rotation und Skalierung des Bildsignals sind.

Um Objekte anhand ihrer lokalen Merkmale erkennen zu können, müssen diese im Kamerabild identifiziert werden. Zu diesem Zweck wurde eine Kombination von Verfahren vorgestellt, die in ähnlicher Form für den SIFT-Algorithmus eingesetzt werden. Dabei werden zuerst initiale Zuordnungen anhand der Deskriptoren getroffen. Anschließend werden durch Hough Clustering und Bestimmung einer Homographie Gruppen von Zuordnungen gesucht, die jeweils eine konsistente Aussage über die Transformation des Objektbildes

treffen. Anhand der Anzahl der verbleibenden Zuordnungen wird entschieden, ob das Objekt in dem analysierten Bild präsent ist.

Zur Erweiterung des SIFT-Algorithmus auf Farbbilder existieren eine Reihe von Verfahren. Diese betreffen unterschiedliche Teilschritte des Algorithmus. Es wurde daher zunächst ein Überblick über die relevante Literatur zu dem Thema gegeben. Zudem wurden vorherige Vergleichsstudien ausgewertet, die die Leistung der Verfahren vergleichen. Anhand dessen wurde eine Reihe von Verfahren ausgewählt, die sich in den bisherigen Evaluationen als besonders geeignet herausgestellt haben.

Dies umfasst zunächst verschiedene Farbräume zur Darstellung der Bildinformationen. Darauf aufbauend wurden die W- und C-Invarianten für die Ableitungen des Bildsignals beschrieben. Diese sind invariant gegenüber bestimmten Klassen von photometrischen Transformationen. Grundlage dafür bildet das dichromatische Reflexionsmodell. Dieses ist ein vereinfachtes physikalisches Modell des Bildentstehungsprozesses. Es stellt eine hinreichende Approximation für die Reflexion von Licht an einer Reihe von typischen Materialien dar. Außerdem wurde ein Verfahren vorgestellt, um einen Ausgleich der Informationsdichte in den einzelnen Farbkanälen zu erreichen, welches in vereinfachter Form in den entwickelten Algorithmus integriert wurde.

Da der SURF-Algorithmus die gaußschen Ableitungen des Bildsignals durch Rechteckfilter approximiert, können die W- und C-Invarianten nicht direkt eingebunden werden. Daher wurde ein Verfahren entwickelt, welches eine Approximation der Invarianten durch die verwendeten Rechteckfilter erlaubt.

Zur Einbindung der Farbinformationen in den Detektionsschritt von SURF wurden verschiedene Verfahren aus der Literatur evaluiert. Da diese eine geringe Eignung im Kontext dieser Arbeit besitzen, wurden zwei alternative Verfahren entwickelt. Eines betrachtet die Bildkanäle getrennt, während das andere eine gemeinsame Bewertungsfunktion auf den Bildkanälen definiert. Zur Zuweisung einer normalisierten Orientierung an Schlüsselpunkte wurden zwei analoge Verfahren entwickelt. Zur Erweiterung des Deskriptors wurde ein Verfahren aus der Literatur adaptiert, wobei die Deskriptoren für die einzelnen Bildkanäle getrennt berechnet und in einem gemeinsamen Vektor kombiniert werden.

Auf der Grundlage der vorgestellten Verfahren wurde im Rahmen dieser Arbeit eine Software entwickelt, welche die Erkennung von Objekten anhand ihrer lokalen Merkmale erlaubt. Dieses wurde in ein Framework eingebunden, welches auch für das Robotersystem Robbie genutzt wird. Dadurch ist eine direkte Integration der entwickelten Komponenten in die entsprechende Robotersoftware möglich. Besonderes Augenmerk wurde dabei auf einen modularisierten Aufbau sowie eine einfache Bedienung, Konfigurierbarkeit und Erweiterbarkeit des Systems gelegt. Die verwendete Architektur sieht die Gliederung der Komponenten in Module und deren Kommunikation über Nachrichten vor. Daher wurde eine Untergliederung des Systems in Module und deren Interaktionskonzepte entwickelt, die sich in dieses Paradigma einpasst.

Um verschiedene Algorithmen und Implementationen zur Berechnung von lokalen Merkmalen vergleichen zu können, wurden generische Schnittstellen und Datenstrukturen definiert. Dies ermöglicht die Anbindung der verschiedenen Verfahren an die Objekterkennungssoftware, ohne deren Kernfunktionalität modifizieren zu müssen. Die Schnittstelle

stellt zudem die Möglichkeit bereit, die Berechnung der Deskriptoren parallelisiert auszuführen, wodurch eine Verkürzung der Laufzeit auf Systemen mit mehreren Prozessoren erreicht wird.

Die Software verfügt über eine grafische Benutzeroberfläche, über die das System gesteuert und die generierten Daten visualisiert werden können.

Zur Evaluation der entwickelten Algorithmen zur Extraktion von lokalen Merkmalen wurde eine frei verfügbare Testsoftware adaptiert, welche bereits in einer Reihe von Publikationen eingesetzt wurde. Dadurch ist die Vergleichbarkeit der erzielten Ergebnisse mit Evaluationen aus der Literatur gewährleistet. Zudem wurde eine Softwarekomponente für das Objekterkennungssystem entwickelt, die dessen Evaluation nach den gleichen Prinzipien ermöglicht.

Die Software zur Evaluation der lokalen Merkmale wurde genutzt, um einen Vergleich von verschiedenen quelloffenen Implementationen von SURF anzustellen. Anhand dessen wurde eine davon als Ausgangsbasis für die Implementation der Farbmerkmale ausgewählt. Als Datenbasis dienten mehrere Bildserien, die verschiedene Transformationen eines Bildes enthalten und auch in vergleichbaren Evaluationen in der Literatur eingesetzt wurden. Weitere Analysen wurden angestellt, um Sachverhalte zu klären, die in den Publikationen zum SURF-Algorithmus nicht hinreichend beschrieben sind.

Zur Evaluation der Farbmerkmale diente zusätzlich eine Datenbank mit Objektbildern namens ALOI, welche ebenfalls bereits in verschiedenen Publikationen Verwendung fand. Verschiedene Evaluationsmethoden aus der Literatur wurden verglichen und auf ihre Eignung für die gegebene Problemstellung und Datenbasis geprüft. Als Datenbasis wurden Bildserien gewählt, in denen sich die Beleuchtungsrichtung ändert. Für beide Datenbanken wurde das selbe Evaluationsverfahren eingesetzt.

Die verschiedenen Erweiterungen des SURF-Algorithmus wurden miteinander kombiniert und in mehreren Testreihen verglichen. Der beste Kompromiss aus Deskriptorlänge, Trefferquote und Genauigkeit wurde durch Einteilung des Deskriptorfensters in 4×4 Teilregionen für den Intensitätskanal und in 2×2 Teilregionen für die Farbkanäle erzielt. Die Einbeziehung der W- und C-Invarianten in die Zuweisung einer Orientierung und Berechnung des Deskriptors resultierte in einer signifikaten Erhöhung von Trefferquote und Genauigkeit bei Änderung der Beleuchtungsrichtung, während diese bei den übrigen Bildsequenzen geringfügig verringert wurden. Der RGB-Gegenfarbraum, der gaußsche Farbraum sowie $YCrCb$ stellten sich als geeignete Repräsentation der Bilddaten in diesem Kontext heraus.

Im Detektionsschritt des Algorithmus erzielte die getrennte Detektion der Merkmale auf den Bildkanälen ohne zusätzliche Invarianzeigenschaften die höchste Reproduzierbarkeit.

Im Vergleich zum originalen SURF-Algorithmus konnte mit dem Gesamtverfahren eine deutliche Erhöhung von Genauigkeit und Trefferquote bei Änderung der Beleuchtungsrichtung erzielt werden, während diese bei den übrigen Bildsequenzen geringfügig geringer ausfiel.

Die Farbmerkmale sowie SURF wurden schließlich in das Objekterkennungssystem eingebunden und in diesem Kontext evaluiert. Als Datenbasis dienten die zuvor verwen-

deten Objektbilder der ALOI-Datenbank. Diese wurden für einen zweiten Vergleich so modifiziert, dass die abgebildeten Objekte vor einem inhomogenen Hintergrund erscheinen. Die Trefferquote lag bei allen Verfahren auf den Originaldaten bei über 90%, bei den Bildern mit inhomogenem Hintergrund jedoch bei unter 50%. Die Maximal erzielbare Genauigkeit lag in allen Fällen bei 98% bis 100%.

Durch Verwendung der Farbmerkmale ohne zusätzliche Invarianz wurde die Trefferquote im Vergleich zu SURF leicht erhöht. Im Gegensatz zu den vorangegangenen Ergebnissen bewirkte die Verwendung der Invarianten jedoch eine signifikante Verringerung dieses Kennwerts. Eine weitere Analyse ergab zudem, dass sich die Laufzeit des Algorithmus bei Einbeziehung der Farbinformationen deutlich erhöht. Das entwickelte Verfahren weist demnach im Kontext der Objekterkennung auf einem Robotersystem eine geringe Eignung auf.

6.1 Ausblick

Da die Unterschiede in der Evaluation der verschiedenen Verfahren in einigen Fällen sehr gering sind, wäre ein Maß für die statistische Relevanz der erzielten Ergebnisse wünschenswert. In [SGS08b] wird dies durch *Bootstrapping* erreicht. Dafür werden eine Reihe von zufälligen Teilmengen des Testdatensatzes gewählt, für die die jeweiligen Kennwerte berechnet werden. Dieser Prozess wird mehrmals wiederholt. Die Varianz der erzielten Ergebnisse erlaubt eine Aussage über deren Signifikanz.

Die Evaluation hat ergeben, dass sich die Erkennungsrate des Objekterkennungssystems deutlich verringert, wenn sich das Objekt vor einem inhomogenen Hintergrund befindet. Dies ist dadurch zu erklären, dass die Deskriptorfenster der Objektmerkmale teilweise über dieses hinausragen und daher Informationen über den Hintergrund enthalten. Bei der Erstellung des Objektdatensatzes ist bekannt, welche Bildbereiche dieses überdeckt. Den verschiedenen Teilen der Deskriptoren für das Objektbild könnte demnach eine Gewichtung zugewiesen werden, die davon abhängt, zu welchem Anteil sie Informationen über das tatsächliche Objekt enthalten. Durch Einbeziehung dieser Gewichtung in das Abstandsmaß zwischen zwei Deskriptoren könnte so der Einfluss des geänderten Hintergrundes auf das Ergebnis minimiert werden.

Eine weitere Schwäche des SURF-Algorithmus ist, dass er nur Texturinformationen berücksichtigt. In das Objekterkennungssystem könnten daher zusätzlich orthogonale Bildeigenschaften, beispielsweise Bildsegmente oder Kanten, einbezogen werden, um eine Verbesserung der Erkennungsleistung von schwach texturierten Objekten zu ermöglichen.

In der Robotik spielt die Laufzeit der Bildverarbeitungsalgorithmen eine wesentliche Rolle. Eine Parallelisierung des Detektorschritts des SURF-Algorithmus könnte hier bei Verwendung eines Mehrprozessorsystems eine Verbesserung bringen.

Literaturverzeichnis

[AB07] Ancuti, Cosmin ; Bekaert, Philippe: SIFT-CCH: Increasing the SIFT distinctness by color co-occurrence histograms. In: *5th International Symposium on Image and Signal Processing and Analysis, 2007* (2007)

[AHF06] Abdel-Hakim, Alaa E. ; Farag, Aly A.: CSIFT: A SIFT Descriptor with Color Invariant Characteristics. In: *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)* 02 (2006), S. 1978–1983

[AKB08] Agrawal, Motilal ; Konolige, K. ; Blas, Morten R.: CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching. In: *ECCV (4)*, 2008, S. 102–115

[Att54] Attneave, Fred: Some informational aspects of visual perception. In: *Psychological Review* 61 (1954), Nr. 3, S. 183–193

[BDE+09] Bradski, Gary ; Darrell, Trevor ; Essa, Irfan ; Malik, Jitendra ; Perona, Pietro ; Sclaroff, Stan ; Tomasi, Carlo: *OpenCV*. `http://sourceforge.net/projects/opencvlibrary/`, 2009

[Bea78] Beaudet, P.R.: Rotationally Invariant Image Operators. In: *International Conference on Pattern Recognition*, 1978, S. 579–583

[BETG08] Bay, Herbert ; Ess, Andreas ; Tuytelaars, Tinne ; Gool, Luc van: Speeded-Up Robust Features (SURF). In: *Journal of Computer Vision* 110 (2008), Nr. 3, S. 346–359

[BG09] Burghouts, Gertjan J. ; Geusebroek, Jan-Mark: Performance evaluation of local colour invariants. In: *Comput. Vis. Image Underst.* 113 (2009), Nr. 1, S. 48–62

[BT607] International Telecommunication Union: ITU-R Recommendation BT.601-5. Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios. 2007. – Forschungsbericht

[BT702] International Telecommunication Union: ITU-R Recommendation BT.709-5. Parameter values for the HDTV standards for production and international programme exchange. 2002. – Forschungsbericht

[BTVG06] Bay, Herbert ; Tuytelaars, Tinne ; Van Gool, Luc: SURF: Speeded Up Robust Features. In: *ECCV* (2006), S. 404–417

[BZM06] Bosch, Anna ; Zisserman, A. ; Munoz, Xavier: Scene Classification via pLSA, 2006

[CRM+09] Chant, Andrew ; Roberts, David ; Mierle, Keir ; Gargallo, Pau ; MacLean, W. J.: *libmv*. `http://code.google.com/p/libmv/`, 2009

[Cro84] Crow, Franklin C.: Summed-area tables for texture mapping. In: *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM, 1984, S. 207–212

[Diz86] Dizenzo, S.: A note on the gradient of a multi-image. In: *Computer vision, graphics, and image processing* 33 (1986), Nr. 1, S. 116–125

[DLS07] Derpanis, K.G. ; Leung, E.T.H. ; Sizintsev, M.: Fast Scale-Space Feature Representations by Generalized Integral Images. In: *IEEE International Conference on Image Processing, 2007*, 2007, S. IV: 521–524

[Eva09] Evans, Christopher: Notes on the OpenSURF Library / University of Bristol. 2009 (CSTR-09-001). – Forschungsbericht

[FA91] Freeman, William T. ; Adelson, Edward H.: The Design and Use of Steerable Filters. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (1991), Nr. 9, S. 891–906

[GBS05] Geusebroek, J. M. ; Burghouts, G. J. ; Smeulders, A. W. M.: The Amsterdam Library of Object Images. In: *Int. J. Comput. Vis.* 61 (2005), Nr. 1, S. 103–112

[GBSD00] Geusebroek, Jan-Mark ; Boomgaard, Rein van d. ; Smeulders, Arnold W. ; Dev, Anuj: Color and Scale: The Spatial Structure of Color Images. In: *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part I*. London, UK : Springer-Verlag, 2000, S. 331–341

[GBSG01] Geusebroek, Jan-Mark ; Boomgaard, Rein van d. ; Smeulders, Arnold W. ; Geerts, Hugo: Color Invariance. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (2001), Nr. 12, S. 1338–1350

[GGB06] Grabner, Michael ; Grabner, Helmut ; Bischof, Horst: Fast Approximated SIFT. In: [NNS06], S. 918–927

[GS97] Gevers, Theo ; Smeulders, Arnold W.: Color Based Object Recognition. In: *ICIAP '97: Proceedings of the 9th International Conference on Image Analysis and Processing-Volume I*. London, UK : Springer-Verlag, 1997, S. 319–326

[HS88] Harris, C. ; Stephens, M.: A combined corner and edge detector. In: *Fourth Alvey Vision Conference*. Manchester, UK, 1988, S. 147–151

[Kin09] King, David: *dlib C++ Library*. `http://dclib.sourceforge.net/`, 2009

[KSH05] Ke, Yan ; Sukthankar, Rahul ; Hebert, Martial: Efficient Visual Event Detection Using Volumetric Features. In: *IEEE International Conference on Computer Vision* 1 (2005), S. 166–173

[LB02] Lowe, D. ; Brown, M.: Invariant Features from Interest Point Groups. In: *BMVC*, 2002

[LBS03] Le Bihan, Nicolas ; Sangwine, S.J.: Quaternion principal component analysis of color images, 2003, S. I: 809–812

[Lin94] Lindeberg, Tony: *Scale-Space Theory in Computer Vision*. Dordrecht, Netherlands : Kluwer Academic Publishers, 1994

[Lin98] Lindeberg, Tony: Feature detection with automatic scale selection. In: *International Journal of Computer Vision* 30 (1998), S. 79–116

[Low04] Lowe, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision* 60 (2004), Nr. 2, S. 91–110

[MCUP02] Matas, J. ; Chum, O. ; Urban, M. ; Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: *British Machine Vision Conference* Bd. 1, 2002, S. 384–393

[Mik02] Mikolajczyk, Krystian: *Interest point detection invariant to affine transformations*, Institut National Polytechnique de Grenoble, Diss., 2002

[Mik09] Mikolajczyk, Krystian: *Affine Covariant Regions*. `http://www.robots.ox.ac.uk/~vgg/research/affine/`, 2009

[MM07] Ming, Anlong ; Ma, Huadong: A blob detector in color images. In: *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*. New York, NY, USA : ACM, 2007, S. 364–370

[Mor80] Moravec, Hans P.: *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. Pittsburgh, PA, Carnegie Mellon University, Robotics Institute, Diss., 1980. – Available as Stanford AIM-340, CS-80-813 and CMU-RI-TR-3

[MP07] Moreels, Pierre ; Perona, Pietro: Evaluation of Features Detectors and Descriptors based on 3D Objects. In: *International Journal of Computer Vision* 73 (2007), Nr. 3, S. 263–284

[MS02] Mikolajczyk, Krystian ; Schmid, Cordelia: An Affine Invariant Interest Point Detector. In: *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*. London, UK : Springer-Verlag, 2002, S. 128–142

[MS04] Mikolajczyk, Krystian ; Schmid, Cordelia: Scale and Affine Invariant Interest Point Detectors. In: *International Journal of Computer Vision* 60 (2004), Nr. 1, S. 63–86

[MS05] Mikolajczyk, Krystian ; Schmid, Cordelia: A performance evaluation of local descriptors. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005), Nr. 10, S. 1615–1630

[MTGM04] Mindru, Florica ; Tuytelaars, Tinne ; Gool, Luc V. ; Moons, Theo: Moment invariants for recognition under changing viewpoint and illumination. In: *Computer Vision and Image Understanding* 94 (2004), Nr. 1-3, S. 3 – 27. – Special Issue: Colour for Image Indexing and Retrieval

[MTS+05] Mikolajczyk, Krystian ; Tuytelaars, Tinne ; Schmid, C. ; Zisserman, A. ; Matas, J. ; Schaffalitzky, Frederik ; Kadir, Timor ; Gool, L. V.: A Comparison of Affine Region Detectors. In: *International Journal of Computer Vision* 65 (2005), Nr. 1-2, S. 43–72

[NNS06] Narayanan, P. J. (Hrsg.) ; Nayar, Shree K. (Hrsg.) ; Shum, Heung-Yeung (Hrsg.): *Computer Vision - ACCV 2006, 7th Asian Conference on Computer Vision, Hyderabad, India, January 13-16, 2006, Proceedings, Part I*. Bd. *3851*. Springer, 2006

[Orl09] Orlinski, Anael: *Pan-o-matic*. `http://aorlinsk2.free.fr/panomatic/`, 2009

[PGP09] Pellenz, Johannes ; Gossow, David ; Paulus, Dietrich: Robbie: A Fully Autonomous Robot for RoboCup Rescue. In: *Advanced Robotics (Robotics Society of Japan)* 23 (2009), Nr. 9, S. 1159–1177

[SFH08] Shi, Lilong ; Funt, Brian ; Hamarneh, Ghassan: Quaternion Color Curvature. In: *Sixteenth Color Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications*, 2008

[SGS08a] Sande, K. E. A. d. ; Gevers, Th. ; Snoek, C. G. M.: Color Descriptors for Object Category Recognition. In: *European Conference on Color in Graphics, Imaging and Vision*, 2008, S. 378–381

[SGS08b] Sande, Koen E. A. d. ; Gevers, Theo ; Snoek, Cees G. M.: Evaluation of color descriptors for object and scene recognition. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 0 (2008), S. 1–8

[SGS08c] Sande, Koen E. A. d. ; Gevers, Theo ; Snoek, Cees G. M.: Evaluation of color descriptors for object and scene recognition. In: *CVPR*, IEEE Computer Society, 2008

[Sha92] Shafer, Steven A.: Using color to separate reflection components. In: *Color* (1992)

[SMB00] Schmid, Cordelia ; Mohr, Roger ; Bauckhage, Christian: Evaluation of Interest Point Detectors. In: *International Journal of Computer Vision* 37 (2000), Nr. 2, S. 151–172

[Thi09] Thierfelder, Susanne: *Objekterkennung durch Hough-Transform-Clustering von Surf-Features*, Universität Koblenz-Landau, Diplomarbeit, 2009

[TM08] Tuytelaars, Tinne ; Mikolajczyk, Krystian: Local invariant feature detectors: a survey. In: *Foundations and Trends in Computer Graphics and Vision* 3 (2008), Nr. 3, S. 177–280

[URL09] *Wikimedia Commons*. `http://commons.wikimedia.org`, 2009

[VJ01] Viola, Paul ; Jones, Michael: Rapid Object Detection using a Boosted Cascade of Simple Features. In: *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 1 (2001), S. 511

[WGB06] Weijer, Joost van d. ; Gevers, Theo ; Bagdanov, Andrew D.: Boosting Color Saliency in Image Feature Detection. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (2006), Nr. 1, S. 150–156

[WGG03] Weijer, J. Van D. ; Gevers, Th. ; Geusebroek, J. M.: Color Edge Detection by Photometric Quasi-Invariants. In: *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*. Washington, DC, USA : IEEE Computer Society, 2003, S. 1520

[WGS04] Weijer, J. Van D. ; Gevers, Th. ; Smeulders, A. W. M.: Robust Photometric Invariant Features from the Color Tensor. In: *IEEE Trans. Image Processing* 15 (2004), S. 2006

[WS06] Weijer, Joost van d. ; Schmid, Cordelia: Coloring local feature extraction. In: *European Conference on Computer Vision* Bd. Part II, Springer, 2006, S. 334–348

[ZMKB08] Zickler, Todd ; Mallick, Satya P. ; Kriegman, David J. ; Belhumeur, Peter N.: Color Subspaces as Photometric Invariants. In: *International Journal of Computer Vision* 79 (2008), Nr. 1, S. 13–30