

Farb- und merkmalsbasierte Objekterkennung mittels Histogrammrückprojektion und Wavelettransformation

Studienarbeit im Studiengang Computervisualistik

vorgelegt von

Patric Lambrecht

Betreuer: Dipl.-Ing. Wolfram Hans, Institut für Computervisualistik, Fachbereich Informatik

Koblenz, im September

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 AusfÃ¼hrungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Richtlinien der Arbeitsgruppe für Studien- und Diplomarbeiten 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

Unterschrift

Inhaltsverzeichnis

- 1 Einleitung** **7**

- 2 Stand der Wissenschaft** **9**
 - 2.1 Merkmalsräume 11
 - 2.1.1 Form 11
 - 2.1.2 Textur 13
 - 2.1.3 Farbe 15
 - 2.1.4 Fazit zur Verwendung von Merkmalsräumen 15

- 3 Eigener Ansatz** **17**
 - 3.1 Motivation und Überblick 17
 - 3.2 Wavelet 18
 - 3.2.1 Wavelet Transformation 20
 - 3.2.2 Erstellung der Merkmalsvektors 22
 - 3.3 Farbe 24
 - 3.3.1 Erstellung des Farb-Histogramm 25
 - 3.4 Rückprojektion 26
 - 3.4.1 Farbrückprojektion 26

3.4.2	Wavelet-Rückprojektion	28
3.5	Kombination von Wavelet und Farbe	29
3.6	Programmierung	30
3.6.1	ColorHistogram class	30
4	Experimente und Ergebnisse	33
4.1	Wavelet Evaluation	33
4.2	Farb-Histogramm Evaluation	35
4.3	Evaluationsschluss	37
5	Zusammenfassung	39
	Literaturverzeichnis	41

Kapitel 1

Einleitung

Dem Menschen ist es ohne Probleme möglich Gegenstände, Umgebungen und Personen mit einem Blick einer von ihm gelernten Semantik zuzuordnen. Dieser Vorgang von Erlernen, Speichern und Erkennen wird in verschiedensten wissenschaftliche Disziplinen erforscht und analysiert.

Auch in der Informatik arbeitet man seit mehreren Jahren an Algorithmen, welche diese menschlichen Fähigkeiten nachbilden. Anwendung finden diese im Bereich der Robotik. Dem Roboter wird durch die Objekterkennung zum Beispiel das Suchen nach verlorenen Gegenständen (werden im weiteren Verlauf als Objekte bezeichnet) ermöglicht. Des Weiteren bildet diese Fähigkeit eine Basis für die Interaktion des Roboters mit seiner Umgebung.

Ein praktisches Beispiel sind die Fußball Roboter des RoboCup, einer internationalen Initiative zur Förderung der Forschung in den Bereichen Künstliche Intelligenz und autonome mobile Systeme. In den vom RoboCup ausgerichteten Turnieren treten Roboter in Fußballspielen gegeneinander an. Um diese Spiele zu bestreiten muss der Roboter nicht nur die Spielsituation (Stellung der gegnerischen Spieler) sondern auch die Begrenzung des Spielfeldes, das gegnerische Tor als auch den Ball erkennen und in der Lage Schlüsse für den Spielverlauf und seine Reaktion daraus zu ziehen.

An diesem Beispiel sieht man deutlich, das sich die Problematik der Objekterkennung in zwei Teilprobleme Objektlokalisierung und Objektklassifikation aufteilen lässt. Da es sich

hier um eine Abbildung des menschlichen Sehens handelt legen wir für die Lösung dieser Teilprobleme Kamerabilder zu Grunde. Um die verschiedenen Objekte in abstrakter Form zu beschreiben, ist es nötig Modelle anzufertigen. Diese Modelle entstehen bis auf wenige Ausnahmen durch beschreibende Eigenschaften des Objekts. Bei der Betrachtung aktueller Verfahren wird hierbei in den meisten Fällen eine Eigenschaft sehr stark fokussiert.

Für unsere beiden Teilprobleme zeigt sich aber, dass wenn eine Modelbeschreibung zur Lösung eines Teilproblems führt nicht automatisch der Lösung des anderen Teilproblem genügt. Diese Arbeit nimmt sich genau dieser Problematik an. Die Idee ist hier eben zwei Objekteigenschaften zu verwenden, welche jeweils einer der beiden Teilprobleme genügen. Die Farbe eines Objektes bildet hierzu die Grundlage für die Objektlokalisierung, da sie sehr gut in der Bildsegmentierung zur Detektion eingesetzt werden kann. Als weitere Eigenschaft bilden die Wavelets, ermittelt aus der Wavelettransformation eine Repräsentation der Objekttextur, welche sich sehr gut zur Objektklassifikation eignet. Diese beiden Eigenschaften werden in einem Algorithmus verwoben und spielen jeweils in ihrem Teilbereich ihre Stärken aus.

Aspekte wie Kamera Kalibrierung welche in diesem Kontext auch eine sehr wichtige Rolle spielen werden in dieser Arbeit nicht behandelt, aber dennoch vorausgesetzt.

Kapitel 2

Stand der Wissenschaft

In der Wissenschaft finden hauptsächlich ansichtenbasierte (Appearance-based) sowie die merkmalsbasierte Objekterkennung (Feature-based) Verwendung. Beide Methoden arbeiten auf Basis eines Objektmodells, welches die reale Komplexität des Objektes in einer Form wandelt in der leichter Vergleiche angestellt werden können. Die ansichtenbasierte Erkennung legt eine Vielzahl von Bildern des Objektes aus verschiedenen Blickwinkeln zu Grunde, während die merkmalsbasierte Erkennung ein abstrahiertes Modell (z.B. 3D Modell) der Merkmale eines Objektes nutzt.

Die Nutzung von abstrakten Objekt-Modellen ist die ältere der beiden Techniken, da sie zumeist mit weniger Vergleichen auskommt und somit auch weniger Rechenleistung benötigt. Das Problem der merkmalsbasierten Objekterkennung liegt in der Wahl des geeigneten Abstraktionslevels. Merkmale wie zum Beispiel Objektflächen und Kanten müssen so im Modell abgebildet werden, dass sie beim Vergleich bei der Erkennung mit einer gewissen Varianz (Gütemaß) eine so geringe Differenz haben, dass das Objekt als richtig anerkannt wird. Hingegen hat die ansichtenbasierte Erkennung die Problematik das die Vergleiche für alle Ansichten, welche zwar durch Transformation und Reduktion des Werteraums minimiert aber dennoch alle angestrengt werden müssen.

Beiden Verfahren ist gemeinsam das sie sich Eigenschaften des Objektes stützen. So stützt sich das Edge Matching Variante der ansichtenbasierten Erkennung im Grunde neben der Textur des Objektes auf dessen Silhouette, die ja auch wieder rum die Objektform wieder-

gibt. Die Objektform wird hingegen in der merkmalsbasierten Erkennung auch mit Hilfe von Geometrie Primitiven angenähert und somit die Identifikation durchgeführt. Daher kann man sagen, dass beide Verfahren beschreibende Eigenschaften des zu erkennenden Objektes nutzen. Ich werde einige dieser Eigenschaften, die ich in dieser Arbeit als Objektmerkmale bezeichne, im Folgenden noch beleuchten.

Eine weitere Gemeinsamkeit besteht darin, dass es beide Ansätze eine Lern- und eine Erkennungsphase haben.

Die Lernphase beinhaltet sowohl die Extraktion der Objektinformationen aus den verschiedenen Ansichten als auch deren Verknüpfung (ansichtenbasierte Erkennung), sowie die Erstellung eines Objektmodells (z.B. 3D Modell der Objektflächen ? merkmalsbasierte Erkennung). Diese erlernte Datenbasis wird in einer für die spätere Verwendung geeigneten Form z.B. in einer Datenbank oder Baumstruktur abgelegt.

In der Erkennungsphase geht es darum die durch Kamerabilder gelieferten Bilder in dieselbe Form zu transformieren um einen Vergleich möglich zu machen. Im Folgenden werde ich dies Als Merkmalsraum bezeichnen.

2.1 Merkmalsräume

Ein Merkmal eines Objektes kann man mit Hilfe eines Vektors repräsentieren. Diese ist ein sogenannter Merkmalsvektor, der in der Erkennungsphase zum Vergleich herangezogen wird. Der Merkmalsraum bezeichnet den Vektorraum der alle möglichen Merkmalsvektoren eines bestimmten Merkmals oder die Kombination mehrere festgelegter Merkmale einschließt.

Ein praktisches Beispiel ist der HSV-Farbraum welcher auch in meinem eigenen Ansatz zum Einsatz kommt. Würde man diesen Farbraum komplett verwenden so ergäbe sich ein dreidimensionaler Merkmalsraum der die drei Werte Hue (Farbwinkel), Saturation (Sättigung) und Value (Helligkeitswert) in allen Belegungen abbilden kann.

In der Erkennungsphase wird dann die Distanz des aus den gelernten Daten und aus dem Eingabebild erzeugten Vektors ermittelt. Anhand dieses Abstandes kann entschieden werden ob es sich bei dem Objekt um das gelernte Objekt aus der Datenbank handelt.

2.1.1 Form

Die Form ist eines der Hauptmerkmale mit denen sich Gegenstände beschreiben lassen. Sie lässt sich in einer zweidimensionalen Ansicht mit Hilfe eines Linienzuges durch ihre Kontur als auch mit der Kombination verschiedener geometrischer Primitive wie Rechteck, Dreieck oder Kreis annähern. Ein dreidimensionales Modell kann zum Beispiel ebenfalls durch die Verknüpfung von 3D-Primitiven wie Quader, Würfel, Pyramide, Zylinder oder Kegel oder etwa durch einen Octree abgebildet werden (Abb:2.1).

Das Problem der Wahl des angemessenen Approximationsgrades tritt hier sehr schnell auf. Versucht man mit einem Modell eine zu starke Annäherung an ein komplexes reales Objekt zu erreichen, wird der Merkmalsvektor meist so groß das ein Vergleichsprozess in der späteren Verarbeitung meist sehr viel Zeit und Rechenleistung in Anspruch nimmt. Der Approximationsgrad muss zu Beginn der Lernphase festgelegt werden damit man im Schritt der Erkennungsphase eine Objektbasis auf einem einheitlichen Merkmalsraum zu Grunde legen kann. Somit muss ein in seiner Form eher einfach zu repräsentierendes Objekt wie etwa ein Stuhl mit dem gleichen Approximationsgrad wie ein Laubbaum repräsentiert

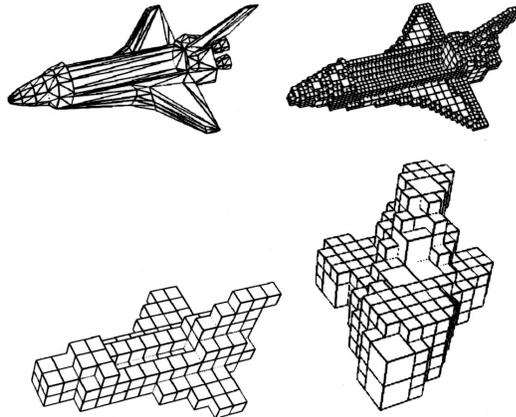


Bild 2.1: Octree-Repräsentation eines Spaceshuttels [KK96]

und in der Datenbasis abgelegt werden. Dieses Problem kann in den meisten Fällen durch die Ablage in einer Baumstruktur in dem der Approximationsgrad mit steigender Tiefe zunimmt.

Bei der Verwendung der dreidimensionalen Repräsentation eines Gegenstandes ergibt sich eine weitere Schwierigkeit. Zumeist liegen die Eingabeinformationen nur als Kamerabild in zweidimensionaler Form vor. Daher müssen hieraus im ersten Schritt zunächst Bildmerkmale, welche Aufschluss über die im Bild enthaltenen Formen geben, extrahiert werden. Zum einen kann dies mit der Durchführung einer Kantendetektion (z.B. Canny-Kantendetektor), die durch den Verlauf und den Zusammenschluss von Kanten die Form in Teilen wiedergeben können, als auch durch die Betrachtung von Farbverläufen zur Ermittlung von runden Formen, geleistet werden (Abb:2.2).

Eine weitere Methode der Formermittlung ist die Annäherung der Objektform auf der Grundlage einfacher geometrischer Formen. Geometrische Grundformen wie Rechteck, Dreieck, Kreis oder einfache 3D-Objekte wie Quader, Pyramide, Kegel etc. werden zusammengefügt und an die Realform angenähert. Der resultierende Verknüpfungsbaum kann mit geringem Datenumfang gespeichert werden. Damit kann schneller Zugriff auf die Daten realisiert werden. Der grundsätzliche Nachteil dieses Verfahrens besteht darin,

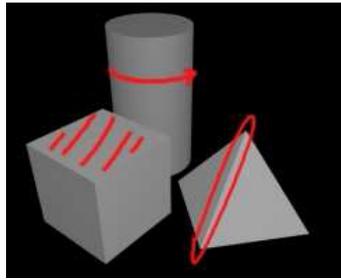


Bild 2.2: Extraktion der Merkmale aus Bildern [CGL03]

dass es im Detailierungsgrad beschränkt ist. Bei komplexeren Objekten entsteht eine stark verzweigte Baumstruktur, die sich nur mit hohem Aufwand traversieren lässt.

2.1.2 Textur

Die Textur eines Objektes zeigt an wie die Oberfläche eines Objektes beschaffen ist, lässt aber auch Rückschlüsse auf die Form des Objektes zu. Texturen eignen sich ähnlich wie die im folgenden Abschnitt betrachtete Farbe ideal zum Segmentieren eines Bildes.

In der Lernphase wird das zu lernende Objekt in Texturflächen zerlegt. Eine Texturfläche bewegt sich in ihrem Oberflächenmuster innerhalb einer gewissen Toleranz. Im Zweiten Schritt gilt es diese Texturen in geeigneter Form abzulegen. Sie könne zum einen als reine Bilddaten gespeichert werden und dann zur Erkennung über das Eingabebild geschoben werden und für die jeweilige Position der Abstand ermittelt werden. Nähert sich dieser Abstand null an so gilt diese Textur an dieser Stelle im Bild als vorhanden. So können bei der Erkennung ebenfalls Flächen mit gleicher Textur gefunden werden. Hätte man an dieser Stelle nur die reinen Bilddaten und nicht die Relationen verschiedener Objekttexturen zueinander gespeichert, könnte man nur eine sehr ungenaue Aussage darüber treffen ob sich das Objekt im Eingabebild befindet oder nicht. Relationen der Texturen können zum einen die Häufigkeit des Vorkommens einer Textur im Vergleich zu einer anderen, und zum anderen die Positionierung im Verhältnis zu einander. Hiermit können dann genauere Aussagen getroffen ob, wo und in welcher Ausrichtung sich das Objekt innerhalb der Szene befindet (Abb:2.3).

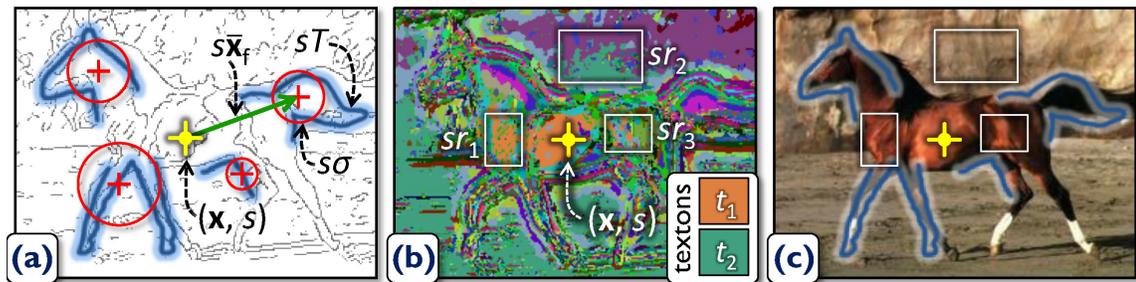


Bild 2.3: (a) Regionen Detektion auf Kantenbasis; (b) Regionen Detektion auf Farb/Texturbasis; (c) Zusammenführung der beiden Detektionen [JML99]

Problematisch wird es wenn wir keine einfache homogene Textur wie zum Beispiel ein Leopardmuster haben, woraus lediglich ein kleiner Ausschnitt genügt um die Textur komplett zu beschreiben. Dann wird das Textur Bild eines Objektes schnell so groß wie das Bild des Objektes selbst. Um so etwas zu vermeiden werden häufig Mikrot Texturen sogenannte Textons oder wie in meinem Ansatz Wavelets genutzt um die Textur eines Objektes zu beschreiben. Als Basis dienen dabei wegen der geringen Größe und die daraus resultierende kleine Anzahl an möglichen Kombinationen, nur wenige Mikrot Texturen. Dadurch kommt es nie dazu, dass eine Mikrot Textur ein Objekt eindeutig beschreibt. Bei der Abbildung durch Mikrot Texturen wird das Objekt hauptsächlich durch die Relationen der Mikrot Texturen untereinander beschrieben.

2.1.3 Farbe

Ein weiteres primäres Merkmal eines Gegenstandes ist seine Farbe. Sie beschreib nicht eineindeutig eignet sich aber ähnlich der Textur sehr gut um ein Bild in seine Segmente zu zerlegen und diese anschließend mit den gelernten Objektfarben zu vergleichen. Ein Unterschied zwischen Farben ist je nach gewähltem Farbraum lediglich ein Vergleich zweier maximal sechsstelliger Vektoren oder zweier Histogramme. Farbe ist jedoch sehr empfindlich gegenüber Veränderungen des Umgebungslichts und ist damit ohne eine Kompensation durch Kalibrierung der Kamera oder Anpassung des Umgebungslichts nur schwer im praktischen Einsatz zu verwenden. Dennoch kann man durch geschickte Wahl eines Farbraums und Reduktion von Farbinformationen.

Diese Idee wird in meinem eigenen Ansatz weiter verfolgt und noch vertieft. Dort stell ich auch das Verfahren der Histogrammrückprojektion vor, welches es ermöglicht die Vergleichsoperation zu überspringen.

Da ich die Objektfarbe in meinem eigenen Ansatz als Merkmal verwendet habe werde ich das hier nicht weiter ausführen und es in einem folgenden vertieft erläutern.

2.1.4 Fazit zur Verwendung von Merkmalsräumen

Wie wir schon anhand dieser drei exemplarisch ausgewählten Merkmalsräume sehen können, gibt es nicht den absoluten Merkmalsraum mit dem es möglich ist ein Objekt zweifelsfrei zu identifizieren. Hinzu kommen noch Probleme, wie eine teilweise Überdeckung (Okklusion), Verschattung oder eine gänzliche Änderung der Lichtverhältnisse die dazu führen kann, dass sich das gewählte Merkmale innerhalb eines Erkennungs-Szenarios anders darstellt, als es sich zum Zeitpunkt des Lernens dargestellt hat. Eine weiterer Punkt der sich hier herausstellt ist das sich einige Merkmalsräume gut in der Segmentierung einsetzen lassen, und somit die Objektdetektion innerhalb der Szene stützen aber das Objekt nicht wirklich immer eineindeutig beschreiben.

Darum bedarf es wie ich auch in meinem Ansatz beschreiben werde einer Kombination dieser Merkmalsräume.

Kapitel 3

Eigener Ansatz

3.1 Motivation und Überblick

Wie der vorherige Abschnitt zeigt gibt es unterschiedliche Merkmalsräume, die geeignet sind, Merkmalsvektoren für die Objekterkennung zu konstruieren. Allerdings hat sich gezeigt, dass keiner dieser Merkmalsräume ein universales Objektmerkmale darstellt. Daher bietet sich eine Kombination von Merkmalsräumen an. Ich hab mich in der vorliegenden Arbeit für die Merkmale Farbe und Textur entschieden.

Zur Repräsentation des Merkmals Textur habe ich Wavelets ausgewählt, deren Einsatz ich im Abschnitt Wavelet noch näher beschreiben werde. Die Objekt Farbe verarbeite ich wie schon erwähnt im HSV-Farbraum, da ich hier eine höhere Konstanz innerhalb der Farbwerte von homogenen Farbflächen habe. Die Werte der Objektfarbe werden in einem Histogramm abgelegt um sie für den Erkennungsschritt leichter verarbeiten zu können.

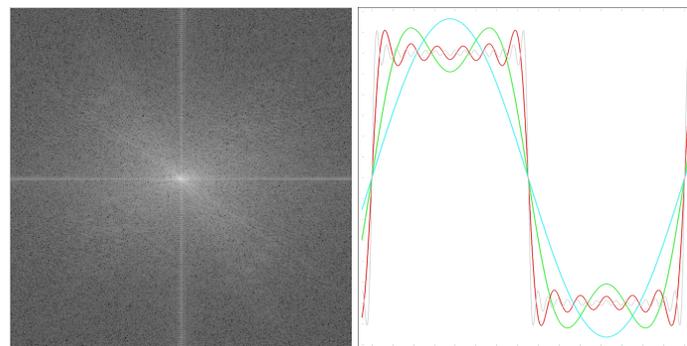
Diese Optimierung und Vereinfachung der Abbildung ist nötig um den kontinuierlichen praktischen Einsatz in der Objekt-Erkennungsphase zu gewährleisten, da wir es zum Beispiel bei der Verwendung in einem Roboter mit bewegten Kamerabilder zu tun haben, von dem jedes einzeln auf vorhandene zu suchende Objekte geprüft werden muss. Die Laufzeit der Lernphase ist hingegen weniger kritisch, da diese komplett entkoppelt von der Erkennungsphase stattfinden kann. Daher kann man diese Zeit für eine fundierte Vorver-

arbeitung der Lernbilder nutzen.

Zur abschließenden Erkennung nutze ich zunächst das Merkmal Farbe zu Detektion und Eingrenzung des Objektes innerhalb der Szene und danach die Wavelets zur Identifikation des Objektes.

3.2 Wavelet

Wavelets haben in Ursprung in der Signalverarbeitung. Sie werden zur Signalkompression genutzt. Hierbei wird versucht das Signal durch Wavelets und deren Skalierung zu beschreiben. Im Gegensatz zur Fourier-Transformation besitzt die Wavelet-Transformation eine Auflösung im Ortsraum.



(a) Ergebnis einer Fourier Transformation
 (b) Rechteck-Funktions Annäherung durch Fourier-Transformation
 Cyan: 1.Näherung; Grün: 2.Näherung; Rot: 6.Näherung; Grau: 15.Näherung

Bild 3.1: Fourier-Transformation

Dieser Umstand ist für die Objekterkennung von Vorteil da somit eine Detektion innerhalb des Bildraums mit Wavelets möglich wird. Um den Idee der Approximation von Fourier- und Wavelet-Transformation in einem Bild zu erklären, Kann man sich eine Pixelzeile als Signal vorstellen. Die Anzahl der Pixel wird durch das Sampling (Abtastrate) bestimmt.

Die Fourier-Transformation versucht sich diesem Signal durch die Verwendung von Sinus und Kosinus Funktion mit verschiedenen Eingangsvariablen anzunähern. So lässt sich dann mit wenigen Werten eine ganze Pixelzeile beschreiben. Bei diesem Schritt geht die Ortsinformation von Frequenzwerten innerhalb der Zeile in der Fourier-Repräsentation verloren, während sie bei der Wavelet-Transformation erhalten bleibt, da diese nicht auf die komplette Pixelzeile sondern nur auf einen jeweils auf Ausschnitte dieser angewandt wird. In Abb 3.1(a) sieht man das Ergebnis einer Fourier-Transformation, welche auf ein Bild. Den hellen Bereich bilden Punkte, die Eingabewerte der Sinus und Cosinus Funktionen anzeigen, die für die Approximation des Bildes angenommen werden müssen. Um die Approximationsfähigkeit dieser Transformation sieht man in Abb 3.1(b) wie eine Rechteckfunktion durch Annäherung mit Hilfe von Sinus und Kosinus Funktion innerhalb der FT dargestellt werden kann.

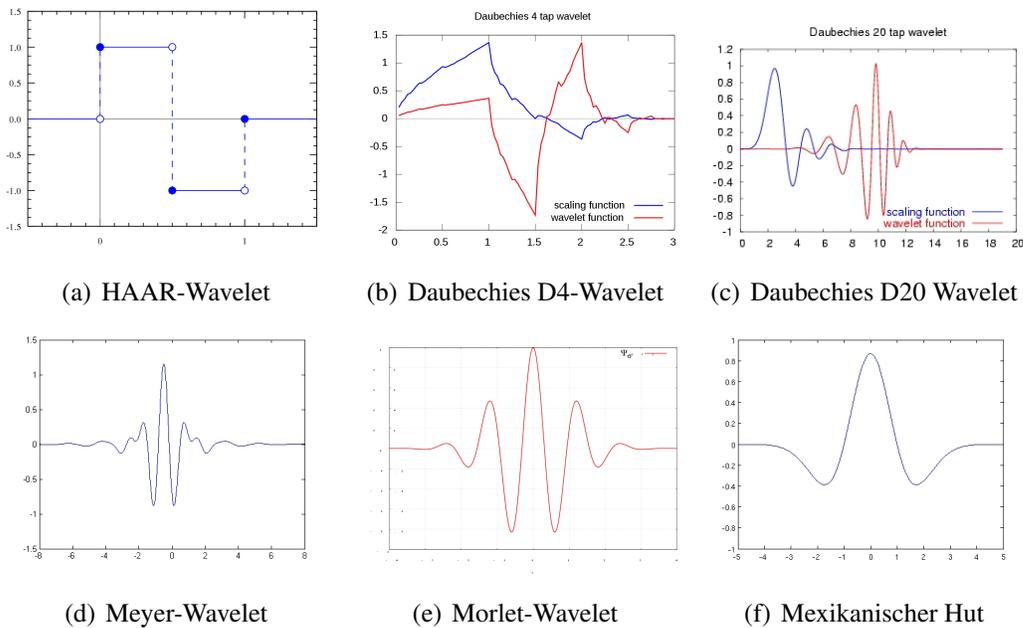


Bild 3.2: Funktionen der Wavelet-Transformation [Wik]

Basis für die Wavelet Transformation bilden ein Skalierungs- und eine Waveletfunktion. Ein Wavelet übersetzt kleine Welle ist ein Wellenfunktion mit der Besonderheit dass ihr Integral immer 0 ergibt. Betrachtet man die verschiedenen Funktionen welche als Wave-

let Funktion bei der Transformation zum Einsatz kommen, erklärt sich auch woher das Verfahren seinen Namen hat. Die Funktionen haben meist die Form von nach außen hinauslaufender immer kleiner werdender Wellen. Abbildung 3.2 zeigt die 6 am häufigsten verwendeten Funktionen in der Wavelet-Transformation.

Da ich mich in meinem Ansatz die diskrete Wavelet-Transformation benutze, verwende ich die HAAR-Funktion. Hierbei handelt es sich um einen Differenzfunktion, welche die Wavelet-Funktion darstellt, und eine Mittelwertfilter der als Skalierungs-Funktion dient. Wie diese beiden Funktionen auf die Bilddaten angewendet werden beschreibe ich im Abschnitt Wavelet-Transformation.

3.2.1 Wavelet Transformation

Als Grundlage für die Berechnung dienen die beiden HAAR-Funktionen sowie ein Grauwertbild mit einer Dimension die einer 2er Potenz entspricht. Diese Voraussetzung wird sich im Folgenden durch die Berechnungsschritte erklären. Da die Differenzfunktion einen Hochpass-Filterung (HP) und die Mittelwertfunktion einen Tiefpass-Filterung (TP) darstellt werde ich diese im Rechenweg so benennen. Eine sogenannte Stage (Wavelet-Transformations-Schritt) besteht aus vier Filterungsabfolgen. Aus dem Ausgangsbild B_{init} werden durch Filterung vier Bilder erzeugt, die sich jeweils in der Dimension halbieren.

- $B_{s1/HP,HP}$ ergibt sich aus der vertikalen und horizontalen Hochpass-Filterung
- $B_{s1/HP,TP}$ ergibt sich aus der vertikalen Hoch- und horizontalen Tiefpass-Filterung
- $B_{s1/TP,HP}$ ergibt sich aus der vertikalen Tief- und horizontalen Hochpass-Filterung
- $B_{s1/TP,TP}$ ergibt sich aus der vertikalen und horizontalen Tiefpass-Filterung

Durch die Differenz Funktion beinhalten die hochpassgefilterten Bilder die feinsten Kanten-Strukturen innerhalb des Bildes, welche bei Geständen meist durch Oberflächenmuster oder Aufschriften entstehen. $B_{s1/HP,HP}$ enthält die Diagonal ausgerichteten Kanten, wohingegen $B_{s1/HP,TP}$ die vertikalen und $B_{s1/TP,HP}$ die horizontalen Strukturen beinhaltet.

Durch die doppelte Tiefpass-Filterung fallen diese Strukturen in $B_{s1/TP,TP}$ weg oder werden abgemildert, so dass sie im nächsten Iterationsschritt, der auf $B_{s1/TP,TP}$ angewendet wird, nicht mehr oder nur noch schwach berücksichtigt werden. Abbildung 3.3 zeigt das Ergebnis eines solchen ersten Wavelet-Transformationsschrittes. In dieser Abbildung kann man sehr gut erkennen wie die Verortung der Kanten auch nach dem Transformationsschritt noch erhalten bleibt.

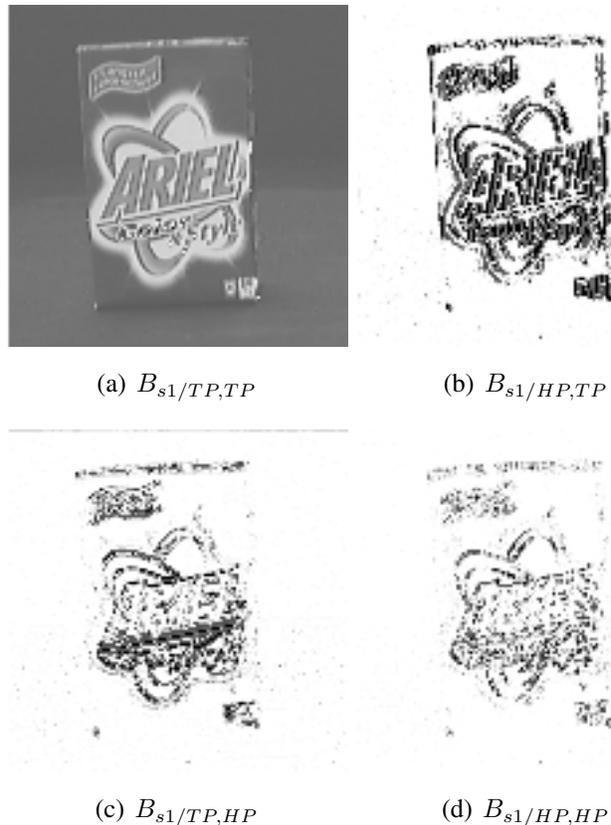


Bild 3.3: Ergebnis eines Wavelet-Transformationsschrittes

In jedem Iterationsschritt werden somit gröbere Strukturen durch die Hochpassfilterung abgebildet. Dieser Vorgang wird n -mal wiederholt bis das Teilbild $B_{sn/TP,TP}$ die Dimension 1 hat. Der Wert dieses einzelnen Pixels gibt die mittlere Helligkeit des Ausgangsbildes wieder. In höheren Stages können hier neben den Kanten auch noch Farbverläufe des Objektes detektiert und verarbeitet werden. In jedem Iterationsschritt werden somit gröbere

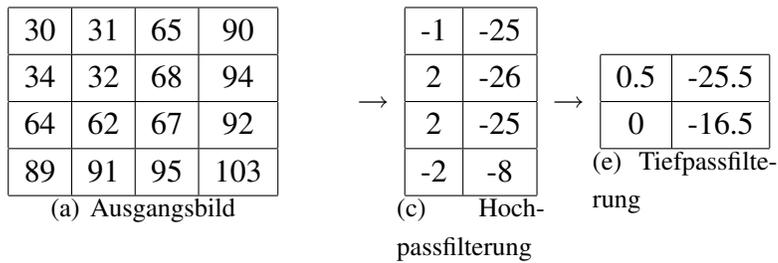


Bild 3.4: Rechenschritte zur Berechnung eines Teilbildes

Strukturen durch die Hochpassfilterung abgebildet. Dieser Vorgang wird n-mal wiederholt bis das Teilbild $B_{sn/TP,TP}$ die Dimension 1 hat. Der Wert dieses einzelnen Pixels gibt die mittlere Helligkeit des Ausgangsbildes wieder. In höheren Stages können hier neben den Kanten auch noch Farbverläufe des Objektes detektiert und verarbeitet werden.

3.2.2 Erstellung der Merkmalsvektors

Ergebnisbilder einer Wavelet-Transformation haben immer den Aufbau, der in Abb. xx zu sehen ist.

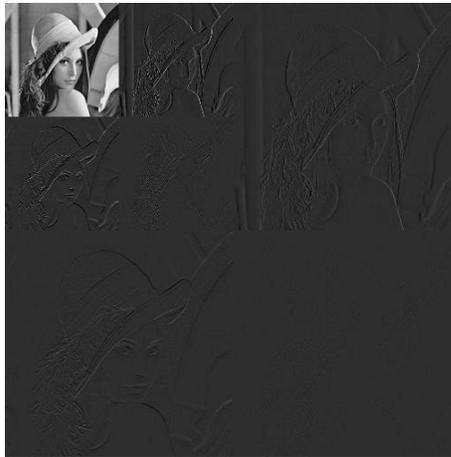


Bild 3.5: Lena zwei Stages

Für den späteren in der Erkennung notwendigen Abgleich ist diese Darstellungsform eher

ungeeignet, daher bietet es sich an die Informationen der einzelnen Stages s und deren Teilbilder $B_{sx/Filter,Filter}$ in einen Merkmalsvektor abzulegen.

Die Daten der Teilbilder werden in ein Histogramm ausgewertet. Hierzu werden zunächst die Werte $v_{y,B_{sx/Filter,Filter}}$ welche aus der Anwendung der Differenz-Funktion (Hochpass-Filterung) resultieren vom Wertebereich $[-256, 256]$ in einen positiven Wertebereich von $[0, 512]$ gebracht. Danach wird das Vorkommen der Werte aufsummiert und normalisiert im Histogramm abgelegt.

Damit erhält man ein für die Wavelet-Transformation eines Bildes einen Merkmalsvektor der Größe $s \times (3 \times g)$ (Abb.3.6). Hierbei beschreibt g die Größe der Histogrammbins. Diese kann variiert werden um mehrere Werte des Teilbildes zusammen zu fassen. Dadurch dass in jedem Schritt das Eingabebild für den nächsten Schritt doppelt Tiefpass gefiltert wird, werden die Ergebniswerte in höheren Stages immer kleiner. Diesen Umstand könnte man leicht korrigieren, wenn es sich denn hierbei um einen Fehler handeln würde. Richtigerweise kann man dies in der Betrachtung welche Merkmale davon betroffen sind als nützlich ansehen. So beschreiben feine Strukturen die in frühen Stages abgebildet werden das Objekt eindeutiger als grobe Strukturen (z.B. Farbverläufe) die erst in den letzten Stages berücksichtigt werden.

$$\begin{pmatrix} v_{0,B_{s1}/HP,TP} & v_{1,B_{s1}/HP,TP} & v_{2,B_{s1}/HP,TP} & \dots & v_{512,B_{s1}/HP,TP} \\ v_{0,B_{s1}/TP,HP} & v_{1,B_{s1}/TP,HP} & v_{2,B_{s1}/TP,HP} & \dots & v_{512,B_{s1}/TP,HP} \\ v_{0,B_{s1}/HP,HP} & v_{1,B_{s1}/HP,HP} & v_{2,B_{s1}/HP,HP} & \dots & v_{512,B_{s1}/HP,HP} \\ v_{0,B_{s2}/HP,TP} & v_{1,B_{s2}/HP,TP} & v_{2,B_{s2}/HP,TP} & \dots & v_{512,B_{s2}/HP,TP} \\ v_{0,B_{s2}/TP,HP} & v_{1,B_{s2}/TP,HP} & v_{2,B_{s2}/TP,HP} & \dots & v_{512,B_{s2}/TP,HP} \\ v_{0,B_{s2}/HP,HP} & v_{1,B_{s2}/HP,HP} & v_{2,B_{s2}/HP,HP} & \dots & v_{512,B_{s2}/HP,HP} \\ \dots & \dots & \dots & \dots & \dots \\ v_{0,B_{sn}/HP,TP} & v_{1,B_{sn}/HP,TP} & v_{2,B_{sn}/HP,TP} & \dots & v_{512,B_{sn}/HP,TP} \\ v_{0,B_{sn}/TP,HP} & v_{1,B_{sn}/TP,HP} & v_{2,B_{sn}/TP,HP} & \dots & v_{512,B_{sn}/TP,HP} \\ v_{0,B_{sn}/HP,HP} & v_{1,B_{sn}/HP,HP} & v_{2,B_{sn}/HP,HP} & \dots & v_{512,B_{sn}/HP,HP} \end{pmatrix}$$

Bild 3.6: Merkmalsvektor einer Wavelet-Transformation, $g = 512$

Diese Berechnung wird innerhalb des Lernprozesses für mehrere Ansichten des Objektes durchgeführt. Die aus den verschiedenen Ansichten entstehenden Merkmalsvektoren

werden Werteweise arithmetisch gemittelt um zum Abschluss der Lernphase einen Merkmalsvektor für das jeweilige Objekt zu erhalten.

Mit diesem Vektor können wir jetzt bestimmen ob sich das Objekt in einer von der Kamera aufgenommenen Szene befindet. Dazu wird von der Szene ebenfalls ein Merkmalsvektor erstellt und mit dem des Objektes verglichen, und der Euklidische-Abstand zwischen den beiden Vektoren ermittelt. Wird dieser Abstand kleiner so steigt die Wahrscheinlichkeit dass sich das gesuchte Objekt in der Szene befindet. In den seltensten Fällen liegt das Objekt in der Szene formatfüllenden vor und bildet sich daher nur schwach in der Vektorbeschreibung der Szene ab. Um dieses Problem zu lösen bedarf es einer Segmentierung des Bildes, diese kann sowohl auf einer Rückprojektion des Objekt-Merkmalsvektors auf die Szene als auch auf der Verwendung von Farbe als Merkmal beruhen. Durch die Segmentierung können uninteressante Bereiche im Bild schon vor der Wavelet-Transformation ausgeblendet werden.

3.3 Farbe

Farbe Das zweite Objektmerkmal welches in meinem Ansatz zum Einsatz kommt ist die Objektfarbe. Hierbei beschränkt sich das Modell nicht nur auf eine Farbe sondern stellt alle Farben, die in dem Gegenstand vorkommen, dar. Farben beschreiben ein Objekt jedoch nicht uneindeutig. Zwar kann man durch die Ermittlung von Relationen zwischen den Objektfarben und der Häufigkeit des Auftretens im Objektfarbraum dieses Verfahren sicherer machen. Aber dennoch ist in den meisten Fällen nicht möglich nur anhand der Farbe einen roten Ball und eine roten Apfel zu unterscheiden. Daher benutze ich das Merkmal Farbe in Kombination mit Textur und Form. Für die Darstellung des Merkmals habe ich den H (Hue - Farbwinkel) S (Saturation - Sättigung) V (Value - Helligkeitswert) Farbraum gewählt. Mit diesem Farbraum lassen sich Farben sehr leicht beschreiben, da klarer ist was passiert wenn sich einer der drei Werte ändert. Betrachtet man im Vergleich den additiven RGB-Farbraum verhält sich dieser sehr viel sprunghafter. Um die Betrachtung weiter zu vereinfachen setzte ich den Helligkeitswert auf 1 fest und betrachte lediglich die Werte H und S. Grundlage für diese Entscheidung ist das ich Objekte beleuchtungs-invariant betrachten möchte, und sich der größte Anteil der Beleuchtung in dem Helligkeitswert

V wieder spiegelt. Dennoch kann man keinesfalls sagen, dass der HSV beleuchtungsinvariant ist, da auch z.B. die Objektfarbe also der Farbwinkel H durch die Farbe des zur Beleuchtung verwendeten Licht verändert wird. Meistens jedoch liegen Bilder im additiven Farbraum RGB vor, welches von Monitoren zur Ausgabe genutzt wird. Somit ist eine Vorverarbeitung in Form eine Farbraum Transformation notwendig. Abb. 3.7 zeigt die in meiner Umrechnung verwendete Formel welche in der Methode CvtColor der Bildverarbeitungsbibliothek OpenCV implementiert ist.

$$V = \max(R, G, B)$$

$$S = \begin{cases} V \neq 0 & | \quad V - \min(R, G, B)) * 255/V \\ \text{sonst} & 0 \end{cases}$$

$$H = \begin{cases} V = R & | \quad (G - B) * 60/S \\ V = G & | \quad 180 + (B - R) * 60/S \\ V = B & | \quad 240 + (R - G) * 60/S \end{cases}$$

$$H < 0 \Rightarrow H = H + 360$$

Bild 3.7: Formel zur Umrechnung von RGB in HSV

3.3.1 Erstellung des Farb-Histogramm

Wie schon im vorangehenden Abschnitt beschrieben werden nur die Werte H und S eines jeden Pixels innerhalb des Bildes betrachtet. Die Verknüpfung dieser beiden Werte darf nicht gelöst werden das sonst die Informationen verloren gehen, daher setze ich an dieser Stelle ein zweidimensionales Histogramm ein. Die Dimensionen sind H und S, wobei eine Bingröße gewählt werden kann. Im Kapitel Evaluation werde ich verschiedene Bingrößen und die Ergebnisse die damit erzielt werden vergleichen und bewerten.

Zum Füllen des Histogramms wird jeder Pixel im Bild betrachtet und um jeweils dem H und S Wert entsprechend das passende Histogrammfeld an der Stelle $[H; S]$ um eins erhöht. Als Ergebnis erhält man eine Farbkarte des Eingabebildes.

Auch hier können wie schon beim Merkmalsvektor der Wavelet in der Lernphase die Histogramme der verschiedenen Ansichten des Objektes arithmetisch gemittelt werden um ein Objekt-Farbhistogramm zu erstellen welches das Farbspektrum des Objekts aus allen Blickwinkeln abbildet.

3.4 Rückprojektion

Eine Rückprojektion ist ein Transformation der Werte aus dem Merkmalsraum zurück in den Bildraum. Sowohl die Merkmalsvektoren aus der Wavelet-Transformation als auch die Farbhistogramme können für dieses Verfahren verwendet werden. Die Merkmalswerte des Eingabebildes werden mit denen des zu erkennenden Objektes gefiltert. Als Ausgabe erhält man ein Bild der Szene in dem alle nicht im Objektmerkmalsraum vorkommenden Werte ausgeblendet sind. Auf Basis dieser Bilddaten kann segmentiert und in den folgenden Schritten detektiert und klassifiziert werden.

3.4.1 Farbrückprojektion

Voraussetzung für die Farbrückprojektion ist, dass die zweidimensionalen Histogramme von Szene H_s und Objekt H_o die gleiche Größe besitzen. Dies ist dadurch gewährleistet, dass das Histogramm der Szene mit demselben Algorithmus berechnet wird, der auch zum Erlernen der Objekte genutzt wurde. Es wird die Bingröße, welche bei der Erstellung des Objekthistogramms gewählt wurde angewendet.

Aus den beiden Histogrammen wird ein Quotienten Histogramm H_q erstellt. Der Wert im Quotienten-Histogramm ergibt sich aus der Division der korrespondierenden Histogrammwerte von Objekt- und Szenehistogramm. Dadurch wird eine Bewertung der Objektfarbe für die Erkennung innerhalb der Szene erreicht. Kommt zum Beispiel eine Objektfarbe in der Szene nur in einem kleinen Bereich vor, so ist sie für die Detektion des Objektes von sehr hohem Interesse, während eine Objektfarbe, welche großflächig in der Szene vorkommt eher von geringem Interesse ist.

$$\forall(x, y)H_q[x, y] = \begin{cases} H_s[x, y] \neq 0 \mid H_q[x, y] = H_o[x, y]/H_s[x, y] \\ H_s[x, y] = 0 \mid H_q[x, y] = 0 \end{cases}$$

Da bei der Erstellung des Histogramms die Ortsinformation verloren geht, benötigt man zur Rückprojektion das in den HSV-Farbraum transformierte Bild der Szene B_{szene} . Dort wird unter Berücksichtigung des Quotientenhistogramms H_q ein Grauwertbild B_{back} erstellt. Es wird geprüft, welcher H bzw. S Wert an der Stelle x vorliegt und im Grauwertbild g der Histogrammwert aus $H_q[H, S]$ eingetragen.

Das Grauwertbild B_{back} ist also eine Interessenskarte, in der Bereiche an denen sich das Objekt mit hoher Wahrscheinlichkeit befinden kann abgebildet sind. Um interessante Regionen in diesem Bild zu finden, muss zunächst ein Schwellwert auf das Bild angewendet werden, wodurch ein binarisiertes Bild entsteht, das nur noch zwei Werte besitzt.



(a) Objektbild

(b) Szenebild

(c) Rückprojektion

Bild 3.8: Farb-Histogram-Rückprojektion

Auf dieses binarisierte Bild wird werden morphologischer Operatoren angewendet. Das Strukturelement, welches auf das Bild angewendet hat eine Dimension von fünf mal fünf Pixel. Um natürliche Formen im Bild zu begünstigen verwende ich eine Maske von runder Form.

Zunächst wird eine Dilatation durchgeführt, welche sich nicht verbundene Flächen verbinden. Darauf folgt eine Erosion mit demselben Strukturelement, was dazu führt, dass

Flächen, die bei der Dilatation durch kleine Störungen entstanden sind, verschwinden. Die Anwendung der morphologischen Operatoren in dieser Reihenfolge nennt man “schließen”, da große, getrennte Flächen zusammenwachsen und kleinere Störungen verschwinden. Somit ergeben sich gut detektierbare zusammenhängende Flächen welche ich im Folgenden als Segmente bezeichne.

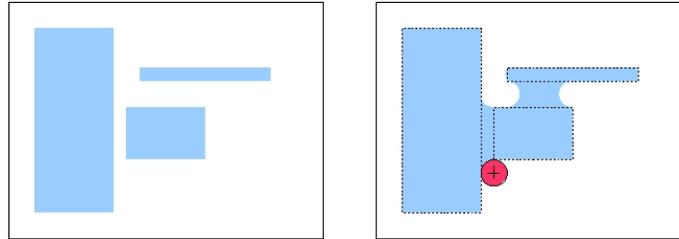


Bild 3.9: Beispiel für das Schließen eines Bildes

Diese Segmente können mit der OpenCV Funktion `cvFindContours()` detektiert werden. Den Algorithmus dieser Funktion werde ich in dieser Arbeit nicht erläutern, er ist in der Dokumentation der OpenCV Bildbearbeitung Bibliothek nachzulesen.

3.4.2 Wavelet-Rückprojektion

Bei der Rücktransformation von einem durch die Wavelet-Transformation erzeugten Merkmalsvektor muss zunächst die Wavelet-Transformation auf das Bild der Szene angewendet werden. Hierbei muss darauf geachtet werden, dass dieselbe Wavelet-Funktion verwendet wird die auch in der Lernphase verwendet wurde (in meinem Ansatz HAAR). Für das Szenenbild gilt auch die Voraussetzung, dass die Dimension ein Vielfaches von zwei sein muss. Um in der Transformation genügend Stages zu erhalten muss die Größe des Szenenbildes mindestens die Größe des Objektbildes in der Lernphase aufweisen.

Nach der Wavelet-Transformation werden die einzelnen Teilbilder Pixel für Pixel durchlaufen und mit den korrespondierenden Zeilen des Objekt Merkmalsvektors analysiert. Kommt ein Wert aus dem Merkmalsvektor vor so wird in einem Rückprojektionsbild

$B_{waveBack}$ der Histogrammwert an die zur Position im Teilbild passende Stelle geschrieben. Diese Stelle ist eine Fläche benachbarter Pixel, und verdoppelt ihre Dimension mit jeder Stage.

Als Ergebnis erhält man ähnlich der Farb-Histogramm-Rückprojektion ein Grauwertbild, das eine Interessenskarte darstellt. Auf dieses Grauwertbild werden dieselben Verarbeitungsschritte angewendet, die auch auf B_{back} im vorangehenden Abschnitt angewendet wurden.

3.5 Kombination von Wavelet und Farbe

Zum Abschluss meines Ansatzes müssen die zwei Verfahren sinnvoll kombiniert werden. Es gibt mehrere mögliche Kombinationen. Während der Programmentwicklung hat sich gezeigt, dass eine vorgelagerte Segmentierung und Detektion von für die Objekterkennung interessanten Bildausschnitten mit der Farb-Histogramm-Rückprojektion nicht nur bessere Ergebnisse liefert sondern auch einfacher umzusetzen ist.

Als Problem haben sich die Voraussetzungen für die Wavelet-Transformation herausgestellt. Da die Dimensionen des Eingabebild ein Vielfaches von zwei sein müssen die Eingabe Bild oft skaliert oder beschnitten werden. Bei beiden Operationen können wichtige Information für die Objekterkennung verloren gehen. Des Weiteren sind die Merkmalsvektoren um ein vielfaches größer als die der Farb-Histogramm-Rückprojektion, was eine längere Rechenzeit für die Verarbeitung zur Folge hat. Filtert man das Bild zunächst mit Hilfe der Farb-Histogramm-Rückprojektion, kann man im zweiten Schritt, die interessanten Bereiche an die Voraussetzungen der Wavelet-Transformation anpassen.

Abb. 3.9 zeigt diese Verarbeitungsreihenfolge. Hier ist zu sehen wie der untere Bildausschnitt, der an der Milchpackung detektiert wurde, an die Wavelet Voraussetzung angepasst wird. Nun kann für diesen Ausschnitt ermittelt werden wie wahrscheinliches es ist das er das gesuchte Objekt beinhaltet.

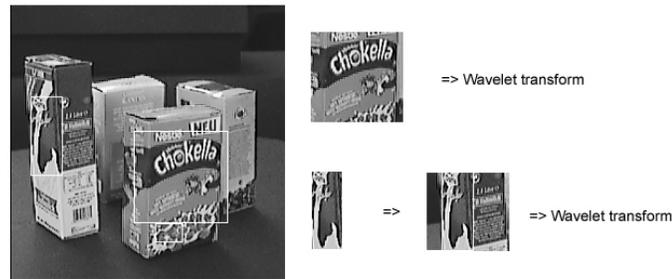


Bild 3.10: Wavelet Transformation auf Segmenten des Farbhistogramm-Rückprojektion

3.6 Programmierung

Ich verwende in meinem Programmcode die quelloffenen Bibliotheken OpenCV und für die Wavelet-Transformation von Bildern entwickelte Bibliothek (<http://sourceforge.net/projects/wavelet/>). Um beide Bibliotheken nutzen zu können, habe ich eine Konvertierungsfunktion geschrieben, die die Bild-Klassen jeweils untereinander konvertiert.

Wavelet Transformation und Farb-Histogramm-Rückprojektion sind im Programmcode separiert, um sie Tests der einzelnen Verfahren zu ermöglichen zu machen. Das Programm ist komplett konsolenbasiert und besitzt eine große Anzahl an Eingabe-Variablen. Um das Projekt kompilieren zu können werden die beiden oben angegebenen Bibliotheken benötigt.

3.6.1 ColorHistogram class

Die Klasse ColorHistogram bietet die Funktion calculate, die das Histogramm eines Bildes erstellt, diesem aber noch weitere Bilddaten hinzufügt. Diese Funktionalität wird beim Lernen von Objekten benötigt, da ein Objekt je nach Ansicht verschiedene Objektfarben besitzen kann, welche sich alle in einem kompletten Objekt-Farbhistogramm wiederfinden müssen.

Um ein gelerntes Histogramm zu speichern nutze ich die Methode save(), die das Hi-

stogramm im CVS-Format in einer Datei ablegt. Der Konstruktor kann mit dieser Datei aufgerufen werden und das Histogramm wird wieder initialisiert.

Die Rückprojektion wird in der Funktion `backprojection()` durchgeführt. Als Parameter wir hier das Bild übergeben, auf welches die Rückprojektion angewendet werden soll. Um die zwei Histogramme, die als Berechnungsgrundlage dienen, in den gleichen Wertebereich abzubilden, wird die `normalize` Funktion verwendet, die das Histogramm anhand des Maximalwertes auf den Wertebereich $[0,1]$ abbildet.

$$h_{max} = \max(H[x, y])$$

$$\forall H_{norm}[x, y] = H[x, y]$$

Die Klasse `ColorHistogram` bietet eine Funktion `draw()`, welche das Histogramm mit dem als Parameter übergebenen Faktor skaliert und auf dem Bildschirm darstellt.

Mit der Funktion `equal()` wurden die unter Abschnitt 4.1 gezeigten Evaluationstabellen berechnet. Die Funktion ermittelt die mittlere Distanz $dist_{mean}$ zwischen dem bestehenden normierten $H1_{norm}$ und einem aus dem übergebenen Bild berechneten normierten Histogramms $H2_{norm}$. Bei der Erstellung des zweiten Histogramms wird dafür gesorgt, dass es die selbe Dimension n besitzt um eine Abstandsberechnung zu ermöglichen.

$$dist_{mean} = \frac{\sum_{x=0}^n \sum_{y=0}^n |H1_{norm}[x,y] - H2_{norm}[x,y]|}{n*n}$$

Kapitel 4

Experimente und Ergebnisse

Testgrundlage bildet die AGAS Bilddatenbank der Universität Koblenz. Es wurden Wavelet- und Farb-Merkmalvektoren von 17 Objekten erstellt. In die Erstellung dieser Vektoren wurden jeweils 10 Bilder eines jeden Objektes verwendet, welche diese aus verschiedenen Blickwinkeln zeigen. Die Beleuchtung ist bei allen Bildern identisch. Alle Objekte wurden vor einem einheitlich grauen Hintergrund erstellt. Der Hintergrund wird für die Histogrammberechnung ausmaskiert und in den Berechnungen nicht mit einbezogen, um das Ergebnis nicht zu verfälschen (Abb 4.1).

4.1 Wavelet Evaluation

Die Auswertung (siehe Tabellen Anhang zu dieser Arbeit) zeigt, dass sich die besten Ergebnisse bei einer Histogrammbingröße von 1 und vierfacher Anwendung der Wavelet Transformation und einer Bildgröße von 256×256 Pixel einstellen.

Es wird bei 17 Objekten mit jeweils 10 Bildern eine Erkennungsrate von 77,06 % erzielt.

Ein Objekt gilt als erkannt, wenn der niedrigste Abstandswert in dem Tabellen-Abschnitt des jeweiligen Objektes liegt. In den Auswertungstabellen ist dieser Bereich durch einen



Bild 4.1: Zugeschnittenes Objektbild aus der Bilddatenbank

grünen Rahmen gekennzeichnet.

Die niedrigen Abstandswerte resultieren aus der starken Ähnlichkeit der Objekte. Alle Objekte haben nahezu die gleiche Form und unterscheiden sich hauptsächlich durch ihre Beschriftung.

Um hier die Ergebnisse noch zu verbessern, kann man bei der Erstellung der Wavelet Transformation andere Wavelet Funktionen verwenden, die eine größere Filtermaske besitzen und somit auch komplexere Strukturen abbilden können. Darauf wird in dieser Arbeit aber nicht weiter eingegangen.

Um die Wavelet Transformation zu ermöglichen war es zunächst notwendig, die Bilder auf eine Größe zu bringen, die einer Zweier-Potenz entspricht. Hierzu habe ich die Objektbilder manuell vor der Verwendung zugeschnitten und in ein Grauwertbild umgewandelt. Von jedem Objekt habe ich 10 Bilder erstellt. Danach wurde von jedem dieser Bilder ein Merkmalsvektor erstellt und die Abstände aller Merkmalsvektoren berechnet. Zur Berechnung habe ich den absoluten Abstand jedes einzelnen Wertes herangezogen.

Object	Ariel	Brandt	Chokella	coral	Domo	Hipp	Iglo	Jod	Kaba
Good/Bad	7/3	9/1	2/8	6/4	10/0	8/2	8/2	9/1	10/0

Object	Kandis	Kinder	Osram	Proper	Suchard	Sunil	Tchibo	Weiss
Good/Bad	9/1	10/0	10/0	6/4	6/4	8/2	7/3	6/4

Bild 4.2: Evaluation - Wavelet Transformation: 4 Stages / Binggröße 1

4.2 Farb-Histogramm Evaluation

Es wird zunächst von jedem Bild ein Histogramm erstellt, welches mit allen Histogrammen verglichen wird. Als Vergleichsergebniswert wird der mittlere Abstand zweier Histogramme genutzt. Ein Objekt gilt dann als richtig erkannt, wenn der niedrigste Abstands-wert innerhalb der 10 Objektbilder liegt, die das Objekt selber zeigen. Der Vergleich des Histogramms mit sich selber wird ausgenommen, da dieser immer null ergibt. Trägt man die Daten in eine Tabelle ein, erkennt man, dass sich alle Werte an der Diagonalen spie-geln, woraus auf eine Kommutativität des Vergleichsverfahrens zu schließen ist, was die Überprüfung der Testergebnisse erleichtert.

ariel1	ariel2	ariel3	ariel4	ariel5	ariel6	ariel7	ariel8	ariel9	ariel10	brandt1	brandt2	brandt3	brandt4	brandt5	brandt6	brandt7	brandt8	brandt9	brandt10	chokella1	chokella2	chokella3	
ariel1	0.000000	0.111842	0.012887	0.012836	0.005274	0.010850	0.011814	0.011881	0.012609	0.007826	0.020702	0.016951	0.018038	0.018641	0.019815	0.018758	0.016019	0.018506	0.017529	0.017909	0.017887	0.016153	0.017629
ariel2	0.011842	0.000000	0.006453	0.009124	0.010252	0.009507	0.007368	0.010085	0.008542	0.019138	0.016289	0.019040	0.019995	0.019604	0.018824	0.014763	0.018077	0.015671	0.014182	0.017465	0.014131	0.016557	0.016557
ariel3	0.012887	0.006453	0.000000	0.005627	0.009063	0.008904	0.007850	0.007488	0.007655	0.020927	0.016804	0.017971	0.020508	0.022653	0.019570	0.015481	0.017709	0.015715	0.015285	0.016537	0.013573	0.014478	0.014478
ariel4	0.012636	0.009124	0.005627	0.000000	0.006884	0.008443	0.009554	0.008510	0.007580	0.022690	0.017422	0.018775	0.021546	0.021444	0.020551	0.016363	0.018138	0.016880	0.016978	0.017422	0.014132	0.014460	0.014460
ariel5	0.005274	0.010252	0.009063	0.006884	0.000000	0.008754	0.010426	0.008811	0.009645	0.007073	0.021175	0.017721	0.018851	0.019998	0.020390	0.019555	0.016803	0.019054	0.017435	0.017102	0.016798	0.014148	0.015411
ariel6	0.010850	0.009507	0.008904	0.008443	0.008754	0.000000	0.008883	0.007372	0.008729	0.008917	0.022372	0.017310	0.018479	0.020524	0.020809	0.020477	0.017286	0.018669	0.017867	0.018632	0.019264	0.015542	0.016870
ariel7	0.011814	0.007368	0.007850	0.009554	0.010426	0.008883	0.000000	0.007966	0.010848	0.009377	0.020266	0.017366	0.019569	0.021233	0.020419	0.018961	0.016249	0.019115	0.017079	0.015677	0.017027	0.013518	0.015951
ariel8	0.011881	0.007368	0.007850	0.009554	0.010426	0.008883	0.000000	0.007966	0.010848	0.009377	0.020266	0.017366	0.019569	0.021233	0.020419	0.018961	0.016249	0.019115	0.017079	0.015677	0.017027	0.013518	0.015951
ariel9	0.012609	0.010085	0.008542	0.009507	0.009645	0.008729	0.010848	0.007966	0.000000	0.008286	0.024221	0.018923	0.019668	0.022632	0.022632	0.021652	0.018123	0.019234	0.018372	0.019754	0.018289	0.015071	0.014259
ariel10	0.007826	0.008542	0.007368	0.009507	0.009645	0.008729	0.010848	0.007966	0.000000	0.008286	0.024221	0.018923	0.019668	0.022632	0.022632	0.021652	0.018123	0.019234	0.018372	0.019754	0.018289	0.015071	0.014259
brandt1	0.020702	0.016951	0.020507	0.022890	0.021175	0.022372	0.020266	0.020954	0.024221	0.021242	0.000000	0.011032	0.012025	0.010789	0.009071	0.012422	0.012207	0.013842	0.013497	0.012221	0.022921	0.022074	0.024797
brandt2	0.016951	0.016289	0.016884	0.017422	0.017721	0.017310	0.017366	0.016652	0.018923	0.016865	0.011032	0.000000	0.008182	0.008160	0.010994	0.009783	0.006091	0.008501	0.012221	0.021664	0.018982	0.021925	0.021925
brandt3	0.018038	0.019040	0.017971	0.018775	0.018851	0.018479	0.019569	0.017474	0.019668	0.018038	0.012025	0.000000	0.006818	0.008898	0.009239	0.010289	0.006674	0.011033	0.015969	0.022112	0.020787	0.021796	0.021796
brandt4	0.018641	0.019995	0.020508	0.021546	0.019998	0.020524	0.021233	0.019708	0.022832	0.020187	0.010789	0.008160	0.008898	0.000000	0.010521	0.010254	0.012603	0.010617	0.012663	0.017478	0.024474	0.023702	0.024688
brandt5	0.019815	0.018758	0.020263	0.021444	0.020390	0.020809	0.020419	0.019531	0.022632	0.020804	0.009103	0.010994	0.010521	0.000000	0.009669	0.013389	0.010596	0.013820	0.015970	0.022157	0.021860	0.022826	0.022826
brandt6	0.018758	0.018824	0.019570	0.020551	0.019555	0.020477	0.019861	0.018641	0.021652	0.020258	0.012422	0.009783	0.009239	0.010254	0.009669	0.000000	0.012424	0.010106	0.012276	0.016429	0.023199	0.022331	0.023521
brandt7	0.016019	0.014763	0.015481	0.016363	0.016803	0.017286	0.016249	0.016150	0.018123	0.015961	0.012207	0.006091	0.010289	0.012603	0.013389	0.012424	0.000000	0.008137	0.007150	0.007929	0.018860	0.016029	0.019513
brandt8	0.018506	0.018077	0.017709	0.018138	0.019054	0.018669	0.019115	0.017594	0.019234	0.018227	0.013842	0.006919	0.009674	0.010617	0.010596	0.010106	0.008137	0.000000	0.007736	0.013378	0.022380	0.019417	0.021780
brandt9	0.017529	0.015671	0.015715	0.016880	0.017435	0.017867	0.017079	0.015962	0.018372	0.016587	0.013497	0.008501	0.011033	0.012663	0.013620	0.012276	0.007736	0.000000	0.007849	0.021168	0.018066	0.020665	0.020665
brandt10	0.017909	0.014182	0.015285	0.016978	0.017102	0.018632	0.015577	0.016662	0.018754	0.016184	0.012221	0.012221	0.015896	0.017470	0.015970	0.016429	0.007529	0.013371	0.006734	0.000000	0.018025	0.016324	0.016986
chokella1	0.017887	0.017465	0.016537	0.017422	0.016798	0.019264	0.017027	0.017868	0.018289	0.016978	0.022921	0.021684	0.022112	0.024474	0.022157	0.023199	0.018850	0.022330	0.021188	0.019025	0.000000	0.008943	0.009442
chokella2	0.016153	0.014131	0.013573	0.014132	0.014480	0.015542	0.013518	0.015514	0.015071	0.013276	0.022074	0.021892	0.020767	0.023702	0.021860	0.022331	0.016029	0.019417	0.018006	0.016324	0.008943	0.000000	0.007803
chokella3	0.017629	0.016557	0.014478	0.014460	0.015411	0.016870	0.015951	0.015436	0.014259	0.014725	0.024797	0.021796	0.024888	0.022826	0.023521	0.019513	0.021780	0.020606	0.019988	0.008442	0.007803	0.000000	0.000000
chokella4	0.016543	0.017583	0.016609	0.015957	0.014498	0.017112	0.017215	0.016886	0.015851	0.014999	0.025797	0.021778	0.021088	0.024390	0.022799	0.023199	0.019979	0.021376	0.021265	0.021665	0.009925	0.010549	0.009784
chokella5	0.013288	0.017588	0.016615	0.018186	0.016835	0.019513	0.017252	0.017796	0.018559	0.017415	0.024022	0.022994	0.023187	0.023582	0.023393	0.024376	0.022682	0.022470	0.021950	0.021950	0.007804	0.011019	0.002250
chokella6	0.016728	0.016797	0.016356	0.018019	0.015364	0.018743	0.016586	0.016132	0.018635	0.016564	0.023096	0.022535	0.023552	0.025576	0.023798	0.024117	0.018963	0.023017	0.020653	0.017752	0.009889	0.008106	0.008064
chokella7	0.016833	0.015523	0.015129	0.016158	0.015318	0.016683	0.014882	0.017149	0.017076	0.014673	0.023056	0.022092	0.022173	0.024730	0.023419	0.023780	0.017031	0.020619	0.018793	0.016979	0.008772	0.007941	0.005931
chokella8	0.017074	0.016545	0.014689	0.014898	0.014938	0.017025	0.016196	0.016148	0.015203	0.014311	0.023056	0.022092	0.022173	0.024730	0.023419	0.023780	0.017031	0.020619	0.018793	0.016979	0.008772	0.007941	0.005931
chokella9	0.016900	0.017325	0.015637	0.015899	0.014642	0.018008	0.017183	0.016943	0.015974	0.015373	0.025886	0.022628	0.022416	0.025103	0.023531	0.020380	0.020308	0.022696	0.021715	0.020978	0.009215	0.009993	0.008302
chokella10	0.014718	0.016496	0.015639	0.016633	0.014286	0.017468	0.015891	0.016555	0.017470	0.015220	0.020985	0.020010	0.020461	0.021706	0.020280	0.021183	0.017103	0.020885	0.019002	0.016978	0.008713	0.010468	0.010589
coral1	0.024307	0.020537	0.020050	0.022271	0.021638	0.024614	0.021308	0.022240	0.022611	0.021693	0.026033	0.026901	0.028425	0.028876	0.029062	0.029214	0.023071	0.028211	0.025401	0.020724	0.017552	0.018032	0.018335
coral2	0.024506	0.020403	0.020302	0.022528	0.021925	0.024794	0.021124	0.022387	0.022743	0.021723	0.026333	0.026798	0.028637	0.029940	0.029303	0.029535	0.023383	0.028475	0.025479	0.021002	0.019448	0.017588	0.020160
coral3	0.024239	0.020888	0.019489	0.021178	0.021587	0.024490	0.022136	0.022184	0.021677	0.021510	0.027205	0.028088	0.028724	0.030579	0.030105	0.030470	0.024582	0.029860	0.026775	0.022702	0.019237	0.018502	0.019437

Bild 4.3: Ausschnitt aus der Farbhistogramm Evaluation

Abbildung 4.3 zeigt einen Auszug aus der Evaluation sowie eine Auswertungstabelle. Im Idealfall lässt sich bei einer Anzahl von 25 Bins eine 91,12 prozentige Erkennungsrate errechnen. Die Treffer werden für die verschiedenen Objekte aufgezählt und unterteilt in Good-Counts, welche die richtigen Entscheidungen und Bad-Counts, welche die falschen Entscheidungen des Algorithmus wiedergeben.

Um ein noch besseres Ergebnis zu erzielen, könnte man die Bingröße für die beiden Dimensionen H und S entkoppeln, da sich deren Wertebereiche in der Größe unterscheiden. Auf diesen Verbesserungsansatz gehe ich aufgrund der erzielten hohen Erkennungsrate in dieser Arbeit nicht ein.

Object	Ariel	Brandt	Chokella	coral	Domo	Hipp	Iglo	Jod	Kaba
Good/Bad	10/0	10/0	8/2	9/1	10/0	10/0	7/3	10/0	10/0
Object	Kandis	Kinder	Osram	Proper	Suchard	Sunil	Tchibo	Weiss	
Good/Bad	8/2	9/1	10/0	10/0	10/0	10/0	6/4	8/2	

Bild 4.4: Evaluation - Farbhistogramme: 25 Bins (komplette Tabelle im Anhang zu dieser Arbeit)

Die kompletten Evaluationstabellen für Farbhistogramme sowie Wavelet-Merkmalsvektoren sind im Anhang dieser Arbeit zu finden.

4.3 Evaluationsschluss

Die Evaluation der beiden Verfahren ergibt, dass sich die Objektfarbe besser zur Bildsegmentierung und damit zu Detektion eignet, während die Wavelet-Transformation bessere Ergebnisse in der Objektklassifizierung liefert. In meinem Ansatz habe ich die einfachste Wavelet Funktion HAAR genutzt. Die Ergebnisse lassen sich durch den Einsatz von stetigen Wavelet-Funktionen wie Daub4, Daub20 oder andere sicher noch verbessern.

Die Ausführungszeiten liegen bei gleicher Anzahl von Objekten um 13 Sekunden pro Ausführungsschritt höher als bei der Farb-Histogramm-Rückprojektion. Ein Ausführungsschritt bezeichnet hier den Vergleich eines Objekt Bildes mit allen anderen Objekten aus den Testwerten. Daher ist es sinnvoll, zunächst die Farbhistogramm Rückprojektion durchzuführen und danach die Wavelet-Transformation auf eine kleineren Bildbereichen, der den Voraussetzungen der Transformation genügt, anzuwenden was die Ausführungszeit drastisch senkt. Dennoch ist zu beobachten, dass die Ergebnisse je nach Objekt sehr stark schwanken, was zeigt das die Merkmale Farbe und Textur nicht für jeden Gegenstand als Beschreibung ausreichend sind.

Da es sich hier vorwiegend um Laborbilder handelt, welche unter normierter Beleuchtung aufgenommen wurden, sind Schwankungen die sich durch ändernde Beleuchtungsverhältnisse ergeben nicht berücksichtigt. So zeigt das Verfahren noch einige Schwächen. Die sequenzielle Ausführung ist hier eine der Hauptschwächen. Schlägt die Farb-Histogramm-Projektion fehl so tut dies auch die Wavelet-Transformation, da sie falsche oder keine Eingabe Daten erhält.

Kapitel 5

Zusammenfassung

Die Nutzung mehrerer Merkmalsräume führt zu einer Steigerung der Erkennungsrate. Jedoch darf die Menge der für einen Merkmalsvektor gespeicherten Daten nicht zu groß werden, da es sonst zu Laufzeitproblemen in der zeitkritischen Erkennungsphase kommt. Der Vergleichsaufwand steigt stark an, da aus dem zu analysierenden Bild annähernd dieselbe Menge Daten generiert werden müssen und mit allen Objektdaten aus der Objektdatenbank verglichen werden müssen.

Hierbei könnte ein Baumstruktur innerhalb der Objektdatenbank helfen, die mittels abstrakter Objekte in den oberen Ebenen eine Vorselektierung anhand bestimmter Merkmalsvektorstärken trifft und damit unnötige Vergleichsoperationen verhindert.

Im Allgemeinen berücksichtigen die wenigsten Objekterkennungsmethoden den semantischen als auch realen Kontext eines Objektes, welche für die menschliche Wahrnehmung eine essentielle Bedeutung hat. Dieses Merkmal könnte man neben den anderen in dieser Arbeit vorgestellten Merkmalen in einen Erkennungsalgorithmus einbringen. Solche Algorithmen, die diese Zusammenhänge verarbeiten, benötigen große semantische Netzwerke, die die Beziehung von Objekten zueinander sowie die Wahrscheinlichkeit des Vorkommens von Objekten im Umfeld von anderen Objekten abbilden. Auch hier müsste natürlich auch darauf geachtet werden dass die Vergleichsoperationen nicht die nahezu echtzeitfähige Ausführung behindern.

Die Entwicklung von Objekterkennungsmethoden wird in Zukunft stark durch die Hardware auf der sie genutzt wird bestimmt. Daraus resultieren auch neue Merkmalsräume. So erhält man zum Beispiel bei modernen Handys die Möglichkeit zu den Kameradaten noch Information über die Position per GPS zu erhalten. Es werden auch mehrere Merkmalsräume benötigt, da diese Arbeit zwar gezeigt hat das die verbundene Nutzung verschiedener Merkmalsräume Vorteile bringt aber noch nicht genügt um Objekte eineindeutig unter allen Bedingungen zu erkennen.

Literaturverzeichnis

- [CGL03] CHRISTIAN GROSSE LORDEMAN, Martin L.: Objekterkennung in Bilddaten. (2003)
- [Gra95] GRAPS, Amara: An Introduction to Wavelets. (1995)
- [JML99] JITENDRA MALIK, Jianbo S. Serge Belongie B. Serge Belongie ; LEUNG, Thomas: Textons, Contours and Regions: Cue Integration in Image Segmentation. (1999)
- [KK96] KITAMURA, Yoshifumi ; KISHINO, Fumio: A parallel algorithm for octree generation from polyhedral shape representation, Proceedings of 13th International Conference on Pattern Recognition (ICPR). (1996), S. 303–309
- [SSP] SHAMIK SURAL, Gang Q. ; PRAMANIK, Sakti: SEGMENTATION AND HISTOGRAM GENERATION USING THE HSV COLOR SPACE FOR IMAGE RETRIEVAL.
- [Val99] VALENS, C.: A Really Friendly Guide to Wavelets. (1999)
- [Wik] WIKIPEDIA: Wavelet. <http://de.wikipedia.org/wiki/Wavelet>