
Brandenburgische Technische Universität Cottbus
Institut für Informatik
Lehrstuhl Datenbank- und Informationssysteme

Masterarbeit

Diskriminanzanalyse und Evaluierung von Detektor-Deskriptor-Kombinationen in einem Multimedia-Retrieval-System

Discriminant analysis and evaluation of detector-descriptor combinations in a multimedia retrieval system

Dominik Müller
Informations- und Medientechnik
Matrikel-Nr.: xxxxxxx

1. Gutachter: Prof. Dr.-Ing. Ingo Schmitt
2. Gutachter: PD Dr. Douglas W. Cunningham

Datum der Ausgabe: 01.06.2011
Datum der Abgabe: 01.11.2011

Inhaltsverzeichnis

1. Einführung

Die Verwaltung und Suche auf riesigen medialen Datenmengen spielt in der heutigen Zeit eine immer größere Rolle. Neben einfachen Texten stehen dabei zunehmend auch komplexere Medien wie Bilder, Video und Audio im Vordergrund. Die Entwicklung von effizienten und vor allem schnellen Algorithmen zur Realisierung von Echtzeitsuchsystemen bildet heutzutage den großen Forschungszweig des Multimedia-Information-Retrievals (MIR, vgl. Definition 2.4), welcher trotz zahlreicher Entwicklungen in den vergangenen Jahren noch immer viel Potential für weitere Forschungsarbeit bietet. Genauer befasst sich diese Masterarbeit mit einem Teilbereich des MIR, dem Rechnersehen (Computer Vision) und insbesondere der Bildanalyse. Vordergründig werden hier lokale Features betrachtet, welche durch sogenannte Feature-Detektoren und Feature-Deskriptoren ermittelt werden. Im Folgenden soll zuerst einleitend das Potential des Feldes MIR aufgezeigt werden, um die Frage zu klären, warum es für viele Anwendungen nicht mehr wegzudenken ist. Nachdem das Themengebiet allgemein motiviert wurde, werden einige Punkte aufgegriffen, welche im Rahmen dieser Abschlussarbeit näher behandelt werden sollen. Dieses Kapitel wird durch einen Abschnitt zum Aufbau der Arbeit abgeschlossen.

1.1 Motivation

Spätestens seit den 90er Jahren nimmt die Flut an digitalen Daten stetig zu [Fen03 S. 2]. Die Ursache dafür liegt im technischen Fortschritt der letzten Jahrzehnte begründet, wodurch Digitalkameras, Camcorder, sowie DVD- und Blue-ray Disc-Player für eine breite Bevölkerungsschicht erschwinglich geworden sind. Heute hat jeder die Möglichkeit praktisch jeden Augenblick seines Lebens digital für die Nachwelt festzuhalten. Gleichzeitig wächst aber auch der Wunsch diese riesigen Datenmengen in geeigneter Weise zu speichern und zu gewünschter Zeit daraus genau die gesuchte Aufnahme zu finden. Aber nicht nur im privaten Bereich spielt die digitale Datenhaltung eine große Rolle. In noch größeren Dimensionen werden solche Suchsysteme auch im industriellen Umfeld eingesetzt. Das bekannteste Beispiel dafür sind Suchmaschinen wie Google und Yahoo, welche nahezu in Echtzeit enorme Datenmengen durchsuchen [Gal09 S. 1]. Information-Retrieval-Systeme (IR-Systeme, vgl. Definition 2.5) kommen auch in vielen innerbetrieblichen Systemen, wie firmeneigenen Intranets, und im Bereich der Forschung an zahlreichen Institutionen und Hochschulen zum Einsatz [Sto07 S. 47].

Spätestens an dieser Stelle wird deutlich, dass die Bewältigung dieses Problems von Hand nicht mehr möglich ist. Es mussten geeignete Speichersysteme für die neuen Anforderungen geschaffen werden. Klassisch werden zur Datenverwaltung Datenbanksysteme eingesetzt [Sch06 S. 1]. Diese wurden bereits in den 60er Jahren entwickelt, waren aber ursprünglich nicht zur Verwaltung der heutigen Mediendaten gedacht [Akm09]. Im Kapitel 2.1 wird erörtert, warum eine Suche auf Medienobjekten mit herkömmlichen Datenbanksystemen unzulänglich ist. Stattdessen werden zu diesem Zweck Retrieval-Systeme eingesetzt, welche über speziell an die neuen Gegebenheiten angepasste Methoden verfügen. Statt direkt nach Bildern zu suchen werden Merkmale (Features vgl. Definition 3.1) aus den Objekten generiert und bei späteren Anfragen auf diesen Werten gearbeitet. Diese Masterarbeit beantwortet unter anderem die Fragen, welche Möglichkeiten der Feature-Beschreibung existieren und inwiefern die einzelnen Features untereinander korrelieren.

1.2 Ziele und Aufbau der Arbeit

Ziel dieser Arbeit ist es Algorithmen zur Ermittlung (Detektion) und Beschreibung (Deskription) von lokalen Features in heutigen Systemen näher zu untersuchen. Dabei sind vorrangig die Fragen zu klären, welche Verfahren sinnvoll kombiniert werden können und welche Detektionsergebnisse stark korrelieren. Ziel dabei ist es möglichst viele orthogonale Features zu bestimmen. Dies ermöglicht ein optimales Suchergebnis durch viele charakterisierte Bildeigenschaften bei minimalem Rechenaufwand, was sich positiv auf die benötigte Suchzeit auswirkt. Alle hierfür getätigten Analysen werden auf dem Lehrstuhlsystem Pythia durchgeführt, welches zum Teil Algorithmen der freien C++-Bibliothek OpenCV nutzt. Diese beinhaltet neben weiteren Methoden zur Bildanalyse bereits eine Vielzahl von implementierten Detektoren und Deskriptoren, welche in dieser Arbeit vorrangig untersucht und jeweils einer Korrelationsanalyse zur Betrachtung ihrer Kombinierbarkeit unterzogen werden. Bei solchen Analysen werden verschiedene Ziele verfolgt. Zum einen steht die Ermittlung besonders schneller Deskriptoren ohne Informationsverlust im Vordergrund. Im Gegensatz dazu gilt es bei den Detektoren möglichst performante Algorithmen mit gering korrelierenden Ergebnismengen zu finden. Die Untersuchungsresultate werden anschaulich in Form verschiedener Diagramme mit Aussagen über die Kombinierbarkeit und deren Qualität in Abhängigkeit der benötigten Rechenzeit aller betrachteten Verfahren zusammengefasst. Die Features sollen zudem anhand aktueller IR-Maße evaluiert werden.

Diese Abschlussarbeit gliedert sich in sechs Kapitel. Zunächst wird im ersten Kapitel das Forschungsgebiet Information Retrieval vorgestellt und die vorliegende Arbeit darin eingeordnet. In Kapitel 2 werden grundlegende Konzepte und Modelle des Information Retrievals besprochen, welche in nachfolgenden Kapiteln Anwendung finden. Außerdem werden hier einige später verwendete Begriffe definiert. Im Mittelpunkt stehen dabei die lokalen Features, sowie aufbauend darauf die hier behandelten Detektoren und Deskriptoren, welche im dritten Kapitel untersucht werden. Es werden hier zudem die mathematischen Konzepte und Algorithmen einiger konkreter Detektions- und Deskriptionsmethoden betrachtet, um die späteren Analyseergebnisse besser einordnen zu können. Des Weiteren soll auch ein Einblick in OpenCV und dessen Struktur gegeben werden. Eine kurze Vorstellung des Lehrstuhlsystems Pythia und die Implementierung der Analyseverfahren darin werden im vierten Kapitel besprochen. Die hier vorgestellten Methoden bilden unter anderem die Grundlage für die im fünften Kapitel durchgeführten Analysen. Das Hauptaugenmerk liegt dabei in der Betrachtung von Laufzeiten, der Kombinierbarkeit und Kombinationsqualität und der Evaluierung der betrachteten Extraktionsmethoden mittels gängiger Retrieval-Kennzahlen. Zusätzlich soll der Einfluss von verschiedenen Parametern und unterschiedlichen Distanz- bzw. Ähnlichkeitsfunktionen auf das Analyseergebnis an dieser Stelle untersucht werden. Im abschließenden Kapitel 6 werden die Ergebnisse dieser Masterarbeit bewertet und ausgehend vom heutigen Stand der Technik ein Ausblick in die weitere Entwicklung gegeben.

2. Überblick Feature-Extraktion

Wie bereits im ersten Kapitel erörtert wurde, spielt das Forschungsgebiet Multimedia Retrieval heute eine bedeutende Rolle. In diesem Abschnitt der Arbeit soll nun ein allgemeiner Überblick über diesen großen Bereich erfolgen. Begonnen wird dabei mit einem kurzen historischen Abriss, der von den Anfängen des MIR bis zum heutigen Zeitpunkt einen Einblick in die Entstehungsgeschichte verschaffen soll. Viele Konzepte aus der Anfangszeit haben noch bis heute Bestand und werden zum Teil in weiterentwickelter Form in aktuellen Systemen eingesetzt. Nach diesem historischen Exkurs wird auf die grundlegenden mathematischen Konzepte der im weiteren Verlauf behandelten Detektoren und Deskriptoren näher eingegangen. Um einen Einstieg in den Forschungsbereich zu erhalten, werden an dieser Stelle auch wichtige Begriffe des Retrievals definiert. Eine besondere Rolle spielen dabei die bildbeschreibenden Merkmale, im Folgenden Features genannt. Diese werden im anschließenden dritten Kapitel umfassend untersucht.

2.1 Entstehung des Text- und Image-Retrievals

Die Geschichte der Retrieval-Systeme begann bereits in den 50er Jahren. Der Amerikaner Calvin N. Mooers verwendete erstmals 1950 den Begriff „Information Retrieval“ [Sto07 S. 38]. Von diesem Zeitpunkt an gewannen die Retrieval-Systeme, im gleichen Maße wie ihre Weiterentwicklung vorangetrieben wurde, an Bedeutung. Die erste nennenswerte deutsche Entwicklung war das Retrieval-System GOLEM, welches in den 60er Jahren von einer Forschungsgruppe bei Siemens entwickelt wurde [Sto07 S. 42]. Aber erst in den 80er und 90er des letzten Jahrhunderts erlebte das Information Retrieval zusammen mit den Internetsuchmaschinen seinen Boom [Sto07 S. 47]. In den frühen 90er wurde dann der Begriff Content based Image Retrieval (CBIR) geprägt [Kel11]. Im Gegensatz zu früheren Systemen, bei denen die Bildsuche hauptsächlich über dem Bild zugeordnete Metadaten realisiert wurde, werden im CBIR query-by-image-content-Anfragen verwendet. Diese basierten auf der Nutzung der im Bild enthaltenen Information.

Heutzutage bedienen sich viele Menschen verschiedener Suchdienstes beinahe täglich, was die Notwendigkeit einer stetigen Weiterentwicklung und Verbesserung der Systeme unterstreicht. Herkömmliche Datenbanken sind nicht für alle Anwendungen der Datenspeicherung gleichermaßen geeignet. Die Ursache dafür liegt unter anderem im konzeptionellen Aufbau einer Datenbank. Die Daten werden in recht starrer Form, oft in Tabellen, abgelegt. Diese Form der Datenhaltung hat sich in vielen Bereichen bewährt, ist allerdings auf Grund der fest vordefinierten Struktur und den konzeptionellen Besonderheit der medialen Objekte selbst ungeeignet zur Speicherung für ein Image-Retrieval-System und findet heute allenfalls noch Anwendung in Text-Retrieval-Systemen. Zu einigen Unterschieden zwischen konventionellen Datenbanksystemen und IR-Systemen wird im Abschnitt 2.2 genauer eingegangen. Da die bislang gängigen Daten-Retrieval-Systeme keine adäquate Lösung darboten, war die Notwendigkeit für ein neues System, welches mit diesen Anforderungen umgehen konnte, unverkennbar. Dies begründete die Entwicklung der Retrieval-Systeme.

Der erste naive Ansatz war eine textuelle Beschreibung des Bildinhaltes zu jedem im System gespeicherten Objekt [Fen03 S. 1], was noch recht nah an den bereits etablierten Systemen war. Das Problem des impliziten Informationsgehalts wird hier auf

eine einfache Textsuche herunter gebrochen, wobei die dargestellte Information nun explizit beschrieben ist. Das Text-Retrieval stellt den Anfang der IR-Entwicklung dar und ist heutzutage durch effiziente Suchalgorithmen und Weiterentwicklungen im Hardwarebereich in angemessener Zeit durchführbar [Gal09 S. 1]. Die Nutzung einer zusätzlichen textuellen Beschreibung bei der Bildsuche funktioniert auch auf größeren Bilddatenbanken, was gängige Internetsuchmaschinen wie Google tagtäglich unter Beweis stellen. Der Lösungsansatz ist allerdings nicht optimal. Der Konflikt hierbei besteht darin, dass das eigentliche Problem lediglich verlagert wurde. Es stellt sich jetzt die Frage, wie die Schlagworte zur Bildbeschreibung ermittelt werden. Eine Möglichkeit besteht darin, die Suchterme jedem Bild manuell zuzuordnen. Bei der Datenflut in heutigen Systemen, ein Beispiel hierfür ist das Internet, ist das keine ernstzunehmende Alternative für den praktischen Einsatz. Eine automatische Zuordnung auf Grundlage der im Bildkontext verwendeten Begriffe erscheint daher geeigneter. Google löst dieses Problem genau auf diese Weise. Für die Indexierung zur späteren Bildsuche bedienen sie sich des Bildkontextes. Anstatt zu analysieren was auf dem Bild dargestellt ist, werden bildbeschreibende Informationen, wie der Bildname, der Alt-Text und der umliegende Text des Bildes, untersucht. Aber auch hier liegen einige Schwierigkeiten bei der Umsetzung. Zum einen wird hier nicht überprüft, ob der Kontext tatsächlich zu dem dargestellten Bild passt. Des Weiteren ist es auch möglich, dass der Bildkontext fehlt, wodurch wieder eine manuelle Verschlagwortung nötig wird [Gal09 S. 1]. Ein weiteres Problem liegt bei oft zu allgemeinen Suchtermen. Auf dem Bild ist beispielsweise ein bestimmtes Auto dargestellt. Es wird hier das Schlagwort „Auto“ gewählt, nicht aber eine konkrete Beschreibung zu Marke und Typ des Automobils. Ein anderes Problem tritt auf, wenn mehrere Objekte auf einem Bild dargestellt sind. Zu jedem Objekt müsste eine detaillierte Beschreibung angefertigt werden. Dabei werden leicht bestimmte Aspekte vergessen. Alles in allem bleibt festzuhalten, dass dieser Ansatz zwar in heutigen Systemen schon recht gut funktioniert, aber keineswegs die Endstation der Forschung im Bereich Multimedial-Retrieval darstellen kann.

Vielmehr stellt es nur einen Zwischenschritt bei der Entwicklung dar. Einen anderen Ansatz verfolgt das Content based Image Retrieval. CBIR ist eine Technik, bei der Bilder anhand zuvor ermittelter Bildmerkmale (Features) gesucht werden. Das Aufspüren dieser Merkmale geschieht durch Detektoren, welche die Grundlage für die weitere Arbeit schaffen. Das Beschreiben der gefundenen Merkmale in einheitlicher Form übernehmen Feature-Deskriptoren. Somit sind die ermittelten Features bei späteren Anfragen vergleichbar. Erreicht die Technik den Stand, dass Objekterkennung unabhängig von Betrachtungswinkel und Bildqualität möglich ist, könnte dies einen Meilenstein im Bereich der medialen Suche darstellen. Suchanfragen könnten präzise und ohne Umwege beantwortet werden. Aber auch in anderen Bereichen wie Überwachungssystemen, in denen die semantische Lücke (vgl. Definition 3.2) zwischen den technischen Möglichkeiten und der semantischen Interpretation der Informationen bislang besteht, wäre dieser Fortschritt bahnbrechend.

Nach dem fundamentalen Einblick in die historische Entwicklung der Information-Retrieval-Systeme werden im folgenden Abschnitt einige Konzepte aus diesem Gebiet näher betrachtet. Genauere Informationen zu der geschichtlichen Entwicklung können beispielsweise dem Werk „Information Retrieval“ von Wolfgang G. Stock [Sto07] entnommen werden.

2.2 Konzepte und Definitionen des Information-Retrievals

In diesem Abschnitt der Arbeit werden zunächst einige Begriffe definiert, um darauf aufbauend grundlegende Prinzipien in IR-Systemen zu erörtern. Anschließend wird der prinzipielle Ablauf des IR-Prozess betrachtet, um darin den Zeitpunkt der Extraktion der Features einzuordnen. Zu guter Letzt werden heute gängige Konzepte diskutiert, welche zur Suche im Information-Retrieval eingesetzt werden.

2.2.1 Begriffsdefinitionen

Bei allen im Folgenden vorgestellten Konzepten geht es im Wesentlichen um eine multimediale Suche. Das bedeutet, dass eine Suche auf Multimedia-Objekten erfolgt. Um sich dem Begriff der multimedialen Suche schrittweise zu nähern, sollte zunächst der Begriff des Mediums eingeführt werden.

Definition 2.1 (Medium)
Ein Medium ist eine zur Kommunikation mit einem Endnutzer genutzte Methode [zur Übermittlung von Information]. Zu Medien zählen unter anderem Video, Audio und Text. [Col02 S. 142]

Bei allen Kommunikationsbeziehungen versteht man das Medium als Informationsträger, aber keinesfalls als die Information selbst [Sch06 S. 3]. Der Bedarf Informationen auszutauschen besteht schon lange. Dementsprechend ist die Geschichte des Mediums selbst schon alt und es existieren die unterschiedlichsten Formen. Im Kontext des IR-Systems spielen dabei aber nur die folgenden Medien eine Rolle.

Der Text ist eine der wichtigsten Arten Information zu übertragen. Dementsprechend nimmt das Text-Retrieval, also die Suche auf Texten, einen eigenen Bereich im Information-Retrieval ein, welcher hier allerdings nicht näher betrachtet werden soll. Ein weiteres bedeutendes Medium ist das Bild, welches bei den im weiteren Verlauf untersuchten Verfahren im Mittelpunkt steht. Im Folgenden ist dabei stets das Rasterbild gemeint, wenn von einem Bild die Rede ist. Audio- und Videoaufnahmen vervollständigen die Liste der gebräuchlichsten digitalen Medien. Diese Auflistung führt direkt zum Begriff Multimedia. Darunter versteht man eine Kombination verschiedener Medien zu einem Informationskontext.

Definition 2.2 (Multimedia)
Multimedia ist die Zusammenfassung verschiedener herkömmlicher Medien […] zu einem einheitlichen Gesamtangebot, auf das über eine einheitliche [Schnittstelle] zugegriffen werden kann. [Bra98 S. 232]

Da der Suchbereich nun definiert ist, kann als nächstes das eigentliche Suchobjekt, das Medienobjekt, klassifiziert werden. Ein Medienobjekt kann dabei Daten eines beliebigen Mediums aus dem multimedialen Bereich enthalten. Auch die Kombination mehrere Objekte eines Typs ist denkbar, was beispielsweise bei Collagen der Fall ist. Werden hingegen mehrere Medien kombiniert spricht man von einem Multimedia-Objekt. In dieser Arbeit werden lediglich Medienobjekte behandelt.

Definition 2.3 (Medienobjekt)
Ein Medienobjekt ist in Dateien gespeicherte Information in Form von Texten, Grafiken, Präsentationen, Audios, Videos, oder ausführbaren Software-Programmen. [Bla05 S. 485]

Der Suchprozess, bei dem eine Zusammenstellung zur Anfrage passenden Medienobjekte erfolgt, wird als Retrieval-Prozess bezeichnet. Der Begriff Retrieval tauchte bereits während dieser Arbeit auf und soll an dieser Stelle formal definiert werden.

Definition 2.4 (Retrieval)
Retrieval ist die Suche in einem (meist elektronischen) Datenbestand. [Bra98 S. 291]

Der Begriff Retrieval leitet sich aus dem englischen „retrieve“, also abfragen oder auffinden ab. Genauer geht es um das Wiederauffinden der abgelegten Informationen durch ein System. Ferner kann der große Bereich Retrieval in verschiedene Teilbereiche unterteilt werden. Daten-Retrieval bezeichnet dabei die Suche ohne zusätzliche semantische Interpretation [Sch06 S. 19], wie sie in herkömmlichen Datenbanksystemen eingesetzt wird. Interessant für diese Arbeit ist der Bereich des Information Retrieval. Hier muss der Informationsgehalt vor der Suche aus den Medienobjekten extrahiert werden. Einen besonderen Bereich des Information-Retrievals, welcher sich mit der Suche auf Multimedia-Objekte befasst, stellt das Multimedia-Information-Retrieval dar.

Definition 2.5 (Information Retrieval)
Das Information Retrieval [ist ein Fachgebiet der Informationswissenschaft, das sich mit] der Technik bzw. dem Prozess der Suche, Wiederauffindung und Interpretation von Information aus einem großen Datenbestand [befasst]. [McG02]

Nachdem die Begriffe nun voneinander abgegrenzt sind, sollen noch einmal wesentliche Unterschiede zwischen Datenbank- und IR-Systemen herausgearbeitet werden, um so die in Kapitel 2.1 angesprochene Daseinsberechtigung des Information-Retrievals zu rechtfertigen. Dabei wird hier nur auf gewisse Teilaspekte kurz eingegangen. Genauer können die Vorteile von Information-Retrieval- gegenüber Daten-Retrieval-Systemen bei der Speicherung von medialen Objekten beispielsweise im Buch „Information Retrieval“ von C. J. van Rijsbergen [Rij79] nachgelesen werden. Der erste wichtige Unterschied liegt bereits in den Daten selbst. Bei vielen Anwendungen liegen die Informationen explizit vor. Handelt es sich zum Beispiel um eine Personendatenbank, so sind von allen zu speichernden Personen der Name, das Geburtsdatum und die Adresse bekannt und können in tabellarischer Form gespeichert werden. Bei vielen Medienobjekten ist dies nicht der Fall. Die Information in einem Bild ist lediglich implizit dargestellt und benötigt eine semantische Interpretation durch das Retrieval-System.

Ein weiterer Unterschied ist die gewisse Unschärfe, die ein IR-System bietet. Bei einer Datenbankanfrage passt ein Datensatz entweder zur Anfrage oder nicht. Am Beispiel bedeutet das, wenn nach allen Personen gesucht wird, die in Berlin wohnen und mit Nachnamen Meier heißen, dann stehen im Resultat auch nur Personen auf die alle Kriterien zutreffen. Beim Information-Retrieval verhält es sich anders, was mit der im-

pliziten Informationsdarstellung zusammenhängt. Wird nach einem Bild mit einem Haus gesucht, kann das System nur zu einer gewissen Wahrscheinlichkeit Objekte bestimmen, die zu dieser Anfrage passen. Eine weitere Ursache dafür liegt in der Komplexität der Objekte. Es werden oftmals auf einem Bild mehrere Objekte dargestellt oder aber der Betrachtungswinkel, die Lichtverhältnisse oder andere Kriterien entsprechen nicht der Anfrage. Über eine Ähnlichkeitsabschätzung wird die Ergebnismenge der Objekte ermittelt, welche mit hoher Wahrscheinlichkeit zum Informationswunsch des Nutzers passen.

Oftmals wird die anfängliche Ergebniskollektion danach weiter überarbeitet, da durch die unscharfe Ähnlichkeitsfunktion auch nicht relevante Medienobjekte im Ergebnis liegen können. Diese iterativen Anfragen sind bei herkömmlichen Datenbanksystemen in der Regel nicht notwendig. Die Unschärfe beinhaltet aber auch Potential gegenüber dem Daten-Retrieval. Da die gefundenen Objekte nicht zu einhundert Prozent mit der Anfrage übereinstimmen müssen, ist eine gewisse Fehlertoleranz gegeben. Der Nutzer hat ursprünglich vielleicht nach einem Haus gesucht, erhält im Anfrageergebnis aber auch Bilder von anderen Bauwerken, welche vielleicht besser seinem Informationswunsch entsprechen. Er kann nun Einfluss auf die weitere Suche nehmen und findet so eher das gewünschte Ergebnis.

Der letzte hier besprochene Unterschied zu Datenbanksystemen (DBS) ist die Ergebnisdarstellung. Da bei DBS alle Datensätze gleich gut zur Anfrage passen, kann kein Ranking vorgenommen werden. Es wird lediglich die unsortierte Ergebnismenge zurückgeliefert. Bei IR-Systemen können die Objekte anhand ihres Relevanzwertes sortiert zurückgegeben werden. Die ersten Objekte in der Ergebnisliste sind dementsprechend potentiell interessanter für den Anwender. Die Unterschiede zwischen Daten-Retrieval- und Information-Retrieval-Systemen sind in der nachfolgenden Tabelle zusammengefasst [Rij79, Sch06 S. 200].

Merkmal	**Daten-Retrieval**	**Information-Retrieval**
Information	exakt	implizit
Rückschluss (Inferenz)	deduktiv	induktiv
Modell	deterministisch	probabilistisch
Anfragesprache	formal	natürlich
Fragespezifikation	vollständig	unvollständig
Anfrage	einmalig,	iterativ
Anfrageergebnis	exakt	partiell (best match)
Ergebniskollektion	Menge	Liste
Fehlertoleranz	sensitiv	insensitiv

Tabelle 2-1: Vergleich Daten-Retrieval und Information-Retrieval

2.2.2 Retrieval-Prozess

Im Folgenden soll der prinzipielle Ablauf bei einem Retrieval-Prozess betrachtet werden. Nachdem der IR-Prozess erarbeitet wurde, wird die Feature-Detektion und -Deskription in diesen Ablauf eingeordnet. Die nachfolgende Abbildung verschafft einen groben Überblick über den Vorgang und ist einer Grafik aus dem Buch „Ähnlichkeitssuche in Multimediadatenbanken“ von I. Schmitt [Sch06 S. 78] nachempfunden.

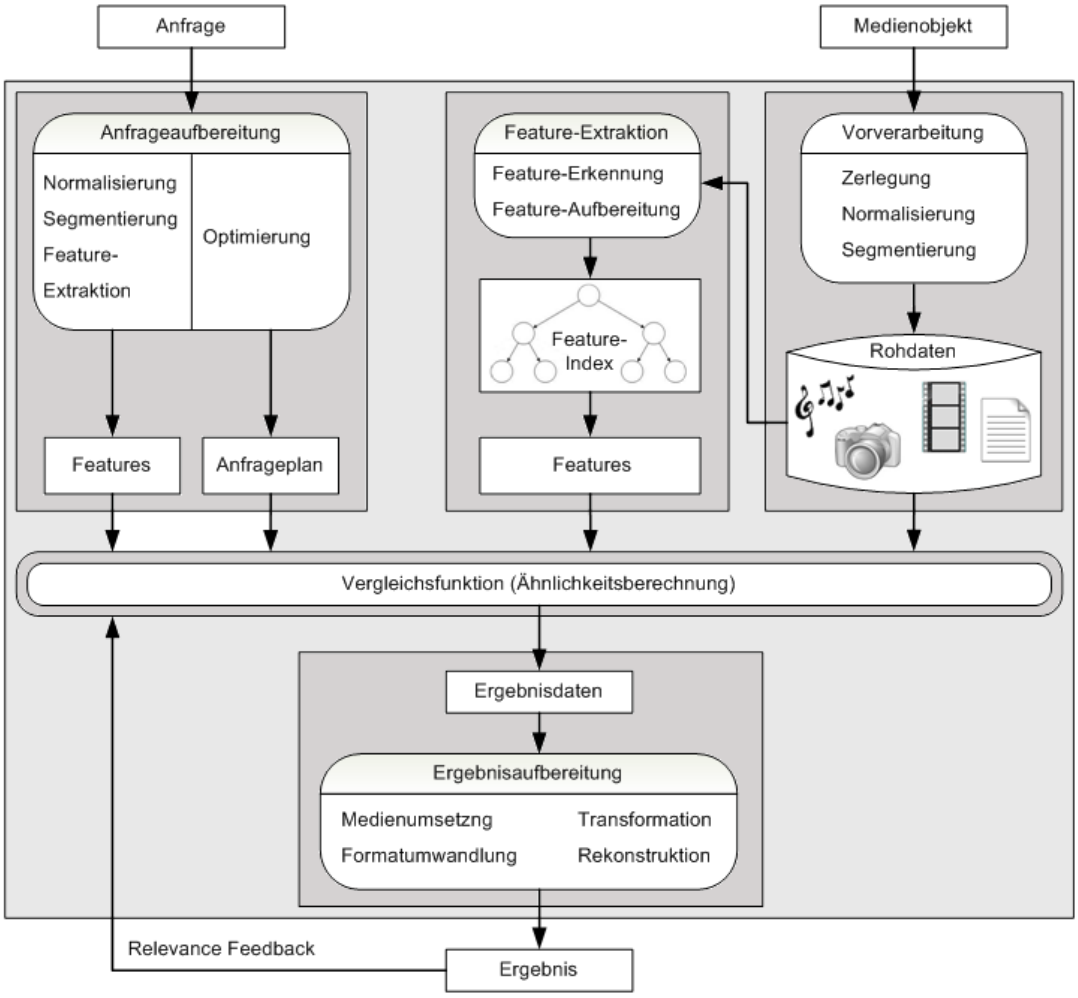

Abbildung 2-1: Der IR-Prozess

Durch den impliziten Informationsgehalt der Medienobjekte müssen sowohl von der Anfrage, als auch von den in das Retrieval-System eingefügten Objekten zunächst Features extrahiert werden. Bei multimedialen Objekten ist eine spezielle Vorverarbeitung notwendig, bevor die gewünschte Information extrahiert werden kann. Diese besteht in der Regel aus der Zerlegung komplexer Objekte, einer eventuellen Normalisierung, um später Störgrößen zu minimieren, und einer weiteren semantischen Zerlegung (Segmentierung). Die Bilddaten sind nun soweit aufbereitet, dass die erforderlichen Merkmalswerte bestimmt werden können. An dieser Stelle setzen die Feature-Detektoren und -Deskriptoren an. Der Detektor lokalisiert die Merkmale und der Deskriptor beschreibt diese in geeigneter Form. Beispielsweise können Bilder als Vektoren ihrer Eigenschaften repräsentiert und bei der Anfrage über diese Vektoren verglichen werden. Die Trennung in Deskription und Detektion hat den Vorteil, dass einzelne Teile ausgetauscht und die Extraktion so speziell an die gegebene Anwendung angepasst werden kann. Des Weiteren ist es möglich einen Index auf den extrahierten Merkmalswert anzulegen, um so eine schnellere Suchanfrage zu realisieren.

Wird eine Anfrage an die Bilddatenbank gestellt, werden bei der Anfragetransformation ähnliche Schritte durchlaufen. Parallel findet eine Anfrageoptimierung statt, damit unnötige Zeiteinbußen vermieden werden. Die Anfrageaufbereitung findet in der Regel nicht zur gleichen Zeit wie die Feature-Extraktion auf den Bildobjekten statt. Hierbei spielen Performanceaspekte eine Rolle. Ein Retrieval-System kann mitunter

riesige Mengen von Bildern und anderen Objekten beinhalten. Würden die Features erst zur Laufzeit der Anfrage berechnet werden, so müsste der Nutzer geraume Zeit auf das Ergebnis warten. Stattdessen werden diese Features bereits beim Einfügen der Bildobjekte berechnet. Zur tatsächlichen Laufzeit finden lediglich die Anfrageaufbereitung und der Vergleich von Anfrage- und Objekt-Features statt.

Die Vergleichsfunktion beruht oft auf einer Distanzberechnung, welche die Ähnlichkeit zwischen der Anfrage und den Objekten der Datenbank durch einen Distanzwert ausdrückt. Dabei kann speziellen Teilen durch Termgewichte höheren Einfluss verliehen werden. Bei vielen Anwendungen schaffen es nur die k besten Objekte, deren Ähnlichkeitswert am größten ist, in das Anfrageergebnis. Dieses kann anschließend entsprechend des Distanzwertes noch sortiert oder die Objekte weiter aufbereitet werden. Zudem kann der Nutzer durch eine Relevanzbewertung (Relevance Feedback) Einfluss auf das bisherige Anfrageergebnis nehmen. Genaueren Einblick in den Retrieval-Prozess bietet unter anderem der Artikel „Modern Information Retrieval“ von Baeza-Yates [Bae99].

Nachdem der grundlegende Prozess zur Extraktion von Features erläutert wurde, stehen im folgenden Abschnitt die Möglichkeiten der Erhebung der Features im Mittelpunkt.

2.2.3 Grundlegende Konzepte der Bilderschließung

In diesem Unterabschnitt werden einige Annotationsverfahren für Bilder vorgestellt, mit dem Ziel einen Überblick über verschiedene Ansätze und Techniken der Bilderschließung zu erhalten. Darunter befinden sich sowohl Konzepte zur Erschließung von Low-Level-Features auf niedriger technischer Ebene als auch von High-Level-Features auf höherer semantischer Ebene. Low-Level-Features werden meistens direkt aus den digitalen Bildern ermittelt und überdecken sich oft nicht mit der menschlichen Wahrnehmung eines Bildes. Die Lücke zwischen High- und Low-Level-Features wird im Kapitel 3.1 genauer beschrieben.

Definition 2.6 (*Bildannotation*)
Bildannotation ist eine visuelle Technik, die es ermöglicht, Bilder semantisch zu annotieren [um ihnen so zusätzlich Informationsmerkmale zuzuweisen, welche bei der späteren Suche verwendet werden können]. [FZI11]

Im Folgenden werden kurz diverse Verfahren vorgestellt. Die Abbildung 2-2 gibt einen ersten groben Überblick über verschiedene Ansätze.

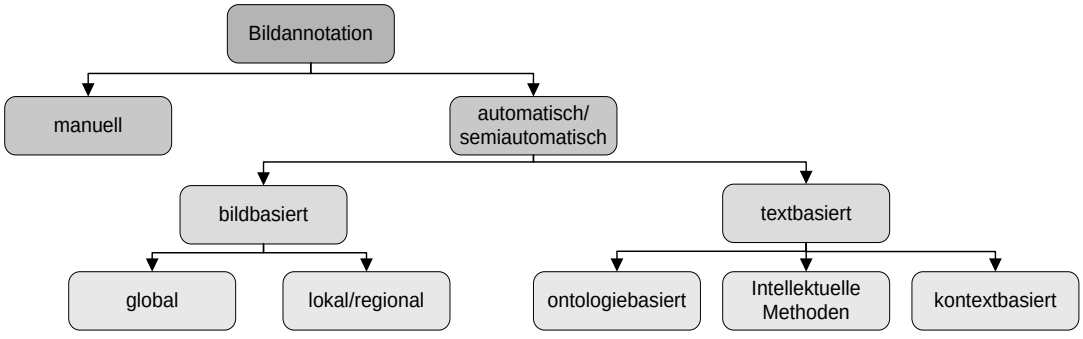

Abbildung 2-2: Konzepte der Bilderschließung [Bla08 S. 65]

Auch im digitalen Zeitalter spielt die manuelle Annotation eine wichtige Rolle. Trotz Werkzeugunterstützung ist dies immer noch ein mühevoller und zeitaufwändiger Prozess [Bla08 S. 65]. Die Entwicklung in diesem Bereich geht vor allem in Richtung Spracherkennung um so den Anwender zu entlasten. Die manuelle Annotation ist nur eine Zwischenlösung im Retrieval-Bereich, da sie für die heute täglich wachsenden Datenmengen zu zeitaufwändig ist. Im Gegensatz zur manuellen Annotation wäre eine automatische Inhaltsextraktion in der Lage mit der gewaltigen Datenflut umzugehen. Hierbei werden die semantischen Informationen direkt aus dem Bildinhalt geschlossen. Allerdings sind automatisierte Methoden auch in naher Zukunft noch nicht dazu in der Lage, zehntausende Fotographien mit akzeptabler Genauigkeit zu verwalten. Das Problem besteht dabei nicht in der Extraktion der Bildeigenschaften, sondern vielmehr in der automatische Erkennung der dargestellten Objekte und deren Kontext. Nichtsdestotrotz können durch automatische Bilderschließung bereits Informationen, wie die Tageszeit anhand einer Farbverteilungsanalyse, erschlossen werden. [Bla08 S. 66f]

Eine semiautomatische Bildbeschreibung ist schlüsselwortbasiert. Der Bildinhalt wird durch vorher festgelegte Schlagwörter beschrieben. Die spätere Suche funktioniert nur, falls die Beschreibung komplett und möglichst genau ist, was durch aufwändige manuelle Annotation oder komplexe automatische Bildbeschreibung erreicht werden kann. Bei der semiautomatischen Bildannotation werden die Informationen neben der Extraktion von low-level Merkmalen auch aus der Analyse des Bildkontextes gewonnen. Hierbei werden mit dem Bild verknüpfte Texte und erhobene Metadaten untersucht (textbasierte Methoden). Zusätzlich wird durch Interaktion mit dem Nutzer die Richtigkeit der vorgeschlagenen Schlagwörter überprüft (Relevance Feedback). Die Anwendung ist zurzeit allerdings meist auf ein kleines Themengebiet spezialisiert.

Auch die Methoden zur Annotation durch Analyse von Bildeigenschaften lassen sich in zwei Untertypen aufteilen. Verfahren dieser Klasse untersuchen entweder lokale oder aber globale Bildeigenschaften. Bei lokalen Anwendungen werden regionale Eigenschaften meist durch Segmentierung oder Partitionierung in semantisch bedeutende Bildteile ermittelt. Anschließend wird die Wahrscheinlichkeit von Kookkurrenzen zwischen den ermittelten Bildteilen und definierten Schlagwörtern oder zu einem Anfragebild berechnet. Methoden zur Ermittlung globaler Bildeigenschaften verzichten dagegen auf Techniken zur Zerlegung in einzelne Objekte. Die Analysen erfolgen auf dem gesamten Bild, wodurch detaillierte Informationen verloren gehen können. Da die Semantik eines Bildes im Allgemeinen durch semantisch bedeutsame Bildregionen repräsentiert wird, sind regionale Verfahren im Allgemeinen vielversprechender. Die Analyse lokaler Feature wird im Kapitel 3.3 näher betrachtet. [Bla08 S. 67f]

3. Features und deren Extraktionsmethoden

Das Konzept der Features bildet ein wichtiges Fundament des heutigen Forschungsgebietes Computer Vision. Es existiert eine Vielzahl unterschiedlicher Features, welche durch verschiedene Arten von Detektoren und Deskriptoren ermittelt und beschrieben werden können. Dies ermöglicht es, ein Bild auf vielfältige Art und Weise zu charakterisieren, um so möglichst viele Informationen über die Abbildung zu erhalten. Dieses Kapitel befasst sich mit dem Konzept des Features. Dabei wird zuerst formal definiert was ein Feature ist, bevor untersucht wird was „gute" Merkmalswerte auszeichnet. Weiterhin wird eine Unterscheidung zwischen globalen und lokalen Features getroffen. Die lokalen Features werden im Mittelpunkt dieser Arbeit stehen. Im Anschluss daran werden diese anhand ihrer Merkmale in Klassen eingeteilt, was den Grundstein für die weitere Betrachtung legt. Die Charakterisierung der Features führt direkt zu den Feature-Detektoren und Feature-Deskriptoren. Dieses Kapitel klärt die Frage zu welchem Zweck diese Extraktionsmethoden eingesetzt werden und geht dabei auf einige Aspekte bei der Extraktion und Transformation der besprochenen Merkmalswerte ein. Abschließend erfolgt in diesem Kapitel eine Vorstellung der C++-Bibliothek OpenCV, welche durch ihre Konzepte zur Feature-Detektion beziehungsweise -Deskription bedeutend für diese Arbeit ist.

3.1 Features im CBIR

Der Grundgedanke von CBIR ist es, im Gegensatz zu den ersten Generationen des Image-Retrievals, die Information nicht aus einer zugehörigen, textuellen Beschreibung zu beziehen, sondern diese aus dem Bildinhalt zu extrahieren. Um sich dem Begriff Feature zu nähern werden zuerst grundlegende Kriterien besprochen, welche für alle heute gängigen Features im Content Based Image Retrieval gelten. Daran anschließend werden Features in feingranularere Einheiten unterteilt.

Da in Kapitel 2.2 bereits erläutert wurde, warum es unerlässlich ist auf Features anstatt direkt auf den Bildern zu arbeiten, wird nun noch erläutert welche Eigenschaften ein Feature auszeichnen. Der Begriff Feature kann formal wie folgt definiert werden.

Definition 3.1 (Feature)
Ein Feature ist ein inhaltstragender Eigenschaftswert von Medienobjekten, der im Retrieval eingesetzt wird. Features müssen effizient berechenbar sein und ermöglichen eine signifikante Unterscheidung verschiedener Objekte [Sch06 S. 79].

Besonders wichtig beim Umgang mit Features ist die Frage, was einen „guten" Eigenschaftswert auszeichnet. An erster Stelle steht dabei, dass er ein Bild adäquat und dabei signifikant zu anderen Bildern beschreibt. Was zeichnet eine gute Beschreibung des Bildinhaltes aus? Eine Möglichkeit dies zu beantworten ist die Betrachtung von Invarianzen. Eine geeignete Feature-Repräsentation ist gegeben, wenn der Merkmalswert unabhängig (invariant) gegenüber Störungen und bestimmten Operationen ist [Sch06 S. 97]. Unter Störungen im Bereich der Bildanalyse versteht man beispielsweise veränderte Lichtverhältnisse, einen anderen Betrachtungswickel oder Verdeckung ei-

niger Bildobjekte. Des Weiteren werden im Allgemeinen drei wesentliche Operationen unterschieden, zu welchen ein Eigenschaftswert möglichst invariant sein soll. Rotationsinvarianz sagt aus, dass die Drehung eines Objektes keinen Einfluss auf das Suchergebnis haben darf. Dies ist zum Beispiel beim Betrachten der Farbverteilung eines Bildes gegeben. Weiterhin sollte die Translationsinvarianz gewährleistet werden. Dies bedeutet, dass die Suche nicht beeinflusst wird, wenn das Bildzentrum verschoben ist. Um dies zu gewährleisten wird oftmals vor der Feature-Berechnung das Bildmassezentrum, also der Bereich im Bild mit den dominanten Bildobjekten, in die Bildmitte verschoben [Som04 S. 12]. Als letztes spielt die Skalierungsinvarianz noch eine große Rolle bei der Feature-Ermittlung. Ein Feature ist skalierungsinvariant, wenn es unabhängig gegenüber Größenänderung des Rasterbildes ist. Dies ist unter anderem bei den meisten auf Farbinformation basierten Features der Fall, da sich die Farbverteilung durch Skalierung des Bildes nicht ändert.

Nachdem die allgemeinen Anforderungen an Features nun definiert sind, widmet sich der nachfolgende Teil dieser Arbeit der Verwendung von Features im CBIR um dabei auf spezielle Probleme einzugehen. Bereits aus der Abbildung 2-1 ist ersichtlich, dass sich der Prozess der Feature-Extraktion in mehrere Schritte unterteilt. Die Gewinnung der Feature ist in den Unterkapiteln 2.4 und 2.5 näher beschrieben. Nun soll vorrangig die Interpretation der bereits gefundenen Merkmalswerte im Mittelpunkt stehen. Ziel dabei ist es eine automatische semantische Interpretation zu erreichen. Könnte das Retrieval-System bei jedem Bild automatisch eine korrekte Aussage darüber treffen, was auf der entsprechenden Abbildung dargestellt ist, wäre das Problem der inhaltsbasierten Bildsuche praktisch gelöst. Beim heutigen Stand der Technik funktioniert eine automatische Interpretation aber nur bedingt [Sch06 S. 93]. Das System kann aufgrund der Farbverteilung einen Sonnenuntergang von einer grünen Wiese und über die Textur eine Ziegelwand von einem Sandstrand unterscheiden. Hierfür werden allerdings meistens zusätzlich Informationen über die Bilddatenbank benötigt. Im allgemeinen Fall ist mitunter nicht einmal eine semiautomatische Interpretation möglich. Zum Beispiel ist es technisch heute meist noch nicht machbar auf einer Datenbank mit Tierbildern einen Hund von einer Katze zu entscheiden.

Das hier aufgetretene Problem in der automatischen Bilderschließung wird häufig als semantische Lücke (semantic gap) bezeichnet. Features mangelt es an der nötigen Ausdruckskraft, um Bilder in einer für den Menschen adäquaten Art und Weise zu beschreiben.

Definition 3.2 (semantische Lücke)

Die semantische Lücke beschreibt den Abstand zwischen der Information, die aus visuellen Daten extrahiert werden kann, und der Interpretation, welche die gleichen Daten bei einem Betrachter hervorrufen [Sme00 S. 1353].

Die Repräsentationen, der aus den Rohbilddaten berechenbaren Bildeigenschaften, können nicht oder nicht hinreichend genau in Repräsentationsformen höherer Ebene, welche die Bedeutung von Bildelementen charakterisieren würden, transformiert werden [Har06 S. 1]. Die folgende Grafik zeigt wie sich die semantische Lücke auf den Suchprozess auswirkt.

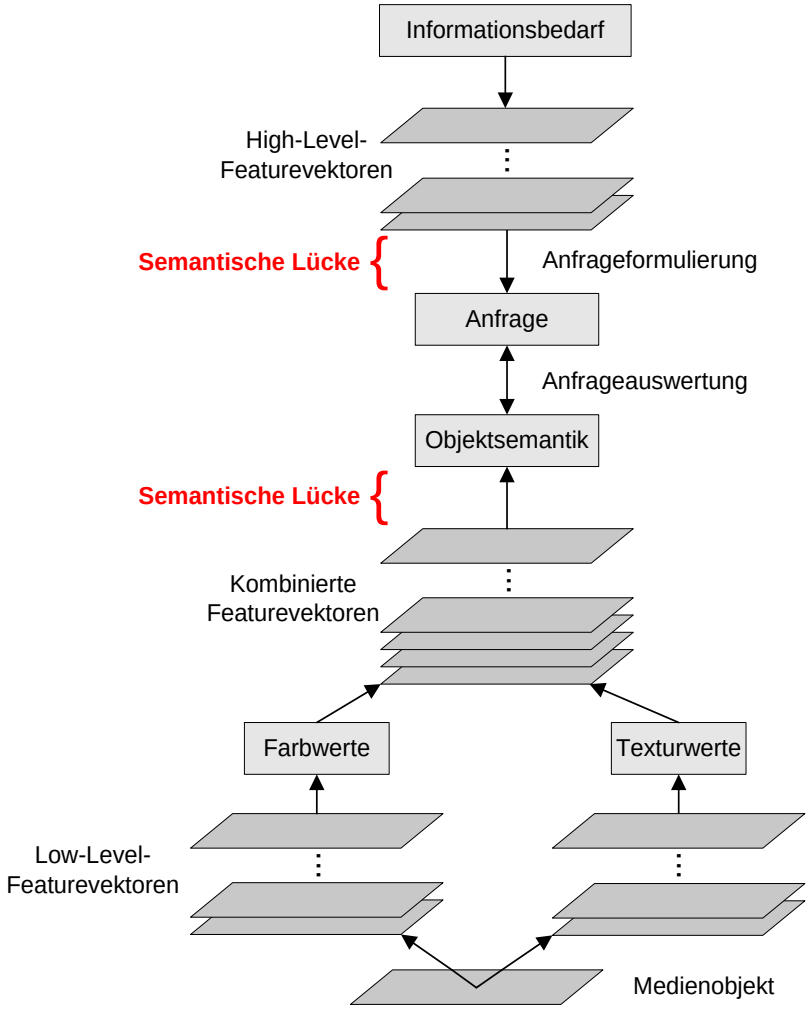

Abbildung 3-1: Einordnung der semantischen Lücke in den IR-Prozess

Wie in der Abbildung 3-1 zu erkennen ist erfolgt sowohl bei der Anfragengenerierung, als auch von jedem gespeicherten Medienobjekt eine Featurewertbildung. In beiden Fällen sollen vergleichbare Merkmalswerte extrahiert werden. Am deutlichsten zeigt sich das Problem der semantischen Lücke bei verbal formulierten Anfragen (semantic retrieval). Hierbei geschieht die Beschreibung des Informationswunsches in wörtliche Form (High-Level Features). Die High-Level Features, welche oft durch Schlüsselwörter umgesetzt werden, arbeiten auf einer viel höheren Ebene als die aus Bildern extrahierten Features (Low-Level Features). Ein Bild von einem Berg besteht auf technischer Ebene aus einer Zusammensetzung verschiedener Farben, Linien und Texturen [Bla08 S. 17]. Die Problematik zeigt sich im Vergleich der Merkmalswerte. Ein Lösungsansatz sieht die Anpassung der Anfrage an die Problemstellung vor. Bei query-by-example-Anfragen (QBE) wird anstatt der natürlichen Anfrageformulierung ein Beispielbild genutzt, welches mit den Inhalten der Datenbank verglichen wird. In die gleiche Kategorie fallen auch query-by-sketch-Anfragen (QBS), bei denen das gewünschte Objekt skizziert und die Bilddatenbank anhand der erstellten Skizze durchsucht wird. Bei beiden Verfahren werden die Features von Anfrage und Bilddatenbank auf die gleiche Weise berechnet und können im Nachhinein direkt verglichen werden. Die Qualität des Anfrageergebnisses hängt dabei im Allgemeinen stark von der Struktur des Datenbestandes ab. Befinden sich sehr vielfältige Bilder in der Datenbank, können als Anfrageergebnis zu einem Foto, das eine grüne Wiese zeigt, auch Bilder eines Waldes zu-

rückgegeben werden. Die Ursache liegt in der bereits beschriebenen fehlenden semantischen Interpretation der Bildobjekte. Je nach Anfrageart verschiebt sich die semantische Lücke weiter in eine bestimmte Richtung, aber das Hauptproblem, der Vergleich zwischen den bestimmten Low-Level Features der Objekte und den gesuchten High-Level-Features der Anfrage, bleibt bestehen. Die Überwindung der semantischen Lücke brächte einige Vorzüge. Auf einer höheren semantischen Ebene können Schlagwörter zur Bildbeschreibung genutzt werden. Die natürliche Charakterisierung einer Abbildung ist deutlich besser an die menschliche Denkweise angepasst. Die gewünschte Suchanfrage kann durch simple Schlüsselwörter formuliert werden anstatt wenig verständliche technische Eigenschaftsvektoren von Bildern oder Bildsegmenten aufzustellen [Bla08 S. 16].

In diesem Abschnitt der Arbeit wurden viele Probleme angesprochen, auf die bei der Berechnung von Features geachtet werden muss. Im folgenden Teil der Arbeit wird die Klassifikation von Features beschrieben. Eine grundlegende Einteilung ist die Unterscheidung der Gruppen der globalen und lokalen Bildeigenschaften, welche nachfolgend betrachtet werden.

3.1.1 Globale Features

Globale Features beschreiben immer das gesamte Bild. Daraus ergibt sich, dass im Vorfeld keine weitere Bildsegmentierung notwendig ist, da die Merkmalswerte über alle Bildpunkte ermittelt werden [Sch06 S.92].

Definition 3.3 (Globale Features)
Globale Features sind eine kompakte Beschreibung eines gesamten Bildes [Gal09 S. 3].

In die Gruppe der globalen Features fallen zum Beispiel Histogramme und Momente, welche meistens die Farbeigenschaften des Bildes beschreiben. Wie bereits erläutert wurde, kann über die Farbe keine genaue Bildklassifikation vorgenommen werden. Bilder zweier vollkommen verschiedener Objekte können die gleiche Farbverteilung aufweisen. Eine Objekterkennung ist rein über die Farbe nicht möglich. Die Semantik eines Bildes wird im Allgemeinen auch durch semantisch bedeutsame Bildregionen repräsentiert [Bla08 S. 67f]. Das bedeutet, dass das Bestimmen einer globalen Bildeigenschaft ohne vorher bedeutsame Regionen zu ermitteln sich negativ auf die Ergebnisqualität auswirken muss. Retrieval-Systeme, die nur auf globalen Features basieren sind daher nicht so leistungsfähig.

Das Potential globaler Features liegt an anderer Stelle. Je nach Anwendung stellen sie nützliche Filter dar, die vor der zeitintensiven lokalen Suche die Ergebnismenge stark einschränken können. Bei den Detektionsverfahren zur Ermittlung der Objektkonturen selbst lässt sich keine weitere Zeit einsparen, da lokale Features unabhängig von einer Anfrage von allen Bildern der Datenbank erhoben werden. Dies geschieht allerdings nicht zur Laufzeit der Anfrage, wodurch die Anfrageperformance unverändert bleibt. Aber auch der Vergleich auf lokalen Features ist teuer. Verglichen mit einer Distanzberechnung auf einem globalen Farbwert ist die Bestimmung der Ähnlichkeit über die viel komplexeren lokalen Bildeigenschaften sehr viel rechenintensiver und benötigt dementsprechend bedeutend mehr Zeit. Wenn die globalen Werte bereits lange vor der eigentlichen Suche für jedes Bild im System berechnet und abgespeichert

wurden, müssen zur Laufzeit lediglich die globalen Features zwischen Anfrage und Objektmenge verglichen werden. Alle Objekte, bei denen dieser Wert zu stark vom Anfragewert abweicht oder anders ausgedrückt bei denen die Ähnlichkeit zur Anfrage unterhalb einer Schwelle liegt, können direkt ausgeschlossen werden. Die längere und kompliziertere Distanzberechnung der lokalen Features muss nun auf einer deutlich kleineren Menge erfolgen.

Hierbei besteht natürlich auch die Gefahr korrekte Kandidaten von vornherein zu eliminieren oder Distanzwerte, welche für nachfolgende Anfragen wichtig wären, nicht zu berechnen. Bilder eines roten Autos werden beispielsweise nicht betrachtet, wenn das Anfragebild ein grünes Auto zeigt, können bei einer allgemeinen Suche nach Autos aber trotzdem relevant für das Ergebnis sein. Der Einsatz globaler Verfahren und das Setzen von Schwellwerten sind deshalb speziell für jede Anwendung getrennt zu überdenken.

3.1.2 Lokale Features

Im Unterschied zu globalen Features beziehen sich lokale Features stets nur auf eine Bildregion, welche sich signifikant von ihrer unmittelbaren Umgebung unterscheidet. Die Ermittlung dieser interessanten Bildteile erfolgt häufig durch Segmentierung oder Partitionierung des Gesamtbildes in Teilregionen, sowie weiterer Vorverarbeitungsverfahren bei denen wichtige Keypoints im Bild bestimmt werden. [Sch06 S.92]

Definition 3.4 (Lokale Features)
Lokale Features sind eine Beschreibung einer Bildregion, die sich von ihrer unmittelbaren Umgebung unterscheidet [Gal09 S. 3].

Zu den wichtigen Bildregionen lokaler Features zählen unter anderem Punkte, Kanten, Ecken (Schnittpunkte von Kanten) und Blobs (kleine Regionen) [Gal09 S. 3f]. Diese werden durch lokale Unterschiede in der Helligkeit oder Farbe ermittelt, beschreiben typischerweise allerdings die Kontur der dargestellten Bildobjekte. Zu dieser und weiteren lokalen Feature-Gruppen wird im darauffolgenden Unterkapitel Bezug genommen. Da lokale Features nicht das gesamte Bild, sondern nur einzelne Bildteile beschreiben sind sie vielseitiger als globale Features. So können in einem Bild beispielsweise mehrere Objekte unterschieden werden. Auch eine Erkennung von Mustern oder Strukturen im Bild, wie den Verlauf von Straßen auf Luftaufnahmen, ist nur mit lokalen Bildeigenschaften möglich [Gal09 S. 4]. Auch für diese Ausarbeitung ist die Verwendung lokaler Feature aufgrund ihrer zahlreichen Vorteile von größerer Bedeutung. Nachfolgende Teile der Arbeit beziehen sich deshalb meist auf lokale Features.

Zusammenfassend kann festgehalten werden, dass lokale Features durch ihr breiteres Anwendungsfeld und die Tatsache, dass die wesentliche Bildinformation in einzelnen Bildregionen enthalten ist, die größere Bedeutung für das CBIR besitzen [Gal09 S. 3]. Die Einteilung in global und lokal ist wichtig und ein erster Schritt in Richtung einer Feature-Klassifikation. Allerdings existieren noch weitere Möglichkeiten Merkmalswerte genauer zu spezifizieren. Im Folgenden wird ein Ansatz vorgestellt, bei dem verschiedene Gruppen von Features entsprechend ihrer bildbeschreibenden Eigenschaften unterschieden werden.

3.2 Feature-Klassifikation nach dem Bildinhalt

Bei der inhaltsbasierten Bildsuche (content based image retrieval) sollen möglichst viele verschiedene Bildmerkmale gefunden werden. Um Features näher zu klassifizieren, muss zunächst analysiert werden, welche Bildinhalte unterschieden werden können. Die Untersuchung welchen Teil des Bildinhaltes ein Feature beschreibt, ergibt eine mögliche Einteilung der Features in Klassen. Generell unterteilt man die dargestellten Bildinformationen in den visuellen und den semantischen Inhalt [Fen03]. Der visuelle Bildinhalt kann dabei entweder allgemein oder aber anwendungsspezifisch sein. In die Gruppe der allgemeinen visuellen Inhalte fallen die Farbe, Form, Umrisse und das Arrangement von Objekten. Die spezifischen Inhalte sind von der jeweiligen Anwendung abhängig und es ist meistens ein Hintergrundwissen über das System notwendig. Beispiele hierfür sind bestimmte Charakteristika in der Gesichts- oder Objekterkennung und der Analyse von biometrischen Daten. Der semantische Inhalt setzt sich aus Informationen, die aus dem visuellen Inhalts abgeleitet wurden, und einer eventuell vorhandenen zusätzlichen textuellen Beschreibung zusammen. Das bedeutet, die semantische Information, das Foto zeigt ein rotes Auto, kann aus einer textuellen Informationen wie dem Bildtitel oder einer visuellen Analyse des Farbspektrums und der Objektkonturen geschlussfolgert worden sein. Die beschriebene Struktur ist im nachfolgenden Schema dargestellt.

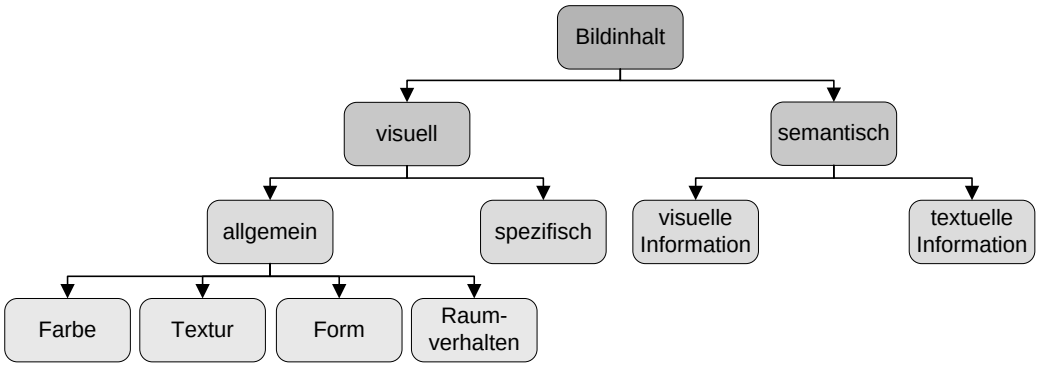

Abbildung 3-2: Klassifikation der Features nach dem Bildinhalt

Besonders interessant im Kontext dieser Arbeit sind dabei die allgemeinen visuellen Features. Im Folgenden werden einige Grundlagen der Nutzung dieser Bildeigenschaften als bildbeschreibende Features gegeben. Die Farbe ist der in Retrieval-Systemen meist verwendete Inhalt [Fen03 S. 4]. Dabei kann jede beliebige Farbe durch einen eindeutig definierten dreidimensionalen Punkt im Farbraum beschrieben werden. Hierfür ist es notwendig im Vorfeld zu definieren welches Farbmodell für die Anwendung genutzt wird. Es existiert eine Vielzahl von Modellen, welche jeweils ihre Vorzüge haben, aber nicht alle sind optimal für ein Retrieval-System geeignet. Eine der wichtigsten Eigenschaften im IR-Umfeld ist die Uniformität der Farbmodelle. Uniform bedeutet, dass die berechenbare Distanz zweier Farben im Modell der subjektiven Wahrnehmung der meisten Menschen zur Ähnlichkeit bzw. Unähnlichkeit dieser Farben entspricht. Die beiden bekanntesten Farbräume sind RGB und CMY. Das RGB-Verfahren beruht auf additiver Farbzusammensetzung, wohingegen CMY einer subtraktiven Farbmischung entspricht. Vor allem in der Computergrafik und der Fernsehtechnik wird das RGB-Verfahren verwendet. CMY spielt dagegen bei allen Printmedien eine sehr große Rolle. Für ein Retrieval-System sind beide Farbräume weniger geeignet, da sie geräteabhängig, also nicht ohne weiteres vergleichbar, und nicht uniform sind. Die

Modelle CIE L*a*b* und CIE L*u*v* sind dagegen geräteunabhängig und wahrnehmungstreu. Beim CIE L*a*b* wird eine subtraktive Farbmischung und beim CIE L*u*v* eine additive Farbmischung angewandt. Weiterhin existiert noch der HSV-Farbraum, bei dem eine Farbe durch ihren Farbton, einem Sättigungswert und dem Helligkeitswert beschrieben wird. Näheres zu allen hier aufgeführten Modellen kann auf den Internetauftritt von Adobe[1] nachgelesen werden. In dieser Arbeit steht eher die Bedeutung der einzelnen Modelle für IR-Systeme im Mittelpunkt. D. Feng, W.C. Sui und H. Zhang stellen in ihrem Buch „Multimedia Information Retrieval and Management" fest, dass CIE L*a*b* und CIE L*u*v* bei Farbmomenten bessere Ergebnisse erzielen, wohingegen bei Farbhistogrammen und Farbvektorverfahren der HSV-Raum eher geeignet ist [Fen03 S. 5f]. Im HSV-Modell steckt die Hauptfarbinformation im Hue-Wert, welcher bei der Quantisierung der Histogramm-Bins entsprechend stärker gewichtet werden kann.

Der nächste hier betrachtete Bildinhalt ist die Textur. Diese lässt sich grundlegend in zwei Kategorien unterteilen [Fen03 S. 7]. Die erste große Gruppe sind die strukturellen oder syntaktischen Texturen. Dabei werden strukturelle Grundformen und wiederkehrende Muster in Bildern erkannt. Besonders effizient arbeitet diese Klasse bei sehr einheitlichen Texturen. Vertreter dieser Kategorie sind zum Beispiel morphologische Operatoren und Adjazenzgraphen, bei denen Strukturen durch betrachten der Nachbarschaft jedes Pixels gefunden werden. Des Weiteren werden Texturen in die Gruppe der statistischen Texturen klassifiziert. Diese werden über die Betrachtung der statischen Verteilung der Bildintensität gefunden. Grob lässt sich diese Texturform wiederum in nicht-transformierende, spektrale und signalverarbeitende Texturen unterteilen. Zu einigen Vertretern dieser Klassen wird später Bezug genommen.

Neben den bereits erwähnten allgemeinen Bildinhalten spielt auch die Form im CBIR eine große Rolle. Anders als Farbe und Textur findet die Beschreibung der im Bild vorhandenen Formen meist erst nach der Segmentierung in Regionen oder Objekte statt [Fen03 S. 12]. Die Qualität der Anwendung unter Ausnutzung der Formen hängt daher stark von der Robustheit und Genauigkeit der Bildsegmentierung ab. Die durch Formdeskription erstellten Features werden in Umriss-basierte und Regionen-basierte Features klassifiziert. Beispielverfahren, die in diese Gruppe fallen, werden ebenfalls im Folgenden diskutiert.

Die räumliche Anordnung von Objekten ist der letzte hier genannte visuelle Bildinhalt. Durch räumliche Beschreibungen können Objekte mit gleicher Farbe und Textur unterschieden werden [Fen03 S. 15]. Ein einfaches Beispiel soll diese Problematik verdeutlichen. Angenommen auf einer Photographie sind unter anderen ein See und ein wolkenloser Himmel dargestellt. Sowohl der See als auch der Himmel besitzen oft nahezu identische Farbhistogramme und sind an einem wolkenlosen Tag auch über die Textur nicht zu differenzieren. Allerdings unterscheidet sich die räumliche Anordnung der beiden Bildobjekte, wodurch mehr Informationen über die zwei „blauen Flächen" gewonnen werden kann. Neben den betrachteten Bildinhalten kann auch die räumliche Position und ortsbezogene Beziehungen zwischen Objekten bei der Suche genutzt werden. Bevor die Relationen zwischen verschiedenen Bildregionen definiert werden können, muss im Vorfeld eine Segmentierung erfolgen. Laut Santini können alle möglichen

[1] http://www.teialehrbuch.de/Kostenlose-Kurse/Adobe-Photoshop/8530-Farben-und-ihre-Darstellung.html

räumlichen Beziehungen dabei in eine der vier folgenden Kategorien eingeordnet werden [San01 S. 251].

Art der Relation	Beispiel
Absolute	Objekt A befindet sich an der Position (x, y)
Distanzbasiert	Objekt A ist in der Nähe von Objekt B, aber weit entfernt von Objekt C
Direktional	Objekt A ist links von Objekt B
Topologisch	Objekt A verdeckt Objekt B teilweise

Tabelle 3-1: Kategorien räumlicher Objektbeziehungen

Für eine optimale Repräsentation der räumlichen Verhältnisse müsste eine genaue Beschreibung aller Relationen der Bildobjekte erfolgen. Dies ist manuell oftmals zu aufwändig, weshalb meistens eine Approximation an die Gesamtmenge erfolgt [San01 S. 246f]. Derzeit existieren keine praktischen Verfahren, die die räumliche Bildanordnung automatisch beschreiben. Aus diesem Grund wird in der Arbeit auch nicht weiter auf den bildraumbasierten Ansatz eingegangen.

Die nachfolgende Grafik zeigt einige Vertreter der vier spezifizierten Feature-Gruppen, erhebt dabei aber keinen Anspruch auf Vollständigkeit. Die bereits im Lehrstuhlsystem Pythia implementierten Features wurden in der Darstellung hervorgehoben. Einige bildbeschreibende Merkmale werden im Folgenden näher betrachtet. Das anschließende Kapitel 3.4 befasst sich danach mit der Frage, wie diese Features aus einem Bild extrahiert werden können.

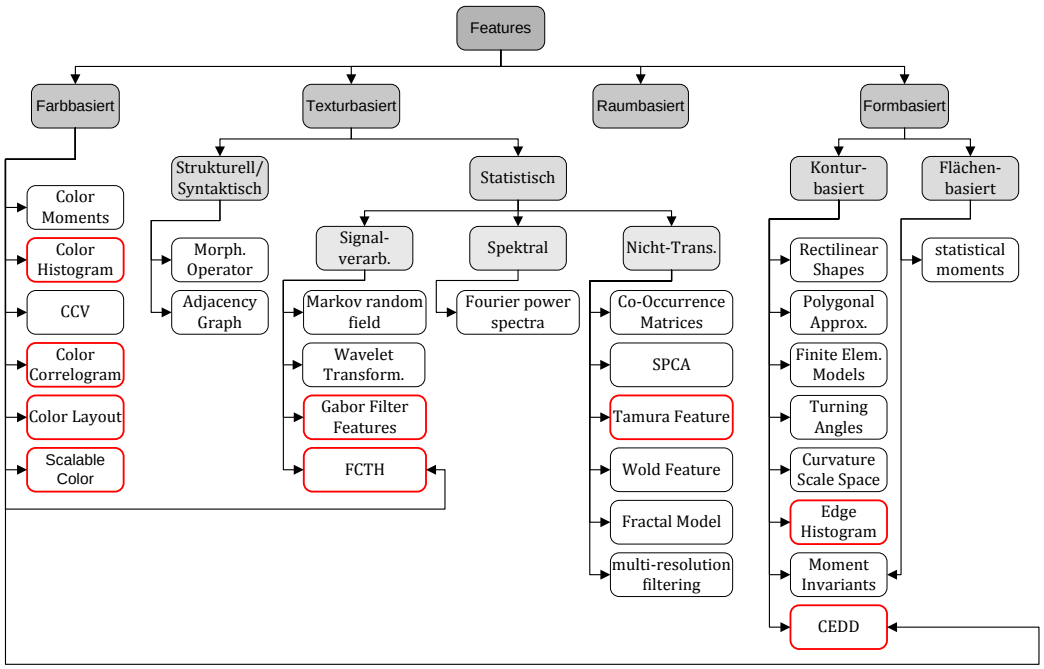

Abbildung 3-3: Einordnung Features in definierte Kategorien

3.2.1 Farbbasierte Features

Farbmomente

Farbmomente werden erfolgreich in vielen Retrieval-Systemen wie QBIC, eine Search-Engine von IBM, eingesetzt. Farbmomente werden gebräuchlicher Weise im

Farbmodell L*u*v* oder L*a*b* berechnet, da diese Modelle bessere Ergebnisse als ein Einsatz im HSV-Farbraum zeigen. Die ersten drei Momente werden dabei wie folgt definiert. [Fen03 S. 5]

Moment erster Ordnung:
Mittelwert (mean)

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} f_{ij} \qquad (3\text{-}1)$$

Moment zweiter Ordnung:
Varianz (varianz)

$$\sigma_i = \left(\frac{1}{N}\sum_{j=1}^{N}\left(f_{ij} - \mu_i\right)^2\right)^{\frac{1}{2}} \qquad (3\text{-}2)$$

Moment dritter Ordnung:
Verzerrung (skewness)

$$s_i = \left(\frac{1}{N}\sum_{j=1}^{N}\left(f_{ij} - \mu_i\right)^3\right)^{\frac{1}{3}} \qquad (3\text{-}3)$$

In den gegebenen Berechnungen stellt f_{ij} den Wert der i-ten Farbkomponente des Pixels *j* und *N* die Anzahl der Pixel im Bild dar. Die Verwendung von Momenten ist eine sehr kompakte Darstellung der Farbinformation im Bild. Lediglich neun Elemente, also jeweils drei Farbmomente für jede der drei Farbkomponenten, dienen der Repräsentation eines Bildes. Durch diese Reduktion gehen aber auch viele Informationen verloren. Beschränkt man sich bei der Suche lediglich auf die Auswahl durch Farbmomente, so sind die Ergebnisse im Allgemeinen sehr ungenau. Momente werden deshalb meist nur als Filter eingesetzt, welche einer Erstauswahl von potentiellen Vergleichskandidaten dienen.

Farbhistogramme

Eine weitere effektive Repräsentation der Farbinhalte eines Bildes sind Farbhistogramme. Dabei wird der Feature-Raum *S* in *M* Regionen S^m mit $m = 1,\ldots,M$ unterteilt. Die empirische Wahrscheinlichkeit, dass ein Datenwert in eine dieser Regionen fällt liegt bei $P(x \epsilon S^m) = K^m/N$, wobei K^m die Anzahl der Datenwerte in dieser Region ist und *N* die Anzahl aller Datenwerte.

Abbildung 3-4: Originalbild (links) [Nob11] mit globalen Histogramm der Farbverteilung (rechts)

Besonders geeignet zur Beschreibung sind Histogramme, wenn die Farbverteilung eindeutig und einzigartig im Vergleich zum übrigen Datenbestand ist. Farbhistogramme sind einfach zu berechnen und können sowohl bei lokalen als auch bei globalen Features eingesetzt werden. Sie sind robust gegen Translation und Rotation und verändern sich auch bei Skalierung nur allmählich. [Des03 S. 18]

Das prinzipielle Vorgehen bei der Erstellung von Histogrammen sieht die folgenden vier Arbeitsschritte vor.

- Auswahl des Farbraums
- Quantisierung der drei Komponenten des Farbraums in Bins
- Ermitteln der absoluten Farbverteilung und Füllen der Bins
- Ermitteln der relativen Farbverteilung

Im Gegensatz zu den Momenten zeigt der HSV-Farbraum bei Histogrammen bessere Ergebnisse als CIE L*a*b* und CIE L*u*v* [Fen03 S. 6]. Im zweiten Schritt, der Quantisierung, wird zu jedem Farbwert der entsprechende Farbeimer (Bin), dem es angehört, berechnet. Hierbei findet oft eine Dimensionsreduktion der Auflösung der entsprechenden Bins statt, um nicht für jede Farbkombination der Komponenten ein Bin schaffen zu müssen. Am Beispiel des HSV-Farbraums hieße das, es müssten rund 3,6 Millionen Bins gespeichert werden. Das ist nicht nur speicher- und rechenintensiv, sondern auch negativ bei der Distanzberechnung und späteren Matching der Ergebnisse, da für eine solche große Anzahl von zu vergleichenden Werten keine effizienten Indexverfahren mehr existieren [Fen03 S. 5]. Darum existiert eine Reihe von Verfahren zur Reduktion der Bin-Anzahl, die vom Herabsetzen der Bildhelligkeit bis zum Bestimmen der *K* besten diskriminierenden Farben eines Farbraums reicht. Ähnlich wie Texte durch wenige Wörtern charakterisiert werden, so machen die k Farben mit den meisten Pixeln oft den Großteil eines Bildes aus. Bins mit zu wenigen Pixeln können eliminiert werden. Das Bestimmen der k besten Farben ist allerdings aufwändig und hat je nach Verfahren mindestens einen quadratischen Zeitaufwand. Eine Lösung für dieses Problem ist die joint-Histogramm-Technik [Fen03 S. 6].

Neben den genannten Vorzügen beinhalten Histogramme aber auch einige Nachteile. Sie sind gegen viele Störgrößen anfällig. Darunter fallen zum Beispiel die Lichtverhältnisse. Je nach Beleuchtung unterscheidet sich das Farbspektrum eines Bildes erheblich. Auch unterschiedliche Bildauflösungen sind beim Vergleich problematisch. Eine Anpassung der Auflösung führt bei Verkleinerung des größeren Bildes zu Datenverlust. Das kleinere Bild kann in der Regel nicht ohne weiteres angepasst werden. Auch Bildstörungen wie Rauschen, Kratzer, Spiegelungen und Reflexionen im Bild beeinflussen das Retrieval-Ergebnis. Ein weiteres großes Problem ist, dass Histogramme wenig über den eigentlichen Bildinhalt aussagen. Bilder mit einer ähnlichen Farbverteilung werden unabhängig vom Bildinhalt als ähnlich eingestuft. Aus dieser Tatsache ergibt sich noch ein weiteres Problem. Mit zunehmender Anzahl von Bildern in der DB wird es immer wahrscheinlicher, dass sich die Histogramme ganz verschiedener Bilder ähneln. Der Vergleich von Histogrammen wird dann wirkungslos. Die Lösung dieser Probleme liegt in der Berücksichtigung der räumlichen Verteilung von Pixeln. So wird neben der Information, welche Farben im Bild verwendet wurden, auch die Information, wo im Bild die Farben dargestellt sind, mit erfasst. Um dies zu berücksichtigen existieren diverse Ansätze wie Color Coherence Vector (CVV), Color Correlogram, Rasterung und Segmentierung, sowie Annular- und Angular Histogramme. Problematisch an

allen Ansätzen sind der Mehraufwand bei der Berechnung und Speicherung der Histogramme, sowie eine stärkere Abhängigkeit gegenüber Translation, Rotation oder Skalierung.

Color Coherence Vector

Der CVV ist ein Versuch räumliche Information in Histogramme zu integrieren. Jedes Bin der Histogramme wird dabei zusätzlich unterteilt in kohärente und inkohärente Pixeleinträge, wobei die kohärenten Pixel zu einer größeren Region gehören und die inkohärenten nicht. Vor allem für Bilder mit großen einheitlichen Farbregionen und Texturen erweist sich dieses Verfahren als günstig und liefert bessere Resultate als einfache Histogramme für gleichartige Bilder. Ein Vektor ist wie folgt definiert.

$<(\alpha_1,\beta_1), (\alpha_2,\beta_2),..., (\alpha_N,\beta_N)>$

α_i: Anzahl kohärenter Pixel der i-ten Farbe eines Bildes

β_i: Anzahl inkohärenter Pixel der i-ten Farbe eines Bildes

Die Rücktransformation in das ursprüngliche Histogramm ergibt sich aus der Summe von kohärenten und inkohärenten Pixeln für jede Farbe.

$$<(\alpha_1+\beta_1), (\alpha_2+\beta_2),..., (\alpha_N+\beta_N)> \quad (3\text{-}4)$$

Color Correlogram

Ein weiterer Ansatz ist das Color Correlogram. Dieses beschreibt die räumliche und farbliche Beziehung von Punktpaaren im Bild. Die Darstellung führt zu einem dreidimensionalen Histogramm. Die ersten beiden Dimensionen *i, j* mit $i,j \in \{1,...N\}$ stellen den Farbwert von zwei Pixeln und die dritte Dimension die Distanz k dieses Pixelpaares dar, wobei $k \in \{1,...,d\}$ ist. Die Wahrscheinlichkeit, dass ein Pixel der Farbe *j* in einer Entfernung *k* von einem Pixel der Farbe *i* im Bild I zu finden ist, wird durch das folgende Farbkorrelogramm beschrieben.

$$\gamma_{i,j}^{(k)} = \Pr_{p_1 \in I_{c(i)}, p_2 \in I} \{ p_2 \in I_{c(j)} || p_1 - p_2 | = k \} \quad (3\text{-}5)$$

I: Gesamtmenge der Bildpixel

$I_{c(i)}/I_{c(j)}$ Menge Bildpixel der Farbe c(i)/c(j)

k: Distanz der Pixel p_1 und p_2

Die Berücksichtigung aller Farbpaare führt zu einem extrem großen Korrelogramm mit einem Aufwand von $O(N^2d)$. Werden hingegen nur die räumlichen Beziehungen zwischen gleichen Farben berücksichtigt, so reduziert sich die Dimensionsanzahl und damit der Rechen- und Speicheraufwand auf $O(Nd)$. Diese Weiterentwicklung heißt color autocorrelogram. Das Autokorrelogramm liefert durch seine höhere Stabilität bessere Retrieval-Ergebnisse als Standardhistogramme oder auch Kohärenzvektoren (CVV), hat aber auch durch die höhere Dimensionalität längere Rechenzeiten. [Fen03 S. 6f]

Rasterung

Eine der einfachsten Methoden räumliche Information in Form von Histogrammen zu speichern ist die Rasterung. Dabei wird das Bild durch ein Raster in verschiedene Regionen aufgeteilt und für jede entstandene Region ein separates Histogramm berechnet. Zwei Bilder, deren Standardhistogramm recht ähnlich ist, können so unterschieden werden.

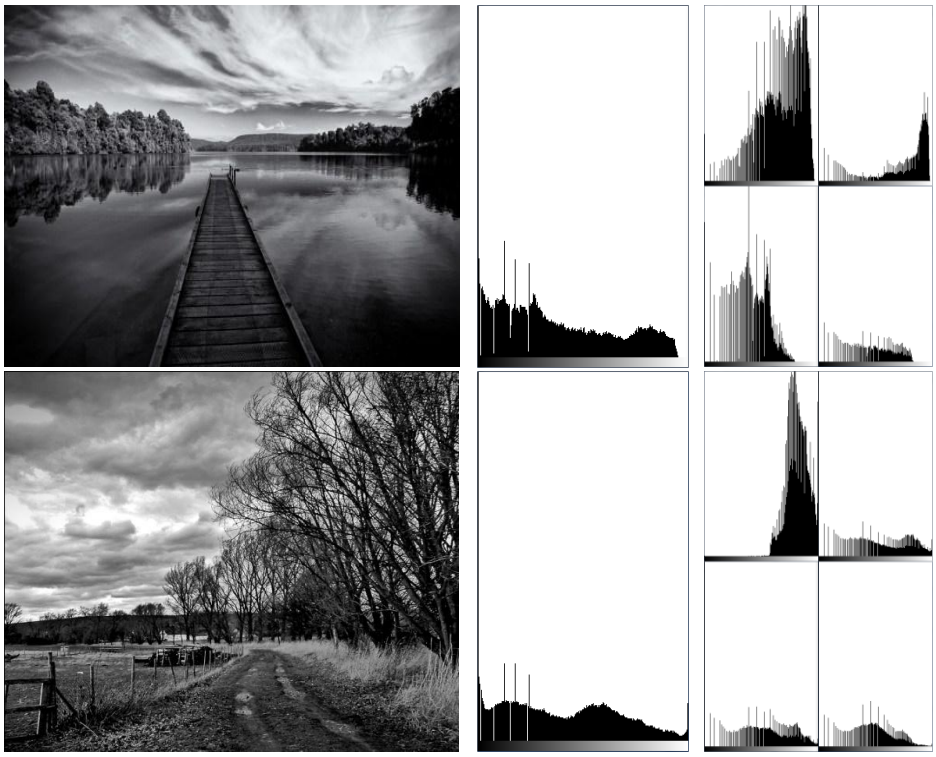

Abbildung 3-5: Vergleich zweier Originalbilder (links) [Nob11, Hol11] mit jeweils einem globalen Histogramm (Mitte) und einem in vier Regionen gerasterten Histogramm (rechts)

Viele Verfahren verzichten auch auf die hier verwendete feste Einteilung der Raster, um das meist wichtigere Bildzentrum stärker zu gewichten. Eine weitere Methode, die ebenfalls in den Bereich der Bildpartitionierung fällt, ist die Segmentierung. Sie ist wesentlich komplexer, da sie näher an der Objekterkennung arbeitet und im Bild semantisch bedeutsame Einheiten isoliert.

Annular und Angular Histogram

Beim Annular Histogram wird das Bild in kreisförmige Regionen eingeteilt. Die Auszählung der Farbanteile von innen nach außen führt zur räumlichen Beschreibung der Farbverteilung in Form eines Vektors [Rao99 S. 183-186].

Der Vorteil dieser Methode liegt in der Abhängigkeit des Retrievals von der räumlichen Verteilung. Dadurch können beispielsweise Sternbilder oder Luftaufnahmen von Inselgruppen voneinander unterschieden werden. Manchmal ist die Unabhängigkeit der Positionierung im Raum allerdings erwünscht. Beispielsweise sollen Bilder einer Bildserie, auf der sich das dominierende Objekt bewegt, nicht als unähnlich erkannt werden.

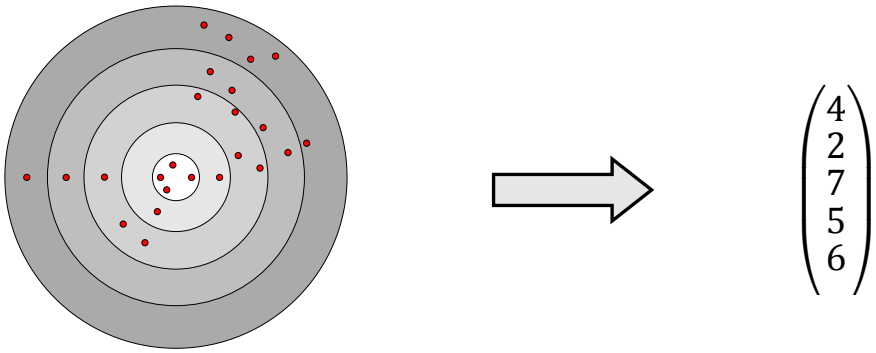

Abbildung 3-6: Annular Histogramm mit Farbverteilungsvektor

Eine ähnliche Technik ist das Angular Histogram. Hierbei wird die Bildfläche durch Winkel unterteilt. Die Auszählung der Farbanteile in den entstandenen Regionen gegen den Uhrzeigersinn ergibt die Vektorbeschreibung. Der Vorteil dieses Verfahrens ist die Abhängigkeit des Retrievals gegenüber vertikaler bzw. horizontaler Ausrichtung [Rao99 S. 183-186].

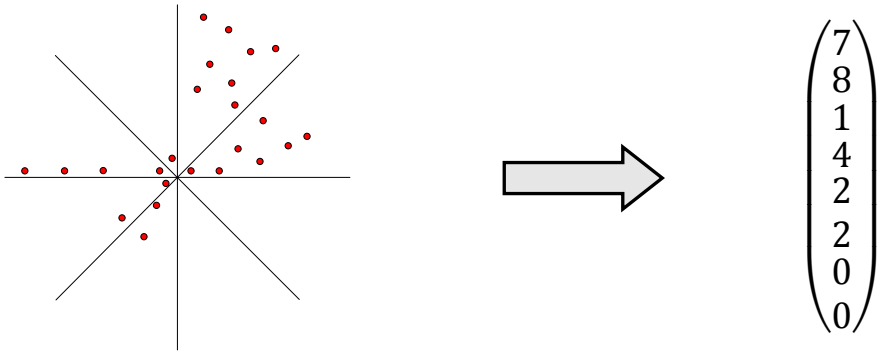

Abbildung 3-7: Angular Histogramm mit Farbverteilungsvektor

Die Vorteile beider Methoden können in einem hybriden Verfahren kombiniert werden. Das Bild wird in kreisförmige Regionen unterteilt, welche zusätzlich durch Winkel separiert werden. Die Auszählung der Farbanteile erfolgt in den Regionen von innen nach außen entgegen dem Uhrzeigersinn.

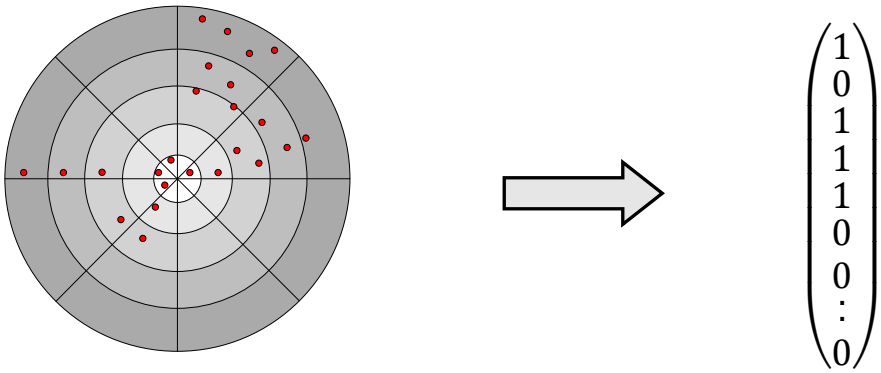

Abbildung 3-8: hybrides Histogramm mit Farbverteilungsvektor

Die Tauglichkeit der hier vorgestellten Verfahren wurde von R. Aibing genauer untersucht und die Ergebnisse aller Techniken miteinander verglichen [Rao99 S. 183-186]. Das Testsetting bestand dabei aus 500 Bildern verschiedener Themengebiete aus Kunst und Photographie. Annulare Histogramme zeigten in diesem Test die beste Performance. Ihnen folgten Angulare und hybride Histogrammverfahren. Der CCV und die Standardhistogramme landeten auf den hinteren Plätzen.

3.2.2 Texturbasierte Features

Co-Occurrence Matrizen

Eine Möglichkeit zur Ermittlung von texturbasierten Features sind Co-Occurrence Matrizen (Grauwert-Matrizen). Sie ermitteln das wiederholte Vorkommen von Grauwertmustern, indem die Häufigkeit zweier Grauwerte *a, b* im Abstand *q* in Richtung *φ* gezählt wird [San01 S. 189f]. Die resultierenden relativen Frequenzen $P_{\Phi,q}(a,b)$ werden in Form einer Matrix gespeichert, wobei jede Vorkommenshäufigkeit über die Texturgröße *M* normiert wird.

$$P_{\varphi,q}(a,b) = \frac{1}{M^2}\left|\left\{(i,j),(h,k): d\big((i,j),(h,k)\big) = q, \tan\varphi = \frac{i-j}{h-k}, f(i,j) = a, f(h,k) = b\right\}\right| \quad (3\text{-}6)$$

Ein einfaches Beispiel soll das Vorgehen veranschaulichen. Im Beispiel sei der Abstand *q* der betrachteten Pixel 1 und der Richtungswinkel *φ* 135°. Die resultierende Matrix ist wie folgt zu lesen. Auf einen Pixel der Farbe 0 folgt in Betrachtungsrichtung im gegebenen Bereich nie ein weiteres Pixel der Farbe 0 und auch kein Pixel der Farbe 2, aber dreimal ein Pixel der Farbe 1. Die Kennzahlen der Farben ergeben sich dabei aus der Durchnummerierung aufsteigend zur hellsten Farbe.

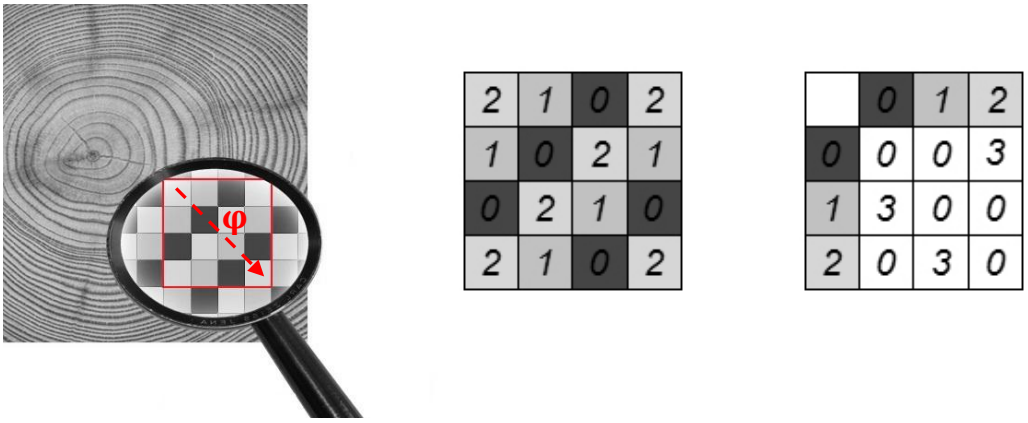

Abbildung 3-9: extrahierte Bildtextur und berechnete Co-Occurence Matrix

Die Co-Occurence wird ähnlich wie die Autokorrelation in der Regel nicht als eigenständiges Feature genutzt, sondern dient als Ausgangspunkt für komplexere Maße. Die folgende Tabelle zeigt eine Übersicht über mögliche Anwendungen.

Maß	Berechnungsformel
Energie	$\sum_{ab} P^2_{\phi,d}(a,b)$
Entropie	$\sum_{a,b} P_{\phi,d}(a,b) \log P_{\phi,d}(a,b)$
Maximale Eintrittswahrscheinlichkeit	$\max_{a,b} P_{\phi,d}(a,b)$
Kontrast	$\sum_{a,b} \lvert a-b\rvert^{\kappa} P^{\lambda}_{\phi,d}(a,b)$
Inverse difference moment	$\sum_{a,b;a\neq b} \frac{P^{\lambda}_{\phi,d}(a,b)}{\lvert a-b\rvert^{\kappa}}$

Tabelle 3-2: Anwendungsbeispiele von Co-Occurence

Tamura Features

1977 stellten H. Tamura, S. Mori, und T. Yamawaki sechs texturbasierte Features vor, welche in vielen Punkten mit der menschlichen Wahrnehmung übereinstimmen. Zu diesen Features gehören Grobheit (coarseness), Kontrast (contrast), Gerichtetheit (directionality), Linienartigkeit (line-likeness), Regelmäßigkeit (regularity) und Unebenheit (roughness). Eine spätere Untersuchung in Form eines Signifikanztests ergab, dass besonders die ersten drei Features für das CBIR wichtig sind, da sie besonders stark mit der menschlichen Wahrnehmung korrelieren [Des03 S. 27]. Im Folgenden wird daher lediglich zu diesen drei Features näher Bezug genommen.

Die Grobheit gibt Auskunft über die Größe der Texturelemente. Je höher der Wert der Grobheit ist, desto grober wirkt eine Textur. Zum Vergleich verschiedener Bilder mit quantitativ gleich stark vertretenen Texturen im Bild wird stets die Textur herangezogen, die den höchsten Grobheitswert aufweist. Die folgende Abbildung zeigt den Vergleich zweier Texturen mit unterschiedlicher Grobheit.

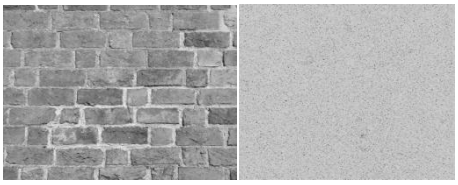

Abbildung 3-10: Vergleich einer Textur mit hoher (links) [Man11] und geringer Grobheit (rechts) [Kun11]

Zum Berechnen des Features für die Grobheit eines Bildes sind die folgenden vier Arbeitsschritte notwendig. [Des03 S. 28]

(1) Für jedes Pixel (x, y) wird der Durchschnitt über seine Nachbarschaft (verschiedener Fenstergrößen 2^k) berechnet

$$A_k(x,y) = \frac{1}{2^{2k}} \sum_{i=1}^{2^{2k}} \sum_{j=1}^{2^{2k}} \mathrm{F}\left(\mathrm{x} - 2^{\mathrm{k}-1} + \mathrm{i}, \mathrm{y} - 2^{\mathrm{k}-1} + \mathrm{j}\right) \tag{3-7}$$

(2) Für jedes Pixel (x, y) wird die absolute Differenz $E_k(x,y)$ zwischen den beiden nicht-überlappenden Nachbarschaftsfenstern in horizontaler und vertikaler Richtung berechnet (Fensterpaare gleicher Größe links und rechts bzw. über und unter dem Pixel)

$$E_k^h(x,y) = \left| A_k\left(x + 2^{k-1}, y\right) - A_k\left(x - 2^{k-1}, y\right) \right| \tag{3-8}$$

$$E_k^v(x,y) = \left| A_k\left(x, y + 2^{k-1}\right) - A_k\left(x, y - 2^{k-1}\right) \right| \tag{3-9}$$

(3) Auswahl des Fensters mit dem höchsten Differenzwert für jedes Pixel

$$S(x,y) = \underset{k=1..5}{argmax} \max_{d=h,v} E_k^d(x,y) \tag{3-10}$$

(4) Der Durchschnitt über die 2^S Werte des Gesamtbildes (X, Y) ergibt das Feature der Grobheit eines Bildes

$$F_{crs} = \frac{1}{XY} \sum_{x=1}^{X} \sum_{y=1}^{Y} 2^{S(x,y)} \tag{3-11}$$

Der Kontrast trägt ebenfalls entscheidend zur Bildqualität bei. Er wird größtenteils durch die vier Faktoren Dynamikbereich der Graustufen, der Verteilung von Schwarz und Weiß, der Kantenschärfe und dem Zeitraum wiederholender Muster bestimmt. Die nachfolgende Abbildung zeigt den Vergleich einer kontrastarmen und einer kontrastreichen Textur. [Des03 S. 28]

Abbildung 3-11: Vergleich einer Textur mit viel (links) [Piq11] und wenig Kontrast (rechts) [ker11]

Die Berechnung des Kontrasts geschieht durch folgende Arbeitsschritte. Dabei wurde experimentell ermittelt, dass der Parameter *z* gegen ¼ determiniert.

(1) Den Vierten Moment des Bildmittelwertes μ berechnen

$$\mu_4 = \frac{1}{XY}\sum_{x=1}^{X}\sum_{y=1}^{Y}(F(x,y)-\mu)^4 \tag{3-12}$$

(2) Den Quotienten aus dem Moment des Mittelwertes und dem Quadrat der Varianz der Grauwerte bilden

$$\alpha_4 = \frac{\mu_4}{\sigma^4} \tag{3-13}$$

(3) Der Quotient aus der mittleren Abweichung und dem berechneten Wert für α ergibt das Kontrast-Feature

$$F_{con} = \frac{\sigma}{{\alpha_4}^z} \tag{3-14}$$

Bei Gerichtetheit ist das Vorhandensein einer Orientierung in der Textur von Bedeutung. Ein Vergleich von gerichteten und nicht gerichteten Texturen ist nachfolgend gezeigt.

Abbildung 3-12: Vergleich einer gerichteten (links) [tha11] und nicht gerichteten Textur (rechts) [Tis11]

Die Berechnung der Gerichtetheit geschieht in folgender Form [Fen03 S. 9].

(1) Das Bild wird mit zwei 3x3-Matrizen gefaltet. Diese können beispielsweise folgende Form besitzen

$$\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \qquad \begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}$$

(2) Betrag und Winkel des Gradientenvektors für jedes Pixel berechnen, wobei Δ_H und Δ_V die horizontalen und vertikalen Differenzen der Faltung darstellen

$$|\Delta G| = \frac{(|\Delta_H| + |\Delta_V|)}{2} \tag{3-15}$$

$$\theta = \tan^{-1}\left(\frac{\Delta_V}{\Delta_H}\right) + \frac{\pi}{2} \tag{3-16}$$

(3) Quantisierung von θ und Pixel zählen, deren Betrag |ΔG| einen gesetzten Schwellwert überschreitet

(4) Konstruktion eines Histogramms H_D aus θ. Dieses Histogramm zeigt für stark gerichtete Bilder ausgeprägte Peaks und ist für schwach-gerichtete Bilder relativ flach

(5) Berechnung des Features für die Gerichtetheit aus der Schärfe des Histogramms (betrachtet Anzahl der Bins zwischen zwei Peaks)

Für das Bild-Retrieval werden die drei Features leicht modifiziert verwendet. Gewünscht ist für jeden Pixel ein eindeutiger Eigenschaftswert. Um dies zu erreichen werden Grobheit und Richtwirkung pixelweise berechnet. Die Berechnung des Kontrasts erfolgt in einer 13x13-Nachbarschaft. Des Weiteren wird ein Sobel-Filter verwendet, um für jeden Pixel die Richtung der Umgebung dieses Pixels zu berechnen. Jeder Pixel erhält so je einen Wert für die Grobheit, den Kontrast und die Richtung der Nachbarschaft. Daraus wird anschließend ein dreidimensionales Histogramm erstellt.

Wold Features

Eine weitere Möglichkeit die Bildtextur zu beschreiben ist die Wold-Zerlegung. Im Wold-Model wird das Bild bzw. ein Bildausschnitt als räumliches, homogenes, zufälliges Feld $\{y(m,n)|(m,n) \in \mathbb{Z}^2\}$ aufgefasst. Dieses Feld kann in drei Komponenten, den harmonischen (harmonic), den flüchtigen (evanescent) und den unbestimmten (indeterministic) Anteil zerlegt werden. Dabei ist zu beobachten, dass periodische Texturen eine starke harmonische Komponente, gerichtete Texturen eine starke flüchtige Komponente und wenig strukturierte (zufällige) Texturen eine starke unbestimmte Komponente besitzen [Fen03 S. 9]. Die drei extrahierten Komponenten sind orthogonal zu einander, wobei folgende Beziehung gilt.

$$y(m,n) = u(m,n) + d(m,n) = u(m,n) + h(m,n) + e(m,n) \tag{3-17}$$

y(m,n):	zufälliges Feld
u(m,n):	nicht deterministische (unbestimmte) Komponente
d(m,n):	deterministische Komponente (weiter zerlegbar)
h(m,n):	harmonische Komponente
e(m,n):	flüchtige Komponente

Diese Zerlegung ist auch auf den Frequenzraum übertragbar. Hierbei stellt F eine spektrale Verteilungsfunktion (spectral distribution functions) der entsprechenden Anteile *y, u, d, h, e* dar.

$$F_y(\xi,\eta) = F_u(\xi,\eta) + F_d(\xi,\eta) = F_u(\xi,\eta) + F_h(\xi,\eta) + F_e(\xi,\eta) \tag{3-18}$$

Zur Bestimmung der Wold-Features existieren in der bildräumlichen Domäne wie auch in der Frequenzdomäne verschiedene komplexe Methoden. Die deterministische Periodizität eines Bildes kann über die Analyse der Autokorrelationsfunktion bestimmt werden. Die korrespondierenden Wold-Features werden aus den Frequenzen und Amplituden der harmonischen spektralen Peaks abgeleitet. Die nicht deterministische (zufällige) Komponente ergibt sich aus Anwendung der MR-SAR-Modellierung (multiresolution simultaneous autoregressive modelling). Für das Retrieval werden die harmonischen Peaks und Distanzen zwischen den MRSAR-Parametern gleichgesetzt. In Experimenten wurde gezeigt, dass die Retrieval-Ergebnisse der Wold-Features besser sind als die Resultate der Tamura-Features [Cas02]. Des Weiteren existieren Ansätze die drei Wold-Feature-Parameter über eine Maximum Likelihood Abschätzung zu bestimmen. Eine Alternative zu den räumlichen Verfahren ist die Verwendung spektraler Transformationen wie der diskreten Fourier-Transformation (DFT), der diskreten Cosinus Transformation (DCT) oder der diskreten Wavelet-Transformationen (DWT). Die Berechnung von globalen Leistungsspektren der DFT ist im Vergleich zu lokalen Features und Fensterverfahren nicht sehr effektiv bei der Texturklassifizierung und im CBIR. Derzeit sind die aussichtsreichsten Methoden für das Textur-Retrieval die Multiresolution-Features mit orthogonalen Wavelet-Transformationen oder mit Gabor-Filterung [Gim06]. Die Merkmale beschreiben räumliche Verteilungen von orientierten Kanten im Bild auf mehreren Skalen.

Gabor Filter Features

Der Gabor-Transformation ist eine weitere Technik zum Extrahieren von texturbasierten Features. Durch die Verwendung einer Gaußfunktion werden Frequenz- und räumliche Informationen erfasst [Des03 S. 25]. Dabei zeichnen sich die Gabor-Filter-Features durch eine besonders hohe Robustheit bei der Beschreibung aus. Das Verfahren zeigt sich robust gegenüber Frequenz- und Raumvariation. Außerdem ist es invariant bei Kontrast- und Helligkeitsänderungen und ist dem menschlichen Sehen angepasst [San01 S. 190f]. Der Gabor-Filter wird oft im Zusammenhang mit im Maßstab veränderbaren Knoten- und Kanten-Detektoren eingesetzt. Die generelle Idee dabei ist die Verwendung einer zweidimensionalen Gabor-Funktion *g(x,y)* mit einer Frequenz ω und der Bandbreite $\sigma_x \sigma_y$.

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} exp\left[-\frac{1}{2}\left(\frac{x^2}{{\sigma_x}^2} + \frac{y^2}{{\sigma_y}^2}\right) + 2\pi\omega x\right] \tag{3-19}$$

Bei diesem Grundwavelet entsprechen σ_x und σ_y der Standardabweichung der Gauß'schen Hüllkurve in x- und y-Richtung. Der Gabor-Filter ergibt sich durch geeignete Streckungen und Drehungen der Funktion *g(x,y)*. [Fen03 S. 11]

$$g_{mn}(x,y) = a^{-m} g(x',y') \quad \text{mit} \quad \begin{aligned} x' &= a^{-m}(x\cos\theta + y\sin\theta) \\ y' &= a^{-m}(-x\sin\theta + y\cos\theta) \\ \theta &= \frac{n\pi}{K} \end{aligned} \tag{3-20}$$

wobei:
- a > 1
- n ∈ 0,1,…,K-1
- m ∈ 0,1,…,S-1

K: Anzahl der Richtungen
S: Anzahl der Skalen

Wavelet-Transformation

Ähnlich zu der Gabor-Transformation verfolgt die Wavelet-Transformation einen Multi-Resolution-Ansatz zur Texturanalyse, bei dem das Bild bei unterschiedlicher Ortsauflösung spektral zerlegt wird [Fen03. S. 11]. Im Gegensatz zum Beispiel der Fourier-Transformation ist es durch die Verwendung von Wavelets möglich gleichzeitig Orts- und Frequenzinformation in einem Merkmalswert zu kombinieren. Dies ist für die Erkennung lokal begrenzter Texturen unabdingbar und ein großer Vorteil der DWT. Des Weiteren ist die Wavelet-Transformation robust gegenüber Größen- und Frequenzänderungen. Durch Verschiebung (Translation) und Skalierung (Dilation) eines Grundwavelet $\psi(x)$ (Mutter-Wavelet) werden eine Reihe von Basisfunktionen $\psi_{mn}(x)$ generiert, welche das Bildsignal beschreiben.

$$\psi_{\mathrm{mn}}(\mathrm{x}) = 2^{-\frac{\mathrm{m}}{2}}\psi(2^{-\mathrm{m}}\mathrm{x} - \mathrm{n}) \tag{3-21}$$

Dabei stellen *m* und *n* die Parameter für die Translation und Dilation dar. Das Bildsignal ergibt sich aus der Aufsummierung der Basisfunktionen.

$$f(x) = \sum_{m,n} c_{mn}\psi_{\mathrm{mn}}(\mathrm{x}) \tag{3-22}$$

Bei der Berechnung der Wavelet-Transformation wird das zweidimensionale Signal in vier Sub-Bänder LL, LH, HL und HH aufgeteilt. Die Bezeichnungen L und H beschreiben das jeweilige Frequenzband. L steht hier für die niedrigen Frequenzen (low frequency) und H für die hohen Frequenzen (high frequency). Zur Texturanalyse werden zwei Grundtypen der Wavelet-Transformation unterschieden [Fen03 S. 12]. Die pyramid-structured wavelet transform (PWT) wird beim untersten Frequenzband und die tree-structured wavelet transform (TWT) für die restlichen drei Bänder eingesetzt. Die wichtigste Information liegt für viele Texturen meist in den mittleren Frequenzbändern. Unter Verwendung des Mittelwertes und der Standardabweichung der Energieverteilung dieser vier Sub-Bänder kann der Feature-Vektor erzeugt werden.

3.2.3 Formenbasierte Features

Zunächst werden die konturbasierten Features (boundary-based features) betrachtet. Der Begriff Kontur wird dabei wie folgt definiert und im Weiteren auch so verwendet.

Definition 3.5 (Kontur)
Die Kontur eines zweidimensionalen Objektes ist eine Reihe eng aufeinander folgender Pixel (x_s, y_s) [Fen03 S. 14]

Der Parameter *s* gibt dabei die Position des s-ten Pixel in der Konturesequenz an und liegt zwischen 0 und *N-1*, wenn *N* die Anzahl aller Konturpixel ist. Die nachfolgende Abbildung zeigt eine mit dem Canny-Edge-Filter extrahierte Objektkontur.

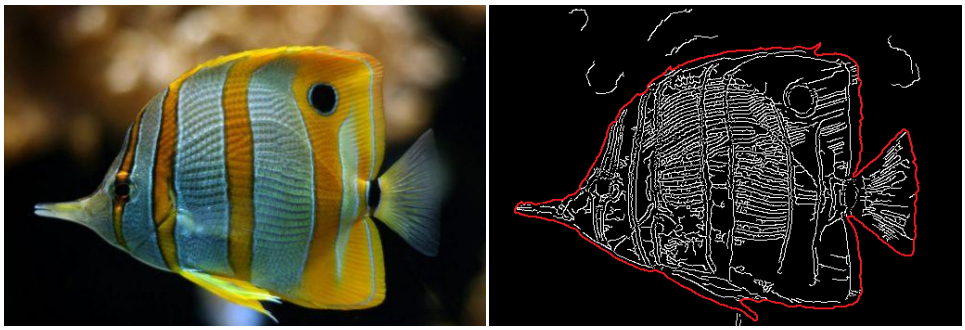

Abbildung 3-13: Originalbild (links) [Vie11] und extrahierte Objektkontur (rechts)

Man unterscheidet drei Arten der Konturdarstellung [Fen03 S. 14]. Die Kontur kann als Bogenlänge (curvature) *K(s)* des Pixels *s* verstanden werden und ist damit als Funktion über der Änderung der Tangentenrichtungen folgendermaßen definiert.

$$K(s) = \frac{d}{ds}\theta(s) \tag{3-23}$$

Die Rotationsfunktion *θ(s)* wird durch die nachfolgende Gleichung bestimmt, wobei *x'* und *y'* die erste Ableitung der Koordinaten eines Konturpixels an der Stelle *s* darstellen.

$$\theta(s) = \tan^{-1}\frac{y'_s}{x'_s} \qquad \text{mit} \qquad y'_s = \frac{dy_s}{ds} \quad , \quad x'_s = \frac{dx_s}{ds} \tag{3-24}$$

Eine weitere Möglichkeit der Konturdarstellung ist die Zentroid-Distanz (centroid distanz) *R(s)*. Diese Distanz ist definiert als Abstand zwischen einem Pixel der Kontur (x_s,y_s) und dem Zentroiden (x_c,y_c) der Form.

$$R(s) = \sqrt{(x_s - x_c)^2 + (y_s - y_c)^2} \tag{3-25}$$

Zuletzt kann die Kontur als Funktion komplexer Koordinaten (complex coordinate function) *Z(s)* definiert werden. Dabei werden die Koordinaten der Konturpixel durch eine komplexe Zahl repräsentiert.

$$Z(s) = (x_s - x_c) + j(y_s - y_c) \tag{3-26}$$

Fourier-basierten Features

Nach der Einführung in diese grundsätzlichen drei Darstellungsformen sollen nun einige konturbasierte Features vorgestellt werden. Begonnen wird dabei mit den Fourier-basierten Features, da die im weiteren Verlauf vorgestellten Verfahren zum Teil auf der Fourier-Transformation aufbauen. Die Features beschreiben die Form eines Objektes mittels einer Fourier-Transformation der drei definierten Konturdarstellungen. Dabei werden drei komplexe Koeffizientenmengen erzeugt, welche die Objektform im Frequenzbereich darstellen. Niedrige Frequenzkoeffizienten symbolisieren dabei allgemeine Eigenschaften der Form während hohe Frequenzen Details der Form charakterisieren.

Fourier-Deskriptor der Bogenlänge

$$f_K = \left[|F_1|, |F_2|, \ldots, \left|F_{M/2}\right|\right] \tag{3-27}$$

Fourier-Deskriptor der Zentroid-Distanz

$$f_R = \left[\frac{|F_1|}{|F_0|}, \frac{|F_2|}{|F_0|}, \ldots, \frac{\left|F_{M/2}\right|}{|F_0|}\right] \tag{3-28}$$

F_i bezeichnet hier die i-te Komponente der Fourier-Transformations-Koeffizienten. Es werden allerdings nur die positiven Koeffizienten betrachtet, da die Transformationen symmetrisch sind, es gilt $|F_{-i}|=|F_i|$. Der dritte Fourier-Deskriptor ergibt sich wie folgt.

Fourier-Deskriptor der komplexen Koordinaten

$$f_Z = \left[\frac{\left|F_{-((M/2)-1)}\right|}{|F_1|}, \ldots, \frac{|F_{-1}|}{|F_1|}, \frac{|F_2|}{|F_1|}, \ldots, \frac{\left|F_{M/2}\right|}{|F_1|}\right] \tag{3-29}$$

F_1 ist der erste Frequenzkoeffizient ungleich null, welcher zur Normalisierung der Transformationskoeffizienten verwendet wird. Im Gegensatz zu den beiden ersten Deskriptoren werden die positiven und negativen Frequenzanteile betrachtet. Der Mittelwert wird hierbei nicht zur Normalisierung verwendet, da er von der Position der Form abhängig ist. Um sicherzustellen, dass die formbasierten Features aller Objekte in der Datenbank die gleiche Länge haben wird der Umriss jedes Objektes vor der Fourier-Transformation auf M Werte normiert. M kann beispielsweise auf $2^m=64$ gesetzt werden. Somit kann die effizientere Fast-Fourier-Transformation (FFT) verwendet werden. Durch weitere Umformungen gewährleisten diese Features ebenfalls Invarianz gegenüber verschiedenen Transformationen. Um Rotationsinvarianz zu erreichen wird beispielsweise die Amplitude der komplexen Koeffizienten genutzt und die Phase wird vernachlässigt. Ist zusätzlich Skalierungsinvarianz gefordert, so wird die Amplitude der Koeffizienten durch die Amplitude des Mittelwertes oder aber den ersten Koeffizienten, welcher ungleich null ist, dividiert. Translationsinvarianz wird bereits durch die Konturdarstellung geliefert.

Turning Angles

Ein weiteres Verfahren, welches auf formbasierten Features basiert sind die Turning Angles. Ein Turning Angle misst den Winkel der Tangenten, gesehen von einem gegebenen Referenzpunkt, entgegen dem Uhrzeigersinn. D. Feng, W.C. Sui und H. Zhang stellen in ihrem Werk „Multimedia Information Retrieval and Management“ [Fen03] hierfür ein Verfahren über die in Gleichung (2-25) eingeführte Formel einer Rotationsfunktion $\theta(s)$ des Bogenmaßes s vor. Dieser Ansatz beinhaltet zwei große Probleme. Erstens sind die extrahierten Features nicht rotationsinvariant und zweitens ist eine eindeutige Auswahl des Referenzpunktes durch diese Herangehensweise nicht gewährleistet. Für einen Referenzpunkt muss hierbei lediglich gelten, dass er auf der Objektkontur liegt. Das Verschieben des Referenzpunktes entlang der Objektkante um t verändert allerdings den berechneten Turning Angle von $\theta(s)$ zu $\theta(s+t)$. Falls das Objekt um den Winkel ω gedreht wird, entsteht ein neuer Turning Angle von $\theta(s)+\omega$. Durch die

mehrdeutige Rotationsfunktion sind zwei Objekte A und B nicht ohne weiteres vergleichbar. Es muss der minimale Abstand über alle möglichen Verschiebungen s und Rotationen ω berechnet werden, was einen erhöhten Rechenaufwand bedeutet. [Fen03 S. 13f]

$$d_p(A,B) = \left(\min_{\omega \epsilon R, t \epsilon [0,1]} \int_0^1 |\theta_A(s+t) - \theta_B(s) + \omega|^p \, ds \right)^{\frac{1}{p}} \qquad (3\text{-}30)$$

Eine weitere Möglichkeit zur Berechnung der Turning Angles, welche nicht die soeben besprochenen Probleme beinhaltet, wird in „A Study of Similarity Measures for a Turning Angles-based Shape Descriptor“ [Zib01] von Carla Zibreira und Fernando Pereira vorgestellt und ist auch im MPEG-7-Standard so umgesetzt. Wieder muss vor dem eigentlichen 3-schrittigem Feature-Extraktionsprozess die Objektkontur durch Segmentierung ermittelt werden. Das anschließende Verfahren ist in der Abbildung 3-14 dargestellt und wird im Folgenden näher diskutiert.

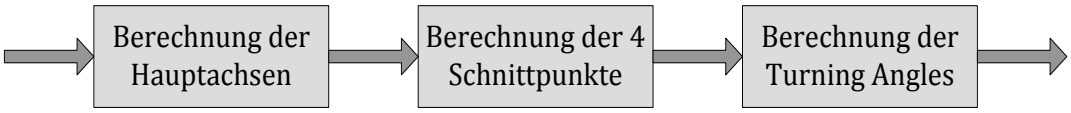

Abbildung 3-14: Schritte zur Extraktion von Turning Angles

Im ersten Schritt werden die zwei Hauptachsen des Objektes berechnet. Diese sind am Objektmassezentrum ausgerichtet und parallel zum kleinsten umschließenden Rechteck (minimal Bounding Box) des Objekts. Mathematisch gesehen ist die Berechnung der Hauptachsen das Bestimmen von Eigenvektoren einer Kovarianz-Matrix V.

$$V = \begin{bmatrix} V_{xx} & V_{xy} \\ V_{yx} & V_{yy} \end{bmatrix} \qquad V_{xx} = \frac{1}{N} \sum_{s=0}^{N-1} (x_s - x_c)(x_s - x_c) \qquad (3\text{-}31)$$

$$V_{xy} = V_{yx} = \frac{1}{N} \sum_{s=0}^{N-1} (x_s - x_c)(y_s - y_c)$$

$$V_{yy} = \frac{1}{N} \sum_{s=0}^{N-1} (y_s - y_c)(y_s - y_c)$$

Dabei entspricht N der Gesamtanzahl von Konturpixeln und (x_s,y_s) den Koordinaten des s-ten Pixels einer Form. Die Koordinaten des Objektmassezentrums (x_c,y_c) werden durch Mittelwertbildung der Konturpixel berechnet.

$$x_c = \frac{1}{N} \sum_{s=0}^{N-1} x_s \qquad y_c = \frac{1}{N} \sum_{s=0}^{N-1} y_s \qquad (3\text{-}32)$$

Im zweiten Schritt wird der Ausgangspunkt für die Berechnung der Turning Angles gewählt. Es besteht die Möglichkeit einen beliebigen Konturpixel zu wählen, was allerdings nicht optimal für das spätere Matching ist. Die Wahl des Referenzpunktes beeinflusst, wie bereits gezeigt, die berechneten Turning Angles, wodurch zwei Objekte nicht ohne weitere Umformungen vergleichbar sind. Um eine eindeutige Zuordnung und

zusätzlich Rotationsinvarianz zu erreichen wird der Ausgangspunkt auf Basis der Form-Momente bestimmt. Als Startpunkt wird der Punkt gewählt, der am weitesten vom Objektmassezentrum entfernt ist. Das Problem nur eines einzigen Startpunktes ist seine fehlende Robustheit. Bereits bei geringen Änderungen des Objektes kann der vom Zentrum am weitesten entfernte Punkt in einen völlig anderen Bildbereich springen, was gravierende Änderungen der Objektbeschreibung nach sich zieht. Stattdessen werden meist vier Ausgangspunkte bestimmt um eine Form zu beschreiben. Dabei handelt es sich um die vier Schnittpunkte zwischen der Objektkontur und den Hauptachsen. Für jeden Startpunkt wird anschließend ein Vektor von Winkeln erzeugt. Der MPEG-7-Standard schlägt folgende Schritte zur Berechnung der Ausgangspunkte vor [Zib01].

(1) Die Hauptachsen des Objektes werden mit S_1 und S_2 bezeichnet
(2) P_1 ist der Punkt auf S_1, welcher am dichtesten am Massezentrum liegt
(3) P_2 ist der Punkt auf S_1, welcher am weitesten vom Massezentrum entfernt ist und zu P1 auf der gegenüberliegenden Seite des Objektmassezentrums liegt
(4) P_3 und P_4 ergeben sich analog auf S_2

Im letzten Schritt wird eine Menge von Konturpixeln gewählt, um die Krümmung der Turning Angles zu berechnen. Jeder Winkel ist durch zwei Vektoren definiert. Der erste Vektor ergibt sich durch zwei aufeinanderfolgende Konturpixel und der andere durch eine Objekthauptachse. Die Berechnung der Turning Angles erfolgt an repräsentativen Punkten. Diese Punkte sind definiert als der Durchschnitt von *k* gleichmäßig verteilt Konturpunkte, wobei der berechnete Durchschnittswert nicht auf der eigentlichen Objektkontur liegen muss (vgl. Abbildung 3-15). Die Anzahl *k* der Konturpixel ergibt sich dabei aus dem Verhältnis zwischen dem Objektumfang und der Anzahl der Winkel *N*, die zur Objektbeschreibung genutzt werden. Im MPEG-7-Standard wird für *N* als Wert 64 vorgeschlagen. Die Bestimmung der Winkel erfolgt entgegen dem Uhrzeigersinn und wird in der nachfolgenden Abbildung illustriert. [Zib01]

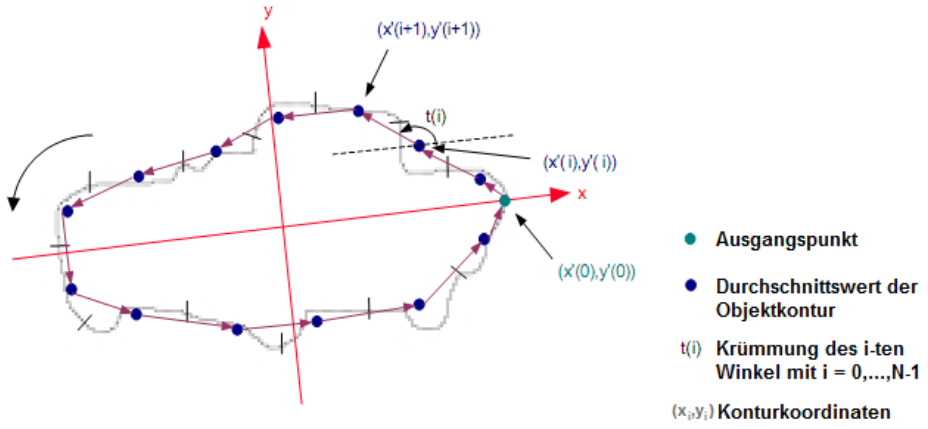

Abbildung 3-15: Schema der Berechnung von Turning Angles [Zib01]

Curvature Scale Space

Das letzte hier betrachtete Feature, welches auf einer Beschreibung der Objektkontur basiert, ist der Curvature Scale Space (CSS). Der CSS gehört zu den wichtigsten objektformbasierten lokalen Features, da durch die Verwendung der Form im Raum nicht

nur die Position, sondern auch der Grad der Ausbuchtungen bzw. Vertiefung einer Objektkontur erkannt wird [Zha01 S. 2]. Grundlage des Verfahrens ist eine Faltung der segmentierten Objektkontur mit einem Gauss-Filter zum Glätten der ermittelten Form. Die Abbildung 3-16 zeigt die notwendigen Schritte zur Berechnung des CSS, welche im darauffolgenden Abschnitt näher erläutert werden [Zha03 S. 7].

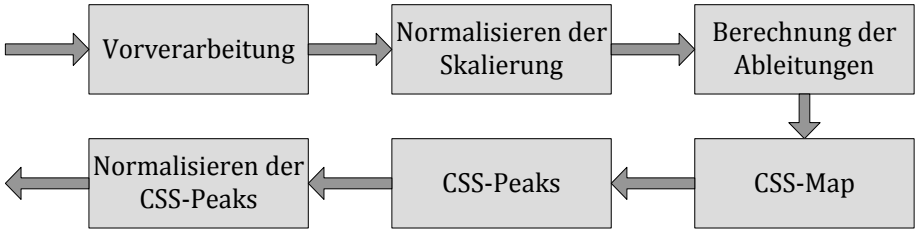

Abbildung 3-16: Schritte zur Extraktion von CSS-Features

Ähnlich zur Fourier-Transformation werden als erstes die Konturkoordinaten *(x_s, y_s)* mit *s = 1,...,N-1* berechnet. Um eine spätere Vergleichbarkeit verschiedener Konturen zu ermöglichen, wird im nächsten Schritt die Anzahl der Konturabtastwerte auf einen festen Wert normiert. Die verwendete Sampling-Technik basiert auf der Bogenlänge der Kontur, da diese die topologische Struktur der Objektgrenzen am besten erhält. Dadurch ermöglicht das CSS-Verfahren auch die Erkennung teilweise verdeckter Objektkonturen. Als nächstes wird die erste und zweite Ableitung der Konturpixel *(x_s, y_s)* berechnet. Mit Hilfe der berechneten Ableitungen wird eine CSS-Kontur-Map erstellt. Die Kontur-Map ist eine Multiskalen-Darstellung der Nullstellen der Konturbogenlänge, welche wie folgt berechnet wird. Durch die verwendete Multiskalen-Strategie zur Repräsentation der Objektform ist diese allgemeiner und weniger anfällig gegenüber Rauschen und anderen Störungen. [Zha03 S. 7]

$$k(s) = \frac{x'(s)y''(s) - x''(s)y'(s)}{(x'^2(s) + y'^2(s))^{3/2}} \tag{3-33}$$

x‘(s), y‘(s): 1. Ableitung an der Stelle s

x‘‘(s), y‘‘(s): 2. Ableitung an der Stelle s

Wird die Faltung mit einem breiteren Gauss-Filter wiederholt, so verschieben sich die Wendepunkte weiter auf den Objektrand bis sie sich gegenseitig auslöschen. Dies entspricht dem Entfernen der konvexen Wölbung, wodurch die Kontur weicher wirkt [Bue08 S. 5].

$$x_n(s) = x_{n-1}(s) \otimes g(s,\sigma) \tag{3-34}$$

$$y_n(s) = y_{n-1}(s) \otimes g(s,\sigma)$$

In jedem Schritt gehen so Details der segmentierten Form verloren. Jeder Iterationsschritt entspricht der Entwicklung der Form auf einer höheren Skala. Der Parameter σ gibt an, auf welcher Skala eine Nullstelle entdeckt wurde. Das Verfahren endet, wenn keine weiteren Nullstellen mehr gefunden werden können. Die resultierende CSS-Map beinhaltet alle gefunden Nullstellen *zc(s, σ)* in Abhängigkeit der Iterationsstufe (vgl. Abbildung 3-17). [Sti06 S. 69]

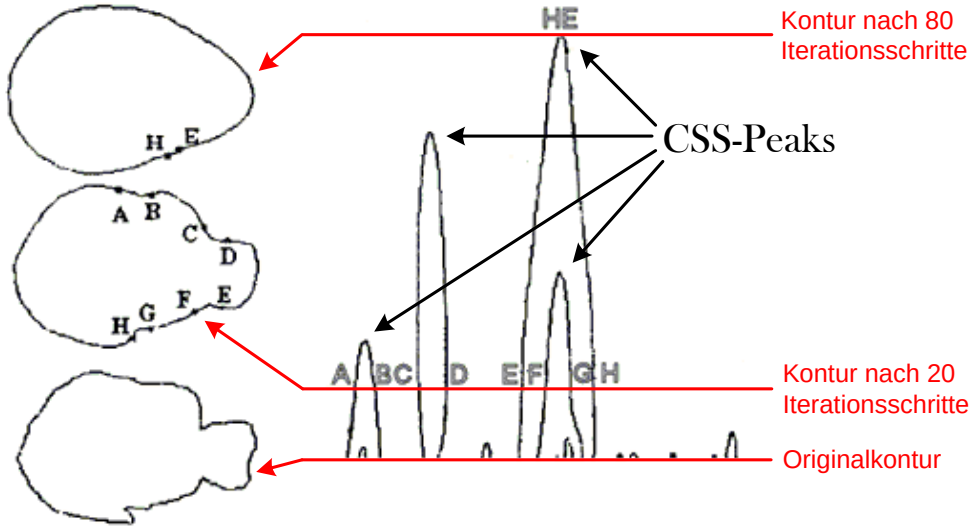

Abbildung 3-17: Darstellung einer Objektkontur im CSS [Sti06]

Der letzte Schritt ist die Extraktion der im CSS vorhandenen Peaks und die Normalisierung dieser [Zha03 S. 10]. Die Abbildung 3-18 zeigt das Resultat des Extraktionsprozesses.

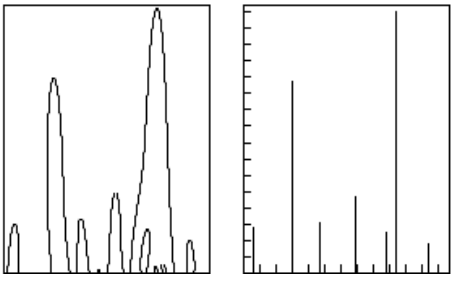

Abbildung 3-18: CSS-Map (links) mit extrahierten CSS-Peaks (rechts) [Zha03]

Die CSS-Features sind translationsinvariant [Zha03 S. 2]. Die durchgeführte Normalisierung in eine feste Anzahl von Konturpixeln liefert zusätzlich Skalierungsinvarianz. Rotationsinvarianz wird durch eine kreisförmige Verschiebung der Peaks in den Ursprung der CSS-Map erreicht. Ein Objektmatching kann direkt auf einzelnen oder mehreren Skalen oder aber indirekt auf der Ähnlichkeit der CSS-Darstellung der zu untersuchenden Segmente durchgeführt werden [Bue08 S. 5].

Alles in allem sind die CSS-Features sehr mächtig und wichtig für die Beschreibung der Ähnlichkeit zwischen Formen, da sie äquivalent zur menschlichen Wahrnehmung sind [Zha03 S. 2]. Die Dimensionalität der CSS ist recht gering und das Verfahren ist sehr robust, aber diese Features beinhalten auch einige Probleme. Zum einen werden lediglich lokale, formbasierte Features erfasst. Globale Merkmale werden hingegen nicht berücksichtigt, können aber durch eine Kombination globaler Funktionen, wie der Zirkularität oder der Anzahl der CSS-Peaks, hinzugewonnen werden. Des Weiteren entspricht die Anzahl der Peaks in Abhängigkeit vom Sampling und den gewählten Schwellwerten nicht der wahren Anzahl der konvexen und konkaven Abschnitte auf der Objektkontur. Außerdem existieren keine CSS-Deskriptoren für weiche, konvexe Formen.

Flächenbasierte Features

Es wurden bereits einige konturbasierte Features vorgestellt. An dieser Stelle soll nun kurz auf Features eingegangen werden, welche auf der Beschreibung der Objektflächen (region-based features) basieren. Im Gegensatz zu den konturbasierten Verfahren wird bei flächenbasierten Features direkt der Objektkörper innerhalb der Konturgrenzen beschrieben. Das Finden homogener Flächen geschieht in der Regel über statistische Momente, welche bereits im Kapitel der farbbasierten Features eingeführt wurden. Für das Beschreiben von Objektformen sind die ersten vier statistischen Momente bedeutend. Der Mittelwert (vgl. Gleichung. 3-1) und die Varianz (vgl. Gleichung. 3-2) geben Auskunft über die Lage und die Variabilität (Streuung bzw. Dispersion) der Regionen im Bild. Auch der dritte und vierte Moment, die Verzerrung (vgl. Gleichung. 3-3) und die Wölbung (Kurtosis) der Verteilung, tragen Informationen über die Lage von Objektregionen. Weicht die Intensität, die Farbe oder die Textur einer Menge von Pixeln in den genannten statistischen Momenten von ihrer Umgebung ab, so handelt es sich um eine Region. Das Bestimmen der Pixelregionen erfolgt üblicherweise analog zu den konturbasierten Features nach der Objektsegmentierung. Die Darstellung der Regionen geschieht in Form von Punktmengen, Blobs oder Skeletons [Gim06]. Die nachfolgende Abbildung zeigt die Segmentierung von Bildregionen mittels gängiger Blob-Detektoren.

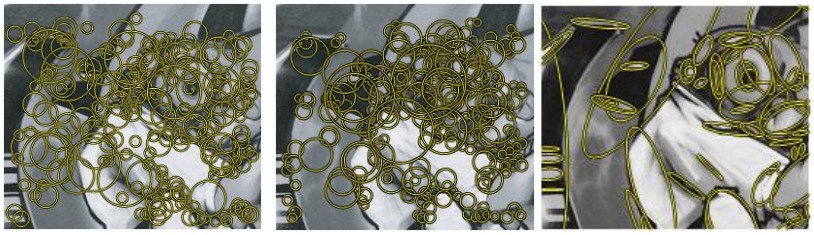

Abbildung 3-19: Detektion interessanter Bildregionen durch SIFT (links), SURF (Mitte) und MSER (rechts) [Tuy08]

Der Skeleton verfolgt einen anderen Ansatz zur Beschreibung dieser Bildregionen. Hierbei wird ein Graph genutzt, welcher die Form in den Ecken der Kontur aufspannt. Beim Verstehen, wie dieser Spanngraph erzeugt wird, hilft folgende Analogie. Die betrachtete Form wird als brennbare Fläche gesehen. An den Rändern der Fläche wird gleichzeitig Feuer gelegt, welches sich mit identischer Geschwindigkeit Richtung Mitte bewegt. Treffen sich dabei zwei Feuerfronten, löschen sie sich gegenseitig aus. Die Menge aller Punkte, an denen sich zwei Fronten treffen (quench points), bildet das Skelett der Form. Die Abbildung 3-20 zeigt einen einfachen Skeleton, der ein gegebenes Rechteck beschreibt. [Fel04 S. 13]

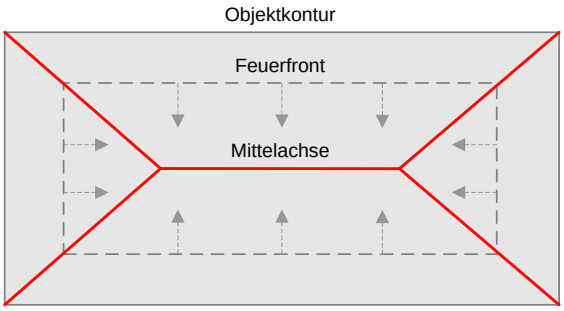

Abbildung 3-20: Skeleton eines Rechtecks [Fel04]

Moment Invariants

Zuletzt sollen noch die Moment Invariants als Vertreter der formbasierten Features untersucht werden. Hierbei handelt es sich um eine hybride Variante, welche sowohl Charakteristika aus den konturbasierten als auch aus den flächenbasierten Features beinhaltet. Klassisch wird ein Moment der Ordnung *(p, q)* wie folgt definiert, wobei die Funktion *f(x, y)* angibt, ob ein Pixel *(x, y)* zu einer geschlossenen Form *R* gehört oder nicht. Liefert die Funktion 1 als Wert zurück, so ist der Pixel enthalten in *R*, bei 0 ist er es nicht. [Che93 S. 683]

$$m_{pq} = \iint_R x^p y^q f(x,y) dx dy \tag{3-35}$$

Aus dieser allgemeinen Darstellung kann das zentrale Moment der Form *R* abgeleitet werden. Dabei ist (x_c, y_c) das Zentrum von *R*.

$$\mu_{pq} = \iint_R (x - x_c)^p (y - y_c)^q f(x,y) dx dy \tag{3-36}$$

mit $\quad x_c = \frac{m_{10}}{m_{00}} \quad , \quad y_c = \frac{m_{01}}{m_{00}}$

Bei den hier betrachteten Rasterbildern entspricht das Integral einer einfachen Aufsummierung der Merkmalswerte über den Spalten *x* und den Zeilen *y* des Bildes.

$$\mu_{pq} = \sum_{(x,y)} \sum_{\in R} (x - x_c)^p (y - y_c)^q \tag{3-37}$$

Bei dieser ersten Überlegung werden die invarianten Momente über die Pixel der Kontur und die durch diese begrenzten Pixel der Objektfläche berechnet. Dies ist recht zeitaufwändig. Eine Weiterentwicklung der klassischen invarianten Momente sieht deshalb vor, die Berechnung lediglich über die Objektkontur *C* durchzuführen.

$$\mu_{pq} = \sum_{(x,y) \in C} (x - x_c)^p (y - y_c)^q \tag{3-38}$$

Das so entstandene zentrale Moment wird zusätzlich normalisiert, um skalierungsinvariant zu sein.

$$\eta_{pq} = \frac{\mu_{10}}{{\mu_{00}}^{\gamma}} \quad \text{mit} \quad \gamma = \frac{p+q}{2} + 1 \tag{3-39}$$

Darauf aufbauend kann eine ganze Reihe weiterer robuster Momente konstruiert werden, welche invariant gegenüber Translation, Skalierung und Rotation sind. Diese können unter anderem im Buch „Multimedia Information Retrieval and Management“ [Fen03 S. 13] nachgelesen werden.

3.3 Lokale Feature-Detektoren

In den vorangegangenen Kapiteln wurden Features als inhaltstragende Bildmerkmale bereits definiert, klassifiziert und teilweise auf deren Extraktionsmethoden eingegangen. In diesem Kapitel sollen die Teile eines Retrieval-Systems betrachtet werden,

welche Features bei einer lokalen Betrachtung eines Bildes ermitteln. Genauer geht es um den Teil des Extraktionsprozesses, welcher die semantisch bedeutsamen Bildregionen durch eine lokale Betrachtung der Pixelumgebung identifiziert, den sogenannten Detektor. Im Wesentlichen erfüllt ein Feature-Detektor zwei wichtige Aufgaben. Zuerst berechnet er eine Abstraktion der im Bild dargestellten Information. Im Anschluss ermittelt der Detektor auf der Basis der durchgeführten Vorverarbeitung interessante Bildregionen, welche sich besonders gut für eine eindeutige Bildbeschreibung eignen.

Analog zu den Features existiert, je nachdem welcher Bildinhalt extrahiert werden soll, eine Vielzahl an Feature-Detektoren. Bevor auf verschiedene Vertreter der einzelnen Gruppen eingegangen wird, soll der Begriff Feature-Detektor formal definiert werden und im Anschluss allgemeine Anforderungen an einen idealen Detektor festgelegt werden.

Definition 3.6 (Feature-Detektor)
Der Begriff Feature-Detektor bezeichnet in der Computer Vision ein Werkzeug zum Ermitteln der zu extrahierenden Features in einem Bild [Tuy08 S. 180].

Die wünschenswerten Eigenschaften für Detektoren überdecken sich weitgehend mit den Anforderungen an Features. Im Wesentlichen ist eine schnelle Ermittlung möglichst aussagekräftiger und robuster Features das Ziel einer guten Detektion. Die Tabelle 3-3 zeigt eine Übersicht über die wichtigsten Merkmale für performante Detektoren. Die Reihenfolge der Merkmale definiert dabei kein Ranking über deren Bedeutung für den Einsatz. [Tuy08 S. 183f]

Merkmal	Erläuterung
Wiederholbarkeit	Der Detektor muss in zwei Bildern mit dem gleichen Bildausschnitt dieselben interessanten Bildregionen ermitteln.
Informationsgehalt	Die ermittelten Merkmale sollen einzigartig sein und das Bild gut charakterisieren.
Quantität	Die Anzahl der ermittelten Features soll den Anforderungen angemessen sein. Zu viele Features können nicht mehr effizient verarbeitet werden. Zu wenige Features erlauben aufgrund des geringeren Informationsgehaltes keinen eindeutigen Rückschluss auf das Bild.
Genauigkeit	Die erfassten Merkmale sollen in Lage, Umfang und ggf. Form genau lokalisiert sein.
Effizienz	Die Erkennung der Regionen soll in angemessener Zeit erfolgen, insbesondere gilt dies in zeitkritischen Anwendungen.
Invarianz	Die Detektion soll gegenüber Bildtransformationen unabhängig sein. Diese Eigenschaft ist besonders wichtig, wenn große Verformungen zu erwarten sind.
Robustheit	Gegenüber geringen Verformungen soll das Detektionsverfahren dagegen unempfindlich sein

Tabelle 3-3: Anforderungen an ideale Detektoren [Tuy08 S. 183f]

Die Entwicklung der vergangenen Jahre im Bereich der Feature-Detektion ging vorwiegend in Richtung formbasierter Detektoren. Auch die in OpenCV implementierten Verfahren arbeiten überwiegend auf der Objektform. Eine Ursache für die Präferenz für dieses Bildmerkmal ist die Tatsache, dass Formen im hohen Maße semantiktragend im Bild sind. Im Gegensatz zu Farbe und Textur beinhaltet sie deutlich mehr Informationen über den Inhalt einer Photographie. Die Betrachtung von dargestellten

Objekten ist näher an der menschlichen Denkweise und daher besser geeignet inhaltlich schwach korrelierte Bilder zu unterscheiden.

Nichtsdestotrotz spielen die übrigen Bildfeatures ebenfalls eine große Rolle bei heutigen Detektoren. Farbe und Textur werden häufig dazu eingesetzt, um formbasierte Features zu ermitteln. Bei vielen auf Farbinformationen basierenden Verfahren handelt es sich dabei um Weiterentwicklungen von Intensitätsbasierten Verfahren. Es wird ermittelt, ob die Farbwertänderung benachbarter Pixel bzw. Pixelregionen einen gesetzten Schwellwert überschreitet und über diese Information eine potentielle Objektkante erkannt. [Tuy08 S.203]

Da Detektionsalgorithmen zur Bestimmung formbasierter Features eine große Rolle in der Ermittlung lokaler Features spielen, liegt der Schwerpunkt dieser Arbeit auf diesem Bereich. Dabei erfolgt zunächst eine Einteilung in Klassen auf Grundlage ihrer Funktionsweise. Ein guter Ansatz dabei ist zu spezifizieren, welchen Teil der Form eines Bildobjektes der Detektor untersucht. Hier lassen sich drei große Gruppen differenzieren. Kanten-Detektoren ermitteln Pixel auf der Objektkontur, wohingegen Ecken-Detektoren nur die Pixel auf der Kontur bestimmen, die an markanten Eckpunkten liegen. Blob-Detektoren berechnen die Regionen (Blobs) eines Objektes. Zudem existieren noch Affin-Invariante Detektoren. Sie stellen oft eine Erweiterung etablierter Verfahren anderer Klassen dar. Ihr Schwerpunkt liegt allerdings auf dem Charakterisieren von Features, die gegenüber affinen Transformationen unveränderlich sind. Affine Transformationen umfassen jegliche maßstabsändernde Abbildungen des Bildraumes, bei denen die Verhältnisse der Bildobjekte beibehalten werden. Dadurch wird der Einfluss von Störgrößen auf den Retrieval-Prozess, wie der Betrachtungswinkel der Aufnahme, vermindert. Die folgende Grafik zeigt einen Überblick über die einzelnen Klassen und gibt Beispiele für Vertreter dieser Gruppen. Die hervorgehobenen Detektoren sind in OpenCV implementiert und werden im Laufe dieser Arbeit näher betrachtet.

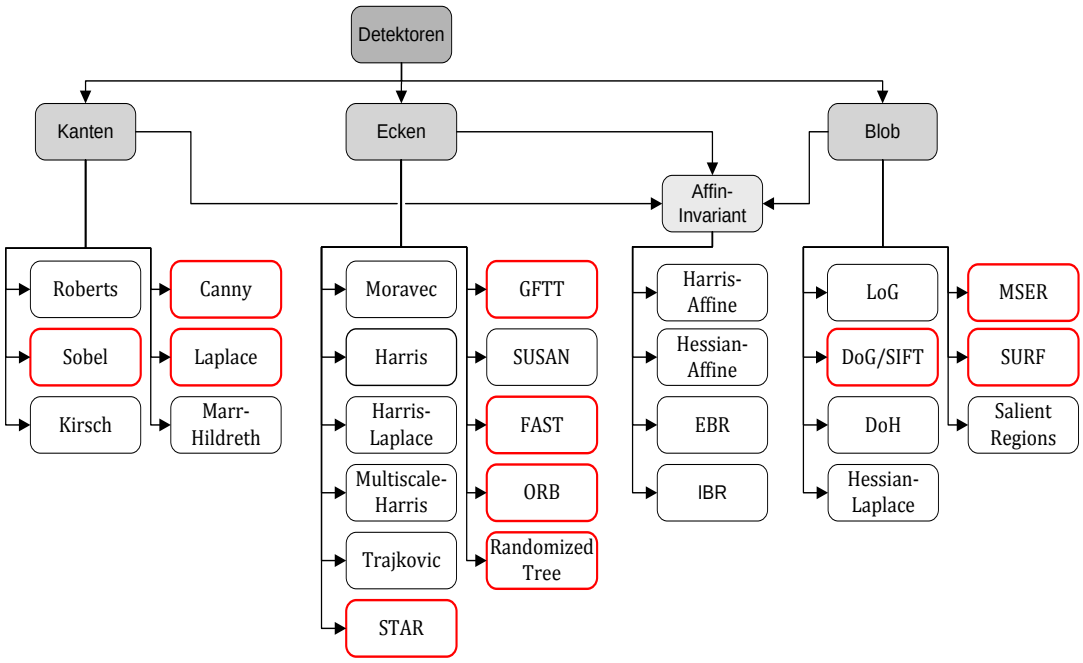

Abbildung 3-21: Einordnung lokaler Feature-Detektoren in definierte Kategorien

Nach dieser allgemeinen Einführung wird im Folgenden die Klasse der Kanten-Detektoren diskutiert.

3.3.1 Kanten-Detektoren

Ziel der Kanten-Detektion ist es mit Hilfe von Kantenoperatoren Umrandungen von Objekten, im Folgenden Kanten genannt, zu finden [Gal09 S. 8f]. Der Begriff Kante kann wie folgt definiert werden.

Definition 3.7 (Kante)
Eine Kante ist eine Diskontinuität einer Funktion der Bildintensität, welche unter anderem an Objektübergängen und Schattenverläufen auftritt [Str03 S. 3].

Viele Objektkanten sind dadurch charakterisiert, dass zwei Flächen mit unterschiedlichen Grauwerten aufeinandertreffen, wodurch die entstehende Kante per Detektionsverfahren erkannt werden kann. Kantendetektionsalgorithmen ermitteln ein Kantenpixel auf der Grundlage von Intensitätsunterschieden eines Pixels im Vergleich zu seiner direkten Nachbarschaft. Zu diesem Zweck ist es im Allgemeinen notwendig das Farbbild in ein Graustufenbild zu transformieren. Für die Helligkeitsverteilung im Bild ergibt sich die Funktion *f(u)*, wobei *u* die Betrachtungsrichtung spezifiziert. Hierbei werden die Frequenzen im Bild durch hellere und dunklere Grauwerte dargestellt. Besonders interessant sind dabei die Bereiche der größten Intensität, das heißt wo sich die Helligkeit des Originalbildes am stärksten ändert. Hierbei kann es sich um Objektkanten handeln, an denen ein Intensitätswechsel stattfindet. Dies äußert sich in der ersten Ableitung durch einen Extremwert. [Jäh05 352f]

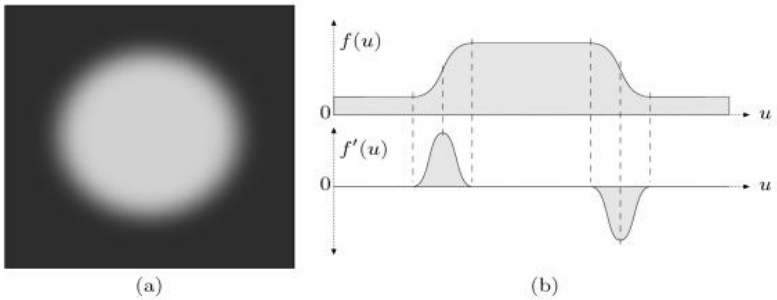

Abbildung 3-22: Originalbild (links) mit Grauwertprofil und erster Ableitung (rechts) [Bur06 S. 118]

Um die Extremwertstellen zu bestimmen, beinhaltet die Kantendetektion die Suche nach Maxima im Betrag des Gradientenvektors. Zu diesem Zweck erfolgt eine Berechnung der ersten Ableitung für jede Bildrichtung. Der ermittelte Gradient zeigt dabei stets in die Richtung ansteigender Grauwerte. Für die in Abbildung 3-22 dargestellte Grafik bedeutet dies, dass die Gradientvektoren vom Bildrand hin zur Bildmitte zeigen.

Definition 3.8 (Gradient)
Der Gradient an der Stelle (x,y) einer kontinuierlichen Funktion *f* ist ein zweidimensionaler Vektor, welcher in die Richtung des steilsten Anstieges der Bildfunktion *f* zeigt [Kir08 S. 17]

Das Ergebnis des Detektionsprozesses ist ein Kantenbild, also ein Binär- oder Grauwertbild, in welchem alle gefundenen Kanten eingezeichnet sind. Das erstellte Kantenbild dient als Ausgangspunkt für weitere Verarbeitungsschritte der Kontur-

oder Texturerkennung. Nachfolgend werden nun einige Kantenoperatoren vorgestellt, wobei detaillierter auf den Detektionsprozess eingegangen wird. Begonnen wird dabei mit dem Sobel-Operator.

Sobel-Operator

Ähnlich zu anderen kantenbasierten Detektionsoperatoren wird beim Sobel-Operator das Rasterbild als eine Matrix A betrachtet. Die Detektion basiert auf einer pixelweisen Faltung eines Paars von 3×3-Matrizen (Faltungsmatrizen) mit einem Ausschnitt der Bildmatrix. Dabei entspricht beim Sobel-Operator die zweite Faltungsmatrix einer 90°-Drehung der ersten Matrix. Bei beiden Matrizen ist zudem eine stärkere Gewichtung der Mittelachse konzipiert. Nach Sobel sind sie wie folgt definiert. [Fis03]

-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

Abbildung 3-23: Sobel-Operatoren S_x (links) und S_y (rechts) [Fis03]

Aus dieser Definition folgt, dass S_x auf vertikale und S_y auf horizontale Änderungen des Gradienten reagieren. Als Ergebnis dieser Faltung wird aus dem Originalbild das Gradienten-Bild erzeugt.

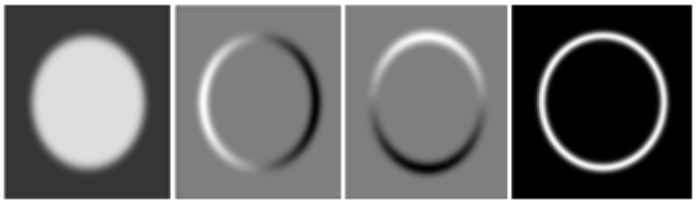

Abbildung 3-24: Originalbild (links), vertikales Faltungsresultat G_x (Mitte links), horizontales Faltungsresultat G_y (Mitte rechts), Gradienten-Bild (rechts) [Bur06 S. 120]

Der resultierende Gradientenvektor G definiert den Gradienten der Intensitätsänderung in beiden Bildachsen. Dabei ergibt sich G_x aus der Faltung von S_x mit A und G_y durch Falten von S_y und A.

$$G = \begin{pmatrix} G_x \\ G_y \end{pmatrix} \tag{3-40}$$

Mithilfe der beiden ermittelten Gradienten G_x und G_y lässt sich der Anstieg einer potentiellen Kante durch einen Pixel errechnen. Es gilt:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \tag{3-41}$$

Ein Pixel hat jedoch lediglich acht Nachbarn und dementsprechend auch nur acht mögliche Gradientenwinkel, wobei jeweils ein Winkelpaar eine lokal berechenbare Kantenrichtung bestimmt. Beispielsweise kann eine horizontale Kante durch den Winkel 0° oder 180° ausgedrückt werden. Daraus ergeben sich vier ermittelbare Winkel von Objektkanten für horizontale, vertikale und die beiden diagonalen Bildrichtungen. Diese werden durch 0°, 45°, 90° und 135° angegeben. Der soeben berechnete Gradientenwinkel wird auf einen der Werte gerundet. Um die absolute Amplitude des Gradienten zu ermitteln wird der Betrag des Vektors gebildet. [Wag06 S. 14]

$$|G| = \sqrt{{G_x}^2 + {G_y}^2} \tag{3-42}$$

Der Betrag des Gradienten ist invariant bei Bilddrehungen und damit unabhängig von der Orientierung der Bildstrukturen. Dies ist wichtig für eine isotrope Kantendetektion, wobei das Operationsresultat nicht von der Richtung der Kante abhängig ist, was auch die Grundlage anderer Kantendetektionsalgorithmen darstellt. [Jäh05 S 352f]

In der Praxis ist, um die Rechenzeit zu verkürzen, die Verwendung eines Pseudo-Faltungsoperators üblich. Die Berechnung des Gradienten erfolgt hierbei für den Anwender in nur einem sichtbaren Schritt, ohne Ausgabe der Teilgradienten G_x und G_y. Zur Erzeugung der Amplitude des Gradienten wird zu jedem Pixel seine 8-Pixel-Nachbarschaft betrachtet

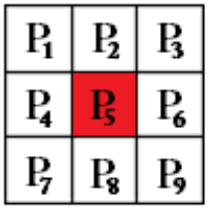

Abbildung 3-25: Pixelmatrix eines Bildausschnitts [Fis03]

Die Faltung entspricht im Frequenzbereich einer punktweisen Multiplikation der gerasterten Matrix des Bildausschnitts mit den Sobel-Operatoren (vgl. Abbildung 3-23 und Abbildung 3-25). Für die Gradienten G_x und G_y ergibt sich folgende Rechnung.

$$\begin{aligned} G_x &= -P_1 + P_3 - 2P_4 + 2P_6 - P_7 + P_9 \\ &= (P_3 + 2P_6 + P_9) - (P_1 + 2P_4 + P_7) \end{aligned} \tag{3-43}$$

$$\begin{aligned} G_y &= P_1 + 2P_2 + P_3 - P_7 - 2P_8 - P_9 \\ &= (P_1 + 2P_2 + P_3) - (P_7 + 2P_8 + P_9) \end{aligned}$$

Die Berechnung des Betrages des Gradientenvektors wird ebenfalls approximiert (vgl. Gleichung 3-42), wodurch sich folgende Rechnung ergibt.

$$\begin{aligned} |G| &= |G_x| + |G_y| \\ &= |(P_3 + 2P_6 + P_9) - (P_1 + 2P_4 + P_7)| \\ &\quad + |(P_1 + 2P_2 + P_3) - (P_7 + 2P_8 + P_9)| \end{aligned} \tag{3-44}$$

Diese Approximation ist allerdings anisotrop, wodurch Kanten in Diagonalrichtung um den Faktor $\sqrt{2}$ empfindlicher detektiert werden als Kanten in Achsenrichtung. [Jäh05 S 352f]

Allgemein gilt der Sobel-Operator in der Literatur trotz seiner Einfachheit als hinlänglich genau. Mit ihm lassen sich recht gute Retrieval-Ergebnisse in akzeptabler Rechenzeit erzielen. Nicht zuletzt diesem Umstand hat der Algorithmus es zu verdanken, dass er auch Bestandteil anderer Verfahren, wie dem Canny-Filter, ist. Im Gegensatz zu beispielsweise dem Roberts-Operator besitzt er größere Faltungsmatrizen. Dies benötigt bei der Berechnung mehr Zeit, liefert allerdings auch bessere Ergebnisse, da die Störanfälligkeit auf diese Weise herabgesetzt ist. Des Weiteren ist beim Sobel-Operator eine stärkere Gewichtung der Mittelachse als beim Prewitt-Operator spezifiziert. Diese gewichtete Mittelung dient der Rauschunterdrückung und verbessert so die Kantendetektion. Außerdem wurden lediglich zwei Matrizen definiert, wobei jede auf Änderungen in einer Bilddimension am stärksten reagiert. [Ebl11, Vis07 S. 12]

Canny-Edge-Detektor

Der Canny-Edge-Detektor umfasst einen Algorithmus zur Kantendetektion, bei dem, um ein bestmögliches Detektionsergebnis zu erzielen, verschiedene Verfahren hintereinander ausgeführt werden. Er wurde von J. Canny, mit dem Ziel einen optima-

len Kantendetektor zu entwickeln, erstellt. Dabei waren drei Kriterien besonders wichtig. Zum einen soll der Detektor lediglich tatsächlich existierende Bildkanten erkennen. Diese sollen weiterhin möglichst nah an der realen Position im Bild detektiert werden. Zuletzt legte Canny großen Wert darauf, dass Kanten nicht mehrfach identifiziert werden. Dies umfasst insbesondere, dass jede gefundene Kante genau ein Pixel breit ist. Um dies zu erreichen wurden bekannte Verfahren mit zusätzlichen Schritten der Bildverarbeitung kombiniert.

Ähnlich zum Sobel-Filter wird aus dem Originalbild im Vorfeld ein Grauwertbild erzeugt, welches die Grundlage für die weitere Intensitätsbetrachtung ist. Wie bereits erwähnt wurde, können die Helligkeitsschwankungen an Objektkanten als Unstetigkeit der Grauwertfunktion aufgefasst werden. Da derartige Unstetigkeitsstellen auch bei Bildrauschen auftreten, verwendet der Canny-Algorithmus einen Gauß-Filter zum Glätten des Bildes. Dabei wird das Originalbild mit Hilfe einer Matrix gefaltet, um sich der Normalverteilung anzunähern und das Rauschen zu unterdrücken. Der neue Grauwert der Pixel ergibt sich dann aus den gewichteten Werten der ihn umgebenden Pixel. Im Allgemeinen gilt, je größer die verwendete Faltungsmatrix gewählt wird, desto robuster wird der Algorithmus gegenüber Rauschen. Ein Beispiel für eine solche Maske ist in der folgenden Gleichung dargestellt. [Fis03, Wag06 S. 13]

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{3-45}$$

Anschließend werden die Gradienten der einzelnen Pixel unter Verwendung des betrachteten Sobel-Operators ermittelt. Der nächste Schritt sieht die Entfernung aller Grauwerte vor, die nicht auf einer Objektkante liegen. Dafür wird für jeden Pixel sein Grauwert mit denjenigen seiner 8 Nachbarn verglichen. Bei diesem Vergleich darf keiner der benachbarten Pixel einen höheren Wert aufweisen, es sei denn, der betreffende Nachbarspixel liegt entlang der zuvor berechneten Kantenrichtung. Ist dies nicht gegeben, wird der Grauwert des betrachteten Pixels auf null gesetzt. Diese Methode heißt non-maximal suppression und dient der genaueren Lokalisierung der Kanten. Somit wird die geforderte Bedingung nach minimal dünnen Kanten eingehalten.

Zusätzlich wird eine Hysterese, ein Zwei-Schwellwerteverfahren zur Unterdrückung nicht relevanter Kanten, durchgeführt. Neben der resultierenden Relevanzbewertung werden in diesem Schritt auch die Helligkeitswerte aller Bildpunkte auf Schwarz oder Weiß gesetzt. Die zwei verwendeten Schwellwerte T_1 und T_2 dienen der Beeinflussung des Detektionsprozess. Dabei ist der erste Schwellwert für die Beurteilung, ob ein Pixel relevant ist oder nicht, wichtig. Der zweite Wert dient hingegen der Vermeidung von Kantenbrüchen. Beinhaltet eine Kante überwiegend Pixel mit einem hohen Grauwert, so gilt sie als relevant, und darf auch an Stellen mit niedrigerem Grauwerten nicht auseinander gerissen werden. Der Tracking-Prozess beginnt in einem Pixel, dessen Wert größer als T_1 sein muss. Entlang der berechneten Kanten werden in beiden Richtungen ausgehend von diesem Punkt alle Pixel markiert, bis deren Grauwert unterhalb T_2 fällt. Nach diesem Schritt liefert der Algorithmus eine Menge von Pixeln, welche den im Bild vorhandenen Kanten entspricht. Im Gegensatz zu vielen anderen Detektoren ist das Detektionsresultat ein Binärbild, welches lediglich die Pixelwerte 0 (Schwarz) und 1 (Weiß) beinhaltet. Die Abbildung 3-26 zeigt eine Übersicht über verschieden Zwischenstufen des Detektionsprozesses. [Wag06 S. 14-17]

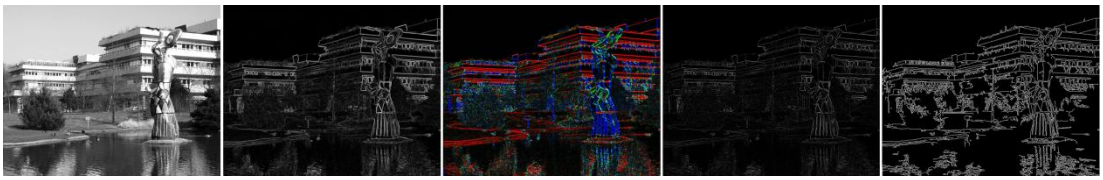

Abbildung 3-26: Originalbild, Kantenbild, Kantenbild mit Gradientenrichtungen, Kantenbild nach non-maximal suppression, Ergebnis des Canny-Filters (v.l.n.r.) [Wag06]

Das berechnete Kantenbild kann auf viele verschiedene Arten verwendet werden, um weitere Informationen aus dem Bild zu extrahieren. Bei der Hough-Transformation wird es zur Erkennung einfacher geometrischer Objekte und beim Waltz-Algorithmus zur Erkennung von dreidimensionalen Objekten im Bild verwendet. Die Wirkung des Canny-Algorithmus hängt von drei Parametern ab. Die Breite des Gauß-Filters bestimmt den Grad der Weichzeichnung. Eine Erhöhung der Filterbreite senkt die Sensitivität des Detektors gegenüber Rauschen, aber vergrößert im gleichen Maße auch die Wahrscheinlichkeit Bilddetails zu verlieren. Außerdem wird der Lokalisierungsfehler der ermittelten Kanten so leicht erhöht. Ebenso wichtig für das Retrieval-Ergebnis ist die Wahl des oberen und unteren Schwellwertes bei der Hysterese. Falls die untere Schwelle zu hoch gesetzt ist, erhöht sich das Bildrauschen. Ist dagegen der obere Schwellwert zu gering konzipiert, vergrößert sich die Anzahl falsch detektierter Objektkanten. Allgemein gilt, für gute Ergebnisse sollte der untere Parameter möglichst tief und der obere hoch angesetzt werden. [Fis03]

Laplace-Operator

Im Gegensatz zum Sobel-Operator handelt es sich beim Laplace-Operator um einen richtungsunabhängigen Hochpassfilter. Er arbeitet sowohl in vertikaler, horizontaler als auch in diagonaler Richtung und basiert auf der zweiten Ableitung der Bildfunktion. Die bisher betrachteten Verfahren rechnen jeweils mit der ersten Ableitung, also dem Gradient der Bildfunktion. Die zweite Ableitung einer Funktion besitzt an einer Extremstelle der ersten Ableitung einen Nulldurchgang, was für die Kantendetektion genutzt werden kann. Der klassische Laplace-Operator wird in der Analysis wie folgt definiert. [Vis07 S. 8, Wag06 S. 10]

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{3-46}$$

Der Operator ist null, wenn beide partiellen Ableitungen null werden, was ein Indiz für eine mögliche Kante an dieser Stelle ist. Um ihn für das Retrieval zu nutzen, muss die Formel aus (3-46) in Matrizenschreibweise überführt werden. Hierfür werden zuerst die zweiten partiellen Ableitungen folgendermaßen diskretisiert.

$$\begin{aligned} \frac{\partial^2 f(x,y)}{\partial x^2} &\approx \frac{\partial\big(f(x+1,y) - f(x,y)\big)}{\partial x} \\ &\approx f(x+1,y) - f(x,y) - \big(f(y,x) - f(x-1,y)\big) \\ &= 1f(x-1,y) - 2f(x,y) + 1f(x+1,y) \end{aligned} \tag{3-47}$$

Diese Rechnung wird analog für die zweite Koordinate durchgeführt und anschließend werden die beiden Faltungsmatrizen konstruiert.

$$H^L = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3-48}$$

Der Filter liefert akzeptable Ergebnisse, allerdings ist er stark rauschanfällig. Der Grund hierfür liegt in der punktförmigen Struktur des Filters, welche den mittleren Bildpunkt im Gegensatz zu seinen Nachbarn stärker gewichtet. Dadurch wird punktförmiges Rauschen eher verstärkt als geglättet. Zusätzlich ist bei der Kantendetektion mittels Laplace-Filter eine anschließende Suche nach einem Vorzeichenwechsel möglich. Dabei werden für jedes Pixel alle Paare von sich gegenüberliegenden Nachbarpixeln betrachtet. Besitzt mindestens ein Paar davon unterschiedliche Vorzeichen, markiert man den aktuellen Pixel als Kantenpixel, ansonsten als Nicht-Kantenpixel. Bei dieser Vorgehensweise enthält man wie beim Canny-Filter ein Binär-, anstatt des sonst üblichen Graustufenbildes. In der Praxis wird meist auf diese Erweiterung verzichtet, da durch die hohe Rauschanfälligkeit des Laplace-Filters die Ergebnisse durch eine Vorzeichenbetrachtung eher verschlechtert als verbessert werden. Stattdessen wird häufig eine abgewandelte Filtermatrix eingesetzt (vgl. Gleichung 3-49). Diese zeichnet sich durch eine bessere Isotropie, also einen geringeren Einfluss der Kantenrichtung auf das Detektionsergebnis aus. [Wag06 S. 11]

$$\hat{H}^L = \begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{3-49}$$

3.3.2 Ecken-Detektoren

Die Erkennung von Ecken ist Teil der Point of Interest-Detektion (POI), aber keinesfalls mit ihr gleichzusetzen, da noch weitere Ausprägungen zu der Gruppe der Points of Interest zählen. Hierbei ist beispielsweise ein isolierter Punkt, welcher sehr hell bzw. dunkel ist, oder das Ende einer Linie zu nennen. Das Ziel von POI-Detektionsverfahren ist es interessante Punkte zu ermitteln. Ein Punkt gilt im Sinne des Retrievals als interessant, wenn er sich von seiner Umgebung durch beispielsweise eine hohe Helligkeit oder eine starke Kontraständerung abhebt. Eine Ecke ist ein besonderer Point of Interest, an dem sich zwei Kanten (vgl. Definition 3.7) schneiden.

Definition 3.9 (Ecke)
Im zweidimensionalen Raum bezeichnet eine Ecke einen Punkt, in dessen unmittelbarer Nachbarschaft zwei dominante Kanten in verschiedenen Richtungen existieren. [Gal09 S. 10]

Allgemein sollte ein Eckpunkt folgende Eigenschaften erfüllen. Ihr Informationsgehalt sollte möglichst groß sein, um nützlich für den Retrieval-Prozess zu sein. Dazu zählt unter anderem die Einzigartigkeit des detektierten Punktes. Nach Möglichkeit sollten die ermittelten Pixel semantische Informationen über das Bild und die dargestellten Bildobjekte enthalten. Ein zweites wichtiges Kriterium betrifft die Anzahl der berechneten interessanten Bildpunkte. Diese sollte möglichst gering sein, um das Bild in komprimierter, semantisch reicherer Form zu repräsentieren und späteren Speicher- und Rechenaufwand bei Vergleichsoperationen zu minimieren. Zuletzt soll die Auswahl für alle Anwender wiederholbar und nachvollziehbar sein. Insbesondere gilt dies auch bei Bildänderungen, gegenüber denen die detektierten Punkte invariant sein sollten. [Gal09 S. 10f]

Im Folgenden werden einige Detektionsverfahren aus der Gruppe der Eckenerkennung diskutiert.

Good Features to Track (GFTT)

Good Features to Track (GFTT) umfasst verschiedene Ansätze, um das Tracking von Objekten in Videos zu verbessern. Das Verfahren wurde 1993 von Jianbo Shi und Carlo Tamasi vorgestellt und ist in der Literatur auch unter dem Namen Shi-Tomasi Corner Detector bekannt. GFTT basiert im Wesentlichen auf der Idee die Features an den Video-Tracker anzupassen. Hierfür werden von Shi und Tomasi verschiedene Ansätze diskutiert und experimentell gezeigt, dass so bessere Ergebnisse erzielt werden können. Da sich der zugrundeliegende Algorithmus in erster Linie auf bewegte Bilder (Videos) bezieht, werden hier nur einige Grundideen angesprochen und für einen tieferen Einblick auf das Originalpaper von Shi und Tomasi verwiesen [Schi93].

Grundsätzlich gilt, dass sich Features, welche gut für das Feature-Tracking geeignet sind, in vielen Punkten von Features anderer Anwendungen unterscheiden. Features, die eine geeignete Beschreibung der Textur einer Region darstellen, können ungeeignet zum Tracking sein. Zum Beispiel ist es bei Bewegungen im dreidimensionalen Raum möglich, dass das Objekt, auf welches sich ein Feature bezieht, durch ein weiteres Objekt verdeckt wird. Deshalb ist es notwendig, die Features und den Feature-Detektionsprozess den neuen Anforderungen anpassen. Der erste Ansatz zur Verbesserung sieht deshalb eine Qualitätsüberwachung (Monitoring) der errechneten Features vor. Dabei wird über ein Unähnlichkeitsmaß die Veränderung der Werte zwischen dem ersten und dem aktuellen Frame berechnet. Überschreitet dieser Wert eine gesetzte Grenze, charakterisiert dies ein schlechtes Feature und es wird ein neuer Eigenschaftswert ermittelt.

Des Weiteren erkannten Shi und Tomasi, dass im Videobereich komplexe Bildänderungen stattfinden, die über einfache Rotation, Translation und Skalierung hinaus gehen. Eine Kernaussage schlussfolgert daraus, dass zwei Bewegungsmodelle notwendig sind, um die Bildänderung zu beschreiben. Bei kleinen Interframe-Änderungen liefert die Bildtranslation brauchbare Ergebnisse. Liegt zwischen den beiden betrachteten Frames allerdings eine große Distanz, so beschreibt das Modell der affinen Bildänderungen die Bewegung deutlich besser. Die komplexe Änderung der Bildintensität bei jedem Frame kann wie folgt berechnet werden, wobei die Variation der Bildfunktion eines bestimmten Pixel *(x,y)* zwischen einem Frame zum Zeitpunkt *t* und einem Frame zur Zeit $t+\tau$ betrachtet wird. [Schi93]

$$I(x, y, t + \tau) = I(x - \xi(x, y, t, \tau), y - \eta(x, y, t, \tau)) \tag{3-50}$$

Die Bildänderung wird dabei durch die Verschiebung $\delta=(\xi,\eta)$ des Pixels *(x,y)* vom Frame *t* zum Frame $t+\tau$ angegeben. Meist handelt es sich hier um eine Verschiebung eines Bildausschnitts (Fenster) im dreidimensionalen Raum, welche durch ein affines Bewegungsfeld in Abhängigkeit der Verschiebung des Fensterzentrums *d* angegeben wird.

$$\delta = Ax + d \quad mit \quad A = \begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix} \tag{3-51}$$

Die Qualität der Beschreibung hängt von der Größe des Fensters, der Bildtextur innerhalb des Fensters und dem Betrag der Kamerabewegung zwischen zwei Frames ab. Allerdings beschreibt auch das affine Bewegungsmodell die Änderungen im Bild nicht perfekt, wodurch die Parameter *A* und *d* so bestimmt werden müssen, dass sie die

Unähnlichkeit minimieren. Die besten Resultate für das Tracking bieten POI. Im Gegensatz zu Kanten, bei denen je nach Kantenart, waagerecht oder senkrecht, lediglich die vertikale bzw. horizontale Komponente Bewegungsinformationen beinhaltet, sind Ecken eindeutig zu detektieren. Die Bestimmung erfolgt über die Eigenwerte der Bildänderungsmatrix, welche in Ecken besonders groß sind. In der Praxis wird dies über einen Schwellwert umgesetzt, über welchen beide Eigenwerte liegen müssen. [Schi93]

Die Methode der Good Features to Track umfasst insgesamt drei Ansätze zur Verbesserung von Objekt-Tracking in Videos. Dies beinhaltet eine Methode zur Feature-Auswahl, einen Tracking-Algorithmus basierend auf affinen Bildänderungen und eine Überwachungstechnik der Features, um stets mit guten Merkmalswerten zu arbeiten. Die Überwachung der Features mit Hilfe des Ähnlichkeitswertes behebt nicht alle Probleme beim Tracking. Durch eine gute Beschreibung können so aber die Features, bis auf wenige Ausreißer, für den Tracking-Prozess verbessert werden. Eine Überlegung wäre nun, ob diese Verbesserung nicht auch auf das einfache CBIR übertragbar ist. Anstatt der Unähnlichkeit zweier Frames könnte diese auch für zwei Bilder bestimmt werden. Der Tracker könnte eingesetzt werden, um bestimmte Strukturen oder Muster im Bild zu erkennen. So kann zum Beispiel die Suche nach speziellen Architekturen oder eine Gesichtserkennung realisiert werden. Auch dieser Ansatz wurde untersucht und festgestellt, dass der Unähnlichkeitswert für verschiedene Bilder wenig aussagekräftig ist [Shi93]. Im Bereich des Bild-Retrievals existieren bereits effektivere Detektionsverfahren.

Features from Accelerated Segment Test (FAST)

Features from Accelerated Segment Test (FAST) ist ein Verfahren, welches von Edward Rosten und Tom Drummond entwickelt wurde und bei dem der Schwerpunkt nicht auf der möglichst genauen Bestimmung der Ecken in einem Bild liegt, sondern auf der Rechenzeit, die hierfür benötigt wird. Um eine Zeiteinsparung zu ermöglichen, wird bewusst eine herabgesetzte Qualität bei der Eckenerkennung akzeptiert, welche sich unter anderem in einer ungenauen Lokalisierung der Ecken zeigt. Die auf diese Weise erkaufte Einsparung der Rechenzeit macht FAST zu einem der momentan schnellsten Feature-Detektoren. Die folgende Tabelle zeigt einen Vergleich von FAST mit anderen Ecken-Detektoren. [Ros05 S. 6, Ros06 S. 4f]

Detektor	Zeit (in ms)	Anteil an Framedauer (in %)
FAST (mit non-max suppression)	1.59	7.95
FAST(ohne non-max suppression)	1.49	7.45
Harris	24.0	120.0
SUSAN	7.58	37.9
DoG	60.1	301.0

Tabelle 3-4: Laufzeitvergleich verschiedener Ecken-Detektoren [Ros06 S. 7]

Bei diesem Test wurde die Zeit gemessen, welche die Verfahren benötigen, um auf einem AMD Opterion 2.6GHz Features in einem Farbbild (768x288 Pixel) zu detektieren. Zusätzlich wurde die Rechenzeit ins Verhältnis zu der zur Verfügung stehenden Dauer eines Frames beim PAL-Fernsehen (20ms) gesetzt, um so die Tauglichkeit für Echtzeit-Trackingsysteme zu untersuchen, wobei FAST nach diesem Test sehr gut geeignet ist.

Ähnlich zu anderen POI-Detektionsverfahren erfolgt die Auswahl eines Pixels bei FAST auf Grundlage der Intensitätsunterschiede des Pixels zu seiner Umgebung. Für

jedes Pixel *p* wird die Nachbarschaft auf einem 16-Pixel-Kreis (Bresenham-Kreis mit Radius 3) um *p* betrachtet. Pixel *p* wird vom Detektor ausgewählt, wenn mindestens *n* seiner Nachbarn eine um den gewählten Schwellwert *t* geringere oder höhere Intensität als *p* besitzen. Gebräuchlich ist *n=12* als Anzahl der nötigen Nachbarn zu wählen. [Ros05 S. 6]

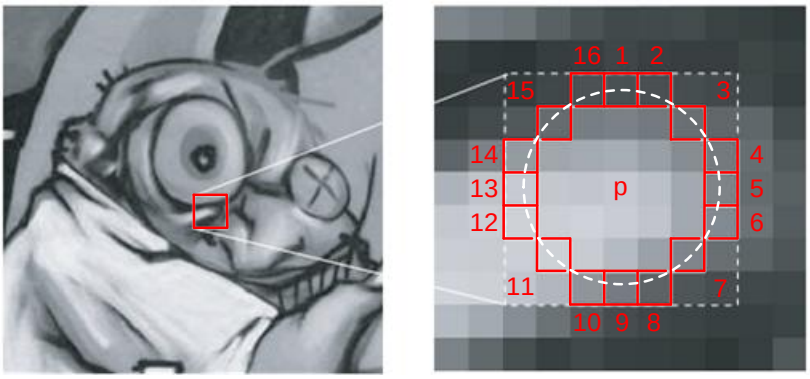

Abbildung 3-27: Pixel-Segment-Test zur Auswahl von Ecken bei FAST [Ros05 S. 6]

Dieser Test kann optimiert werden, indem zuerst nur einige Nachbarpixel getestet werden. Hierfür werden die Pixel 1, 5, 9 und 13 auf den Bildhauptachsen von *p* gewählt. Liegt die Intensität nicht bei mindestens drei dieser Pixel über bzw. unter der von *p*, so kann der Kandidat *p* direkt ausgeschlossen werden. Lediglich für die Bildpunkte, welche diese erste Vorauswahl absolviert haben, wird die komplette 16-Pixel-Nachbarschaft betrachtet. Über den Schwellwert *t* kann bei diesem Verfahren die Geschwindigkeit und die Anzahl der detektierten Features beeinflusst werden. Dabei nimmt die Berechnungszeit zu und die Anzahl der gefundenen Ecken ab, wenn *t* erhöht wird (vgl. Abbildung 3-28).

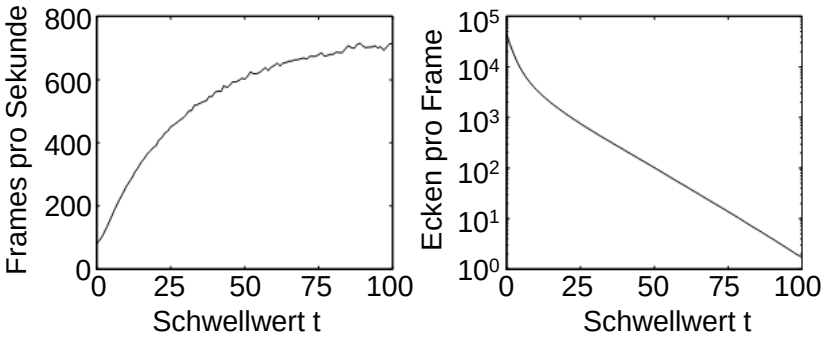

Abbildung 3-28: Beeinflussung der Rechenzeit und der Anzahl der detektierten Ecken durch den Schwellwert *t* [Ros05 S. 6]

Die Detektionsergebnisse können zusätzlich durch die Anwendung der non-maximal suppression verbessert werden. Dabei wird zu jeder detektierten Ecke geprüft, ob diese eine stärker ausgeprägte adjazente Ecke besitzt. Falls dies der Fall ist, wird der betrachtete Pixel verworfen. Die Überprüfung auf adjazente Ecke geschieht mittels einer Vergleichsfunktion *V*, auf welche anschließend die non-maximal suppression durchgeführt werden kann. [Ros06 S. 6]

$$V = max\left(\sum_{x \in S_{bright}} \left| I_{x \to p} - I_p \right| - t \, , \sum_{x \in S_{dark}} \left| I_p - I_{x \to p} \right| - t \right) \tag{3-52}$$

mit

$$S_{bright} = \{x | I_{x \to p} \geq I_p + t\}, \qquad S_{dark} = \{x | I_{x \to p} \leq I_p - t\}$$

V entspricht dabei der Summe der absoluten Differenz zwischen der Intensität des Pixels *p* und seiner sechssehn Kreisnachbarn. Die Nachbarschaft wird in die Mengen S_{bright} und S_{dark} unterteilt, je nachdem ob die Intensität des Nachbarpixels höher oder niedriger als die von *p* ist.

Eine weitere Detektor-Deskriptor-Kombination, welcher auf dem FAST-Detektor aufbaut, ist oriented BRIEF (ORB). Diese besteht aus einem orientierten FAST-Detektor zum Finden interessanter Bildbereiche und einer modifizierten Version des BRIEF-Deskriptors (vgl. Kapitel 3.4). Der Detektor, welcher bei ORB verwendet wird, unterscheidet sich lediglich durch die zusätzliche Berechnung einer Orientierungsrichtung an den detektierten Ecken vom eingeführten FAST-Detektor. Hierzu werden an jedem Eckpunkt verschiedene Momente des Intensitätswertes der Ecke anhand folgender Formel berechnet. Trotz des zusätzlichen Rechenaufwands ist ORB um bis zu hundert Mal schneller als SIFT und etwa zehn Mal schneller als SURF, was in der Wahl von FAST als Basis der Detektion begründet liegt. [Bra11 S. 42-46]

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \tag{3-53}$$

Die Orientierung ergibt sich aus der Orientierung in x-Richtung und in y-Richtung, welche mittels der Gleichung 3-54 berechnet werden kann.

$$C_{ori} = \tan^{-1}\left(\frac{c_y}{c_x}\right) \tag{3-54}$$

mit

$$c_x = \left(\frac{M_{10}}{M_{00}}\right) \qquad , \qquad c_y = \left(\frac{M_{01}}{M_{00}}\right)$$

FAST wird vor allem bei zeitkritischen Anwendungen eingesetzt, da es anderen gängigen Ecken-Detektionsverfahren im Bezug auf die Rechenzeit deutlich überlegen ist. Neben der dadurch höheren Ungenauigkeit bei der Ecken-Detektion beinhaltet FAST den weiteren Nachteil, dass es nicht skalierungsinvariant ist. [Tuy08 S 252]

Star-Detektor (STAR)

Der Star-Detektor wurde von dem US-amerikanischen Unternehmen für Robotertechnologie Willow Garage, welches derzeit auch die OpenCV-Bibliothek pflegt (vgl. Kapitel 3.5), entwickelt. Er resultiert aus dem Versuch zwei zentrale Ziele der Feature-Detektion zu kombinieren. Auf der einen Seite sollen die detektierten Eckpunkte möglichst nah an den realen Ecken liegen, auf der anderen Seite ist auch eine hohe Robustheit bei der Detektion wichtig. Eine hohe Genauigkeit besitzen auch andere eckenbasierte Verfahren wie der Harris-Detektor. Ihr Schwachpunkt ist dagegen die Robustheit insbesondere gegenüber Skalierungsänderungen. Im Gegenzug besitzen skalierungsinvariante Feature-Detektoren meist Defizite bei der Genauigkeit, da sie nicht auf den einzelnen Bildpixeln arbeiten, sondern auf abstrahierten Objektformen. Beide Kriterien spielten bei der Entwicklung von STAR eine Rolle. Der Detektor ist bei einer hohen De-

tektionsgenauigkeit invariant gegen Bildänderungen und zudem durch effiziente Berechnungen für Echtzeitsysteme geeignet. [Agr08 102-104]

Das Grundprinzip des Star-Detektors stellt eine möglichst genaue und schnell zu berechnende Approximation an bereits bestehende, digitale Filterverfahren dar. Die Basis bildet dabei der Center Surround Extremas Filter (CenSurE), welcher wiederum eine verallgemeinerte Variante der Extrempunktsuche nach Laplace darstellt. Diese ist sehr stabil gegenüber Skalierung und wird aus diesem Grund auch bei anderen Detektoren, wie dem Harris-Laplace und dem Hessian-Laplace, eingesetzt. Bei beiden Beispielen wird nach der Ecken-Detektion eine Laplace-Filterung durchgeführt, was sehr rechenaufwändig ist. STAR führt dagegen zuerst die CenSurE-Filterung über alle Bildpunkte aus und eliminiert anschließend durch verschiedene Verfahren einige der gefundenen Features. Ein Censur-Filter ist ein bipolares Filter (bi-level filter), da es aus einem inneren und einem äußeren Kern mit entgegengesetzter Polarität besteht. Für jeden Pixel werden die Grauwerte seiner Nachbarn im inneren Kern mit +1 und die Nachbarschaft im äußeren Kern mit -1 multipliziert. Die Summe dieser Werte bildet das Feature des Pixels, welcher sich an Ecken stark von denen seiner Nachbarpixel unterscheidet. Das Hauptproblem bei der Center-surround Filterung ist es, ein Filter zu entwerfen, der skalierungs- und rotationsinvariant ist und gleichzeitig schnell zu berechnen ist. Die Abbildung 3-29 zeigt eine Übersicht über mögliche Filterkerne.

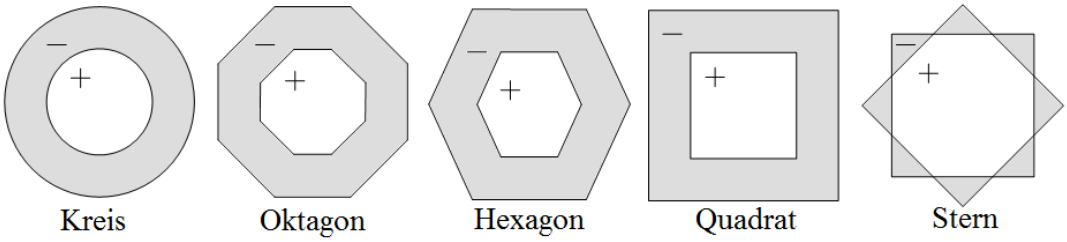

Abbildung 3-29: Center-Surround bi-level Filter [Agr08 S. 106]

Der Kreisfilter liefert bei der Laplace-Filterung die besten Ergebnisse, ist im Gegenzug aber auch bei der Berechnung am zeitaufwändigsten und deshalb für zeitkritische Anwendungen nicht geeignet. Ein quadratischer Filterkern wäre dagegen einfach zu berechnen. Allerdings sind diese Filter nur gegenüber 90°-Drehungen invariant. Eine gute Approximation des Kreisfilters ist der Oktagon-Filterkern. Er ist relativ robust gegenüber Drehungen und trotzdem schnell zu berechnen. Beim Star-Detektor wird eine Variante des Oktagon-Filters, der Stern, genutzt.

Um Skalierungsinvarianz zu erreichen, muss die Filterung auf verschiedenen Skalierungsebenen erfolgen. Bei STAR wird standardmäßig ein siebenstufiger Skalierungsfaktor für die Filtermasken verwendet und der Merkmalswert für jeden Bildpunkt auf allen Skalierungen berechnet. Dies ist bei quadratischen Filtern durch die Verwendung von Integralbildern effizient berechenbar. Integralbilder dienen der schnellen Berechnung von Pixelsummen innerhalb rechteckiger Bildbereiche. Der Wert eines Pixels in einem Integralbild $I_{\Sigma}(x,y)$ entspricht der Summe aller Pixel der Nachbarschaft im Originalbild. Für einen Bildausschnitt wird es folgendermaßen berechnet.

$$I_{\Sigma}(x) = \sum_{i=0}^{i \le x} \sum_{j=0}^{j \le y} I(i,j) \tag{3-55}$$

Nachdem das Integralbild ermittelt wurde, sind lediglich vier Operationen notwendig, um eine rechteckige Fläche beliebiger Größe zu berechnen. Die schnelle Be-

rechnung ermöglicht Bildskalierungen unabhängig von der Größe des Integralbildausschnitts, was in der folgenden Abbildung dargestellt ist. Komplexere Filterkerne lassen sich durch mehrere geneigte quadratische Formen zusammensetzen und so ebenfalls über modifizierte Integralbilder berechnen. [Tuy08 S. 248f]

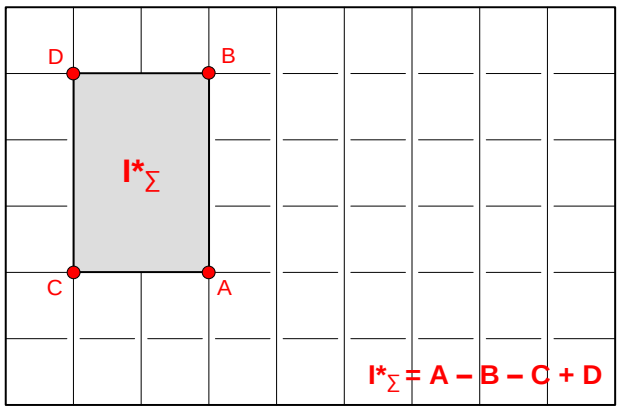

Abbildung 3-30: Berechnungsschema von Integralbildern mit rechteckiger Grundfläche [Tuy08 S. 249]

Nach der Filterung wird eine non-maximal suppression angewandt. Dies dient der Suche nach skalierungsinvarianten Extremwerten in der lokalen Nachbarschaft und stellt eine erste Auswahl der detektierten Ecken dar. Dabei wird ein Wert unterdrückt, wenn in seiner Nachbarschaft ein anderer Wert höher (maximaler Fall) bzw. niedriger (minimaler Fall) auf einer Skalierungsebene ist. Die Amplitude der Werte dient zudem als Maß für die Stabilität eines Features bei der Skalierung. Die nachfolgende Abbildung zeigt die Detektion von Pixeln, die in ihrer Nachbarschaft ein Maximum oder Minimum darstellen. [Agr08 S. 105-108]

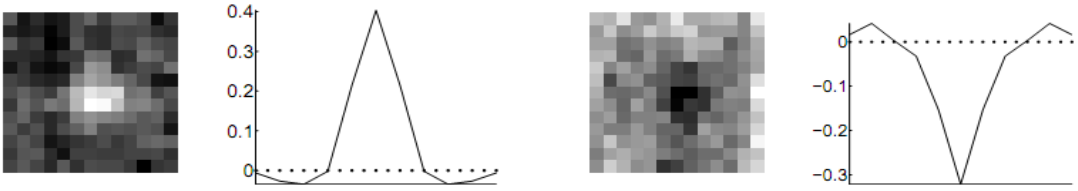

Abbildung 3-31: non-maximal suppression beim Star-Detektor [Kie07 S. 207]

Der letzte Schritt bei der Star-Detektion sieht die Eliminierung von Features vor, welche auf einer Linie oder Kante liegen, da diese bei Änderung des Bildausschnittes meistens nicht stabil sind. Im Gegensatz zu SIFT, bei welchem ein Hessian-Filter verwendet wird, nutzt STAR hierzu einen Harris-Kantenfilter. Dieser ist rechenintensiver als Hessian, liefert aber bessere Ergebnisse bei der Kantenunterdrückung. Der Mehraufwand bei der Berechnung entspricht allerdings nur einem kleinen Teil der Gesamtrechenzeit, da durch vorhergehende Berechnungen bereits viele Punkte als Ecken ausgeschlossen werden konnten und die Harris-Filterung nur für relativ wenige Features durchgeführt werden muss.

STAR ist momentan der einzige existierende Ecken-Detektor, der volle räumliche Auflösung auf allen Skalierungsebenen ermöglicht. Am nächsten daran sind SIFT und SURF. Sie suchen aber jeweils nur stichprobenartig, was eine sinkende Genauigkeit auf höheren Skalierungsebenen zur Folge hat. Eine volle Suche würde bei beiden Verfahren zu viel Rechenzeit in Anspruch nehmen und die Detektion ineffizient werden lassen.

Die folgende Tabelle vergleicht STAR, SIFT und SURF. Auf SIFT und SURF wird im folgenden Kapitel 3.3.3 genauer eingegangen. [Agr08 S. 104-108]

Detektor	Räumliche Auflösung über Skalierung	Skalierungs-methode	Kanten-Filter	Rotations-invariant
STAR	voll	CenSurE	Harris	approximiert
SIFT	stichprobenhaft	DoG	Hessian	ja
SURF	stichprobenhaft	DoB	Hessian	nein

Tabelle 3-5: Vergleich von STAR, SIFT und SURF [Agr08 S. 104]

Des Weiteren hat Willow Garage die Leistungsfähigkeit von STAR in zahlreichen Messungen mit der von SIFT und SURF verglichen. Die Ergebnisse dieser Tests können in grafischer Form auf der Internetseite des Unternehmens[2] eingesehen werden. Dabei zeigt STAR eine durchschnittlich 10-20% höhere Reproduzierbarkeit bei Änderungen des Betrachtungswinkels. Bei Bilddrehungen ist dagegen der rotationsinvariante SIFT-Detektor besser im Test. Er zeigt eine ca. 30% bessere Wiederholbarkeit. STAR ist hier nur dem SURF-Detektor leicht überlegen. Auf einem Niveau sind SURF und STAR ebenfalls bei Skalierungsänderungen. SIFT hat hier bei einem höheren Skalierungsfaktor leichte Defizite. Neben der hohen Robustheit gegenüber Bildänderungen spricht vor allem die schnelle Berechnung der Eigenschaftswerte beim Star-Detektor für einen Einsatz in zeitkritischen Systemen. Willow Garage setzt ihren Star-Detektor selbst bei der Steuerung von Outdoor-Robotern ein.

Randomized tree

Bei randomized trees handelt es sich nicht um einen Feature-Detektor, sondern um eine Datenstruktur zum Lösen multidimensionaler Entscheidungsprobleme, welche bei der Detektion eingesetzt werden kann. Die Grundidee ist, die Features so zu speichern, dass einerseits verschiedene Bilder schnell verglichen werden können und andererseits die extrahierten Features robust gegenüber unterschiedliche Bildtransformationen sind. Da der randomized tree häufig im Zusammenhang mit eckenbasierten Detektoren verwendet wird, wurde er in diese Kategorie eingeordnet. Der Einsatz mit Ecken-Detektoren ist allerdings nicht zwingend erforderlich. Der Hauptanwendungsbereich dieses Verfahrens liegt in der Erkennung zuvor analysierter Objekte. Bei der Verwendung von randomized tree existieren keine festen Richtlinien wie die Features zu bestimmen sind oder anhand welcher Testbedingungen das Entscheidungsproblem zu lösen ist. Aus diesem Grund wird im Folgenden eine mögliche Variante der Nutzung besprochen, welche sich mit der Erkennung von Bildobjekten befasst. [Lep06]

Der eigentliche Detektionsprozess teilt sich bei der Verwendung von randomized trees in zwei Phasen auf. In einer Trainings- oder Lernphase wird zuerst ein solcher Baum erstellt. Hierfür werden verschiedene Features eines gewählten Objektes anhand gesetzter Testbedingungen innerhalb des Baumes angeordnet. Zur Bestimmung der Features wird das Baumverfahren mit anderen Extraktionsmethoden kombiniert. Gut eignen sich zu diesem Zweck auch Regionen-basierte Detektionsalgorithmen wie SIFT. Bevor die extrahierten Features allerdings im Baum abgelegt werden, erfolgt eine Eliminierung instabiler Merkmalswerte. Ein Beispiel für variante Features ist ein Pixel p, dessen Grauwert ähnlich zu der Intensität zweier Nachbarpixel n_1 und n_2 ist, welche sich diametral auf einer Kreisbahn um p befinden. Die folgende Abbildung zeigt einen

[2] http://pr.willowgarage.com/wiki/Star_Detector

Ausschnitt aus der Lernphase, in welcher Features für ein zu detektierendes Buch ermittelt werden. [Lep06 S. 10-14]

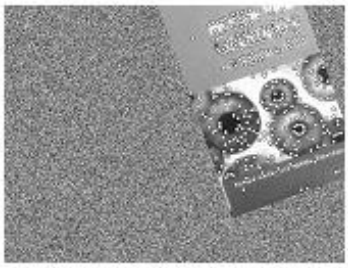

Abbildung 3-32: Extrahierte Features für ein Buchcover in der Lernphase des randomized tree-Verfahrens [Lep06 S. 6]

Nach der Trainingsphase ermöglicht der erstellte Entscheidungsbaum in der Arbeitsphase eine Objekterkennung des erfassten Gegenstandes unabhängig von Skalierung, Translation und Rotation. Die Abbildung 3-33 zeigt die Erkennung des Buchcovers von verschiedenen Blickwinkeln.

Abbildung 3-33: Erkennung des Buchcovers in der Arbeitsphase des randomized tree-Verfahrens [Lep06 S. 2]

Gewöhnliche Entscheidungsbäume enthalten in jedem inneren Knoten einen Test wodurch der Datenraum gesplittet wird. Beim Klassifizieren eines Features wird der gesamte Baum durchlaufen und an jedem Knoten ein elementarer Test durchgeführt, der entscheidet, ob in die eine oder andere Richtung weitergelaufen wird. Um eine Klassenzugehörigkeit zu ermitteln muss für alle Features der gesamte Baum durchsucht werden. Dies ist ineffizient und für eine Echtzeitsuche nicht geeignet. Beim randomized tree wird dieses Problem durch das Aufspannen mehrerer Bäume gelöst. Jeder Baum deckt dabei einen anderen Bildbereich ab. Anschließend werden alle Teilbäume kombiniert, wie es in Abbildung 3-34 dargestellt ist. [Lep06 S. 11f]

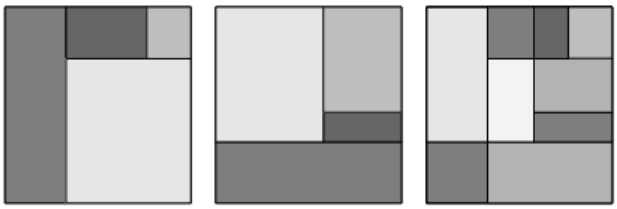

Abbildung 3-34: Klassifikation beim randomizes tree durch zwei Zerlegungen, welche in der Kombination eine feinere Zerlegung ergeben [Lep06S. 11]

Mit zunehmender Tiefe und Anzahl der Einzelbäume wird die Aufteilung des Gesamtbaumes immer feiner. Dies ermöglicht eine bessere Abschätzung, verursacht aber auch höhere Rechen- und Speicherkosten. Tiefe und Anzahl der Einzelbäume bildet somit sowohl ein Maß für die Effektivität (Erkennungsrate), als auch die Effizienz (Ressourcenverbrauch) des erstellten Gesamtbaums.

Die Leistungsfähigkeit des randomized tree hängt von den Verfahren ab, mit denen er kombiniert wird. Hierzu zählen unter anderem die verwendete Feature-Extraktionsmethode und der definierte Knotentest. Ein randomized tree kann gegebe-

nenfalls auch mehrere verschiedene Tests enthalten. Durch eine effiziente Berechnung, welche Echtzeitanwendungen ermöglicht, und der hohen Robustheit gegenüber Änderung der Lichtverhältnisse und des Blickwinkel, der Skalierung, sowie gegen Verdeckungen, ist der randomized tree ein nützliches Hilfsmittel im Bereich der Computer Vision.

3.3.3 Regionen-Detektoren

Nach Kanten- und Ecken-Detektoren ist die Klasse der Feature-Detektoren, welche auf der Beschreibung von Bildregionen basiert, die dritte und letzte hier betrachtete Gruppe. In diese Kategorie fallen auch Blob-Detektoren. Blobs sind Bildregionen oder Punkte, wie beispielsweise Intensitätsmassezentren, die sich ähnlich wie Ecken durch Helligkeitsunterschiede von ihrer Umgebung abheben. Aus diesem Grund können Blob-Detektoren gleichzeitig in die Gruppe der POI-Detektoren eingeordnet werden.

Definition 3.10 (Blob)
Ein Blob ist ein Punkt oder eine Bildregion, die sich von ihrer unmittelbaren Umgebung durch seine Helligkeit abhebt [Gal09 S. 12]

Durch Blobs können auch Bildbereiche erfasst werden, welche zu stark geglättet sind, um durch Ecken-Detektoren ermittelt zu werden. In der Praxis werden Blob-Detektoren häufig eingesetzt, um Zusatzinformationen über eine Bildregion zu erhalten. So können beispielsweise Objekte oder Texturen im Bild erkannt werden. Im Folgenden werden einige Vertreter dieser Detektoren-Klasse diskutiert. [Gal09 S. 12]

Scale Invariant Feature Transform (SIFT)

In der Literatur ist der SIFT-Detektor auch unter Namen Difference of Gaussian (DoG) bekannt. Der DoG-Algorithmus wurde 2003 von David Lowe vorgestellt und entspricht einer Approximation des Verfahrens Laplacian of Gaussian (LoG). DoG ist ein skalierungsinvarianter Detektor, welcher die Blobs in einem Bild durch eine Laplace-Filterung extrahiert. Über die Diffusionsgleichung aus der Skalenraumtheorie kann gezeigt werden, dass Laplace einer Ableitung der Bildfunktion in Skalenrichtung entspricht. Da die Differenz zwischen benachbarten Punkten in einer vorgegebenen Richtung die Ableitung in dieser Richtung annähert, ist auch die Differenz zwischen Bildern verschiedener Skalierungen eine Approximation der Ableitung im Bezug auf die Skalierungsebene. Beim SIFT-Algorithmus wird die Laplace-Filterung deshalb durch die Differenz zweier Gauß-geglätteter Bilder approximiert, was in der Abbildung 3-35 dargestellt ist. [Tuy08 S. 246f]

Abbildung 3-35: Approximation von Laplace durch die Differenz zweier Gauß-geglättete Bilder [Tuy08 S. 247]

Das Originalbild *I(x,y)* wird mehrfach durch Faltung mit verschiedenen Gauß-Filtermasken *g(x,y,σ)* geglättet, um so mehrere Bildversionen verschiedener Skalierungsebenen *L(x,y,σ)* zu erzeugen. Die entstandenen geglätteten Kopien des Eingabebildes werden zu Oktaven gruppiert und anschließend paarweise kombiniert, um so mehrere DoG-Bilder *D(x,y,σ)* zu berechnen. In diesen Differenzbildern wird mithilfe der non-maximal suppression nach Pixeln gesucht, die ein lokales Maximum bzw. Minimum in der Bilddimension und den Skalierungsebenen darstellen. Dazu wird jeder Punkt mit seinen acht Nachbarn des gleichen Bildes und den neun Nachbarn in jedem Bild der adjazenten Skalen verglichen. Ist der Wert des Punktes größer oder kleiner als alle Pixelwerte der betrachteten Punkte, wird er als Feature gewählt. Die Position der gefundenen Extremas wird durch quadratische Interpolation verfeinert. Dabei werden Features, die in Bildstrukturen mit wenig Kontrast detektiert wurden, gelöscht, da sie zur Instabilität neigen. Im Anschluss werden die restlichen Bildoktaven gesampelt. Da die Laplace-Filterung stark auf Kanten reagiert, wird ein zusätzlicher Schritt eingefügt, um diesen Einfluss zu minimieren. Hierbei werden durch die Eigenwerte einer Hesse-Matrix der Fundort und das Ausmaß der interessanten Punkte geschätzt, indem die Differenz benachbarter Abtastpunkte gebildet wird. Diese Berechnung wird nur für sehr wenige Punkte durchgeführt und beeinflusst die Gesamtlaufzeit des Algorithmus unwesentlich. Der gesamte Ablauf des SIFT-Verfahrens wird in folgender Abbildung zusammenfassend dargestellt. [Tuy08 S. 246-248, Ame10 S. 36-38]

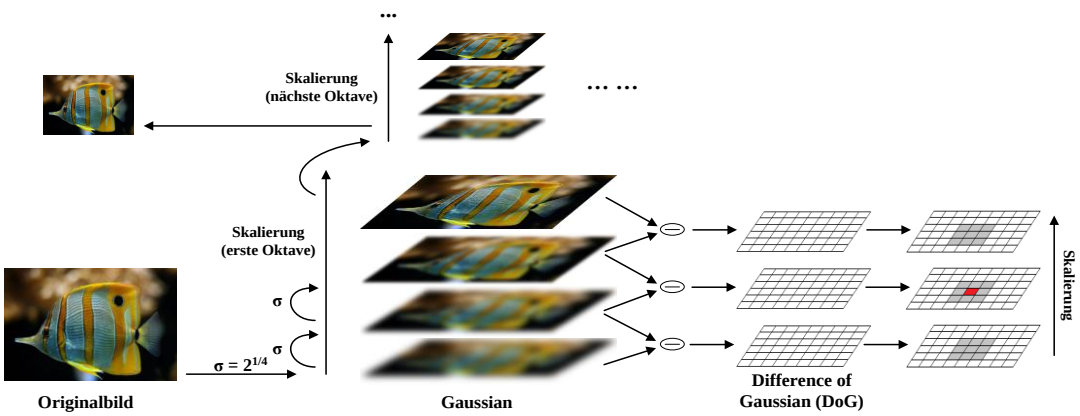

Abbildung 3-36: Schema des SIFT-Detektionsprozesses [Tuy08 S. 247]

Der SIFT-Detektor ist invariant gegenüber Rotation und Standpunktänderungen (Translation). Das Ergebnis des Detektionsprozesses ist eine Approximation des Laplacian of Gaussian. Diese benötigt für die Berechnung der DoG weniger Rechenzeit und ist somit deutlich effizienter als LoG. Dies ermöglicht den Einsatz von SIFT in Sensorsystemen zur Robotersteuerung, Navigation und Objekterkennung. State of the Art SIFT-Softwareimplementierungen werden häufig zusammen mit High-End Prozessoren in eingebetteten Systemen eingesetzt. Aber auch eine hardwarebasierte Realisierung in mobilen Systemen ist möglich. [Gal09 S. 12, Bla07 S. 1-2]

Speeded Up Robust Features (SURF)

Das Speeded Up Robust Features Verfahren (SURF) wurde von Herbert Bay et al. vorgestellt. Zur Detektion von POIs wird eine Approximation des Difference of Hessian Detektors (DoH) genutzt, weshalb der SURF-Detektor auch als Fast-Hessian bezeichnet wird. Die Determinante der Hesse-Matrix wird zur Bestimmung der Position und der

Skalierung der detektierten Features verwendet. Bei SIFT und anderen Verfahren ist ein Gauß-Filter zum Glätten bei Bildskalierung notwendig. Diese Filterung ist optimal für die Skalierung geeignet, ist bei großen Filterkernen aber recht rechenaufwändig. Die Verwendung von kleinen und effizienten Kernen erzeugt hingegen Bildartefakte, welche sich negativ auf den Detektionsprozess auswirken. Zur Steigerung der Effizienz bei Beibehaltung der Retrieval-Qualität wird die Matrix beim SURF-Verfahren durch eine Reihe von Mittelwert-Filtern (Box-Filter) approximiert, wodurch kein Glätten des Originalbildes notwendig ist. Aus diesem Grund ist diese Variante von SURF auch unter dem Namen Difference of Boxes (DoB) bekannt. Die in Abbildung 3-37 dargestellten 9x9-Filter (D_{xx},D_{yy},D_{xy}) nähern die Gauß'sche zweite Ableitung mit einem Skalierungsfaktor von $\sigma = 1{,}2$ in horizontaler, vertikaler und diagonaler Richtung an. [Gal09 S. 22, Ame10 S. 38f, Tuy08 S. 248-250]

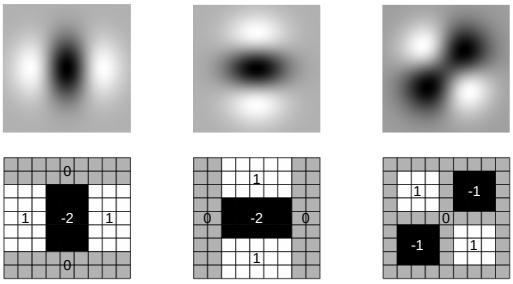

Abbildung 3-37: SURF-Filterkerne zur Approximation von LoG (oben) in x-Richtung (unten links), y-Richtung (unten Mitte) und xy-Richtung (unten rechts)

Durch die Verwendung von Integralbildern können die Filterkerne sehr effizient und unabhängig von der Filtergröße berechnet werden. Trotz dieser groben Annäherung ist die Leistung mit einer Gauß-Diskretisierung vergleichbar. Die relative Gewichtung in xy-Richtung muss theoretisch zudem bei der Skalierung angepasst werden, weshalb der Filterkern D_{xy} zusätzlich durch einen Faktor ω balanciert wird. In der Praxis wird dies meist durch einen konstanten Faktor $\omega=0{,}9$ umgesetzt, was gegenüber einem angepassten balancierten Faktor keinen signifikanten negativen Einfluss auf das Ergebnis hat [Bay06]. Die Determinante der Hesse-Matrix wird durch die Box-Filter beim SURF-Verfahren wie folgt approximiert. [Ame10 S. 39, Tuy08 S. 250]

$$det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \qquad (3\text{-}56)$$

Während bei traditionellen Verfahren wie SIFT verschiedene Gaußfilter verwendet werden, um das Bild in der Größe zu variieren, nutzt SURF Filterkerne von zunehmender Größe. Das Originalbild bleibt unverändert. Die kleinste Filtereinheit besteht dabei aus einem 9x9-Filter (vgl. Abbildung 3-37). Skalenänderungen werden durch Skalierung der Filter unter Beibehaltung des gleichen Filterverhältnisses umgesetzt. [Ame10 S. 39]

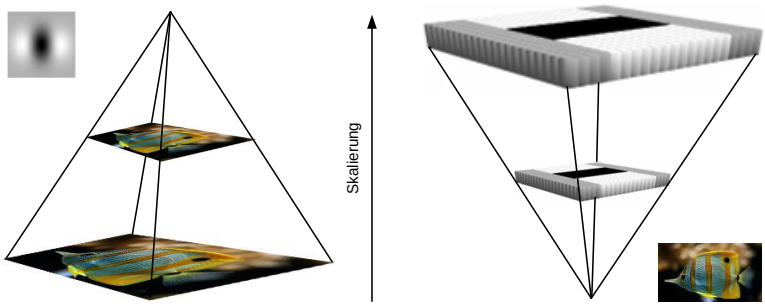

Abbildung 3-38: Modell für traditionellen Ansatz einer Skalenraum-Pyramide (links) und die Nachbildung durch Variation der Filtergröße bei SURF (rechts) [Ame10 S. 40]

Eine Extremwertsuche nach lokalen Maxima über Raum und Skalierung liefert die zu detektierenden Features. Um skalierungs- und rotationsinvarianten Features zu lokalisieren, sind nach der Detektion die drei folgenden Schritte notwendig. Zuerst werden alle Werte gelöscht, die unterhalb eines gesetzten Schwellwertes t liegen. Danach wird durch non-maximal suppression sichergestellt, dass es sich bei allen verbliebenden Eigenschaftswerten um lokale Maxima handelt. Hierzu wird jeder Pixel mit seinen 26 Nachbarn verglichen. Lediglich Bildpunkte mit einem größeren Wert als die ihn umgebenen Pixel werden weiter betrachtet. Im letzten Schritt wird über eine Taylor-Expansion unter Verwendung von Integralbildern die Genauigkeit der Bestimmung der Feature-Position durch Pixelinterpolation erhöht. Um Invarianz gegen Bildänderungen zu erreichen wird außerdem zu jedem errechneten Feature die dominierende Orientierung berechnet. [Ame10 S. 40]

In vielen Ansätzen gleichen sich SIFT und SURF. Der größte Unterschied ist der Einsatz von Box-Filtern zur Skalierung des Bildes, was die Berechnungszeit des SURF-Algorithmus erheblich senkt. Es ist verglichen mit SIFT mehr als fünfmal schneller. Neben der erhöhten Effizienz bietet es noch einen weiteren Vorzug. Im Gegensatz zu SIFT existiert für SURF kein Patent und Implementierungen sind frei im Internet zugänglich. [Tuy08 S. 250, Gal09 S.22]

Maximally Stable Extremal Regions (MSER)

J. Matas et al. schlugen die Maximally Stable Extremal Regions (MSER) als ein weiteres Verfahren vor, um Features zu bestimmen, welche gegen affine Transformation der Bildintensität invariant sind. Eine Maximally Stable Extremal Region ist ein Bereich, in dem alle Pixel entweder heller oder dunkler als die angrenzenden Pixel sind. Die MSER kann folgendermaßen formal definiert werden. [Gal09 S. 14]

Definition 3.11 (Maximally Stable Extremal Region)
Eine Extremal Region Ω ist eine Region, für die für alle Pixel $p \epsilon \Omega$ $I(p)>I(q)$ (maximum intensity regions) oder $I(p)<I(q)$ (minimum intensity regions) als die angrenzenden Pixel $q \epsilon \Omega'$ gilt, wobei Ω' die Begrenzung der Region Ω ist. [Ame10 S. 44]

Im Kontext der MSER bedeutet extrem, dass alle Pixel innerhalb der Region eine höhere (helle extremal regions) oder niedrigere (dunkle extremal regions) Intensität als die angrenzenden Bildpunkte besitzen. Maximal stabil beschreibt die Eigenschaft, welche bei diesem Schwellwertverfahren optimiert wurde. Die Menge aller extremen

Regionen ε muss dabei folgende drei Kriterien erfüllen. Monotone Änderungen der Bildintensität beeinflussen ε nicht. Des Weiteren verändern auch kontinuierliche geometrische Transformationen von Bildkoordinaten die MSERs nicht. Dies schließt die Invarianz gegenüber affinen Transformationen, wie Bildverzerrungen, ein. Zuletzt gilt, dass ein Bild nie mehr MSERs als Bildpixel besitzt. [Tuy08 S. 239f]

Die Größe der Maximally Stable Extremal Regions wird über einen Schwellwert *t* gesteuert, welcher nicht global festgelegt, sondern anhand der Stabilität der detektierten Komponenten evaluiert wird. Die Berechnung der MSERs ist sehr effizient. Der Aufwand nimmt annähernd linear zur Pixelanzahl des Bildes zu. Dabei müssen die Pixel zuerst entsprechend ihres Intensitätswertes sortiert werden, was beispielsweise mit einem Sortierverfahren wie BINSORT einen Berechnungsaufwand von *O(n)* hat. Anschließend werden die Pixel im Bild entsprechend ihres Intensitätswertes in auf- oder absteigender Reihenfolge neu platziert. Durch Vereinigungssuchalgorithmen (union-find algorithm) werden die Bildpunkte im ursprünglichen Bild nun zu zusammenhängenden Komponenten zusammengefasst. Dies besitzt eine Komplexität von *O(n log(log(n)))*. Bei der Vereinigung zweier Komponenten werden die Pixel der kleineren Komponente der größeren Region zugeführt und die kleine Komponente im Anschluss zerstört. Der Algorithmus liefert als Ausgabe eine Menge von MSERs, wobei jeder MSER durch die Position eines lokalen Intensitätsminimums bzw. –maximums und einem Schwellwert *t* gekennzeichnet ist. [Ame10 S. 44, Mik05 S. 53, Mat02 S. 387]

Die nachfolgende Abbildung zeigt das Ergebnis einer MSER-Detektion anhand von Beispielbildern mit verschiedenen Betrachtungswinkeln. Im Gegensatz zu vielen anderen affin invarianten Detektoren sind die berechneten Regionen keine Ellipsen. Durch eine weitere Verarbeitung basierend auf dem ersten und zweiten Form-Moment, dem Mittelwert und der Varianz, können allerdings, wie in der Abbildung dargestellt, entsprechende Ellipsen berechnet werden. Diese sind besser beschreibbar und für die weitere Berechnung geeignet. [Tuy08 S. 241]

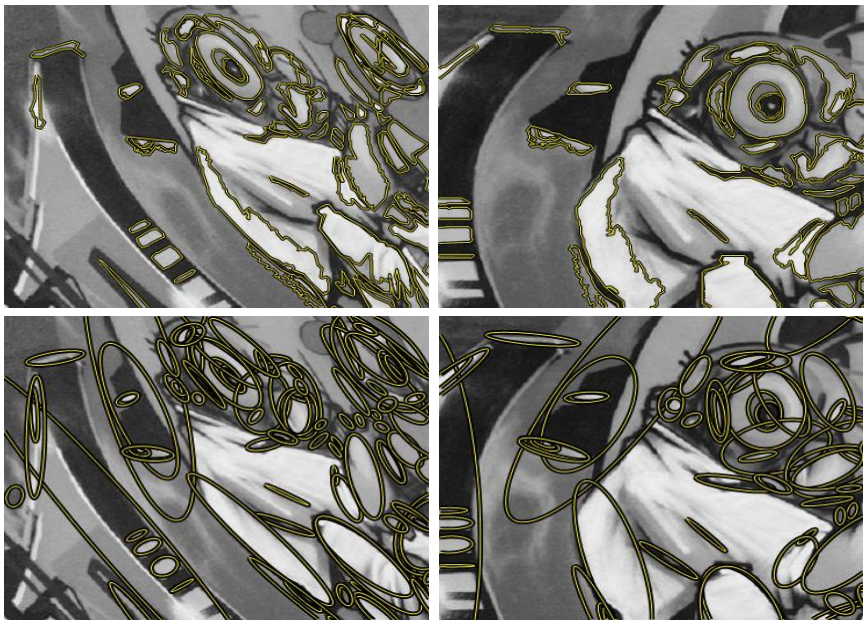

Abbildung 3-39: detektierte MSER (oben) und durch Ellipsen approximierte MSER (unten) aus zwei verschiedenen Betrachtungswinkeln [Tuy08 S. 241f]

Indem die globale Anzahl korrespondierender Bildflächen berechnet wird, können MSERs eingesetzt werden, um in zwei Bildern mit unterschiedlichem Betrachtungs-

winkel Ähnlichkeiten zu bestimmen (vgl. Abbildung 3-39). Dabei gibt es mehrere Möglichkeiten die detektierten Regionen zu vergleichen. Oft wird der Zentroid der MSERs hierfür genutzt.

Krystian Mikolajczyk et al. [Mik05] verglichen in einem ausführlichen Test mit mehreren Kriterien sechs verschiedene affine Regionen-Detektoren. Ziel dabei war es die Leistungsfähigkeit von Harris-affine, Hessian-affine, MSER, EBR und anderen Detektoren zu vergleichen. Ein Ergebnis dabei war, dass MSER bei einer Wiederholbarkeit von 92% die größte Dichte der Regionen aufweist. Im Gegenzug tendiert das Verfahren allerdings auch dazu, viele kleine Regionen zu erkennen, was ein späteres Matching erschwert. Des Weiteren übertrifft MSER die übrigen Detektoren bei Standpunktänderungen und Änderungen der Lichtverhältnisse mit einer hohen Wiederholbarkeit. Bei Skalierungsänderungen ist hingegen der Hessian-affine Detektor besser geeignet. Sehr sensibel reagierte MSER im Test auf Glättung (Blur), wodurch die anderen Detektoren hier überlegen sind. Ein letzter großer Vorteil von Maximally Stable Extremal Regions ist, dass Implementierungen auch frei im Internet erhältlich sind [Mik05, Gal09 S. 14]

3.4 Feature-Deskriptoren

Neben den Feature-Detektoren handelt es sich bei den Feature-Deskriptoren um das zentrale Konzept der Feature-Extraktion. Meist liegen die durch den Detektor ermittelten Daten in einer für die Anwendung ungeeigneten Art und Weise vor und bedürfen einer speziellen Nachbearbeitung. Die Feature-Deskription arbeitet die durch den Detektor ermittelten Keypoints so auf, dass sie für spezifische Anwendungen, wie das Berechnen einer Distanzfunktion, verwendbar sind. Erst danach handelt es sich im Sinne des Retrievals um Features. Sie können beispielsweise für einen effizienten Vergleich verschiedener Merkmalswerte bei der Suche auf den analysierten Bildern oder bei der Auswertung dieser Features zur Steuerung und Überwachung von Systemen eingesetzt werden. Ein wesentliches Ziel dabei ist die Reduktion der Dimensionalität. Oft sind viele, der durch den bisherigen Prozess ermittelten Daten, redundant. Das bedeutet, es herrscht ein Ungleichgewicht zwischen Datenmenge und Informationsinhalt. Die Eingabedaten werden durch den Deskriptor in eine Form transformiert, welche nur noch minimale Redundanz enthält und sich optimal für die anschließende Auswertung eignet. Eine solche Form kann beispielsweise ein Feature-Vektor oder eine Matrix sein.

Definition 3.12 (Feature-Deskriptor)
Ein Feature-Deskriptor bezeichnet in der Computer Vision ein Werkzeug zum Beschreiben der durch den Detektionsprozess ermittelten interessanten Bildregionen in geeigneter Form. [Gal09 S. 19]

Ähnlich zur Detektion existieren viele verschiedene Methoden, um Feature-Vektoren zu extrahieren, welche die ermittelten interessanten Punkte beschreiben. Ein wesentliches Ziel dabei ist eine Darstellung der Objekteigenschaften, die unabhängig von der spezifischen Detektionsstelle ist. Invarianz ist für lokale Deskription ebenso wichtig wie für die Detektion. Idealerweise sollte das Deskriptionsergebnis von verschiedenen Beleuchtungen, Skalierungen und Rotation unabhängig sein. Da Invarianz meist schwer zu erreichen ist, wird oft die schwächere Eigenschaft der Robustheit, welche geringfügige Änderungen erlaubt, gefordert. Ein weiteres Ziel ist eine hohe Unterscheidungskraft des Deskriptionsergebnisses, um möglichst eindeutige Features zu

erhalten. Beide Kriterien gleichzeitig zu einhundert Prozent zu erfüllen ist unmöglich, da ideale Features nicht existieren. Der Lösungsansatz sieht vor, Features mit einer hohen Unterscheidungskraft zu ermitteln und die Invarianz durch verschiedene Ansätze zu verbessern. Einige dieser Ansätze wurden bereits bei den Detektionsverfahren erläutert. Eine höhere Skalierungsinvarianz wird unter anderem durch die Verwendung mehrerer Skalierungsebenen erreicht. Invarianz gegenüber Rotation kann durch die Berechnung von Histogrammen oder der dominanten Richtung ermöglicht werden. Die Unabhängigkeit gegen Beleuchtungsänderungen wird durch die Nutzung der Gradientenstärke anstatt des absoluten Grauwertes oder über eine adaptive Grauwertnormierung realisiert. [Ulg06 S. 15-17]

Im Folgenden werden die wichtigsten in OpenCV implementierten Deskriptionsverfahren diskutiert.

Scale Invariant Feature Transform Deskriptor (SIFT)

Neben dem Keypoint-Detektor existiert auch ein lokaler Scale Invariant Feature Transform (SIFT) Deskriptor, welcher ebenfalls von Lowe vorgeschlagen wurde. Bei diesem Deskriptor steht in erster Linie eine hohe Robustheit gegenüber Änderungen durch Rotation, Skalierung, Rauschen und kleinen Perspektivenänderungen im Vordergrund. Da der Deskriptor relativ zur Orientierung und Skalierung berechnet wird, besitzen die detektierten Features trotz verschiedener Bildtransformationen die gleichen Deskriptoren und können einander zugeordnet werden. Somit ist es mit hoher Wahrscheinlichkeit möglich einen Bildausschnitt in verschiedenen Aufnahmen wiederzuerkennen.

Um Rotationsinvarianz zu erreichen, wird zu jedem detektierten Punkt zunächst die Amplitude und die Richtung der Gradienten aller Pixel in der lokalen Umgebung berechnet. Daraus wird ein Richtungshistogramm für den detektierten Keypoint ermittelt, in dem von allen Nachbarn die Gradientenrichtung eingetragen wird. Jede Gradientenorientierung wird mit dem Betrag des Gradienten und einem Gauß-Fenster gewichtet. Die Gewichtung des Fensters steht dabei in Abhängigkeit von der Entfernung vom Zentrum des Fensters, welches sich im Keypoint befindet. Das resultierende Histogramm hat sechsunddreißig Bins, wobei jeder Bin 10° der 360°-Umgebung des Feature-Punktes abdeckt. Die Peaks des Histogramms entsprechen den dominanten Richtungen eines Keypoints. [Gal09 S. 19f]

Nachdem der Ort, die Skalierung und die dominante Orientierung jedes ermittelten Schlüsselpunktes berechnet wurden, erfolgt die eigentliche Deskription des Features. Zum Berechnen des Deskriptors wird die Umgebung um den Feature-Punkt in mehrere Regionen unterteilt. Die Gradienten der 16x16-Umgebung werden anschließend in Gruppen von 4x4-Regionen zu Orientierungshistogrammen mit jeweils 8 Bin zusammengefasst. Das Schema ist in Abbildung 3-40 dargestellt, wobei die Helligkeitsunterschiede die Gewichtung durch das Gauß-Fenster andeuten.

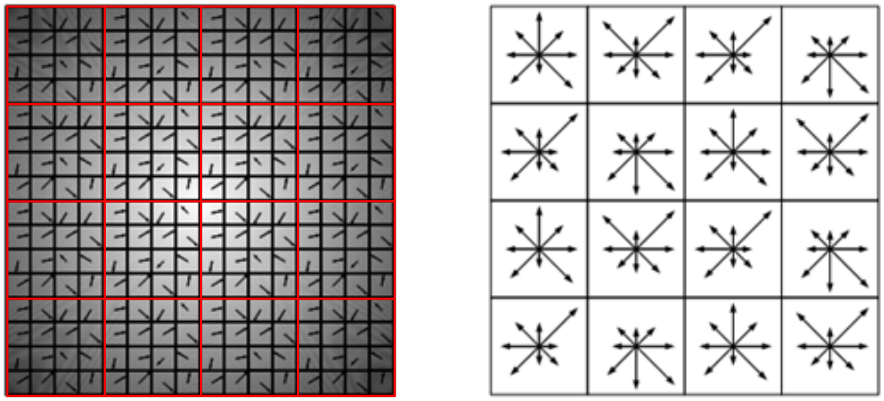

Abbildung 3-40: Berechnungsschema des SIFT-Deskriptors [Ame10 S. 45]

Der von Lowe beschriebene Deskriptor besteht aus sechszehn Richtungshistogrammen mit acht Bins. Eine Konkatenation der sechszehn Histogramme ergibt einen 128-elementigen Vektor, welcher bezüglich seiner Länge normalisiert wird. Eine Kontraständerung des Bildes ändert die Beträge aller Gradienten im gleichen Maße, was durch die Normalisierung herausgefiltert wird und somit keinen Einfluss auf die Suche nimmt.

Inzwischen gibt es viele Varianten des SIFT-Deskriptors von Lowe. Bei **PCA-SIFT** schlagen die Autoren vor, durch die PCA, auch bekannt als Karhunen-Loeve-Transformation, die Größe der Deskriptoren zu reduzieren. Die Reduktion bringt eine Einsparung bei der Rechenzeit, allerdings werden schlechtere Ergebnisse als beim 128-Bin SIFT-Deskriptor erreicht. Bei **rg-SIFT** werden Deskriptoren für die r und g Komponenten des normalisierten RGB-Farbraumes berechnet. **Opponent-SIFT** beschreibt die Farbkanäle in einem dem RGB-Farbraum entgegengesetzten Farbraum und Van Sande beschreibt eine andere Variante des SIFT-Deskriptors, den **Color-SIFT.** Bei diesem Algorithmus werden ebenfalls Farbinformationen zur Berechnung der SIFT-Deskriptoren hinzugezogen. Zuletzt wird auf der Computer Vision Konferenz 2009 von Morel und Yu eine SIFT-Variante vorgestellt, die **Affine-SIFT** (ASIFT) bezeichnet wird. Der Algorithmus soll bei affinen Transformationen wesentlich bessere Ergebnisse als ein einfaches SIFT liefern, welcher lediglich gegen Skalierung, Rotation und Translation invariant ist. ASIFT berücksichtigt zusätzlich die Kameraorientierung. [Gal09 S. 19f]

All diese Varianten basieren auf dem ursprünglichen SIFT-Deskriptor von Lowe, welcher für seine guten Ergebnisse bekannt ist. Durch die Detektion verschiedener Skalierungsebenen ist er skalierungsinvariant. Des Weiteren ist der Deskriptor durch die Schätzung der dominanten Richtung rotationsinvariant. Zuletzt ist er robust gegenüber Beleuchtungsänderungen, da zur Berechnung die Gradienten anstatt der Bildintensitäten verwendet werden. [Ulg06 S. 16]

Speed Up Robust Features Deskriptor (SURF)

Das SURF-Deskriptionsverfahren wurde von Bay et al. vorgeschlagen. Es basiert teilweise auf ähnlichen Ideen wie der SIFT-Deskriptor, hat allerdings einen wesentlich geringeren Berechnungsaufwand. SURF beschreibt mittels Haar-Wavelet-Antworten wie die Pixelintensitäten innerhalb der skalenabhängigen Nachbarschaft des Interessanten Punktes (POI) verteilt sind. [Gal09 S. 22, Ame10 S. 45f]

Wie beim SURF-Detektor bereits angedeutet wurde, wird vor der Deskription die dominante Richtung am detektierten Keypoint berechnet. Dies dient der Bestimmung einer reproduzierbaren Ausrichtung und somit der Berechnung rotationsinvarianter Features. Dafür werden die horizontalen und vertikalen Haar-Wavelet-Antworten d_x und d_y in einem Radius von 6σ um den POI ermittelt. Durch die Verwendung von Integralbildern ist eine schnelle Berechnung der Wavelets in x- und y-Richtung auf allen Skalierungsebenen mit lediglich sechs Operationen möglich. Die Wavelet-Antworten werden mit einem über dem Schlüsselpunkt zentrierten Gauß-Fenster gewichtet. Die dominante Richtung ergibt sich aus der Summe aller Wavelet-Antworten. [Bay06 S. 6f]

Für die Deskription wird ein quadratischer Bereich um den Keypoint relativ zur berechneten Orientierung gebildet. Dieses quadratische Feld besitzt eine Größe von 20σx20σ, wobei σ der Skalierung des detektierten POI entspricht. Die Region wird gleichmäßig in 4x4 Teilbereiche unterteilt, wobei jede Teilregion nun 5x5 Samplepunkte beinhaltet. Um die Robustheit der zuvor berechneten Wavelet-Antworten gegenüber geometrischen Transformationen zu erhöhen, werden d_x und d_y zunächst mit einem Skalierungsfaktor von $3{,}3\sigma$ gewichtet. Anschließend werden beide für jede Teilregion separat aufsummiert. Um zusätzliche Information über die Polarität der intensitätswechsel zu erhalten werden auch die Beträge $|d_x|$ und $|d_y|$ aufsummiert. Jede Teilregion besitzt nun einen vierdimensionalen Vektor $v=(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ der verschiedenen Summen der Wavelet-Antworten. Die Gesamtheit der Vektoren aller sechszehn Regionen ergibt den 64-dimensionalen SURF-Deskriptor. Der komplette Deskriptionsprozess ist in nachfolgender Abbildung schematisch dargestellt. [Bay06 S. 7f, Ame10 S. 46]

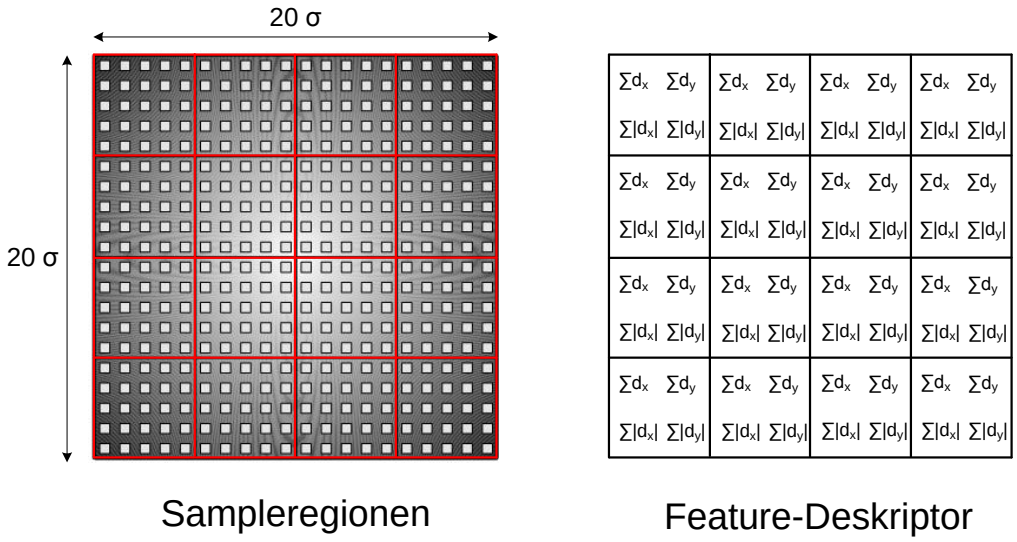

Abbildung 3-41: Berechnungsschema des SURF-Deskriptors [Ame10 S. 46]

In [Bay06] wird gezeigt, dass die Verwendung der hier beschriebenen vier Werte bei Experimenten die beste Performance zeigt. Auch die Nutzung der dargestellten Aufteilung in Teilregionen ist für die meisten Anwendungen optimal. Es wird auch ein SURF-36-Deskriptor mit nur 3x3 Teilregionen diskutiert. Dieser ist zwar schneller, zeigt im Test aber auch ein schlechteres Suchergebnis. Des Weiteren wird eine Variante mit einem 128-dimensionalen Vektor vorgestellt. SURF-128 zeigt bessere Ergebnisse und benötigt bei der Berechnung nicht wesentlich mehr Zeit. Zu bedenken dabei ist auch, dass die Vergleichsoperationen auf Grund der verdoppelten Anzahl an Features deutlich langsamer sind. [Bay06 S. 8]

Ähnlich wie beim SIFT-Verfahren existieren auch zu SURF bereits zahlreiche Varianten. **Color-SURF** verwendet lokale Farbhistogramme zur Berechnung des Deskriptionsvektors. Bei **Upright-SURF** (USURF) werden die dominanten Orientierungen nicht berechnet. Dies spart Rechenzeit, zerstört allerdings auch die Rotationsinvarianz. Dieses Verfahren ist besonders geeignet, wenn der Kamerastandpunkt praktisch unverändert bleibt. **Phase-space based SURF** (PSURF) arbeitet im Phasenraum, wodurch laut den Autoren C. Liu, J. Yang und H. Huang die Performance gegenüber dem ursprüngliche SURF erhöht werden kann. Trotzdem benötigt es weniger Rechenzeit als SIFT, bei dem die Berechnungen direkt auf den Gradienten stattfinden.

Binary Robust Independent Elementary Features Deskriptor (BRIEF)

Der Binary Robust Independent Elementary Features Deskriptor (BRIEF) wurde von Michael Calonder et al. entworfen. Er basiert auf der Verwendung von binären Zeichenketten (binary strings), welche laut den Autoren auch bei relativ wenigen Bits höchst deskriptiv sind. Im Allgemeinen ist die beste Möglichkeit die Rechenzeit zu verkürzen und so das Matching zu beschleunigen die Dimensionalität der berechneten Feature-Vektoren zu reduzieren. Einerseits geht dies durch die Eliminierung redundanter Informationen wie bei der Karhunen-Loève Transformation. Das BRIEF-Verfahren geht einen anderen Weg. Hier wird die Laufzeit von Deskriptoren mit guten Ergebnissen durch die Abbildung des Deskriptors auf eine Hash-Funktion verbessert. Die Ähnlichkeit des resultierenden binären Deskriptors kann mittels Hamming-Distanz verglichen werden, welche sehr schnell über einfache Bitoperationen berechnet werden kann. Um weitere Rechenzeit zu sparen, wird bei der Berechnung des Binärstrings nicht der Umweg über die Ermittlung des SIFT-Deskriptors gegangen, sondern diese direkt aus dem Bild extrahiert. Somit ist nicht nur das Matching, sondern auch die Berechnung beim BRIEF-Deskriptor sehr schnell. Zudem reichen relativ wenig Bits meist für eine gute Bildbeschreibung aus. Je nach Anforderungen benötigt BRIEF 128, 256 oder 512 Bit. Dies spart wiederum Rechen- und Speicherplatzaufwand, wobei gerade der benötigte Speicherplatz bei der Berechnung mehrerer Millionen Deskriptoren entscheidend ist. Alle hier diskutierten Kriterien machen BRIEF zu einem der schnellsten existierenden Deskriptoren. Auch bei der Wiedererkennungsrate steht er laut Calonder den anderen Verfahren in nichts nach. [Cal10 S. 1-3]

Im Folgenden wird der Deskriptionsablauf beim BRIEF-Verfahren genauer erläutert. Vor der Berechnung des Deskriptors wird das Bild geglättet. Hierfür wird ein 9x9 Pixel großer Filterkern verwendet, da sich dieser in Tests bewährt hat. Durch die Filterung wird die Störanfälligkeit herabgesetzt, was wiederum die Stabilität und die Reproduzierbarkeit des Deskriptors erhöht. Im Anschluss wird der Binärstring berechnet, welcher die Helligkeitsverteilung des Bildes beschreibt. Jedes Bit darin steht für das Ergebnis eines paarweisen Vergleichs der Pixelintensitäten zweier Punkte. Der Suchbereich der zu vergleichenden Pixel wird am detektierten Keypoint ausgerichtet. Im Gegensatz zu anderen Verfahren wird nicht die Intensität des POI mit seinen Nachbarn, sondern die zweier Nachbarpixel verglichen. Abbildung 3-42 zeigt das schematische Vorgehen dabei.

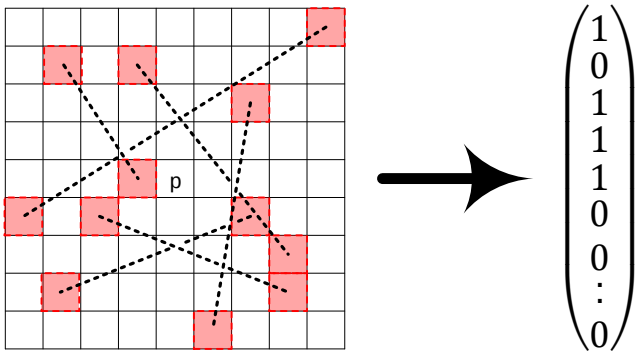

Abbildung 3-42: Schema der Vergleichsoperationen beim BRIEF-Deskriptor

Das Ergebnis des Vergleichs wird in den Feature-Vektor eingetragen. Ist der Intensitätswert des ersten betrachteten Pixels kleiner, so wird eine 1 geschrieben, ist er größer eine 0. Der Vergleich τ zweier Pixel p_1 und p_2 ist in folgender Gleichung formal definiert.

$$\tau(p_1, p_2) = \begin{cases} 1 & wenn\ I(p_1) < I(p_2) \\ 0 & sonst \end{cases} \qquad (3\text{-}57)$$

Die Anzahl der nötigen Test n_d ist je nach Anforderungen an Qualität und Geschwindigkeit 128, 256 oder 512. Da sich die Länge des Deskriptionsvektors direkt aus der Anzahl der Vergleiche ableitet, ergibt sich daraus auch der Speicherverbrauch eines Deskriptors, wonach die einzelnen BRIEF-Versionen auch klassifiziert werden. BRIEF-k mit $k=n_d/8$ bezeichnet einen BRIEF-Deskriptor, welcher k Byte Speicher benötigt. [Cal10 S. 3f]

Eines der wichtigsten Kriterien bei diesem Deskriptor ist die Wahl der paarweise zu vergleichenden Pixel. Calonder et al. schlagen hierfür fünf verschiedene Modelle vor. Diese reichen von einer Gleichverteilung über eine zufällige Gauß- bzw. einer isotropen Gauß-Verteilung bis hin zu einer symmetrischen Anordnung, bei der auf einer Kreisbahn um den Keypoint alle möglichen entgegengesetzten Pixelintensitäten verglichen werden. Die fünf Ansätze sind in folgender Grafik dargestellt.

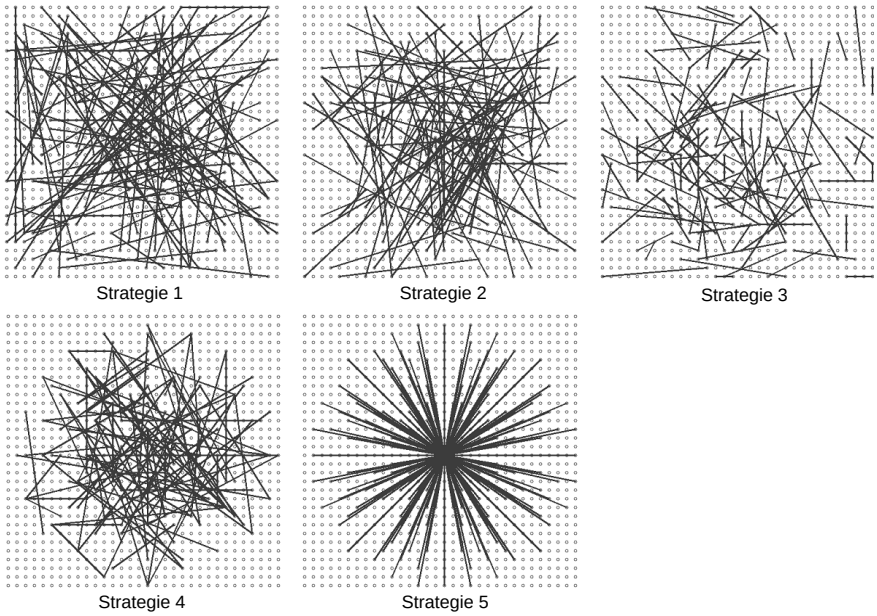

Abbildung 3-43: Verschiedene Modelle zur Wahl der zu vergleichenden Pixel [Cal10 S. 5]

Die schlechtesten Ergebnisse im Test liefert die regelmäßige, symmetrische Strategie 5. Am besten erweist sich die zweite Methode, wobei alle zufälligen Verfahren recht gute Wiedererkennungsraten besitzen. In der Praxis wird meist die Gauß-Verteilung (Strategie 2) verwendet. [Cal10 S. 5f]

Calonder et al. stellten insgesamt vier verschiedene Arten des BRIEF-Deskriptors vor. Bei der hier besprochenen Variante handelt es sich um den **upright BRIEF** (U-BRIEF). Dieser ist die Basisform von BRIEF und nicht rotationsinvariant. **Orientation sensitive** BRIEF (O-BRIEF) ist zumindest teilweise rotationsinvariant. Vor allem bei kleinen Bilddrehungen von ca. 10-15° erwies sich diese Form als gut. Das Verfahren ist dabei zum Teil auch anderen Deskriptoren wie SURF überlegen. Dieser ist hingegen bei großen Rotationen deutlich besser. Eine weitere Variante ist **scaled BRIEF** (S-BRIEF). S-BRIEF ist skalierungsinvariant, wobei diese sich genau wie die Rotationsinvarianz bei OBRIEF nur über den verwendeten Detektor realisieren lässt. Das letzte Verfahren ist **database BRIEF** (D-BRIEF). Dieses bietet im Gegensatz zu den anderen Varianten volle Invarianz gegenüber Rotation und Skalierung. Erreicht wird diese durch mehrere UBRIEF-Deskriptoren zu verschiedenen Bildtransformationen, welche als Template in einer Datenbank gehalten werden.

One-Way Deskriptor

Der One-Way Deskriptor wurde von S. Hinterstoisser et al. entworfen, um einen Bildausschnitt in anderen Bildern der Datenbank wiederzuerkennen. Um eine effiziente Nutzung in interaktiven Systemen zu gewährleisten, kombiniert dieses Deskriptionsverfahren eine offline und eine online Trainingsphase. Zur Laufzeit wird bei diesem Verfahren kein Deskriptor mehr berechnet. Dies geschieht bereits während dem Offline-Modus, in welchem die berechnete Deskription in der Datenbank gespeichert wird. Dies ist auch der Grund für die Bezeichnung One-Way Deskriptor.

Die Grundidee hinter diesem Deskriptor ist die Speicherung mehrerer Ansichten auf ein Objekt. Laut den Autoren sind etwa dreihundert Blickrichtungen auf einen Bildausschnitt (Patch) für ein gutes Deskriptionsergebnis notwendig. Zudem werden von diesen Patches nur wenige aus Eingabebildern entnommen. Die übrigen Ausschnitte, welche für die weitere Berechnung notwendig sind, werden als verzerrte Kopien der extrahierten Patches berechnet. Da perspektivischen Verzerrungen recht rechenaufwändig sind, werden sie während der offline Trainingsphase berechnet und in der Datenbank gehalten. Ein weiteres Basiskonzept des One-Way Deskriptors ist die Verwendung von Mittelwert-Patchen. Da das Matching zur Laufzeit geschieht und der Vergleich eines Mittelwertes deutlich weniger Zeit benötigt als alle Ansichten einzeln zu vergleichen, wird der Mittelwert des *i*-ten Patches gebildet. Hierfür wird die Summe des Patches p_i über die N Ansichten auf diesen Bildausschnitt gezogen, wobei ein bestimmter Blick H_h auf p_i durch die Verzerrungsfunktion $w(p_i H_h)$ beschrieben wird. [Hin09 S. 3-5]

$$\overline{p_{i,h}} = \frac{1}{N}\sum_{j=1}^{N} w(p_i, H_{h,j}) \tag{3-58}$$

Diese Mittelwerte sind sehr stabil gegenüber Rauschen und ersetzen rechenaufwändige Glättungsverfahren, wie die Gauß-Filterung, welche bei anderen Deskriptionsverfahren zum Erreichen von Skalierungsinvarianz eingesetzt werden. Wie bereits erwähnt teilt sich die Trainingsphase, in welcher die Mittelwert-Patches bestimmt werden, in eine online und eine offline Komponente. Ziel dabei ist es, zur Laufzeit möglichst

wenige Berechnungen durchzuführen. In der Offline-Phase erfolgt zunächst eine Hauptkomponentenanalyse, bei welcher der Patch in seine Hauptvektoren zerlegt wird. Hierfür wird die Transformation *H* auf mehrere Patches angewendet. Dies liefert eine Menge von Paaren aus originalen und transformierten Patches. Durch lineare Regression kann daraus eine Matrix berechnet werden, welche bei Anwendung auf einen beliebigen Intensitätsvektor eine gekrümmte Version des originalen Vektors berechnet. In der Praxis werden oft mehrere Matrizen unterschiedlicher Größe für einen Patch erzeugt. Die Berechnung neuer Patches ist nun eine Projektion in den Eigenvektorraum. Diese hängt nur proportional von der Anzahl der Basisvektoren und nicht mehr von der Anzahl der notwendigen Ansichten auf den Patch ab. Der Hauptaufwand in der Online-Phase besteht darin die Dimensionalität der neu berechneten Patches zu verringern. Hierfür wird beim One-Way Deskriptor die Principal component analysis (PCA) verwendet. Nach der Trainingsphase wird das Objekt durch eine Reihe von Mittelwert-Patches beschrieben. Jeder dieser Mittelwerte beschreibt das Objekt von einer anderen Ansicht. Die gesamte Menge der Mittelwerte bildet den Deskriptor. Beim Matching wird der Eingabe-Patch mit allen in DB gespeicherten Patches aller Ansichten verglichen. Dabei werden oft Suchheuristiken, wie der KD-Baum, eingesetzt. [Hin09 S. 3-5]

3.5 Einführung in OpenCV

OpenCV ist eine freie Bibliothek, welche für die Programmiersprache C/C++ zur Verfügung steht und kompatibel zu verschiedenen Betriebssystemen ist. Als freie Software unterliegt sie den Lizenzbedingungen der BSD-Lizenz und ist somit sowohl im akademischen als auch im kommerziellen Gebrauch frei zugänglich [Ope11]. Die Firma Intel initialisierte 1999 die Entwicklung dieser Bibliothek, heute wird sie allerdings hauptsächlich durch das US-amerikanische Unternehmen für Robotertechnologie Willow Garage gepflegt. Das Kürzel CV im Namen der Bibliothek steht für Computer Vision und weist auf den großen Umfang der Software an Algorithmen im Bereich der Bildverarbeitung und des maschinellen Sehens hin. Das Markenlogo von OpenCV (vgl. Abbildung 3-44) greift dies im Design auf. Es zeigt die drei Großbuchstaben O, C, V in Form eines Dreiecks und kombiniert so Motive aus dem Bereichen Open Source und Computer Vision. Die Dreiecksform erinnert dabei an die berühmten optischen Täuschungen von Kanizsa, bekannt als *Kanizsa triangle*. Die Farben des Logos symbolisieren den RGB-Farbraum.

Abbildung 3-44: Markenlogo von OpenCV [Ope11]

Die Version 1.0 von OpenCV wurde im September 2006 herausgegeben. Seit Ende September 2009 wird mit der Weiterentwicklung OpenCV 2.0 gearbeitet. Diese liegt aktuell in der Version 2.3 vor. OpenCV basiert auf der Image Processing Library (IPL) und erweitert diese um komplexere Funktionen [Kub05 S. 2-6]. Zum Funktionsumfang gehören unter anderen Bildfunktionen zum Erzeugen und Zerstören von Bildern und Funktionen zum Finden, Anzeigen um Manipulieren von Bildkonturen. Des Weiteren

bietet OpenCV eine Reihe von Funktionen zur statistischen Bildauswertung wie Mittelwertbildung, Normen und Momente, sowie Schwellwertbildung und Histogramme. Aktuell bietet die Sammlung mehr als 500 Funktionen und ist damit ein nützliches und zudem kostenfreies Tool für viele Anwendungen im Bereich des CBIR. Dabei werden Objekt-Identifikation, Segmentierung, Gesten- und Gesichtserkennung und Tracking von Bewegungen unterstützt, um nur einige Anwendungsgebiete zu nennen. Im Fokus dieser Arbeit stehen allerdings die Algorithmen zur Feature-Erkennung und Auswertung.

Die genannten Funktionen sind in der OpenCV-Bibliothek in mehreren Paketen organisiert. Die folgende Abbildung beschreibt die wichtigsten Pakete und ihre Zusammenhänge.

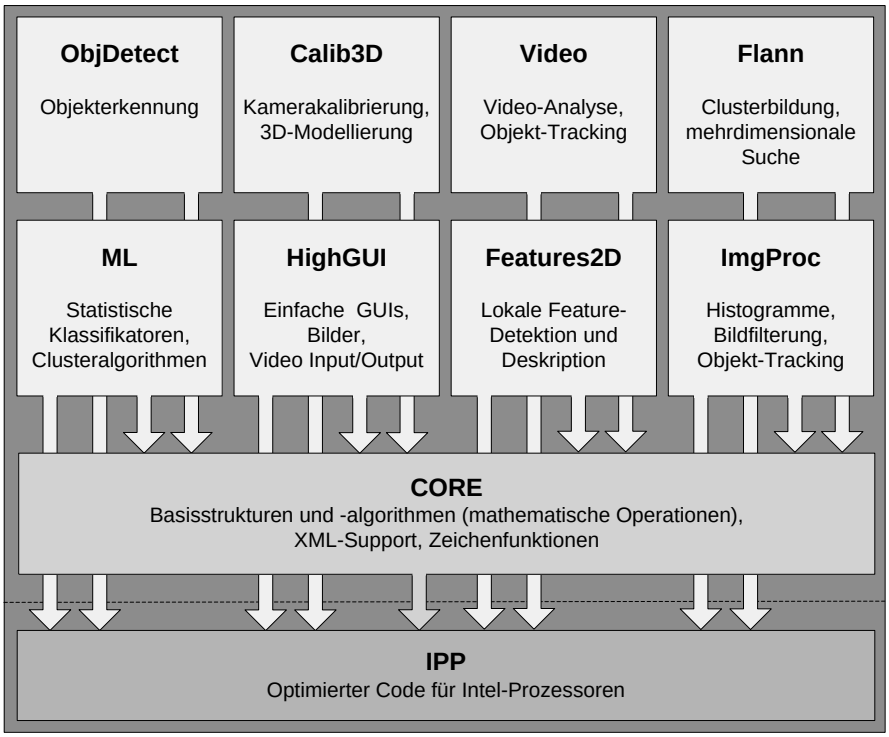

Abbildung 3-45: Basispakete der OpenCV-Bibliothek [Wie08 S. 2]

Auf der untersten Ebene befindet sich die Integrated Performance Primitives (IPP), welche zur Optimierung des Codes bei der Ausführung genutzt werden kann. Dieser Teil stammt noch aus der Zeit als Intel OpenCV betreute, wird aber für die Nutzung der Bibliothek nicht zwingend benötigt. In der Schicht darüber befindet sich das Paket `Core`. `Core` enthält die grundsätzlichen Datenstrukturen, welche für die Arbeit mit OpenCV benötigt werden. Dies beinhaltet insbesondere Punkte, Skalare, Matrizen und Bilder. Es werden aber auch komplexere Datenstrukturen wie Bäume oder Sequenzen bereitgestellt. Auf der obersten Ebene stehen unter anderem die Pakete `Features2d`, `Ml` und `Highgui`. `Highgui` bietet Plattformunabhängige Möglichkeiten zum Erzeugen einfacher GUIs und zum Laden und Speichern von Bildern und Videos. `Ml` beinhaltet Algorithmen des Machine Learnings und `Features2d` enthält Algorithmen zur Detektion und Deskription von lokalen Features, welche im Kapitel 3.3 und 3.4 erläutert werden. Das letzte Paket ist für diese Arbeit besonders wichtig. Neben der Berechnung der Algorithmen zur Feature-Extraktion bietet OpenCV in diesem Modul auch zahlreiche Methoden zum Kombinieren verschiedener Verfahren und zum Matchen der berechneten Features. [Wie08 S. 2f]

Die Stärken von OpenCV liegen besonders in der Performance, der großen Anzahl und der Aktualität der Algorithmen aus den neusten Forschungsergebnissen. Die Bibliothek wird in zahlreichen Ländern der Welt eingesetzt. Es wurde bereits mehr als 2 Millionen Mal heruntergeladen und hat rund 40 Tausend Nutzer in der Usergroup

[Ope11], was die Bedeutung von OpenCV im Bereich Computer Vision unterstreicht. Die Graphik 3-46 gibt einen Überblick über die stetig wachsende Zahl der OpenCV-Nutzer. Dabei verzeichnet die Bibliothek einen sprunghaften Anstieg der Downloadzahlen in den letzten Jahren, was in etwa mit dem Update auf Version 2.0 zusammenfällt. Darüber hinaus zeigt die Abbildung, dass überwiegend Windows-Versionen der Bibliothek genutzt werden und dass an erster Stelle Japan in den Downloadcharts steht.

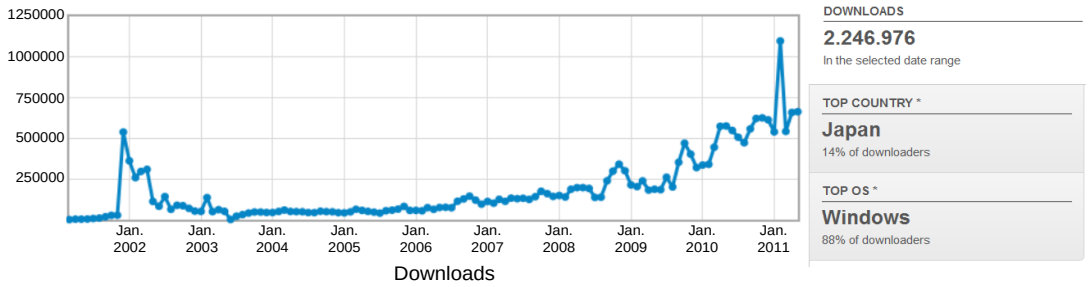

Abbildung 3-46: Downloadzahlen OpenCV von 2002 bis 2011 (Stand 12.08.2011) [Sou11]

Die Nutzerforen von OpenCV und das umfassende Wiki bieten viele nützliche Informationen zu allen implementierten Funktionen und unterstützen die zahlreichen Anwender bei der Arbeit. Neben der Willow Garage stellen die vielen Nutzer ein wichtiges Standbein für die stetige Weiterentwicklung der Programmbibliothek dar. Eingetragener User können eventuell gefundene Bugs oder gewünschte Features der Willow Garage zusenden und so die weitere Entwicklung unterstützen. Nicht zuletzt aus den oben genannten Gründen wurde OpenCV als Grundlage für diese Arbeit gewählt.

4. Einbindung der Verfahren zur Analyse der Detektions- und Deskriptionsmethoden in Pythia

Ziel dieser Arbeit ist es die Verwendung von Detektion-Deskriptions-Kombinationen unter verschiedenen Gesichtspunkten zu untersuchen. Eine Betrachtungsweise dabei ist der Zusammenhang verschiedener Verfahren, um zu ermitteln, welche Extraktionsmethoden zusammen einzusetzen sind. Der Einsatz von möglichst verschiedenartigen Feature-Mengen bringt einige Vorteile, welche in Kapitel 5.1.2 diskutiert werden. Zudem werden an dieser Stelle Analyseansätze vorgestellt, um diese Mengen zu ermitteln.

In diesem Kapitel soll zuerst das Lehrstuhlsystem, auf welchem alle Berechnungen und Analysen erfolgen, vorgestellt werden. Im Anschluss wird Schritt für Schritt ein Analysewerkzeug entworfen, welches die im fünften Kapitel vorgestellten Evaluierungsverfahren umsetzt. Dieser Entwurf bildet die Grundlage für die anschließende Implementierung und die im nächsten Kapitel durchgeführten Analysen.

4.1 Lehrstuhlsystem Pythia

Pythia ist die interne Bezeichnung für ein Retrieval-System, welches im Rahmen eines ForMaT-Projektes am Lehrstuhl Datenbank- und Informationssysteme (DBIS) entwickelt wird. Für dieses Forschungsprojekt arbeitet der Lehrstuhl DBIS zusammen mit dem Lehrstuhl Marketing und Innovationsmanagement (MuI) der BTU Cottbus. Forschung für den Markt im Team (ForMaT) ist ein Förderprogramm des Bundesministeriums für Bildung und Forschung, welches Ergebnisse aus öffentlichen Forschungsprojekten besser und schneller für die Wirtschaft nutzbar machen soll. Dabei sollen gezielt Forscher aus verschiedenen Bereichen zusammen tätig sein und so einen Technologietransfer speziell in den neuen Bundesländern ermöglichen. Derzeit arbeiten neun Mitarbeiter, die Lehrstuhlinhaber, sowie einige studentische Hilfskräfte beider Lehrstühle an der Entwicklung von Pythia. Im Rahmen dieser Forschung werden verschiedene Ziele verfolgt. Der Lehrstuhl MuI erhofft sich beispielsweise in erster Linie Nutzergruppen anhand von Bildanalysen erstellen zu können, um diese für Marketingzwecke zu verwenden. Am Lehrstuhl DBIS erhofft man sich dagegen ein CBIR-System zu entwickeln, welches ein nutzerfreundliches Relevanz-Feedback ermöglicht und so dazu beiträgt die semantische Lücke auf beiden Seiten zu verringern (vgl. Kapitel 2). Weitere Forschungsschwerpunkte sind effiziente Indexstrukturen basierend auf der verwendeten Logiksprache CQQL, Polyrepräsentation von Bildern durch mehrere Features, (semi-) automatisches Tagging von Bildern mittels Clusterverfahren und die intelligente Selektion von Features, welche auch im Mittelpunkt dieser Arbeit steht.

Der Name Pythia stammt aus der griechischen Mythologie. Dabei handelte es sich um eine antike Priesterin, welche den Ratsuchenden im Tempel von Delphi weissagte (Orakel von Delphi). Das Besondere an diesem Orakel war die Vieldeutigkeit der Weissagungen, welche einer speziellen Deutung bedurften. Diese Analogie ist auch auf das Lehrstuhlsystem Pythia zu übertragen. Durch verschiedene Verfahren trifft das System eine Aussage zur gestellten Nutzeranfrage, welche mit entsprechender Retrieval-Kenntnis eine hilfreiche Antwort auf den Informationswunsch geben kann.

Grundsätzlich ist Pythia für die Verwendung unterschiedlicher Medientypen konzipiert. Da derzeit allerdings nur Bilder verarbeitet werden können, wird sich im weite-

ren Verlauf auf die Betrachtung dieses Medienobjektes beschränkt. Das System nutzt für die Suche den im zweiten Kapitel beschrieben QBE-Ansatz. Dabei gibt der Anwender ein Beispielbild vor, zu welchem das System ähnliche Bilder ermittelt. Der prinzipielle Ablauf folgt dabei dem in Abbildung 2-1 dargestellten Schema. Beim Pythia-System bestehen hierbei einige Besonderheiten, wie die Verwendung mehrerer Features und die dadurch notwendige Aggregation der Extraktionswerte. Die nachfolgende Abbildung verdeutlicht den IR-Prozess im Lehrstuhlsystem.

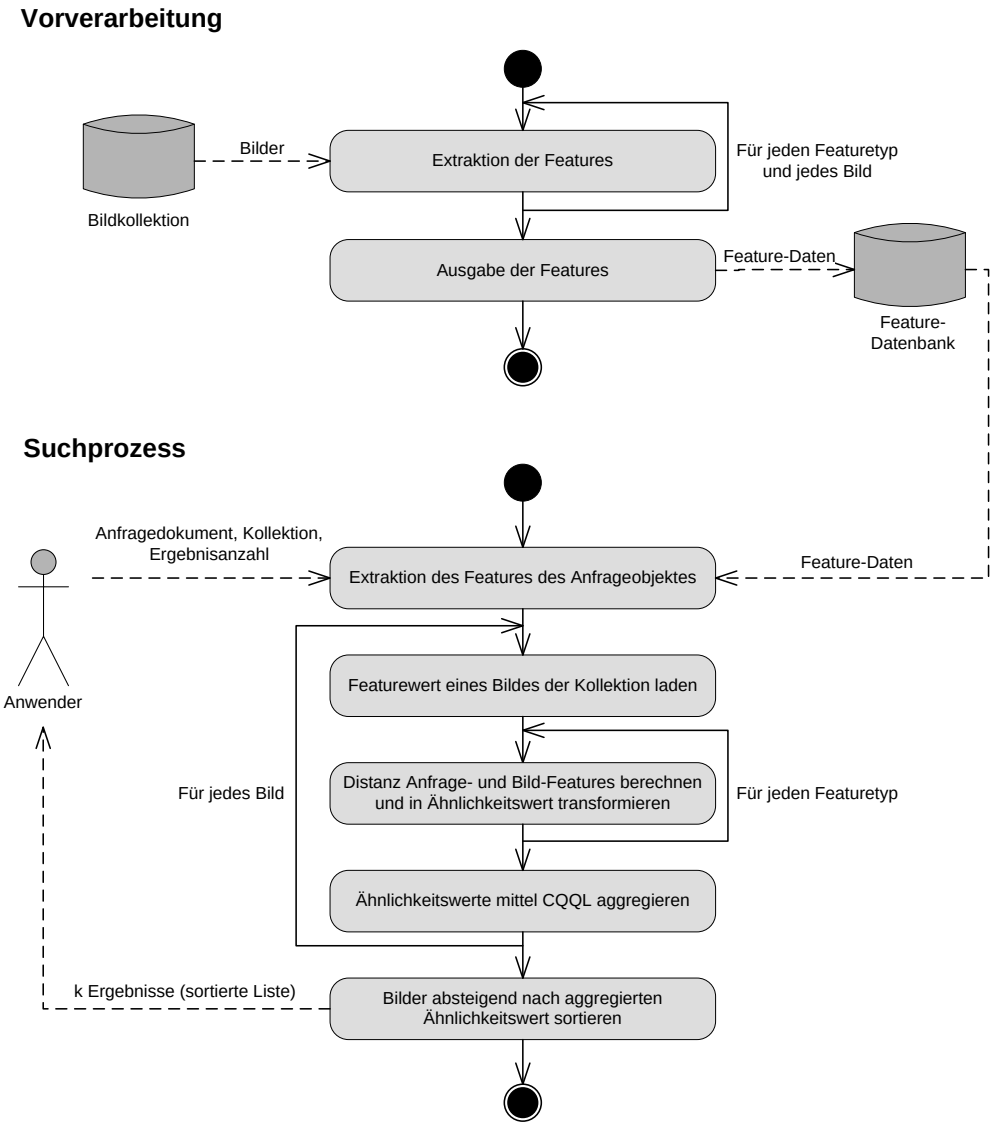

Abbildung 4-1: IR-Prozess bei Pythia [Ber11 S. 8]

Es ist hier, wie im zweiten Kapitel erläutert, eine Trennung zwischen Extraktion von Features und Nutzeranfrage zu erkennen. Zudem werden bei Pythia für jedes Bild einer Kollektion mehrere Features berechnet und für zukünftige Anfragen gespeichert. Bei einer Nutzeranfrage gibt der Anwender ein Anfragebild, eventuell die Anzahl *k* der gewünschten Suchergebnisse und die Bildsammlung, auf der gesucht werden soll, vor. Daraufhin werden die gespeicherten Features jedes Kollektionsbildes gelesen und die Distanz zum Anfragebild ermittelt. Die Ähnlichkeiten der unterschiedlichen Retrieval-Verfahren werden hierfür mittels der Logiksprache CQQL aggregiert. Nach der Distanzberechnung und einer Transformation in Ähnlichkeitswerte für alle Bilder werden diese entsprechend der aggregierten Gesamtähnlichkeit sortiert. Dem Anwender werden

dann die *k*-besten Suchergebnisse ausgegeben. Dieser kann nun die einzelnen Treffer unterschiedlich gewichten und die Ergebnisliste so verändern, um diese seinem Informationswunsch anzunähern.

Im Folgenden werden einige Implementierungskonzepte im Pythia-System kurz erläutert. Wie in der Einführung zu Pythia dargestellt, besteht ein Forschungsschwerpunkt in der Entwicklung eines nutzerfreundlichen Relevanz-Feedback-Verfahrens. Das Prinzip dabei ist, durch eine spezielle Gewichtung der Ähnlichkeiten den Suchprozess zu beeinflussen. Hierfür steht dem Anwender eine am Lehrstuhl DBIS entwickelte Nutzeroberfläche zu Verfügung, deren Prototyp in der folgenden Abbildung illustriert ist.

Abbildung 4-2: Nutzeroberfläche von Pythia [Ber11 S. 9]

Der Nutzer erhält nach der Suche eine geordnete Liste von Bildern, welche er entsprechend der Relevanz für seine Suche innerhalb der geschachtelten Quadrate anordnet. Dabei haben die Bilder in den inneren Quadraten eine höhere Relevanz für die Suche. Zudem kann der Anwender auch Bilder in den Papierkorb verschieben. Diese gelten als irrelevant für die Suchanfrage. Mathematisch gesehen steht hinter diesem Lernverfahren eine Halbordnung, aus welcher mittels eines Simplex-Algorithmus die neuen Gewichte berechnet werden. Da bei dem Verfahren nur lokale Extrema gesucht werden, muss es sich bei der gefundenen Gewichtung nicht um die optimale Lösung handeln.

Wichtig für diese Arbeit ist allerdings der Suchprozess einschließlich der Extraktion von Merkmalswerten. Dieses Retrieval-System unterscheidet Detektoren, Deskriptoren und Extraktoren bei der Ermittlung von Features. Dabei entsprechen Detektoren und Deskriptoren den im dritten Kapitel eingeführten Begriffen (vgl. Definition 3-6 und Definition 3-12). Der Extraktor übernimmt bei Pythia die Aufgaben von Detektoren und Deskriptoren, solange deren Zuordnung eineindeutig ist. Ist dies nicht der Fall, so nutzt der Extraktor für die Detektion eine externe Klasse, welche das `DetectorInterface` implementiert. Für jeden Feature-Typ existiert weiterhin ein Outputter, welcher das vom Extraktor erzeugte Datenformat für die Serialisierung vorbereitet und die Daten in einem spezifischen Format ausgibt. Derzeit existieren hierfür zwei XML-Formate. In Zukunft ist die Ablage der Features in einer Datenbank geplant. Die Ausgabedaten werden bei einer Anfrage durch einen Parser wieder eingelesen und die Features so für die Suche genutzt. Nachdem die Features des Suchbildes extrahiert und die Features der Bildkollektion geladen wurden, wird mittels einer Distanzfunktion die Unähnlichkeit

zwischen Anfrageobjekt und Kollektionsbildern berechnet. Dabei nutzen nicht alle Feature-Arten die gleiche Distanzfunktion. Mitunter sind für ein Feature auch mehrere Distanzfunktionen implementiert. Die berechneten Distanzen werden auf das Intervall [0,…,1] normiert und anschließend in einen Ähnlichkeitswert transformiert. Die Ähnlichkeitswerte werden durch eine CQQL-Formel aggregiert und dem Anwender werden zu seiner Anfrage die k-besten Kollektionsbilder zurückgeliefert. [Ber11 S. 3-5]

In Pythia stehen für die Analyse zurzeit zwei Werkzeuge bereit. Das Kommandozeilen-Tool `extractFeatures` ermittelt die Features, indem der beschriebene IR-Prozess durchlaufen wird. Als Ergebnis der Extraktion wird zu jedem Bild der Bildkollektion eine XML-Datei mit den ermittelten Features erstellt. Diese Datei wird von `calcSimilarities`, einem zweiten Tool zur Berechnung der Ähnlichkeit zwischen Anfrage und Bildsammlung, eingelesen und verarbeitet. Als Ausgabe liefert `calcSimilarities` eine Textdatei, welche das Ergebnis der Suchanfrage enthält. Der Aufbau dieser Datei richtet sich nach dem TREC-Format. TREC ist die Text Retrieval Conference, auf welcher verschiedene Richtlinien für den Bereich der Suchsysteme beschlossen werden. Ein Großteil der hier durchgeführten Untersuchungen, insbesondere die Zusammenhangsbetrachtung des im Anschluss vorgestellten Analysewerkzeugs, findet auf dieser Textdatei statt, weshalb zuerst die Struktur dieses Dokumentes näher erarbeitet werden soll. [Ber11 S. 5f]

Die Ergebnisdatei besitzt einen tabellarischen Aufbau, wobei in den Spalten die Anfrage-ID, der Bildname und dessen Rang, seine aggregierte Relevanzbewertung, sowie einige weitere Informationen stehen. In den Zeilen sind dagegen die Bilder absteigend nach ihrer aggregierten Relevanzbewertung angeordnet. Wichtig in dieser Arbeit ist unter anderem der Zusammenhang der verschiedenen Feature-Typen untereinander. Da die Auflistung der Bildähnlichkeitswerte aber in erster Linie Auskunft über die Ähnlichkeiten der Bildkollektion zum Anfragebild liefert, kann hieraus nicht direkt auf diesen Zusammenhang geschlossen werden. Diese Information ist aber implizit im Anfrageergebnis enthalten, da die Ähnlichkeitswerte nach jedem einzelnen Feature aufgeschlüsselt sind. Jeder im Retrieval-Ergebnis enthaltene Ähnlichkeitswert (Score) besitzt vier Dimensionen. Dazu zählt das Verfahren, mit dem der Score ermittelt wurde, das Bild, zu welchem er ermittelt wurde, das Anfragebild und zuletzt die Höhe des Ähnlichkeitswerts (vgl. Abbildung 5-5). Letztere Dimension bildet die Grundlage für weitere Berechnungen und die ersten drei Merkmale legen fest, inwiefern die einzelnen Werte verknüpft sind. Das Ergebnisdokument enthält folgende Struktur, welche Aussagen über diese vier Dimensionen erlaubt.

$$\begin{pmatrix} s_{11} & s_{21} & \cdots & s_{m1} \\ s_{12} & s_{22} & \cdots & s_{m2} \\ \vdots & \vdots & & \vdots \\ s_{1n} & s_{2n} & \dots & s_{mn} \end{pmatrix} \tag{4-1}$$

Die Matrix zeigt *m x n* Ähnlichkeitswerte s_{xy}, welche zu den *n* Bildern mithilfe der *m* Retrieval-Verfahren berechnet wurden. Auf der Grundlage dieser Struktur basieren alle der in dieser Arbeit entwickelten Verfahren zur Untersuchung des statistischen Zusammenhangs.

4.2 Entwurf eine Analysewerkzeugs

Wie bereits erläutert, soll in diesem Kapitel ein Analysewerkzeug, um die Beziehung zwischen verschiedenen Features zu analysieren, schrittweise entworfen und implementiert werden. Dazu werden zwei unterschiedliche Ansätze verfolgt, welche in Kapitel 5.1.2 näher erläutert werden. Das erste Verfahren betrachtet Differenzen der einzelnen Relevanzbewertungen verschiedener Extraktoren. Die Idee dahinter ist, die Abweichung der Bewertungen als ein Maß für die Unähnlichkeit der Verfahren anzusehen. Aus dieser Differenz wird ein Distanzwert berechnet und die Features auf dessen Grundlage geclustert. Features mit ähnlichen Bewertungen besitzen einen geringen Distanzwert und erhalten den gleichen Cluster zugewiesen. Ziel dieses Prozesses soll sein, das der Anwender einen Vertreter pro Feature-Gruppe für die Berechnung auswählt. Diese Auswahl kann beispielsweise auf Basis der benötigten Rechenzeit der Extraktionsmethode erfolgen, welche im fünften Kapitel ebenfalls untersucht wird. Das zweite Evaluierungsverfahren besitzt einige Analogien zum ersten Ansatz. Anstatt direkt die Distanzen zwischen den Verfahren zu berechnen wird zuerst eine Korrelationsanalyse durchgeführt. Die berechneten Korrelationskoeffizienten werden anschließend für das Clustering in Distanzwerte transformiert. Der Schwellwert für das Clustering wird durch die Korrelationsanalyse bestimmt, da lediglich ein starker Zusammenhang betrachtet werden soll (vgl. Kapitel 5.1).

Im vorgestellten Entwurfsprozess kommen verschiedene Modelle aus dem Bereich der semantischen Analyse und der Unified Modeling Language (UML) zum Einsatz. Der Prozess gliedert sich in mehrere Schritte. Zuerst wird der Ablauf beider Evaluationsansätze definiert, um daraus die Programmanforderungen ableiten zu können. Die geforderten Programmfunktionen werden im Anschluss in Form einer Funktionshierarchie gegliedert. Der nächste Schritt umfasst die Identifikation der Datenflüsse zwischen den einzelnen Funktionen. Auf der Grundlage dieser beiden Modelle werden die Programmfunktionen zu Funktionseinheiten zusammengefasst, welche die Basis für spätere Klassen bilden. Im letzten Entwurfsschritt wird der Ablauf bei konkreten Anwendungsszenarios modelliert.

Vor der Datenanalyse und der Wahl der Analysemethode müssen die zu analysierenden Daten geladen werden. Dazu wird das Programm über eine Konfigurationsdatei (config file) initialisiert, die von Pythia erzeugte Ergebnisdatei mittels eines Textparser gelesen und daraus die benötigten Informationen extrahiert. Nachdem die Daten eingelesen wurden, kann ein Zusammenhang zwischen den Features durch die beiden beschriebenen Evaluierungsmethoden untersucht werden. Hierzu müssen eine Funktionsauswahl, sowie die Berechnung der Distanz- beziehungsweise Korrelationswerte möglich sein. Zudem ist eine Ausgabe und Speicherung dieser Werte angedacht. Die berechneten Werte werden im Folgenden zur Gruppierung der Features genutzt. Hierzu werden Clusterverfahren verwendet. Das Clusterverfahren muss für den Nutzer auswählbar sein, ein Schwellwert für die Bildung eines Clusters muss festgelegt werden können und die Cluster müssen berechnet und zuletzt ausgegeben werden. Die folgende Funktionshierarchie fasst die erläuterten Programmfunktionen grafisch zusammen. Zur besseren Übersicht wurden die Funktionen einer Ebene durch gleiche Graustufen gekennzeichnet.

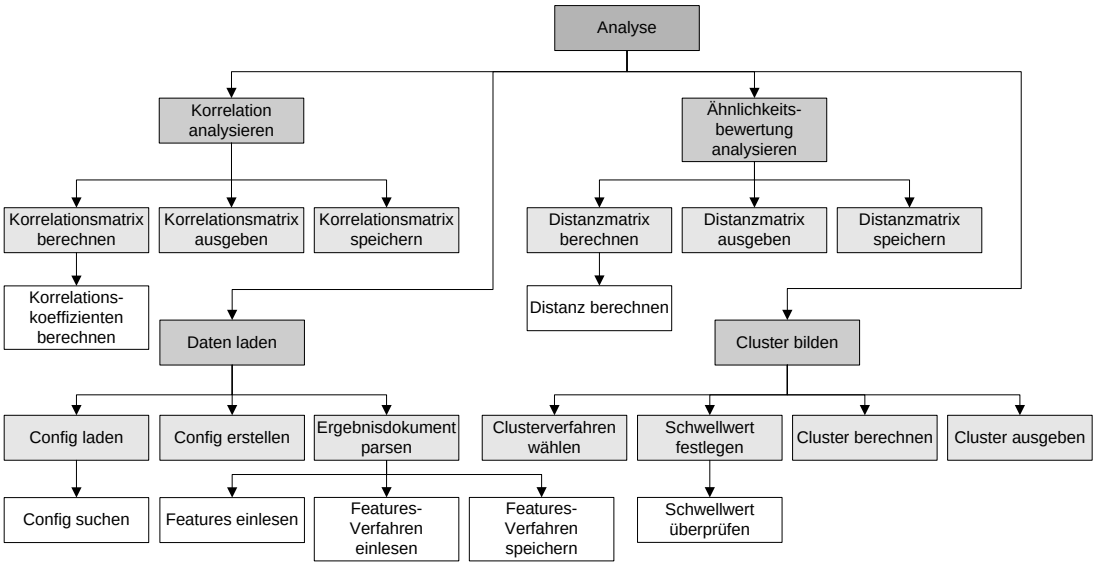

Abbildung 4-3: Funktionshierarchie des Analysewerkzeugs

Um mittels dieser Funktionsübersicht ein Programm implementieren zu können, bedarf es weiterer Entwurfsschritte. Beispielsweise müssen die Schnittstellen des Systems zur Außenwelt definiert werden. Das Analysewerkzeug besitzt im Wesentlichen zwei Zugangspunkte. Der erste beschreibt den Zugriff auf Daten der Festplatte und der zweite die Interaktion mit dem Anwender, welche entsprechend zu modellieren sind. Außerdem müssen die Datenströme zwischen den spezifizierten Funktionen betrachtet werden. Die Abbildung 4-4 zeigt ein Datenflussdiagramm (DFD), in welchem die verschiedenen Systembereiche markiert wurden. Dieses Diagramm beinhaltet den Datenfluss zwischen den einzelnen Funktionen und beschreibt die notwendigen Datenschnittstellen.

Im Folgenden soll der Datenfluss bei einem Anwendungsszenario beschrieben werden. Dabei wird das DFD im Uhrzeigersinn durchlaufen. Die hier dargestellte Analysefunktion bildet eine zentrale Komponente im Ablauf und regelt die Interaktion mit dem Anwender. Die physische Schnittstelle zur Festplatte ist durch Dateizugriffe modelliert. Zwischen Nutzer und Analysefunktion werden verschiedene Daten ausgetauscht. Der Anwender wählt die gewünschten Funktionen aus und gibt notwendige Parameter ein. Im Gegenzug werden ihm die Analyseergebnisse ausgegeben. Bei der Funktionsdefinition wurde bereits erläutert, dass zuerst die Konfigurationsdatei geladen werden muss, was im rechten Bereich der Abbildung 4-4 modelliert ist. An den Dokumentparser wird im Anschluss eine Leseanfrage geschickt und die Datei eingelesen. Der Parser liefert die bei der Extraktion verwendeten Verfahren und die Ähnlichkeitswerte des Retrieval-Ergebnisses zurück, wobei die Verfahren zusätzlich in einer Datei abgelegt werden. Der Anwender hat nun die Auswahl zwischen den beiden Evaluationsmethoden zu treffen. Abhängig von der Wahl wird die entsprechende Funktion aufgerufen und die Distanz- beziehungsweise Korrelationsmatrix berechnet. Die ermittelten Werte werden in Form einer Matrix an die Nutzerschnittstelle für die Ausgabe und an die Clusterfunktion für die Gruppierung der Features übermittelt. Zudem wird die Matrix in Dateiform für eventuelle weiterführende Analysen abgelegt. Bei dem distanzbasierten Clustering werden eine Auswahl des Clusterverfahrens und eine durch den Nutzer definierte Festlegung eines Schwellwertes für die Gruppierung von Extraktionsverfahren verwendet. Zu diesem Zweck werden neben den Distanzwerten noch die Art des Clusterings und der Schwellwert an die Gruppierungsfunktion übertragen.

Bei der Korrelationsanalyse erfolgt diese Auswahl nicht. Der notwendige Schwellwert wird hier durch die Korrelation vorgegeben, da nur Werte über einer gewissen Grenze als stark korrelierend gelten (vgl. Kapitel 5.1.2). Abschließend werden bei beiden Methoden Gruppen gebildet und dem Nutzer ausgegeben, sowie in einer Datei gespeichert.

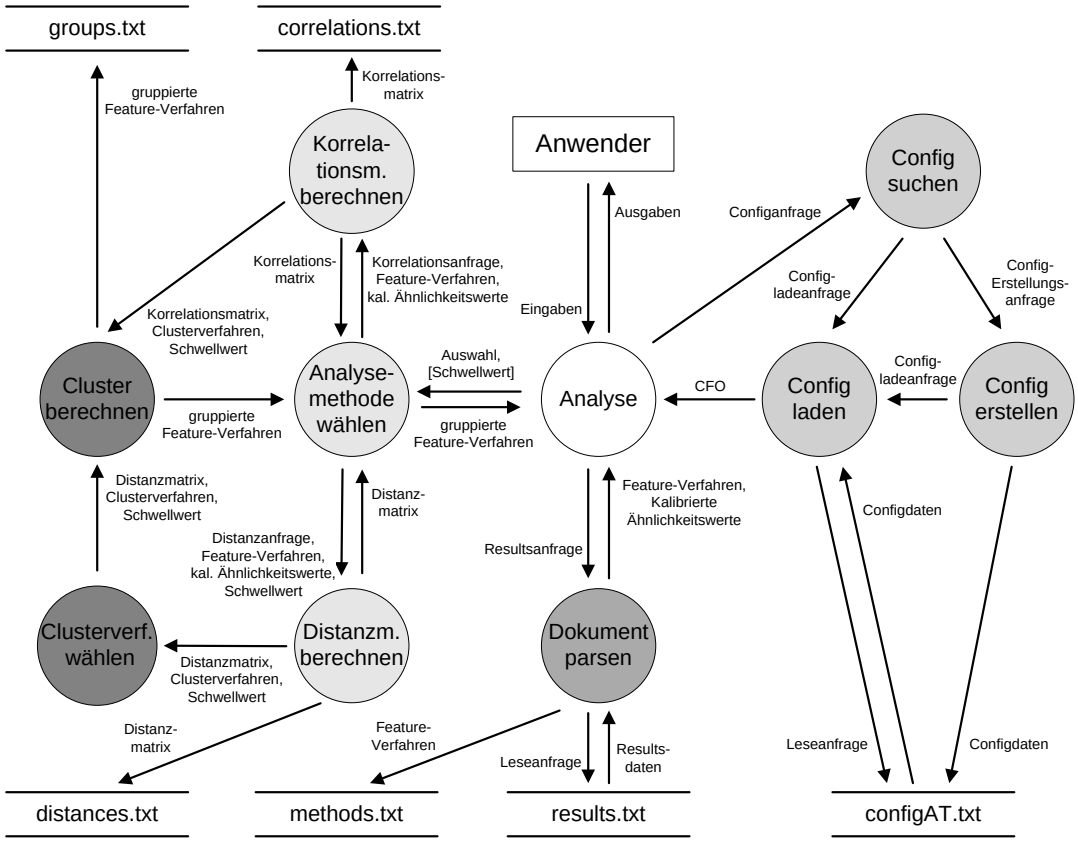

Abbildung 4-4: Datenflussdiagramm des Analysewerkzeugs

Die Kennzeichnung der Funktionen des Datenflussdiagramms lassen bereits eine mögliche Klasseneinteilung erkennen. Demnach sind insgesamt fünf Klassen angedacht. Die erste Klasse regelt die Interaktion mit dem Anwender und dient dem Datenaustausch zwischen den übrigen Komponenten. Zwei weitere Einheiten sind für das Laden der Konfigurationsdatei und das Parsen des Feature-Dokumentes zuständig. Die vierte Klasse berechnet die eigentlichen Distanz- und Korrelationswerte und übermittelt sie an die verschiedenen Stellen. Die letzte Komponente gruppiert die Feature-Methoden in Abhängigkeit der vorherigen Eingaben.

Nachdem die Klassen des Systems identifiziert wurden, sollen zum Abschluss zwei konkrete Anwendungsfälle betrachtet werden. Diese werden durch Sequenzdiagramme modelliert, wobei das Laden der Konfigurationsdatei nicht betrachtet wird. Im ersten Szenario wählt der Nutzer nach dem Parsen der Ergebnisdatei das distanzbasierte Clustering-Verfahren aus. Das System berechnet die Distanzmatrix und bietet dem Anwender eine weitere Funktionsauswahl an. Dieser entscheidet sich zunächst für die Ausgabe der Distanzmatrix, um anschließend Average-Link als Clusterverfahren zu wählen. Das System liefert das nach dem Average-Link-Verfahren berechnete Dendogramm und fordert zur Eingabe des Schwellwerts für das Bilden von Clustern auf. Nach Eingabe dieses Schwellwertes werden auf Basis des Dendogramms die Verfahrensgruppen berechnet und dem Nutzer ausgegeben.

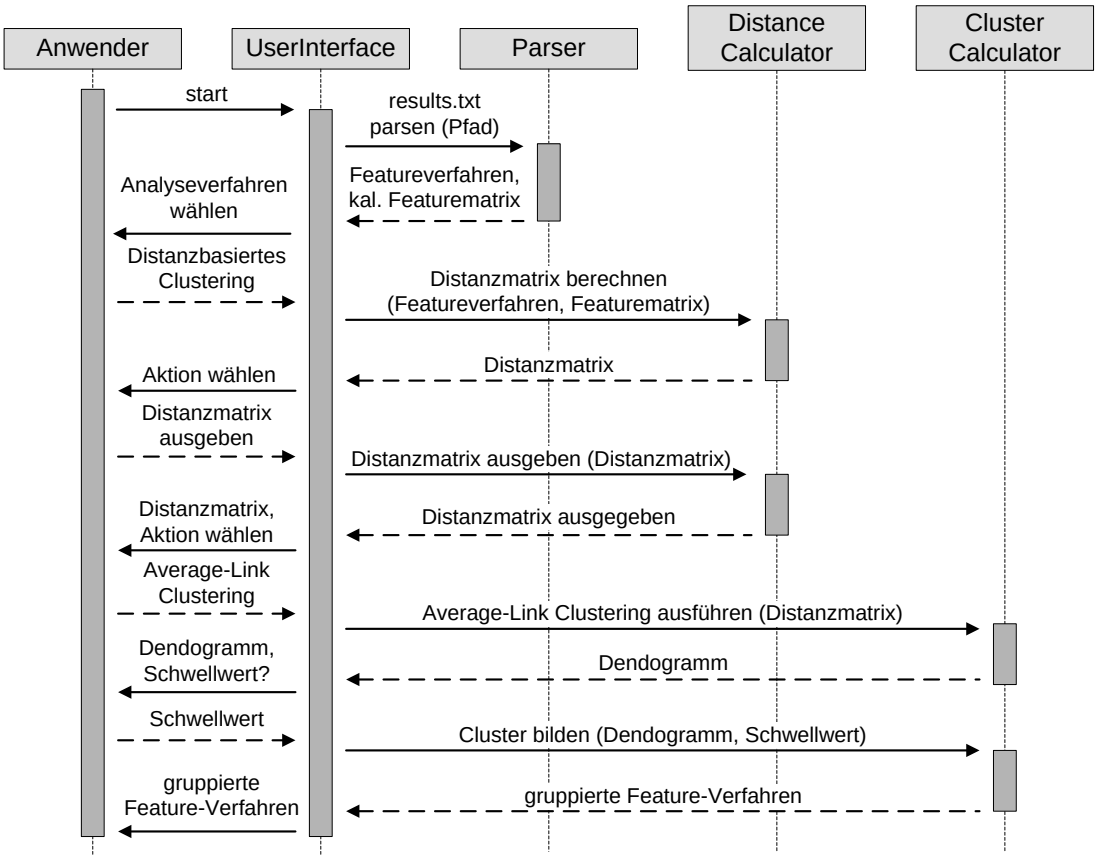

Abbildung 4-5: Szenario distanzbasiertes Clustering

Im zweiten Anwendungsfall wählt der Nutzer stattdessen die Korrelationsanalyse aus. Das System berechnet dieses Mal Korrelationskoeffizienten. Diese werden anschließend in Distanzwerte transformiert, damit sie für das Clustering verwendet werden können. Wie bereits erörtert, entfallen bei dieser Methode die Auswahl des Clusterverfahrens und die Eingabe eines Schwellwertes. Stattdessen werden sofort die gruppierten Features dargestellt.

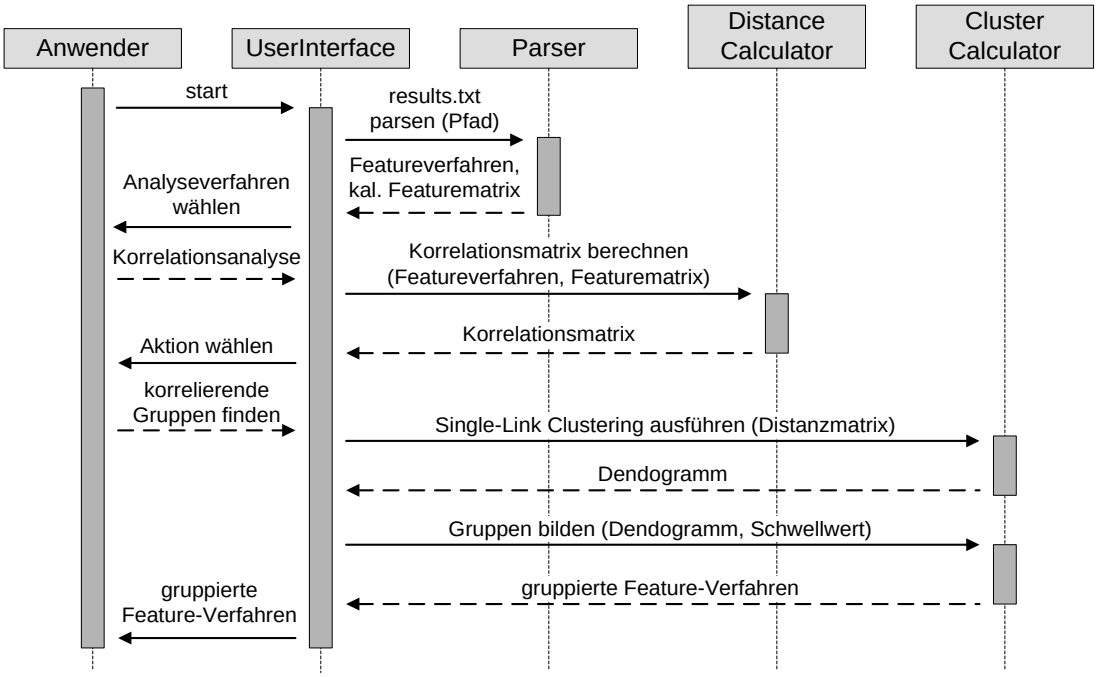

Abbildung 4-6: Szenario Korrelationsanalyse

Auf der Grundlage der hier getätigten Entwürfe wurde das Analysewerkzeug zur Untersuchung des Zusammenhangs der verschiedenen Extraktionsverfahren implementiert. Die Umsetzung soll im anschließenden Kapitel erläutert werden.

4.3 Implementierung

In diesem Kapitel soll die Implementierung des im vorangegangenen Abschnitt entworfenen Programms erläutert werden. Ähnlich wie die beiden anderen in Kapitel 4.1 vorgestellten Anwendungen von Pythia handelt es sich beim Analysewerkzeug um ein Kommandozeilen-Tool, welches über Konsoleneingaben gesteuert wird. Diese Analysesoftware umfasst zurzeit sechs Klassen und zwanzig Methoden. Im Folgenden soll die Implementierung der Klassen separat beschrieben und ihre Funktionsweise dabei erläutert werden.

UserInterface

Die primäre Aufgabe der Klasse `UserInterface` (`UI`) ist die Führung des Anwenders durch den Analyseprozess. Zu diesem Zweck werden dem Nutzer von dieser Komponente über die Konsole Menüoptionen und die berechneten Analyseergebnisse ausgegeben, sowie seine Eingaben entgegengenommen. Der folgende Screenshot zeigt das Hauptmenü des Analysewerkzeugs, welches der Benutzer nach Programmstart sieht.

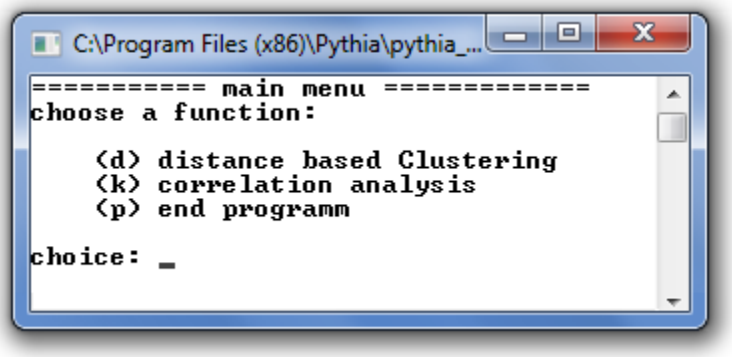

Abbildung 4-7: Screenshot des Hauptmenüs vom Analysewerkzeug

Durch Menüwahl entscheidet sich hier, welches Analyseverfahren genutzt werden soll. Die Wahl einer Funktion erfolgt durch Eingabe des davor stehenden Buchstaben. Dieser wird vom Programm abgefangen und interpretiert, um die entsprechenden Funktionen aufzurufen. Nach diesem Grundprinzip funktionieren auch alle weiteren Untermenüs des Nutzerinterfaces. Das nachfolgende Struktogramm verschafft einen Überblick über das Implementierungsprinzip.

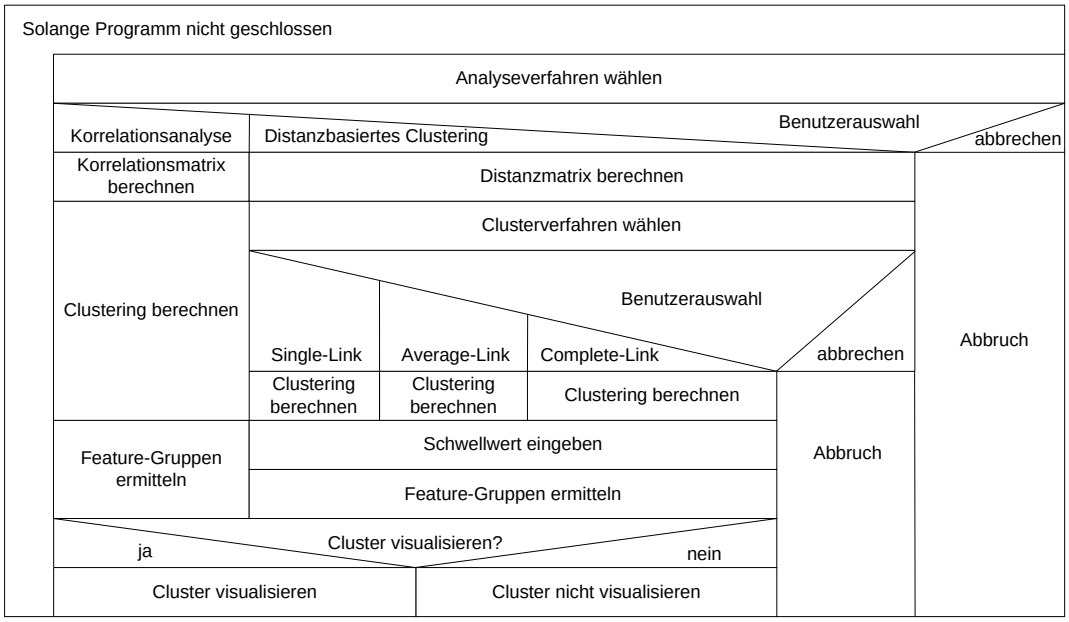

Abbildung 4-8: Struktogramm Menüstruktur

Neben der Interaktion mit dem Anwender erfüllt die Klasse `UI` eine weitere wichtige Aufgabe. Da sie die vom Anwender gewünschten Analyseprozesse startet, ist diese Klasse ein verbindendes Element der übrigen Programmkomponenten. Aus diesem Grund wird sie neben der Nutzerführung auch zum Datenaustausch zwischen anderen Klassen eingesetzt. Dabei ist das Nutzerinterface bewusst vom übrigen Programm separiert. Auf diese Weise ist es austauschbar und könnte bei späteren Weiterentwicklungen beispielsweise durch eine grafische Oberfläche oder einer Konfigurationsdateibasierten Programmsteuerung ersetzt werden.

DistanceCalculator

Beim Programmstart wird, wie beim Entwurf beschrieben, zuerst das Programm durch die Klasse `ConfigReader` initialisiert und von der Klasse `Parser` die Relevanzbewertungen eingelesen. Diese können nicht direkt dazu verwendet werden, um den Zusammenhang verschiedener Features zu evaluieren. Zuvor müssen sie in geeignete Maße transformiert werden. Dies übernimmt die Klasse `DistanceCalculator`, wobei abhängig vom ausgewählten Analyseverfahren entsprechend den Gleichungen 5-5 und 5-7 ein Distanz- oder Korrelationswert zu jeder Kombination zweier verglichener Features berechnet wird. Realisiert ist diese Kalkulation über mehrere verschachtelte Schleifen, in denen die Scores aus dem IR-Prozess nach der im fünften Kapitel beschriebenen Rechenvorschrift aufsummiert werden. Die Ermittlung der Distanzmatrix wird im folgenden Struktogramm beschrieben. Die Berechnung der Korrelationswerte folgt dem gleichen Schema. Zusätzlich werden diese allerdings im Anschluss für das angestrebte Clustering in Distanzwerte umgerechnet.

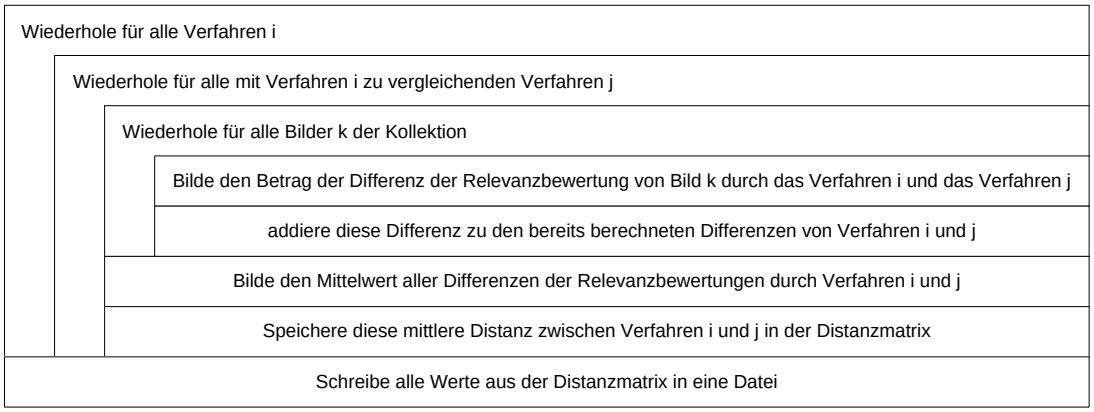

Abbildung 4-9: Struktogramm Berechnung der Distanzmatrix

Die berechneten Matrizen werden an die Klasse `UI` übergeben und hier zur Ausgabe und für weitere Berechnung genutzt. Zudem wird die Matrix noch in einer Datei abgelegt, um so auch für externe Kalkulationen nutzbar zu sein.

ClusterCalculator

Nach der Distanzberechnung werden die Distanzmatrizen an die Klasse `ClusterCalculator` übergeben, welche für die Gruppierung ähnlicher Features zuständig ist. Wie im Entwurf beschrieben geschieht dies unter Verwendung verschiedener Clusteralgorithmen. Hierfür wird die „C Clustering Library[3]“, eine C-Bibliothek, welche frei im Internet verfügbar ist, verwendet. Entwickelt wurde diese Sammlung an der Universität von Tokyo. Zudem ist sie kompatibel zu Windows-, Linux- und MacOS-Systemen.

Zum Bestimmen von Feature-Gruppen sind zwei Schritte notwendig. Zuerst wird nach einem zuvor gewählten Verfahren anhand eines hierarchischen Clusterings ein Dendogramm erzeugt, welches alle Extraktionsmethoden und deren Distanzen zueinander beinhaltet. Dazu wird von der Klasse `UI` die Methode `cluster` des `ClusterCalculators` mit der Distanzmatrix und der gewählten Clustermethode als Übergabeparameter aufgerufen. Die Wahl des Clusterverfahrens hängt dabei von der Art der Analyse ab. Bei einem distanzbasierten Clustering wählt der Nutzer das Verfahren über die Menüauswahl, wie im Abschnitt Nutzerinterface beschrieben, aus. Bei der Korrelationsanalyse gibt das Programm hingegen das Single-Link-Clustering vor. Das berechnete Dendogramm wird im Anschluss an die Methode `findCluster` übergeben. Diese erstellt in Abhängigkeit des angegebenen Schwellwertes die Feature-Gruppen. Eine Gruppe ist dabei als Struktur aus einer `ClusterID` und einem Vektor von Clusterelementen implementiert. Der Schwellwert *t* gibt die maximale Distanz zweier Cluster bei der Vereinigung an. Da im Dendogramm alle Cluster solange verschmolzen werden bis nur noch ein großer Cluster besteht, wird dieser Baum in der Methode `findCluster` solange durchlaufen, bis *t* überschritten wird. Alle im Anschluss getätigten Verschmelzungen würden ebenfalls oberhalb des Schwellwertes liegen. Somit enthält der aktuelle Stand bei Abbruch genau die Cluster, deren Elemente eine Distanz kleiner *t* zueinander besitzen. Ein Auszug des Gruppierungsprozesses ist im Folgenden dargestellt.

[3] http://bonsai.hgc.jp/~mdehoon/software/cluster/software.htm

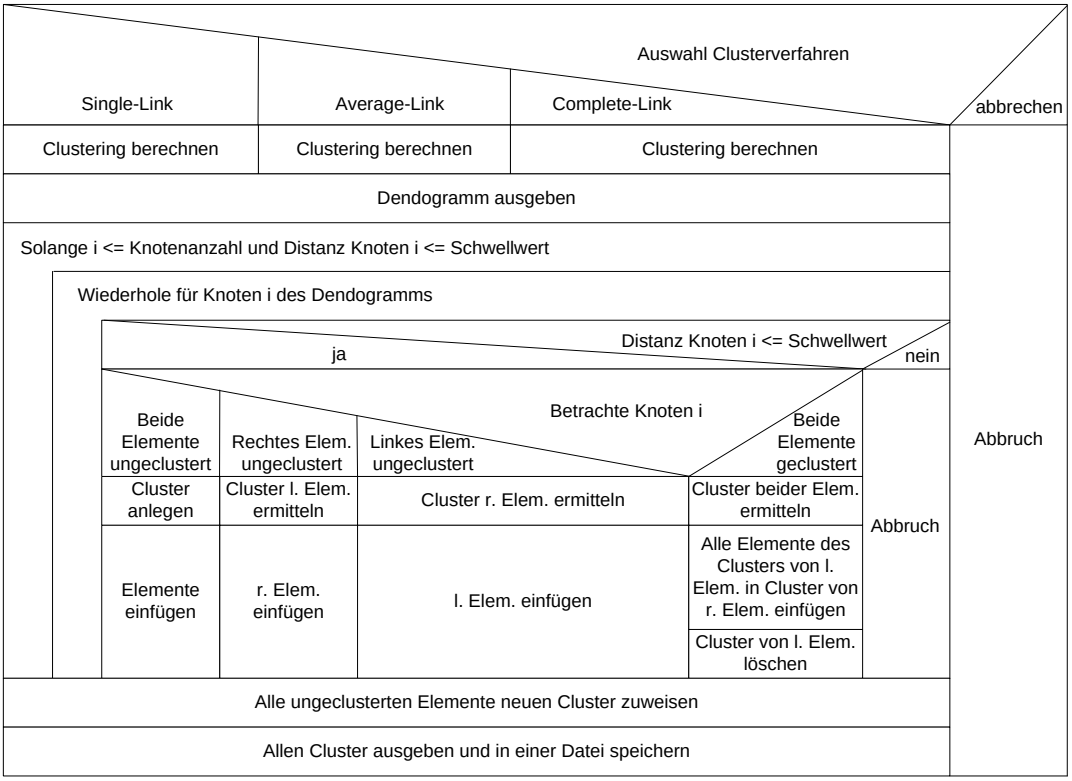

Abbildung 4-10: Struktogramm Clusterberechnung

Wie im Struktogramm gezeigt, werden beim Durchlaufen des Dendogramms alle möglichen Fälle abgefangen und die Clusterelemente entsprechend eingeordnet. Zur Fallüberprüfung wird die Knotennummer der beiden im Dendogramm vereinigten Cluster genutzt. Diese ist positiv durchnummeriert, wenn das Knotenelement bislang keinem Cluster zugewiesen wurde. Stammt ein Clusterknoter bereits aus einer Verschmelzung und besitzt somit mindestens zwei Elemente ist sie negativ durchnummeriert. Neben der Dateiausgabe können die Cluster auch grafisch visualisiert werden. Hierfür wird die Graphvisualisierungssoftware aiSee3 der Firma AbsInt[4] verwendet. Diese ist kompatibel zu Windows, Linux und MacOS und kann kostenfrei heruntergeladen werden.

[4] http://www.absint.com/aisee/index_de.htm

5. Analyse der Detektions- und –Deskriptionsverfahren

Ziel dieser Arbeit ist es verschiedene Aspekte der Detektions- und Deskriptionsalgorithmen, welche im dritten Kapitel eingeführt wurden, zu untersuchen. Dies beinhaltet in erster Linie die Qualität des Suchergebnisses und die dafür benötigte Rechenzeit. Aber auch weitere Kriterien, wie der statistische Zusammenhang der Feature-Mengen, sind bei der Auswahl von bei einer Suche zu verwendenden Verfahren wichtig.

In diesem Kapitel wird der Einsatz der Features im Retrieval-System Pythia nach verschiedenen Gesichtspunkten untersucht, um so eine optimale Wahl treffen zu können. Dazu werden einige Evaluationsmethoden vorgestellt und bewertet. Jede der ausgewählten Methoden analysiert einen anderen Bereich, wie die benötigte Rechenzeit für die Extraktion der Features und die Berechnung der Distanzwerte, die Qualität des Retrieval-Ergebnisses oder die Untersuchung eines statistisch messbaren Zusammenhangs zwischen den Relevanzbewertungen der einzelnen Verfahren. Auf Grundlage der getätigten Analysen sollen Aussagen darüber getroffen werden, welche Verfahren gute Ergebnisse liefern, schnell zu berechnen sind oder sinnvoll kombiniert werden können. Die Ergebnisse werden in Form von Diagrammen und Korrelationsmatrizen veranschaulicht und abschließend in einer kurzen Zusammenfassung beurteilt. Vor der Präsentation der Analyseergebnisse werden nun einige Evaluierungsansätze diskutiert und auf ihre Tauglichkeit für die verschiedenen Analyseziele hin bewertet. Einige dieser Verfahren wurden bereits im vierten Kapitel aus Implementierungssicht vorgestellt.

5.1 Evaluierungsverfahren

Bei der Evaluierung von Retrieval-Systemen sind vor allem zwei Faktoren entscheidend. Einerseits ist die für die Suche benötigte Zeit wichtig. Hierfür existieren verschiedene Messwerte, wie beispielsweise die Gesamtzeit, welche zwischen dem Formulieren der Anfrage und dem Zurückliefern des Suchergebnisses vergeht, oder aber die Zeit bis zum Finden des ersten korrekten Suchergebnisses. Das zweite wesentliche Kriterium für die Güte von Bildsuchsystemen ist die Qualität des Anfrageergebnisses. Diese ist deutlich schwieriger zu evaluieren, da sie im Gegensatz zu der Zeit nicht ohne weitere Maßnahmen gemessen werden kann. Die meisten Evaluierungsverfahren nutzen zur Beurteilung der Ergebnisqualität die Relevanz des Suchergebnisses für die Anfrage als zugrundeliegendes Bewertungskriterium. Dabei beschreibt die Relevanz die Beziehung zwischen der Suchanfrage und dem Suchergebnis näher.

Definition 5.1 (Relevanz)
Relevanz ist ein Maß für die inhaltliche Übereinstimmung zwischen der Anfrage (Informationsbedarf) und dem Suchergebnis (Informationsangebot) [Sal83 S. 163]

Das Hauptproblem beim Retrieval besteht in der Definition was relevant in einem bestimmten Kontext ist. Dabei ist für jeden Nutzer etwas anderes als relevant zu bezeichnen. Dies hängt zum einen stark vom jeweiligen Kenntnisstand des Anwenders ab, was an einem Beispiel aus dem Text-Retrieval verdeutlicht werden soll. Ein Profi auf einem Themengebiet möchte keine ausführliche Einführung finden und für einen Anfänger ist ein Fachartikel oft nur wenig hilfreich. Des Weiteren kann sich die Relevanz eines Suchergebnisses während der Suche ändern. Mitunter wird ein späteres Ergeb-

nisdokument erst durch den Informationsgewinn der bisherigen Suche relevant. Diese Probleme aus anderen Retrieval-Bereichen existieren im Bild-Retrival nicht in diesem Maße, da die Spanne zwischen Anfänger- und Fortgeschritteneninformation nicht so groß ist. Die wesentlichen Bildinhalte, wie die dargestellten Bildobjekte, sind für den Betrachter meist leicht zu erschließen. Aber auch im CBIR tragen zusätzliche Faktoren, wie das Wissen um den Bildkontext, mit zur Relevanz der Suchobjekte bei.

Neben diesen beiden Bewertungskriterien ist auch ein einheitlicher Bewertungsmaßstab sehr wichtig. Wenn jede Evaluierung auf anderen Grundannahmen beruht, sind diese untereinander nicht vergleichbar. Ein standardisiertes Evaluierungsverfahren schließt sowohl eine Standardtestdatenbank, auf welcher das CBIR-System arbeitet, als auch einheitliche Standardbewertungsmaße mit ein. In der Literatur existieren hierfür verschiedene Ansätze, wovon sich allerdings bisher keiner als weltweiter Standard durchgesetzt hat. Es existiert eine ganze Reihe von Bewertungsmaßen. Die meisten von ihnen beziehen sich auf die Relevanz des Suchergebnisses für die Anfrage. Zu den wichtigsten Evaluierungsmaßen wird im Kapitel 5.1.1 Bezug genommen. Auch für standardisierte Testdatenbanken existieren bereits Lösungsvorschläge. Die bekanntesten Bild-DBs sind `Brodatz` und `Vistex`, welche überwiegend verschiedene Texturen beinhalten, sowie `Corel`. Bei `Corel` sind jeweils 100 ähnliche Bilder in einer sogenannten Klasse zusammengefasst. Beispielkategorien sind *Afrika*, *Strand*, *Gebäude*, *Busse*, *Dinosaurier*, *Blumen*, *Elefanten*, *Pferde*, *Essen*, *Berge* und noch viele weitere. Manche dieser Bildbibliotheken stehen frei im Internet zur Verfügung und können unentgeltlich zur Evaluierung von Bild-Retrieval-Systemen genutzt werden. `Corel` existiert in verschiedenen Ausführungen. `Corel1000` umfasst 10 dieser Bildkategorien mit jeweils 100 Bildern pro Klasse. Es existiert auch die größere Version `Corel10000` mit 100 Kategorien. [Des03 S. 57-59]

Allerdings genügt die Verwendung einer standardisierten DB allein nicht für vergleichbare Evaluierungsergebnisse, wie Henning Müller et al. in „The truth about corel“ zeigen. Selbst, wenn die gleiche Bildkollektion mit den gleichen Messwerten genutzt wird, können bei einem CBIR-System durch die Wahl der Anfragen oder anderer Faktoren völlig unterschiedliche Resultate erzielt werden. Das größte Problem der `Corel`-Datenbank besteht in der Vielfalt der Daten. Die Firma Corel bietet über achthundert verschiedene Foto-CDs mit jeweils hundert Bildern an. Es ist dem Entwickler bei der Bewertung seines Systems selbst überlassen, welche Bildkategorien er wählt. Die Wahl der Bildkollektion ist die erste Möglichkeit auf das Retrieval-Ergebnis Einfluss zu nehmen. Bei manchen Arbeiten wird speziell die Sammlung genutzt, für welche das CBIR-System die besten Ergebnisse liefert. Mitunter werden sogar Bilder aus einer Kollektion entfernt, da sie von System trotz ihrer Irrelevanz für eine Kategorie gefunden werden und so das Evaluationsergebnis verschlechtern. Auch die Erstellung neuer Bildgruppen birgt viele Risiken. Beim Erstellen neuer Kollektionen bewertet der Anwender welche Bilder in einem bestimmten Kontext relevant sind. Die Neubewertung von Relevanz ist höchst subjektiv und sollte nie von Entwickler des zu testenden Systems gemacht werden. Eine weitere Möglichkeit der Einflussnahme ist die Wahl des Anfragebildes. Dieses sollte die Bildklasse gut repräsentieren und sich nicht stark von allen anderen Bildern unterscheiden. Allgemein gilt, dass für eine objektive und vergleichbare Bewertung von Detektions- und Deskriptionsverfahren zur Bildsuche eine einheitliche DB mit einer fest definierten Anfragekollektion verwendet werden sollte. Diese sollte weiterhin ebenso wie die Beurteilung der Relevanz der Bildsammlung (relevance judgment) von

einem unabhängigen Institut vorgegeben und nicht durch den Entwickler eines Systems definiert werden. [Mue02 S. 1f]

In dieser Arbeit wird die Bildsammlung 101Categories verwendet. Diese wurde vom California Institut of Technologie herausgegeben und besteht aus 101 Bildkategorien, wovon jede zwischen vierzig und achthundert Bildern beinhaltet.

Im Folgenden werden verschiedene Methoden vorgestellt, um die Retrieval-Qualität und die Rechenzeit zu evaluieren. Begonnen wird dabei mit der Vorstellung der klassischen Retrieval-Maßen, welche zur Bewertung verschiedener Suchsysteme eingesetzt werden. Im weiteren Verlauf wird dann speziell auf die Untersuchung eines Zusammenhangs zwischen einzelnen Features eingegangen. Hintergrund ist, dass bei gleichartigen Verfahren Rechenzeit gespart werden kann, indem weniger Features bzw. Distanzen berechnet werden, ohne dass sich das Retrieval-Ergebnis stark ändert.

5.1.1 Information-Retrieval-Maße

Einer der wichtigsten Maßstäbe für die Qualität eines Suchsystems ist die Erwartungstreue des Suchergebnisses. Ist die Kluft zwischen den Erwartungen des Nutzers und dem tatsächlichen Retrieval-Ergebnis zu groß, kennzeichnet dies eine mindere Qualität. Diese kann beispielsweise durch Nutzerbefragungen erhoben werden. Eine technische Möglichkeit das Suchergebnis qualitativ zu beurteilen sind Information-Retrieval-Maße. Dabei handelt es sich um Bewertungsmaße, die im Bereich des Retrievals eingesetzt werden, um die Güte von IR-Systemen zu klassifizieren. Häufig basieren diese Werte auf der Relevanz der gefundenen Suchobjekte für die Anfrage. Beispielsweise ist bei der Evaluierung wichtig, wie schnell relevante Objekte gefunden werden oder wie hoch die Relevanz der interessanten Bilder vom System bewertet wird.

Definition 5.2 (IR-Maße)
IR-Maße sind objektive und quantitative Messwerte, mit deren Hilfe die Qualität von Suchergebnissen im Bereich der Information Retrieval evaluiert werden kann. [Des03 S. 57]

Zur Messung der Retrieval-Effektivität von IR-Systemen hat die Informationswissenschaft in den letzten fünfzig Jahren ein umfangreiches Instrumentarium entwickelt. Durch Anpassung und Weiterentwicklung etablierter Verfahren können viele dieser Evaluierungsmaße auch außerhalb des Text-Retrievals angewendet werden. Neben der Unterteilung des Suchergebnisses anhand ihrer Relevanz kann diese Unterscheidung weiter gesplittet werden. Dazu wird jeweils eine Aussage darüber getroffen, ob das Bild im Suchergebnis zurückgeliefert wird oder nicht. Daraus ergeben sich vier unterschiedliche Trefferbereiche (hits, misses, noise, rejects), welche in der Tabelle 5-1 dargestellt werden. [Lew07 S. 6]

	relevant	**nicht relevant**	**total**
im Ergebnis	a (hits)	b (noise)	a+b (alle gefundenen Obj.)
nicht im Ergebnis	c (misses)	d (rejects)	c+d (alle nicht gefundenen Obj.)
total	a+c (alle rel. Obj.)	b+d (alle nicht rel. Obj.)	a+b+c+d (alle Objekte)

Tabelle 5-1: Ergebnisbereiche beim Information Retrieval

Das ideale Suchergebnis würde ausschließlich aus allen relevanten Objekten bestehen. Es erreicht also sowohl hinsichtlich der Präzision der Suchergebnisse als auch hinsichtlich ihrer Vollständigkeit das Maximum. Ein solches System existiert allerdings, vor allem im Bereich des Bild-Retrievals durch den komplizierten Informationsextraktionsprozess, nicht. Durch die Retrieval-Maße kann evaluiert werden, inwieweit das Suchergebnis an dieses Ideal heran reicht. Der gemeinhin am häufigsten betrachtete Performancewert ist die Precision. Sie gibt den Anteil der relevanten Treffer an allen im Ergebnis befindlichen Objekten an. Die Precision kann zum Beispiel durch Pooling-Verfahren erhoben werden. Dabei bewerten Juroren für jeden Treffer ob er relevant für die Anfrage ist. Diese Bewertung wird ausgezählt und der Anteil gebildet. Bei großen Treffermengen wird die Precision nur bis zu einem Cut-off-Wert bestimmt, da die ersten Ergebnisse des Rankings für den Anwender am bedeutendsten sind. Das Hauptproblem dieser Methode liegt darin, dass die Relevanzbeurteilung nicht immer auf eine zweiwertige Einteilung abstrahiert werden kann.

Definition 5.3 (Precision)

Die Precision P ist die Wahrscheinlichkeit, mit der ein gefundenes Objekt relevant ist. Diese ergibt sich aus dem Anteil der relevanten Objekte am gesamten Ergebnis [Lew07 S. 6]

$$P = \frac{|Menge\ gefundener\ Bilder \cap Menge\ relevanter\ Bilder|}{|Menge\ gefundener\ Bilder|} = \frac{|a|}{|a+b|}$$

Ein weiteres klassisches Retrieval-Maß ist der Recall. Er ergibt sich aus dem Anteil relevanter Treffer an der Gesamtanzahl aller relevanten Objekte. Der Recall beschreibt somit die Vollständigkeit eines Suchergebnisses. Umso größer die verwendete Datensammlung ist, desto schwieriger ist es die Gesamtanzahl aller relevanten Objekte zu bestimmen. Häufig wird diese durch Stichproben und Hochrechnungen angenähert. Bei der Verwendung kleinerer Testsysteme, wie bei dieser Arbeit, ist die Anzahl allerdings leicht bestimmbar und bereits durch die ground truth vorgegeben. [Lew07 S. 6]

Definition 5.4 (Recall)

Der Recall R ist die Wahrscheinlichkeit, mit der ein relevantes Dokument gefunden wird. Diese ergibt sich aus dem Anteil der relevanten Objekte im Ergebnis an der Gesamtanzahl aller relevanten Objekte. [Lew07 S. 6]

$$R = \frac{|Menge\ gefundener\ Bilder \cap Menge\ relevanter\ Bilder|}{|Menge\ relevanter\ Bilder|} = \frac{|a|}{|a+c|}$$

Des Weiteren hat sich der Fallout als ein klassisches Bewertungsmaß im IR-Umfeld etabliert. Er gibt den Anteil der ausgegebenen, aber nicht relevanten Objekte an der Gesamtanzahl aller nicht relevanten Objekte an und bildet somit das Gegenstück zum Recall. Der Fallout beschreibt die Ungenauigkeit eines Suchergebnisses.

Definition 5.5 (Fallout)

Der Fallout F ist die Wahrscheinlichkeit, mit der ein irrelevantes Dokument gefunden wird. Diese ergibt sich aus dem Anteil der nicht relevanten Objekte im Ergebnis an der Gesamtanzahl aller nicht relevanten Objekte. [Lew07 S. 6]

$$F = \frac{|Menge\ gefundener\ Bilder \cap Menge\ irrelevanter\ Bilder|}{|Menge\ irrelevanter\ Bilder|} = \frac{|b|}{|b+d|}$$

Zur besseren Anschauung werden Precision *P* und Recall *R* oft in einem PR-Graphen kombiniert, welcher angibt aus wie vielen relevanten und irrelevanten Objekten sich das Suchergebnis zusammensetzt. Dabei werden die ersten *n* Treffer ausgewählt und für diese nacheinander *P* und *R* bestimmt. Da die Betrachtung jedes weiteren Bildes im Ranking die Genauigkeit hebt oder senkt, hat der resultierende Verlauf eine Sägezahnform. Diese wird durch die Berechnung der maximalen Precision zu jedem Recall-Wert geglättet, um so eine Funktion der Precision über den Recall zu erhalten. Im letzten Schritte werden die Funktionsverläufe verschiedener Anfragen kombiniert und der arithmetische Mittelwert gebildet. Gewöhnlich erfolgt eine Angabe der interpolierten Precision an den 11 Recall-Level 0%, 10%, ..., 100%. Dies ermöglicht es zwei Systeme miteinander zu vergleichen. Die Abbildung 5-1 zeigt den PR-Graph für zwei hypothetische Systeme. [Des03 S. 57f]

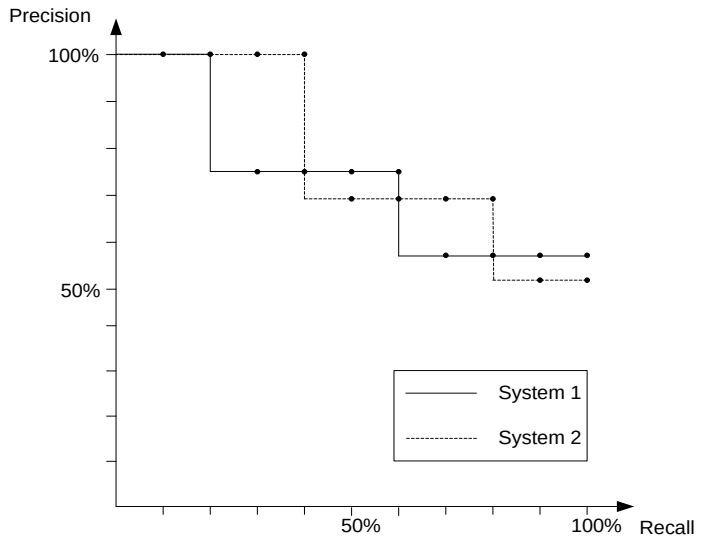

Abbildung 5-1: Vergleich des Recall-Precision-Graphen zweier Systeme

Dieses Beispiel zeigt einen stark vereinfachten Verlauf eines PR-Graphen für zwei unterschiedliche Systeme. Beide besitzen eine Precision, die stets über 50% liegt. Für die meisten Anwendungen ist das System 2 zu präferieren, da auf den vorderen Ranking-Positionen im Vergleich zum ersten System mehr relevante Objekte platziert sind und der Nutzer so das Gesuchte schneller findet. Beim ersten System werden dagegen zuerst alle relevanten Objekte ermittelt, was bei vielen in der Praxis genutzten Suchsystemen allerdings von untergeordneter Bedeutung ist.

Neben den klassischen IR-Maßen existieren noch eine Reihe weiterer Werte, die das Suchergebnis genauer klassifizieren. Henning Müller et al. stellten in „Pattern Recognition Letters“ [Mue02] verschiedene dieser Bewertungsmaße vor. Eine grobe Übersicht bietet die folgende Tabelle.

Symbol	Bedeutung
N_R	Anzahl der relevanten Dokumente
t	Zeit, die zum Ausführen der Anfrage benötigt wird
$Rank_1$	Rang des ersten relevanten Bildes im Suchergebnis
$\overline{Rank}, \widetilde{Rank}$	Mittelwert und normalisierter Mittelwert des Rangs relevanter Bilder
$P(1), P(2), ..., P(N_R)$	Precision auf Rang 1 bis zum Rang mit dem letzten relevanten Bild
$R(100), R(P=0.5)$	Recall auf Rang 100 und Recall bei eine Precision von 50%

Tabelle 5-2: Auswahl von Performance-Messwerten mit kurzer Erläuterung [Mue02 S. 3f]

Die Berechnung aller hier aufgeführten Werte ist in der Regel für die Evaluierung eines Retrieval-Verfahrens nicht notwendig. Auf der einen Seite ist die Bedeutung mancher Kennziffern geringer als die anderer. Beispielsweise sind Messwerte, die vor allem die Qualität der zuerst gerankten Suchergebnisse beurteilen, meist wichtiger. Kennzahlen wie *R(P=0.5)* oder *P(R=1)* sind für eine Beurteilung meist uninteressant. Auf der anderen Seite korrelieren viele der in Tabelle 5-2 aufgeführten Werte stark, was in der nachfolgenden Tabelle veranschaulicht ist. Sie basiert auf den Forschungsergebnissen von Thomas Deselaers, welcher in „Features for Image Retrieval" [Des03] unter anderem den Zusammenhang verschiedener Features evaluiert. Seine Untersuchungen wurden dabei auf der WANG-Bilddatenbank, welche auf der DB von Corel mit 1000 Bildern basiert, und auf der medizinischen Bildsammlung IRMA-1617, welche 1617 Röntgenbildern umfasst, durchgeführt. [Des03 S. 59f]

	P(1)	ER	P(50)	R(P=0.5)	R(100)	$Rank_1$	$\widetilde{Rank}$	P(R=P)	PR-Fläche	P(R=0)	P(R=0.1)	P(R=0.5)	P(R=0.9)	P(R=1)
P(1)	100	-100	94	62	94	-91	-90	86	90	98	95	83	73	31
ER		100	-94	-62	-94	91	90	-86	-90	-98	-95	-83	-73	-31
P(50)			100	57	91	-88	-87	77	81	93	93	73	62	14
R(P=0.5)				100	48	-58	-54	55	58	61	48	55	52	21
R(100)					100	-96	-97	94	96	97	95	92	84	50
$Rank_1$						100	99	-94	-95	-96	-86	-92	-85	-54
$\widetilde{Rank}$							100	-97	-97	-96	-88	-95	-89	-59
P(R=P)								100	99	92	-85	99	95	72
PR-Fläche									100	95	88	98	94	67
P(R=0)										100	94	89	81	42
P(R=0.1)											100	82	72	31
P(R=0.5)												100	96	76
P(R=0.9)													100	81
P(R=1)														100

Tabelle 5-3: Korrelation verschiedener IR-Maße [Des03 S. 56]

Ein etablierter Performancewert ist beispielsweise die Fehlerrate *ER* (error rate), welche mit den meisten anderen vorgestellten Werten stark korreliert und so viele Aspekte des Retrieval-Ergebnisses bewertet. Sie ergibt sich aus dem Reziproken von *P(1)*. [Des03 S. 59]

Für den Anwender ist es meist wichtiger viele relevante Ergebnisse unter den ersten Suchergebnissen zurückgeliefert zu bekommen, anstatt alle relevanten Bilder im Ergebnis vorzufinden. Das IR-Maß, welches dieses Verhalten am besten bewertet ist die Precision. Aus diesem Grund wird bei der Evaluierung in dieser Arbeit vorrangig die Precision betrachtet. Diese wird zum einen an bestimmten Schwellen erhoben, zum anderen wird die „mean average precision" (MAP) berechnet. MAP ist eines der häufigsten Bewertungsmaße im Retrieval-Bereich. Es beinhaltet sowohl Recall-, als auch Precision-orientierte Aspekte. Sie ergibt sich aus dem Mittelwert der durchschnittlichen Precision jeder Anfrage und wird folgendermaßen berechnet.

$$MAP = \frac{1}{Q}\sum_{q \in Q}\left(\frac{1}{N_r}\sum_{n=1}^{N_r} P_q(R_n)\right) \quad (5\text{-}1)$$

Zuerst wird der Mittelwert der Precision-Werte aller relevanten Objekte zu einer Anfrage berechnet. Dafür wird die Precision jedes relevanten Objektes r_n aufsummiert und durch die Anzahl N_r aller relevanten Objekte geteilt. Die average precision aller Anfragen q wird im Anschluss ebenfalls aufsummiert und durch die Anzahl Q aller Anfragen geteilt.

5.1.2 Mathematische Verfahren zur Evaluierung des statistischen Zusammenhangs zwischen Features

Im Gegensatz zu den Retrieval-Maßen, welche in erster Linie die Ergebnisqualität und die Zeit, die zum Finden eines Ergebnisses für unterschiedliche Extraktionsmethoden notwendig ist, untersuchen, wird bei der angestrebten statistischen Betrachtung der Zusammenhang zwischen den Verfahren beurteilt. Besteht ein solcher Zusammenhang, kann bei gleichbleibender Qualität des Suchergebnisses auf einzelne Features verzichtet werden. Dies führt zu der erwähnten Einsparung in der Rechenzeit. Neben diesem offensichtlichen Vorteil einer Korrelationsanalyse existieren noch eine Reihe weiterer Gründe für eine solche statistische Beurteilung, welche im Folgenden diskutiert werden sollen.

Bei Pythia werden meist mehrere Feature-Extraktionsverfahren kombiniert eingesetzt. Die Ähnlichkeitsbewertung eines Kollektionsbildes ergibt sich aus den aggregierten Ähnlichkeitswerten jedes einzelnen Verfahrens. Eine Konsequenz daraus ist, dass bei Erhöhung der Anzahl der zu berechnenden Methoden nicht nur der Mehraufwand für die Extraktion der Features und die Berechnung der Distanzen, sondern auch für das Aggregieren der einzelnen Distanzwerte vergrößert wird. Dies wirkt sich zudem auf den Relevanz Feedback-Prozess aus. In Kapitel 4-1 wurde erläutert, inwiefern der Anwender Einfluss aus die Relevanzbewertung des Systems nehmen kann. Für eine solche Bewertung werden die einzelnen Distanzen unterschiedlich gewichtet, damit das Anfrageergebnis näher an den vom Nutzer als relevant bewerteten Kollektionsbildern liegt. Der notwendige Aufwand zum Lernen der Aggregationsgewichte erhöht sich ebenfalls mit zunehmender Zahl der Features. Des Weiteren ergibt sich bei der Aggregation von mehreren Verfahren ein zusätzlicher Aspekt. Werden bei einer Anfrage viele stark korrelierende Features und nur wenige schwach korrelierende Verfahren verwendet, ist der Einfluss auf das Suchergebnis der schwach korrelierenden Features relativ gering. Bei einer Suchanfrage ist es dagegen erwünscht möglichst verschiedene Kriterien der Bilder zu betrachten, um so eine umfassende Einschätzung der Ähnlichkeit der Kollektionsbilder zum Anfragebild zu erhalten. Um den Einfluss jedes unterschiedlichen Verfahrens zu maximieren, sollten daher gleiche Verfahren erkannt und eliminiert werden. Ein weiteres Problem, welches bei vielen Retrieval-Anwendungen im Zusammenhang mit zu vielen zu betrachtenden Dimensionen immer wieder auftaucht ist der Fluch der hohen Dimensionen. Dabei handelt es sich um das Phänomen, dass alle effizienten Indexstrukturen bei höheren Dimensionen versagen. Bei den meisten Baumverfahren sind dies beispielsweise in etwa zwanzig Dimensionen. Die Ursache des „Fluchs“ ergibt sich aus der Distanzverteilung und kann nicht vollständig behoben,

sondern maximal verringert werden. In Folge dessen ist bei hoher Dimensionalität eine lineare Suche meist schneller als die Verwendung von Indexstrukturen. Könnte die Dimensinalität hingegen durch die Beschränkung auf einige schwach korrelierende Extraktionsverfahren reduziert werden, ist der Einsatz von schnelleren spezifischen Indexstrukturen für eine geeignete Menge von Features und somit eine Steigerung der Effizienz möglich.

An dieser Stelle wird deutlich, welche Vorteile eine Reduzierung stark korrelierender Verfahren bietet. Aus diesem Grund werden zunächst verschiedene Bewertungsansätze diskutiert, um daraus eine Evaluierungsmöglichkeit des Zusammenhangs zwischen den Extraktionsverfahren zu erstellen.

Diskriminanzanalyse

Im Bereich der Statistik existieren eine Reihe von Verfahren zur Evaluierung und Klassifizierung, aber nicht alle sind gleichermaßen für die durchzuführende Bewertung der Features geeignet. Im Folgenden sollen einige gängige Verfahren auf ihre Eignung hin überprüft werden. Eines der am häufigsten eingesetzten statistischen Analyseverfahren zur Klassifikation ist die Diskriminanzanalyse. Dabei werden Objekte auf der Grundlage ihrer individuellen Merkmalskombination in definierte Klassen eingeordnet. In dieser Arbeit wird der Begriff des Klassifikationsverfahrens dementsprechend aufgefasst. Klassifikationsverfahren bilden keine Klassen, sondern ordnen Objekte in vorgegebene Klassen ein. Dabei sind meist zwei Phasen notwendig. In der Lernphase werden aus einer Datenbank zufällige Objekte mit Angabe der dazugehörigen Klassen ausgewählt. Aus diesen Trainingsdaten wird ein Modell, ein Satz von Regeln, definiert. Auf dieser Grundlage können in der darauffolgenden Klassifikationsphase unbekannte Objekte den bestehenden Klassen zugewiesen werden. Da in dieser Arbeit allerdings ein unbekannter Zusammenhang zwischen Features untersucht werden soll, ist weder die Datenverteilung, noch die Art der Klassen bekannt. Aus diesem Grund eignen sich Klassifikationsverfahren, wie die Diskriminanzanalyse, für das Finden von Extraktionsverfahrensgruppen nicht. Anwendungsbeispiele für die Klassifikation sind hingegen die Erkennung von Schriftzeichen, das Diagnostizieren einer Krankheit oder die Überprüfung der Kreditwürdigkeit. Für das hier bestehende Problem werden stattdessen Verfahren zur Klassifizierung, also dem Bilden von Klassen, eingesetzt. Dies umfasst in erster Linie Clusterverfahren, auf welche im Folgenden näher eingegangen wird. [Los02 S. 7]

Distanzbasiertes Clustering

Clustering-Algorithmen stammen aus dem Fachgebiet des Data-Mining. Dieses umfasst einen Bereich der Datenanalyse, in dem verschiedene Verfahren eingesetzt werden, um durch systematische Analyse bisher unbekannte Muster und Zusammenhänge im Datenbereich aufzudecken. Dadurch ist im Gegensatz zu Klassifikationsverfahren, bei denen Klassen vorgegeben werden müssen, beim Clustering nicht zwingend zusätzliches Wissen über die Einteilung oder Verteilung der Daten notwendig. Aus diesem Grund eignet sich dieses Analyseverfahren für das bestehende Klassifizierungsproblem. Durch das Clusterverfahren werden Strukturen entdeckt und die analysierten Daten in die bis dato unbekannte Datenstruktur eingeordnet. Das Clustering gruppiert dabei Daten mit ähnlichen Eigenschaften zu größeren Einheiten, wobei die Datenblöcke als (Daten-) Cluster bezeichnet werden.

Definition 5.6 (Clustering)
Clustering ist das Zusammenfassen von ähnlichen Daten anhand ihrer Merkmale zu Gruppen (Clustern) [Los02 S. 2]

Die Gruppierung soll dabei so erfolgen, dass die Objekte innerhalb eines Clusters sich möglichst ähnlich, die Cluster untereinander aber möglichst unähnlich sind. Der Prozess der Clusterbildung umfasst mehrere Schritte. Zunächst müssen die Anforderungen an die Dateneinteilung bestimmt werden. In der Regel werden disjunkte Partitionierungen gesucht. Das heißt, nach Abschluss des Verfahrens soll jedes Objekt zu genau einem Cluster gehören. Diese Anforderung ist auch an das hier betrachtete Clustering zu stellen. Ziel der Analyse ist es Gruppen von gleichartigen Features zu entdecken, um so nicht notwendige Extraktionsverfahren ausschließen zu können. Falls ein Feature mehreren Gruppen angehört, ist dies allerdings nicht ohne Weiteres möglich. Weiterhin existieren verschiedene Clusterverfahren. Die Wahl eines Verfahrens hängt zum einen von dem Ausgangsdatenbestand und zum anderen von den Anforderungen an die neu zu bildenden Cluster ab. Die wichtigsten Arten von Clusteralgorithmen sind partitionierendes, dichtebasiertes und hierarchisches Clustering. Auf weitere Vorgaben, wie die Anzahl der Cluster, die Anzahl der Objekte in einem Cluster oder die maximale Rechenzeit kann hier verzichtet werden. Diese sind auf Grund der wenigen zu clusternden Elemente als gering einzuschätzen. [Los02 S. 2]

Nachdem die Anforderungen an das Clustering getroffen wurden, sieht der folgende Schritt die Aufbereitung der Daten vor. Dies umfasst beispielsweise eine Normalisierung der Daten, sowie das Entfernen redundanter Attribute. Das Entfernen von einzelnen Werten aus dem Ähnlichkeitsvektor wird hier nicht angestrebt, da die Bewertung aller Bilder in die Berechnung einbezogen werden soll. Weiterhin sind die zur Analyse verwendeten Ähnlichkeitswerte bereits auf das Intervall [0,...,1] normiert. Allerdings unterscheidet sich die Verteilung der Bewertungen der einzelnen Retrieval-Verfahren auf diesem Intervall stark voneinander, was in der folgenden Abbildung für ausgewählte Features beispielhaft dargestellt ist. Die Abbildung zeigt die Anzahl von berechneten Merkmalswerten auf dem definierten Intervall.

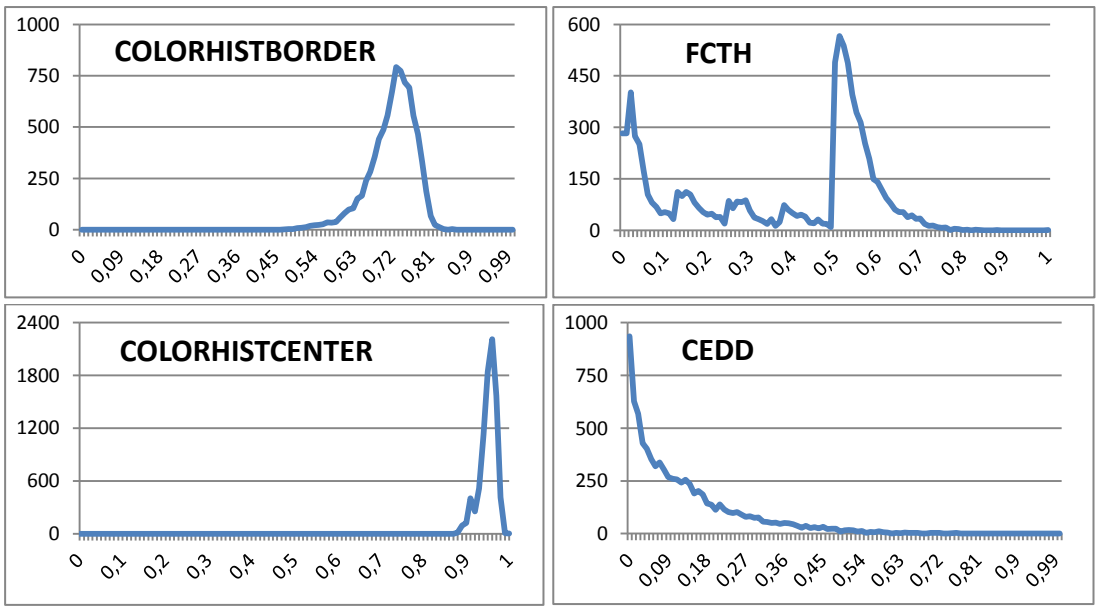

Abbildung 5-2: Datenverteilung der durch Pythia ermittelten Ähnlichkeitswerte für verschiedene Feature

Die in Abbildung 5-2 dargestellten Datenverteilungen entsprechen den häufigsten im Test beobachteten Verteilungen von Ähnlichkeitsbewertungen. Die meisten Features weisen dabei eine Gaußverteilung, ähnlich der von COLORHISTBORDER, auf. Allerdings unterscheiden sich sowohl die Lage, als auch die Streuung der einzelnen Verfahren zum Teil stark voneinander. Dies wird am Beispiel von COLORHISTCENTER deutlich, dessen Verteilungskurve nah eins verschoben und deutlich gestreckt ist. Darüber hinaus sind auch nicht normalverteilte Features unter den Testverfahren. Beispiele hierfür sind FCTH und CEDD, welche eine weitaus größere Wertespanne abdecken. Für einen Vergleich, wie er bei dem entwickelten Clustering-Verfahren angestrebt wird, sollte zuvor eine einheitliche Bewertungsverteilung bei allen Features geschaffen werden. Dies ist nötig, da das Gruppieren der Werte auf dem Abstand der einzelnen Relevanzbewertungen durch die verschiedenen IR-Verfahren basiert. Deckt das Verfahren A beispielsweise hauptsächlich Werte zwischen 0 und 0,2 und das Verfahren B Werte zwischen 0,8 und 1 ab, wäre die berechnete Distanz beim Clustern zwischen beiden Features sehr hoch. Dies spiegelt allerdings nicht immer den tatsächlichen statistischen Zusammenhang beider Testverfahren wieder, sondern basiert allein auf den unterschiedlichen Wertebereichen. Um diesen Störfaktor zu minimieren, müssen die Ähnlichkeitswerte vor der Analyse kalibriet werden. Das bedeutet, der Mittelwert, die Varianz und die prinzipielle Datenverteilung sollten bei allen Features möglichst ähnlich sein. Durch Ermittlung der passenden Funktion wäre es denkbar stets eine Normalverteilung der Werte zu kreieren. Diese Funktion müsste allerding aufwändig bei jedem Test für jedes einzelne Verfahren bestimmt werden. Stattdessen wird im Folgenden eine einheitliche Transformation vorgestellt, welche die Datenverteilung aller Retrieval-Bewertungen anpasst, um diese möglichst genau anzunähern.

Der erste Schritt umfasst die Anpassung der einzelnen Erwartungswerte *E(x)*. Bei der ursprünglichen Verteilung liegen diese zwischen null und eins. Um den Mittelwert anzunähern, wird der Definitionsbereich der Funktion auf das Intervall [0, …, 0.25] eingeschränkt. Diese Transformation ist in folgender Abbildung dargestellt.

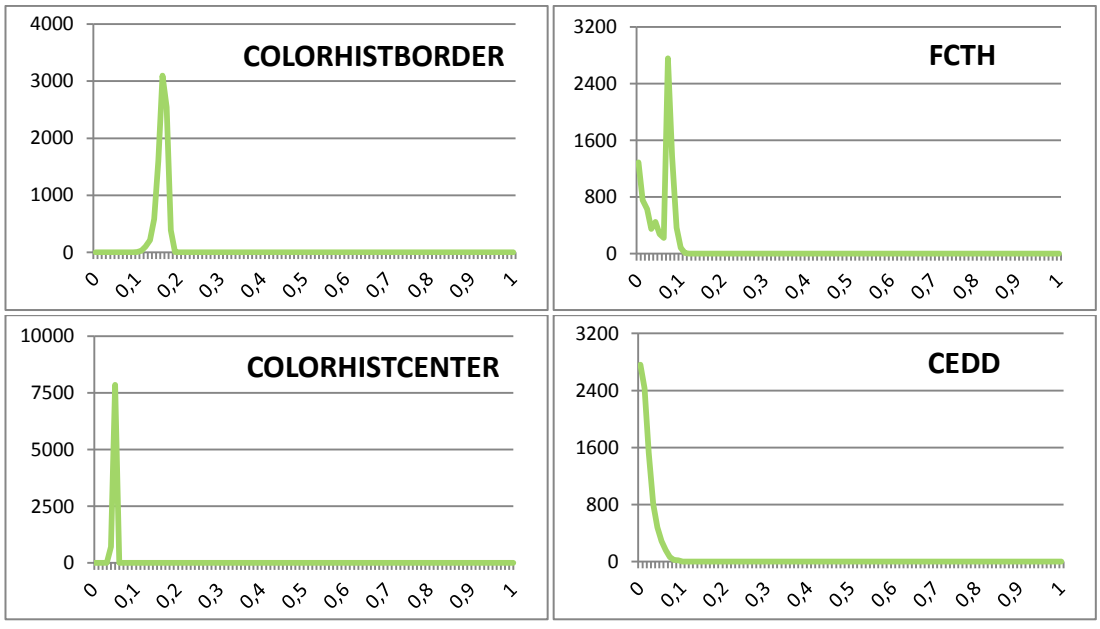

Abbildung 5-3: Datenverteilung der durch Pythia ermittelten Ähnlichkeitswerte nach Beschränkung des Definitionsbereichs

Die Verschiebung der Funktion in den gewünschten Bereich wurde durch Ermittlung des folgenden Korrekturfaktors erreicht, mit welchem jeder Funktionswert multipliziert wird. Dies führt zur in Abbildung 5-3 dargestellten neuen Datenverteilung.

$$k = \begin{cases} E(x) - 0{,}5 & \text{für } E(x) > 0{,}5 \\ 0{,}5 - E(x) & \text{für } 0{,}25 < E(x) \leq 0{,}5 \\ E(x) & \text{für } E(x) \leq 0{,}25 \end{cases} \quad (5\text{-}2)$$

Durch die Einschränkung des Definitionsbereichs werden die einzelnen Funktionen gestreckt. Dies führt dazu, dass die bei einigen Features ohnehin geringe Varianz weiter verkleinert wird. Für die Korrelationsanalyse ist hingegen eine größere Streuung der Werte erwünscht. Unterscheidet sich die Bewertung aller Bilder bei einem Verfahren nur gering von einander, so werden mehr Features als ähnlich erkannt. Um das Korrelationsergebnis nicht zu stark zu beeinflussen und die ursprünglichen Varianz wiederherzustellen, werden die Verteilungsverläufe anschließend um einen weiteren Faktor gestaucht. Die komplette Transformation inklusive der ersten Korrekturvariablen *k* ist in der Gleichung 5-3 dargestellt.

$$Z = \left| kX + \frac{0{,}25(X - E(X))}{\sqrt{S(X)}} \right| \quad (5\text{-}3)$$

In der abgebildeten Gleichung entspricht *Z* dem neu berechneten Funktionswert der Verteilung, *k* dem Verschiebungsfaktor (vgl. Gleichung 5-2), *E(X)* ist der Erwartungswert und *S(X)* die Standardabweichung des alten Funktionswertes *X*. Da der Wertebereich auf ein Viertel des ursprünglichen Bereichs verkleinert wurde, wird dies auch im Stauchungsfaktor berücksichtigt. Die Stauchung wird dabei durch Betrachtung des Abstands zwischen dem alten Wert *X* und seinem Erwartungswert *E(X)* bestimmt. Diese Differenz wird zudem durch die Wurzel von *S(X)* normiert. Das Ergebnis der Stauchung ist in nachfolgender Abbildung gezeigt.

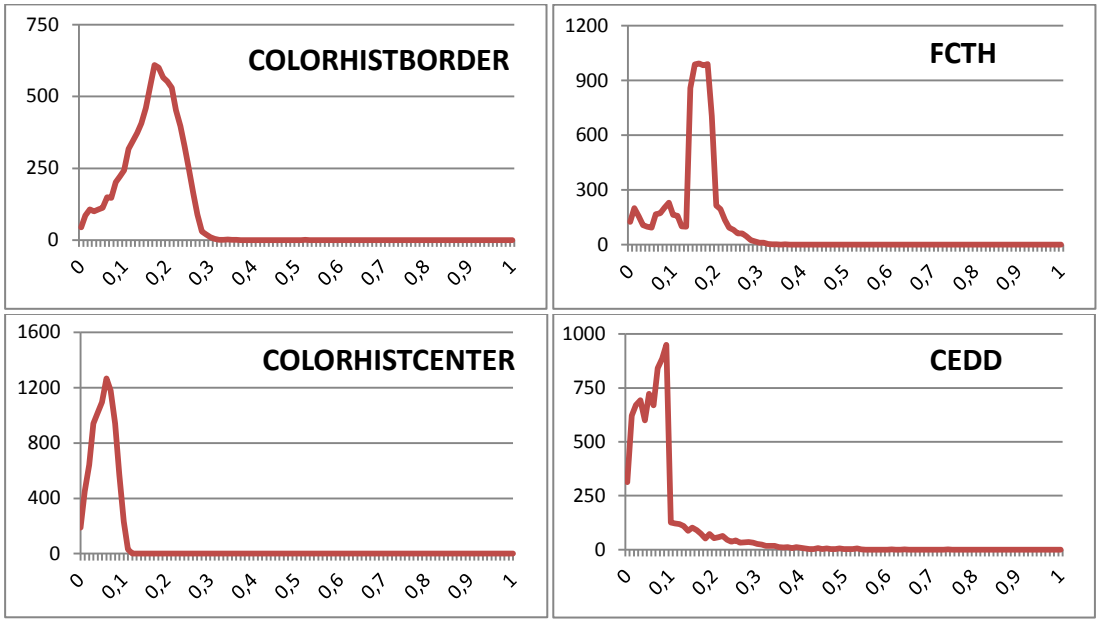

Abbildung 5-4: Datenverteilung der durch Pythia ermittelten Ähnlichkeitswerte nach der Kalibrierung

Nach der Transformation der Werte kann im Anschluss der Zusammenhang zwischen den Features bestimmt werden. Beim verwendeten Clustering werden die Daten aus dem Retrieval-Prozess herangezogen. Der Zusammenhang zweier Retrieval-

Verfahren definiert sich über die unterschiedliche Bewertung der Ähnlichkeit der einzelnen Kollektionsbilder zum Anfragebild. Im Allgemeinen wird hierfür eine Ähnlichkeits- oder Distanzfunktion genutzt, welche die Ähnlichkeit bzw. Unähnlichkeit zwischen den Datenvektoren beider Extraktionsmethoden berechnet. Die entscheidende Rolle bei der Auswahl dieser Funktion spielt das Skalenniveau der Merkmale. Euklidische Distanzen oder allgemeiner L_p-Metriken sind nur bei metrischen, das heißt intervall- oder verhältnisskalierten, Daten sinnvoll. Ordinale Daten können dabei in Form von Rangreihen metrischen Distanzmaßen unterworfen werden. Bei nominalen Daten wird als Ähnlichkeitsmaß grundsätzlich die Anzahl der Übereinstimmungen verwendet, welche auf unterschiedliche Weisen normiert werden kann. Bei Pythia besitzen die Ähnlichkeitswerte aus dem Ranking der Bilder ein metrisches Skalenniveau, weshalb eine L_p-Distanz genutzt werden kann. Genauer wird bei dem verwendeten Clustering die L_1 (Manhattan-Distanz) als Maß für die Unähnlichkeit verwendet, da die Scores bereits auf [0,…,1] normiert sind. Dabei wird für jedes Merkmal des Vektors einer Extraktionsmethode die Differenz zum gleichen Merkmal eines zweiten Verfahrens bestimmt. Dies entspricht dem Unterschied zweier Methoden in der Bewertung der Ähnlichkeit eines Kollektionsbildes zu dem Anfragebild. Zudem besteht die Möglichkeit den Einfluss der einzelnen Summanden durch eine Gewichtung zu ändern. Dadurch können beispielsweise die Differenzen der Bilder auf den vorderen Rängen den Abstand zweier Verfahren stärker prägen. Die gewichtete Manhattan-Distanzfunktion ist wie folgt definiert. [Los02 S. 2f]

$$\delta = \sum_{i=1}^{N} w_i \cdot |x_i - y_i| \tag{5-4}$$

Auf den Einsatz von Gewichten wurde bei dieser Arbeit verzichtet. Somit fließt die Unähnlichkeit der Bewertung jedes Kollektionsbildes zu gleichen Teilen in die Distanz zweier Features ein. Da die mittlere Bewertungsabweichung für den Vergleich der Extraktionsalgorithmen bedeutender als die absolute Differenz ist, wird nach der Aufsummierung der Beträge der Teildifferenzen die Summe durch die Anzahl N der Bilder dividiert. Die resultierende Abstandsfunktion kann folgendermaßen berechnet werden, wobei $s_{x.}$ und $s_{y.}$ den Vektoren zweier Retrieval-Verfahren entsprechen (vgl. Gleichung 4-1).

$$\delta_{XY} = \frac{1}{N} \cdot \sum_{i=1}^{N} |s_{xi} - s_{yi}| \tag{5-5}$$

Die definierte Distanzfunktion liefert die Unähnlichkeit zweier Feature X und Y als Distanzwert im Intervall [0,…,1]. Im Gegensatz zu den Scores des Retrieval-Ergebnisses (vgl. Gleichung 4-1) sind die berechneten Distanzen δ_{xy} dreidimensional. Die beiden Dimensionen x und y geben die verglichenen Extraktionsmethoden an und die dritte Dimension misst die Ähnlichkeit der Verfahren. Diese Distanz wird für alle Methoden über alle Bilder berechnet und in Form einer Distanzmatrix wiedergegeben. Abbildung 5-5 verdeutlicht das Transformationsschema der extrahierten Features zu einer Distanzmatrix zwischen den Verfahren. Die berechnete Matrix bildet die Grundlage für das anschließende Clustering.

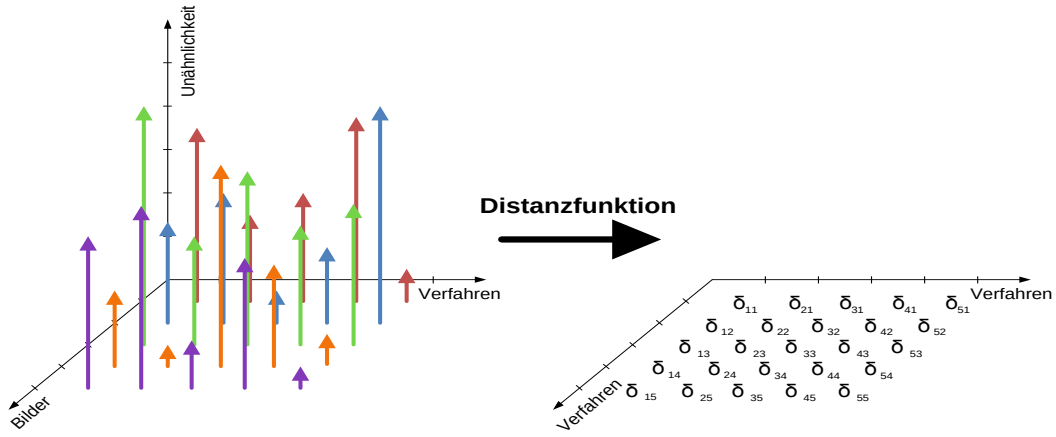

Abbildung 5-5: Transformationsschema der Retrieval-Daten in die Distanzmatrix

Bereits in der Einführung zu den Clusterverfahren wird erwähnt, dass verschiedene Arten des Clustering existieren. An dieser Stelle soll kurz die Eignung der drei wichtigsten Klassen für die angestrebte Analyse diskutiert werden. Partitionierende Verfahren, wie k-medoid oder k-means, sind für die durchzuführende Analyse nicht optimal geeignet, da initial eine Anzahl *k* von zu bildenden Clustern notwendig ist. Ziel dagegen ist es unbekannte Zusammenhänge zwischen den Features zu analysieren, was ebenfalls eine unbekannte Anzahl von Clustern mit einschließt. Auch Verfahren, die nur auf der Verteilungsdichte der Clusterelemente basieren, sind nicht anwendbar. Eine Abbildung auf einen geometrischen Raum, in welchem eine Dichtefunktion definiert werden kann, ist aufgrund eines fehlenden lokalen Bezugssystems für die vorhandenen Daten nicht möglich. Stattdessen wird bei dieser Arbeit ein hierarchisches Clustering verwendet, um die Verfahren einzuordnen. [Los02 S. 2]

Hierbei werden zwei Arten unterschieden. Agglomerative hierarchische Verfahren gehen von einem initialen Zustand aus, bei dem jedes Element ein eigenes Cluster bildet. Alle Cluster werden dann paarweise verglichen und in jedem Schritt die beiden Cluster mit der höchsten Ähnlichkeit verschmolzen. Der Prozess endet, wenn alle Elemente sich in einem großen Cluster befinden. Im Gegensatz zu dieser Bottom-up-Methode existiert auch ein Top-down-Ansatz. Divisive hierarchische Verfahren gehen von einem Cluster mit allen Elementen aus und teilen diesen sukzessive in kleinere maximal unähnliche Cluster. Diese Methode terminiert, wenn alle Elemente ein eigenes Cluster bilden, ist dabei allerding rechenintensiver als agglomerative Verfahren. Die Darstellung dieser Hierarchie erfolgt meist in Form eines Dendogramms. Ein Dendogramm ist ein Baum, dessen Knoten jeweils Cluster darstellen. Jeder Knoten besitzt einen linken und einen rechten Teilbaum, was dem Verschmelzen zweier Cluster entspricht. Die Distanz zwischen den Clustern nimmt von den Blättern zur Wurzel hin zu. Das bedeutet, nah der Blattebene werden zuerst Cluster mit der kleinstmöglichen Distanz verschmolzen. Dies bietet den Vorteil, dass der Anwender, nachdem das Dendogramm erstellt wurde, die Anzahl der Cluster über einen Schwellwert für die Clusterdistanz beeinflussen kann. Dies ist für die betrachtete Analyse wichtig, da nicht der Wurzelcluster, welcher alle Verfahren enthält, sondern die Cluster mit den Verfahren einer gewissen Ähnlichkeit notwendig sind. [Los02 S. 3f]

Das agglomerative hierarchische Clustering beinhaltet wiederum verschiedene Clusterverfahren. Drei gängige hierarchische Verfahren sind Single-Link, Average-Link und Complete-Link. Bei Single-Link ist die Distanz zweier Cluster als die minimale Entfernung zwischen allen Kombinationen der Elemente dieser beiden Cluster definiert.

Im Gegensatz dazu wird beim Complete-Link Clustering die Distanz durch den maximalen Abstand der Clusterelemente betrachtet. Die Distanz beim Average-Link Verfahren ergibt sich aus der Differenz der Mittelwerte aus allen Clusterelementen. Die folgende Abbildung verdeutlicht die Distanzbildung bei diesen drei Methoden. [Zho04 S. 5f]

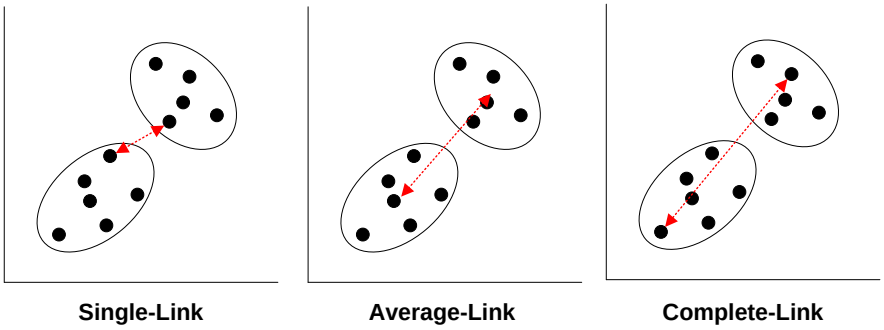

Abbildung 5-6: Distanzberechnungsschema bei Single-Link, Average-Link und Complete-Link

Das Ergebnis jedes dieser Clusterverfahren ist ein Dendogramm, welches die schrittweise Bildung der Cluster zeigt. Da die verschiedenen Verfahren die Distanzen zweier Cluster unterschiedlich berechnen, können sich die jeweiligen Dendogramme in der Reihenfolge der Clusterbildung und in den Distanzwerten bei der Clusterbildung unterscheiden.

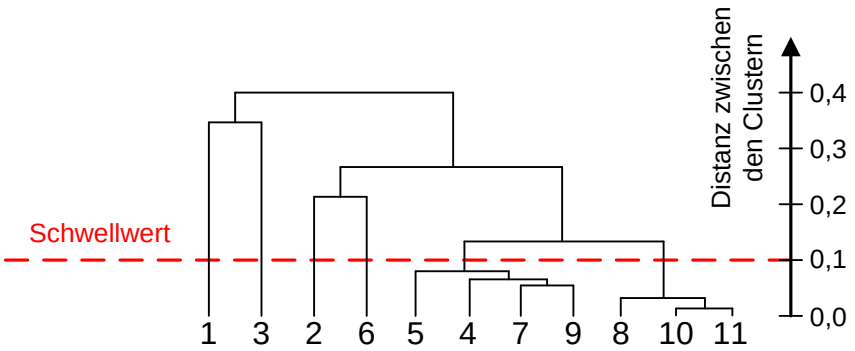

Abbildung 5-7: Dendogramm mit Schwellwert für Abbruch des Clusterverfahrens

Da die Distanzen der zu vereinigenden Cluster von den Blättern zur Wurzel hin zunehmen, kann die Anzahl der Cluster über einen gesetzten Schwellwert bestimmt werden (vgl. Abbildung 5-7). Hierzu wird der Baum von den Blättern an durchlaufen und die Clusterbildung bei Erreichen des Grenzwertes abgebrochen. Auf dieser Grundlage sollte das für die Anwendung ideale Clusterverfahren gewählt werden. Beim Complete-Link Clustering werden durch die Verwendung der maximalen Distanzen bei gleichem Schwellwert im Vergleich zu den beiden anderen Verfahren weniger Cluster gebildet. Die Ursache hierfür ist, dass die Distanz zweier Cluster bei der Vereinigung wesentlich höher ist. Aus diesem Grund ist das Verfahren für die Analyse zwar geeignet, tendiert allerdings zu vielen kleinen Clustern. Dadurch können nach der Analyse nur wenige Extraktionsverfahren ausgeschlossen werden. Single-Link beinhaltet ein anderes Problem. Es betrachtet stets den minimalen Fall bei der Clusterbildung. Zwei Cluster werden auf der Grundlage je eines Vertreters vereinigt, wobei die übrigen Elemente des Clusters nicht betrachtet werden. Bei einer ungünstigen Datenverteilung besteht die Möglichkeit, dass die Randelemente des Clusters eine um ein Vielfaches höhere Distanz zueinander besitzen als der Schwellwert für die paarweise Vereinigung erlaubt. Ein Beispiel soll dies verdeutlichen. Im ersten Schritt werden zwei Cluster vereinigt, da Element x und Element y eine Distanz von rund 0.1 besitzen. Im drauf folgen-

den Schritt wird dieser Cluster mit dem von Element z verschmolzen, da z eine Distanz von 0,1 zu y hat. Bei einer strahlenförmigen Datenverteilung und *n* Verschmelzungsschritten ist die theoretisch mögliche Distanz zwischen Element x und dem zuletzt aufgenommenen Element 0,1n. Diese beiden Elemente sind nur noch wenig korreliert und sollten bei einem niedrigeren Schwellwert nicht in einem gemeinsamen Cluster liegen. Aus diesem Grund ist Single-Link nicht für dieses Analyseverfahren geeignet. Bei Complete-Link besteht dieses Problem aufgrund der Verwendung der maximalen Distanzen nicht und bei Average-Link wird es durch die Mittelwertbildung abgeschwächt. Zudem besitzt Average-Link Clustering den größten Rechenaufwand der drei Verfahren. Dieser ist bei den wenigen Verfahren, die geclustert werden, aber vernachlässigbar. Sowohl Complete-Link, als auch Average-Link sind für die angestrebte Analyse zu empfehlen, da bei beiden Verfahren alle Elemente zur Clusterbildung betrachtet werden und auch sonst keines der Verfahren entscheidende Vorteile bietet.

Kosinus-Maß

Das Kosinus-Maß ist ein Ähnlichkeitsmaß aus dem Bereich des Information Retrieval. Im Text-Retrieval wird es dazu eingesetzt, die Ähnlichkeit zwischen einem Anfrage- und einem Dokumentenvektor zu berechnen. Mathematisch betrachtet wird hierbei das Skalarprodukt dazu genutzt, den Kosinus des Winkels zwischen den beiden Vektoren zu berechnen. Dabei entspricht die maximale Unähnlichkeit null einem Winkel von 90°. Die Vektoren sind in diesem Fall orthogonal zu einander. Identische Vektoren mit einem Winkel von 0° besitzen dagegen ein Kosinus-Maß von eins. Das Kosinus-Maß zwischen dem Vektor X und dem Vektor Y wird wie folgt berechnet.

$$\cos\alpha = \frac{\sum_{i=1}^{n} X_i \cdot Y_i}{\sqrt{\sum_{i=1}^{n} {X_i}^2} \cdot \sqrt{\sum_{i=1}^{n} {Y_i}^2}} \tag{5-6}$$

Dieses Ähnlichkeitsmaß soll nun dazu genutzt werden, den Winkel zwischen den Vektoren zweier Retrieval-Verfahren zu berechnen (vgl. Gleichung 5-6) und dadurch auf die Ähnlichkeit der Extraktionsmethoden zu schließen. Die zu erwartenden Ergebnisse sind allerdings eher schlecht. Eine Ursache dafür ist, dass die Berechnungen im hochdimensionalen Raum stattfinden. Dazu kommt die Tatsache, dass die Relevanzbewertungen der verschiedenen Verfahren zu ähnlich sind. Jeder n-dimensionale Vektor besitzt Werte zwischen null und eins, wobei nur sehr wenige Dimensionen eine null beinhalten. Da jede Dimension eine Bewertung zwischen Anfrage- und Kollektionsbild zeigt, müsste für eine Nullbewertung eine große Unähnlichkeit zwischen beiden Bildern bestehen. In Folge dessen ist der Bereich, in welchem alle Dokumentvektoren liegen sehr klein. Da alle Vektoren nahezu in die gleiche Richtung zeigen, ist der Differenzwinkel, welcher mittels Kosinus-Maß berechnet wird, auch sehr klein. Dementsprechend ergibt sich eine hohe Ähnlichkeit zwischen allen Features, weshalb der Ansatz in dieser Form für die Analyse nicht verwendbar ist.

Im Information Retrieval liefert das Kosinus-Maß hingegen gute Ergebnisse, da jede Dimension einem Wort des Indexvokabulars entspricht. Dadurch enthalten sowohl Anfrage- als auch Dokumentvektor viele Nullwerte. Dies ist bei den hier verglichenen Verfahrensvektoren nicht der Fall. Trotzdem kann die Idee den Winkel zwischen zwei Vektoren als Maß für deren Ähnlichkeit zu betrachten in abgeänderter Form auch hier verwendet werden. Eine Möglichkeit brauchbare Ergebnisse zu erhalten, ist den Bezugspunkt zu verändern (vgl. Gleichung 5-4 und 5-5). Anstatt zum Nullpunkt vergleicht

die Korrelation, welche im Anschluss betrachtet wird, die Differenz jeder Dimension zum Mittelwert aller Dimensionen. Dadurch wird der berechnete Winkel gestreckt und erhält mehr Aussagekraft über die Ähnlichkeit der Extraktionsverfahren.

Korrelationsanalyse

Bei einer Untersuchung der Ähnlichkeit bestimmter Objekte, bietet sich ebenfalls eine Betrachtung der Korrelation der jeweiligen Resultate an. Ähnlich wie bei dem vorgestellten Clusterverfahren werden bei der Korrelationsanalyse wieder die Bewertungsergebnisse der einzelnen Verfahren aus dem Extraktionsprozess herangezogen. Im Gegensatz zum Clusteransatz erfolgt die Klassifizierung nicht auf dem Unterschied, sondern auf dem Zusammenhang der entsprechenden Daten. Die Korrelationsanalyse untersucht, ob eine Beziehung zwischen zwei metrischen Variablen besteht, wie stark diese ist und welche Richtung sie hat.

Definition 5.7 (Korrelationsanalyse)
Eine Korrelationsanalyse ist ein statistisches Verfahren, welches den Zusammenhang zweier zufälliger Variablen anhand mehrerer Stichproben untersucht. [Mes10]

Die zu vergleichenden zufälligen Variablen sind hierbei die paarweise verglichenen Features und die Stichproben bilden die ermittelten Ähnlichkeitswerte für die Anfrage auf einer Bildkollektion. Zur Analyse der Beziehung zweier Verfahren wird der Korrelationskoeffizient der beiden Variablen bestimmt, welcher den Zusammenhang zwischen ihnen charakterisiert. Dabei gilt, je größer die Stichprobe, also die Bildkollektion ist, desto genauer beschreibt der Korrelationskoeffizient den statistischen Zusammenhang.

Der bekannteste Koeffizient zum Berechnen der Korrelation ist der Pearson'sche Produktmoment-Korrelationskoeffizient, welcher auch in dieser Arbeit Anwendung findet. Er kann Werte zwischen -1 und 1 annehmen, wobei Werte um 1 einen proportionalen Zusammenhang aufzeigen, -1 kennzeichnet eine Antiproportionalität und ein Korrelationskoeffizient nahe 0 deutet auf eine geringe oder keine Beziehung zwischen den Variablen. Der Korrelationskoeffizient r wird nach Pearson folgendermaßen berechnet: [Mes10]

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\left(\sum_{i=1}^{n}(X_i - \bar{X})^2\right)\left(\sum_{i=1}^{n}(Y_i - \bar{Y})^2\right)}} \tag{5-7}$$

Dabei werden für die Verfahren X und Y die Abweichungen der einzelnen Merkmale X_i und Y_i zum jeweiligen Mittelwert $\bar{X}$ bzw. $\bar{Y}$ multipliziert und anschließend aufsummiert. Zudem erfolgt eine Normierung des Koeffizienten durch die Einzelstandardabweichung auf dem Intervall [-1,…,0,…,1]. [Mes10]

Da bei dieser Untersuchung lediglich die Höhe der Abhängigkeit und nicht deren Richtung wichtig ist, kann der Betrag von r gebildet werden. Auf die Retrievalverfahren bezogen bedeutet dies, dass Methoden mit geringem Betrag des Korrelationskoeffizienten zusammen eingesetzt werden sollten, da sich ihre Bewertungsergebnisse stark unterscheiden. Besitzen sie hingegen einen hohen Betrag, besteht ein signifikanter Zusammenhang zwischen den Extraktionsergebnissen und es genügt der Einsatz lediglich eines Vertreters dieser Gruppe. Um eine Aussage dazu treffen zu können, muss der berechnete Korrelationskoeffizient klassifiziert werden. Hierfür existieren in

der Literatur verschiedene Auffassungen. In dieser Arbeit wird ein Koeffizient von (0<r≤0,4) als schwache, (0,4<r≤0,7) als mittlere und (0,7<r≤1) als eine starke Abhängigkeit betrachtet. Die folgende Abbildung zeigt die mögliche Verteilung der Daten bei unterschiedlichen Korrelationskoeffizienten. [Mes10]

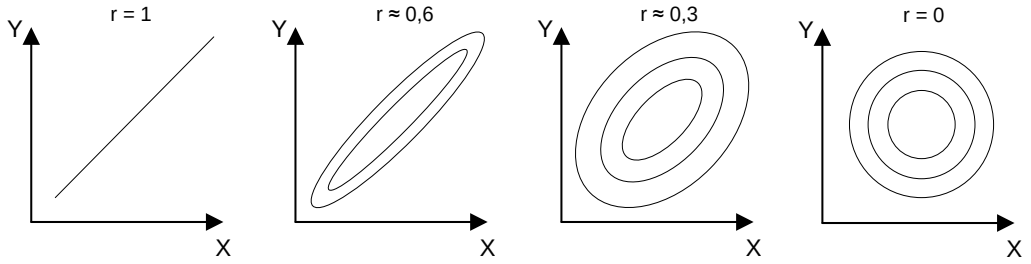

Abbildung 5-8: Korrelation zweier Zufallsvariablen X und Y

Zuletzt muss überprüft werden, ob der ermittelte Zusammenhang bedeutsam oder zufällig ist. Hierzu wird eine Signifikanzanalyse durchgeführt. Die Signifikanz ist eine Kennzahl, welche die Wahrscheinlichkeit eines systematischen Zusammenhangs zwischen den Variablen bezeichnet. Sie drückt aus, ob ein scheinbarer Zusammenhang rein zufälliger Natur sein könnte oder mit hoher Wahrscheinlichkeit tatsächlich vorliegt. Bei einer Signifikanzanalyse wird zunächst das Signifikanzniveau festgelegt. Dabei handelt es sich um die maximale Irrtumswahrscheinlichkeit bei Zurückweisung der Null-Hypothese, welche besagt, dass keine Korrelation vorliegt. Üblich sind dabei 5%, 1%, 0,1% oder 0,01% als Irrtumswahrscheinlichkeit zu wählen. Bei der durchgeführten Korrelationsanalyse mit relativ kleinen Stichproben wird ein Signifikanzniveau von $\alpha=0,01$ gewählt. Um die Signifikanz eines Korrelationskoeffizienten zu testen, wird eine t-Statistik verwendet. [Mes10]

$$T = r\frac{\sqrt{n-2}}{\sqrt{1-r^2}} \tag{5-8}$$

Diese Teststatistik folgt einer t-Verteilung (Student-Verteilung). Von dem Korrelationskoeffizienten wird angenommen, dass er statistisch signifikant ist, wenn der berechnete Wert größer als der kritische Wert einer t-Verteilung mit einem Signifikanzniveau von $\alpha/2$ und n-2 Freiheitsgraden ist. Dieser kritische Wert kann in entsprechenden Tabellen[5] nachgeschlagen werden.

Die Berechnung der Korrelationskoeffizienten aller Kombinationen von zu vergleichenden Features ergibt eine Korrelationsmatrix. Anhand dieser Matrix kann für jede Kombination entschieden werden, ob ein signifikanter Zusammenhang besteht (vgl. Abbildung 5-8). Auf dieser Basis können korrelierende Verfahren zusammengefasst werden, um so die notwendige Anzahl der zu berechnenden Methoden zu reduzieren. Hierfür bieten sich wieder Clusterverfahren an. Der Hauptunterschied zum vorgestellten Clusteransatz ist, dass die Korrelation nicht den Unterschied zweier konkreter Ähnlichkeitsbewertungen betrachtet, sondern untersucht, ob der prinzipielle Verlauf aller Ähnlichkeitsbewertungen der verglichenen Verfahren gleich ist. Ähnlich zum distanzbasierten Clustering kann eine Korrelationsanalyse die Ergebnisqualität eines Verfahrens selbst nicht beurteilen und somit das Suchergebnis nicht verbessern. Durch die Reduktion der notwendigen Berechnungen wird dagegen die Gesamtrechenzeit verkürzt und somit die Effizienz der Suche gesteigert.

[5] http://psydok.sulb.uni-saarland.de/volltexte/2004/268/html/tvert.htm

5.2 Ergebnisqualität nach Retrieval-Maßen

Eines der wichtigsten Kriterien für ein gutes Retrieval-System ist die Qualität des Suchergebnisses. Dieses soll in diesem Abschnitt mittels der in Kapitel 5.1.1 eingeführten Retrieval-Kennzahlen evaluiert werden. Zur Ermittlung wird bei dieser Arbeit das Analysetool TRECEVAL[6] verwendet. Ähnlich den Retrieval-Maßen wurde TRECEVAL ursprünglich für das Text-Retrieval entwickelt und gilt hier als Standardwerkzeug der TREC-Community. Um dieses Tool verwenden zu können, müssen zwei Bedingungen erfüllt werden. Zum einen müssen ground truth-Daten für die verwendete Bildkollektion vorliegen. Dies bezeichnet Daten, die klassifizieren, welche Bilder zu einer bestimmten Anfrage als relevant zu werten sind. Erst durch diesen Regelsatz ist eine automatische Relevanzbewertung möglich. Für die in dieser Arbeit verwendete Bildsammlung 101Categories liegen diese ground truth-Daten durch die Einteilung der Bilder in Bildkategorien vor und können zur Analyse herangezogen werden [Cal03]. Zum anderen muss die analysierte Ergebnisdatei dem TREC-Format entsprechen. Dies ist bei den Ergebnisdokumenten, welche von Pythia zurückgeliefert werden, der Fall. Die nachfolgende Tabelle gibt den Aufbau einer Zeile dieses Formats wieder.

Spalte	Bedeutung
query-number	Nummer der Anfrage zur Identifikation der relevanten Dokumente
Q0	Konstante, welche von manchen Evaluierungsprogrammen genutzt wird
document-id	Einzigartige ID zur Identifikation der Dokumente
rank	Rang, welchen das Dokument durch die Bewertung erhalten hat
score	Relevanzbewertung eines Dokumentes durch das IR-System
Exp	Konstante, welche von manchen Evaluierungsprogrammen genutzt wird

Tabelle 5-4: Aufbau einer Ergebnisdatei im TREC-Format

Neben diesem gezeigten Schema werden die Relevanzbewertungen im Pythia-Ergebnisdokument zudem nach jedem verwendeten Feature aufgeschlüsselt. Dies ermöglicht es, das System nach allen Features einzeln zu bewerten. Es spielt hier keine Rolle, ob sich die Retrieval-Ergebnisse auf eine Textsuche oder andere Medientypen beziehen. Da die beiden genannten Kriterien im Pythia-System erfüllt sind, kann bei dieser Arbeit TRECEVAL zur Analyse der Ergebnisqualität eingesetzt werden.

Mit dem Evaluierungstool können verschiedene Kennzahlen berechnen werden. Darunter die für diese Arbeit verwendete Precision an bestimmten Stellen *P@X* und die average precision AP, welche zur Berechnung von MAP erforderlich ist. Bei einem Suchergebnis sind meist nur die ersten Treffer wichtig. Außerdem umfasst die Bildkategorie der 101Categories mit den wenigsten Elementen rund dreißig Bilder. Eine Betrachtung höherer Retrieval-Werte würde das Evaluationsergebnis verfälschen. Aus diesem Grund werden für alle weiteren Berechnungen lediglich die ersten dreißig gerankten Ergebnisses in die Rechnung einbezogen. Um die Evaluationsergebnisse zu validieren, könnten diese zusätzlich mit dem in Abschnitt 5.1.2 beschriebenen t-Test überprüft werden. Da die Kennziffern allerdings, wie in Abschnitt 5.1.2 gezeigt wird, nicht bei allen Features normalverteilt sind, entfällt diese Möglichkeit.

Vor der Analyse werden an das Retrieval-System verschiedene Anfragen gestellt, um das IR-Ergebnis im Anschluss auswerten zu können. Zu diesem Zweck werden bei

[6] http://ir.iit.edu/~dagr/cs529/files/project_files/trec_eval_desc.htm

dieser Arbeit zu jeder der 101 Bildkategorien der verwendeten Datenbank die ersten zehn Bilder als Anfragebild ausgewählt. Mit jedem dieser gewählten Dokumente wird gegen alle 9197 Bilder der Bildkollektion angefragt. Die nachfolgende Abbildung zeigt die durchschnittlichen Precision an markanten Stellen für die verschiedenen in Pythia implementierten Features. Die Entsprechenden Messwerte sind außerdem in der Tabelle 7-1 aufgeschlüsselt. Als Distanzfunktion wurde dabei für jedes Verfahren seine in Tabelle 7-5 und Tabelle 7-6 definierte Standarddistanz verwendet.

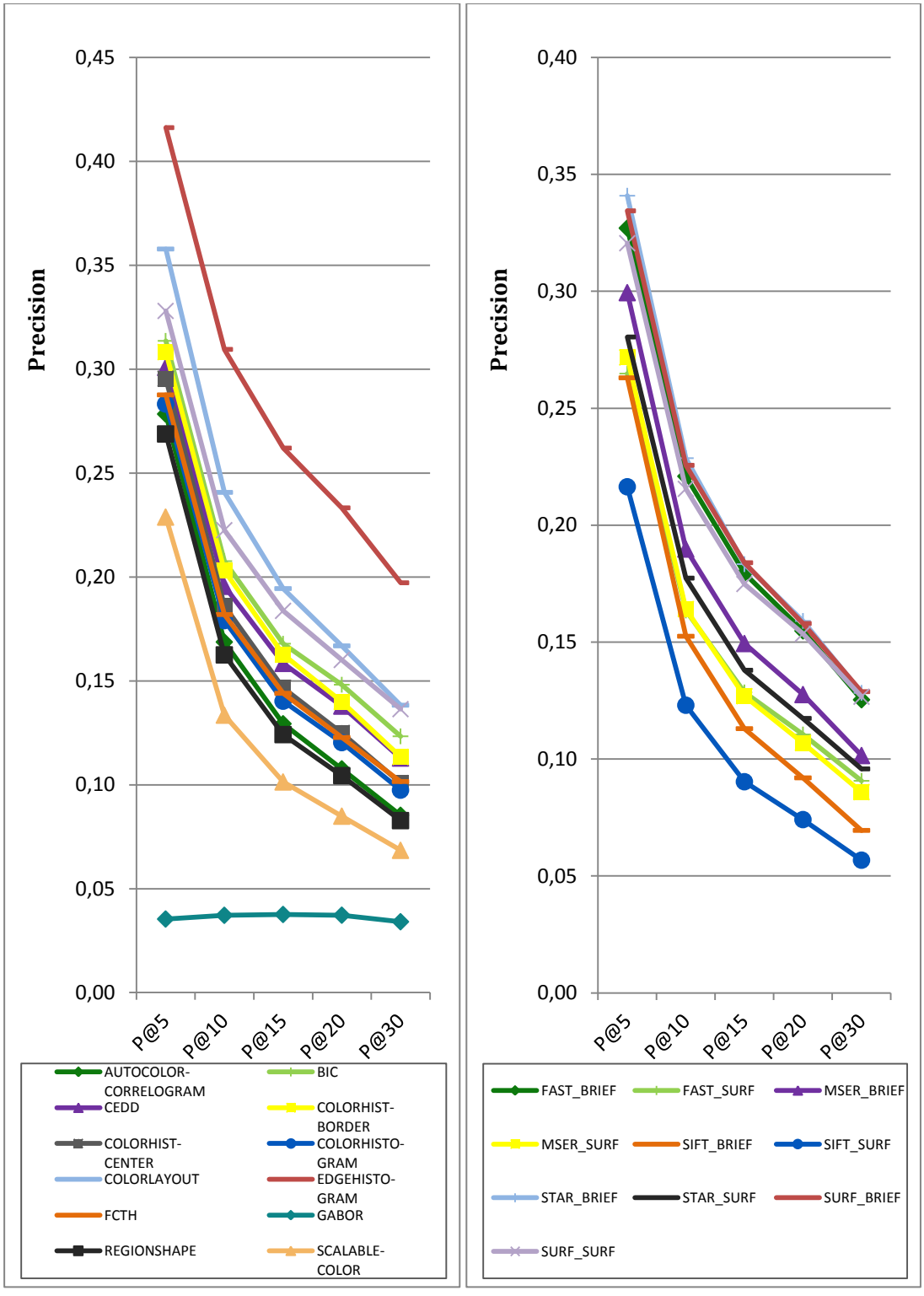

Abbildung 5-9: Gegenüberstellung der Precision@X verschiedener lokaler und globaler Features in Pythia

In Abbildung 5-9 ist die durchschnittliche Precision an bestimmten Stelle der Ergebnisliste abgebildet. Diese wurde für jedes Feature aus allen getätigten Anfragen ermittelt. Es ist zu erkennen, dass der Verlauf der Precision@X-Werte bei allen getesteten Features recht ähnlich ist. Die einzige Ausnahme ist GABOR, welches insgesamt sehr schlecht bei diesem Test abschneidet. Eine Ursache hierfür könnte sein, dass es sich bei GABOR um ein texturbasiertes Feature handelt, die verwendete Bilddatenbank allerdings nicht auf Texturinformationen ausgerichtet ist. Auch die übrigen Verfahren zeigen keine guten Ergebnisse. Dabei existieren zwischen lokalen und globalen Features nur geringe Unterschiede. Nach dem fünften Bild liegt die Precision bei fast allen Features zwischen 25% und 35%. Das bedeutet, nur bei gut jedem vierten Bild auf den vorderen fünf Rängen handelt es sich um einen relevanten Treffer. Am besten ist dabei das EDGEHISTOGRAM mit P@5 von über 40%. Danach nimmt der Precision-Wert bei allen Methoden weiter ab, was zeigt, dass sich auch unter den anschließenden Treffern wenig relevante Suchergebnisse befinden. Eine mögliche Ursache für dieses Ergebnis ist eine schlechte Wahl der Suchparameter oder der verwendeten Distanzfunktion. Um den Einfluss der Distanzfunktion auf die Ergebnisqualität zu evaluieren, wird diese im Kapitel 5.5 variiert. Auch die Wahl der Bildkollektion kann das Ergebnis manipulieren. Beispielsweise könnte die Verwendung von ausschließlich farbigen Bildern für die überwiegend auf Farbinformationen definierten Features oder einer auf Objektkonturen basierenden Bilddatenbank die ermittelten Kennzahlen bei dem einen oder anderen Feature stark ändern. Des Weiteren wurde im Versuch stets nur ein Feature zur Ermittlung des Ranking eingesetzt. In Pythia wird stets eine Kombination verschiedener Verfahren zur Verbesserung der Suchergebnisses verwendet.

Neben der Erkenntnis, dass der Werteverlauf der Features recht ähnlich ist, lassen sich weitere Aussagen über die Retreval-Qualität treffen. Im Test haben das EDGEHISTOGRAM, CALORLAYOUT und TAMURA bei den globalen und STAR_BRIEF, SURF_BRIEF und FAST_BRIEF bei den lokalen Features am besten abgeschnitten. Die schlechtesten globalen Features im Test sind REGIONSHAPE, SCALABLECOLOR und GABOR. Bei den lokalen Verfahren handelt es sich um MSER_SURF, SIFT_BRIEF und SIFT_SURF. Insgesamt sind bei der Untersuchung der lokalen Methoden alle BRIEF-Deskriptoren den SURF-beschriebenen Features überlegen. Dies ist insofern interessant, da die SURF-Beschreibung mit doppelt so vielen Werten genauer als die BRIEF-Deskription sein sollte. Diese Beobachtung lässt sich durch eine schlechtere Verteilung der Werte bei SURF erklären. Dadurch, dass bei diesem Test die Distanzen von BRIEF und SURF zusammen ermittelt werden und somit für beide der gleiche Schwellwert angesetzt wird, gruppieren sich viele Ähnlichkeitsbewertungen bei SURF um eins. Bei einem an SURF angepassten Threshold liefern auch viele SURF-beschriebene Features mitunter bessere Resultate (vgl. Kapitel 5.6).

Als nächstes soll die mean average precision aller betrachteten Features untersucht werden. Da diese sich aus dem Mittelwert der Precision@X-Werte bildet, ist ein ähnliches Ergebnis zu erwarten. Die folgende Abbildung gibt den MAP-Wert aller Verfahren wieder. Die Abbildung 5-10 bestätigt die Ergebnisse des vorherigen Diagramms. Das EDGEHISTOGRAM hat auf den ersten 30 gerankten Bildern eine MAP von 0,27 und ist damit besser als alle übrigen Verfahren im Test. Auf den weiteren Positionen sind die oben auf den vorderen Plätzen genannten Features. Am schlechtesten schneiden wieder SIFT_SURF, COLORHISTOGRAM und GABOR ab.

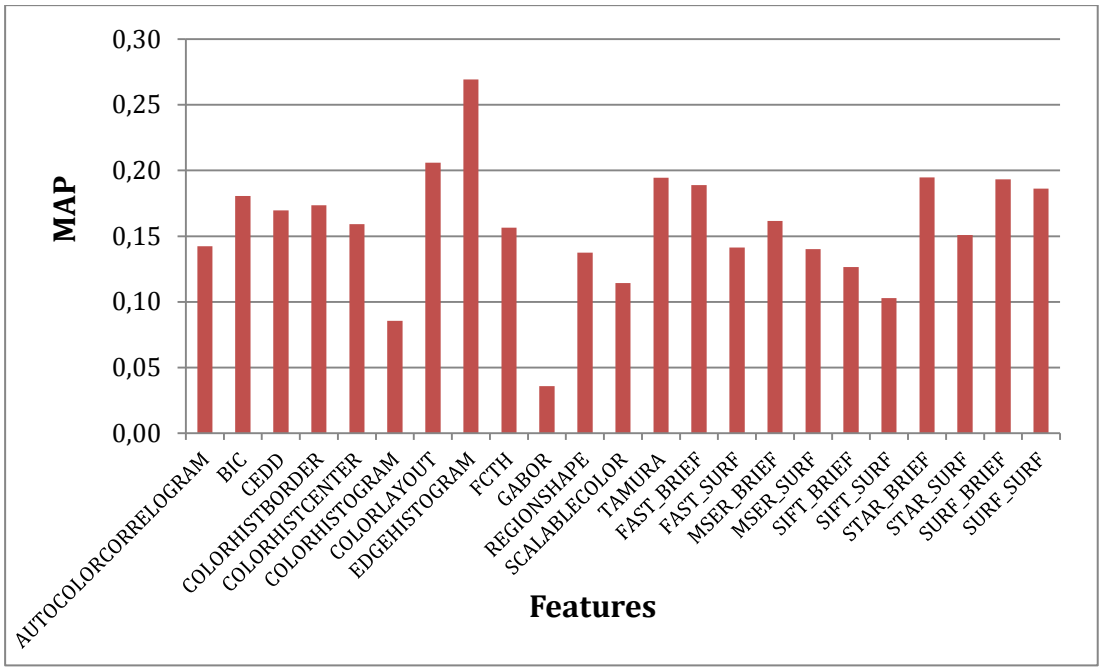

Abbildung 5-10: Auflistung der mean average precision verschiedener Features in Pythia bei der Relevanzuntersuchung

Alles in allem sind die Einzel-Retrieval-Ergebnisse aller im Test untersuchten Features relativ ähnlich. Allein auf der Grundlage der Relevanz des Suchergebnisses lässt sich somit keine Entscheidung über die Wahl der zu nutzenden Features treffen. Hierfür wichtige Faktoren sind auch die Rechenzeit, welche vor allem zwischen lokalen und globalen Features stark variiert, und der Zusammenhang zwischen den einzelnen Features zur Auswahl einer geeigneten Feature-Menge. Wie erwähnt lässt sich das Ergebnis durch eine Kombination möglichst heterogener Verfahren noch verbessern. Sowohl die Rechenzeit, als auch die Korrelation der Retrieval-Verfahren wird im Anschluss untersucht.

5.3 Laufzeitmessungen zur Extraktion und zur Distanzberechnung

Neben der bereits betrachteten Ergebnisqualität, spiel die benötigte Rechenzeit zum Finden eines Suchergebnisses im Retrieval-Bereich eine sehr große Rolle. Diese soll ebenfalls gemessen werden, da sie als ein Kriterium bei der Auswahl der zu verwendenden Feature-Menge herangezogen werden kann. Beispielsweise kann auf Grundlage der Laufzeit eine Entscheidung über den Einsatz eines oder mehrerer Verfahren getroffen werden, falls verschiedene Features ähnlich gute Ergebnisse liefern oder bei der Korrelationsanalyse eine Gruppe bilden. Zudem ist die Rechenzeit in laufzeitkritischen Systemen der entscheidende Faktor bei der Feature-Auswahl.

Für die Analyse der Laufzeit wurden aus der Bildsammlung 101Categories zufällig die folgenden 10 Bildkategorien *accordion, brain, chandelier, dolphin, flamingo, ibis, lotus, pagoda, strawberry* und *wrench* ausgewählt. Aus diesen 10 Klassen wurden wiederum 10 Bilder entnommen, wodurch die Testdatenbank für die Laufzeitanalyse insgesamt 100 Bilder der genannten Kategorien umfasst. Mit jedem dieser Bilder wird auf der Testmenge angefragt und die jeweiligen Laufzeiten bestimmt, wobei bei jeder Anfrage stets nur ein Feature berechnet wird. Als Distanzfunktion werden die in Kapitel 5-

2 genannten Distanzen beibehalten. Die Messungen erfolgen auf einem 64-Bit Windows7 Betriebssystem. Das Testsystem besitzt einen Intel(R) Core(TM)i7-2600K CPU@3.40GHz Quadcore-Prozessor, mit einer NVIDIA GeForce GTX 570 Grafikkarte und 16,0 GB RAM. Zur Erhöhung der Schreib- und Lesegeschwindigkeit wurden alle Berechnungen zudem auf einer C300-CTFDDAC128MAG SSD-Festplatte durchgeführt. Zu erwähnen ist außerdem, dass vor der Extraktion von lokalen Features alle Bilder in Pythia auf maximal 512 Pixel skaliert werden, um die Rechenzeit für zur Berechnung der Features und der Distanzen zu weiter senken.

Wie in Abbildung 4-1 dargestellt unterteilt sich der Suchprozess in Pythia in zwei Phasen. Aus diesem Grund soll die Rechenzeit zum Extrahieren und zur Berechnung von Distanzwerten separat betrachtet werden. Bei der Feature-Extraktion wird dabei zum einen die benötigte Zeit zum Berechnen der Features bestimmt. Zum anderen wird auch die Zeit zum Schreiben der XML-Dokumente gemessen. Analog wird bei der Distanzberechnung die Zeit zum Einlesen der XML-Dateien losgelöst vom Bestimmen der Distanzwerte analysiert. Somit lässt sich später ein Rückschluss auf die Zeiten verschiedener Prozessabschnitte ziehen. Die nachfolgende Abbildung zeigt eine Übersicht der Rechenzeiten zur Extraktion der Merkmalswerte für unterschiedliche Features. Dabei sind im Diagramm lokale und globale Extraktionsmethoden getrennt dargestellt. Die entsprechenden Messwerte sind in der Tabelle 7-4 veranschaulicht.

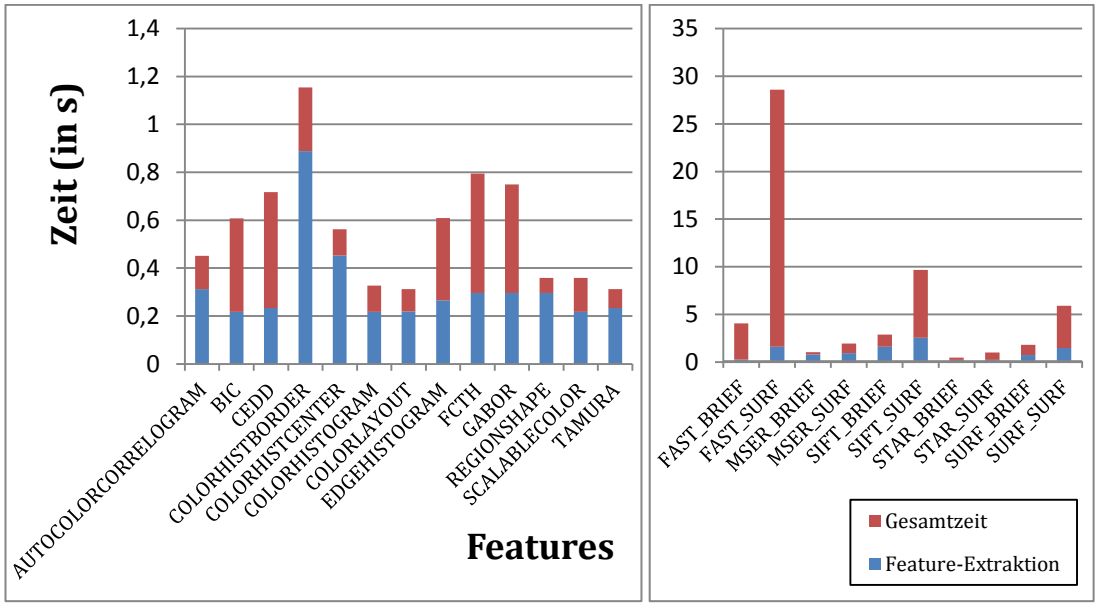

Abbildung 5-11: Rechenzeit zur Feature-Extraktion in Pythia

Die Abbildung 5-11 stellt die notwendige Rechenzeit zum Extrahieren von verschiedenen in Pythia implementierten Features dar. Die angegebene Zeit bezieht sich jeweils auf die Feature-Extraktion eines Bildes, wobei sich dieser Messwert aus dem Mittelwert der benötigten Zeit aller hundert Bilder ergibt. In der Abbildung ist zu jedem Feature neben der gesamten gemessenen Zeit ebenfalls die davon verwendete Zeit zur Detektion und Deskription der Features aufgeführt. Die Differenz zwischen Gesamtzeit und Zeit für die Extraktion ergibt die benötigte Zeit zum Schreiben der XML-Dokumente. Aus der Darstellung wird ersichtlich, dass die Berechnung der globalen Features im Vergleich zu den lokalen Werten, welche im rechten Diagrammbereich abgebildet sind, relativ wenig Zeit benötigt. Des Weiteren sind die Rechenzeiten aller globalen Features recht ähnliche und liegen um 250ms für die Extraktion und weitere 250ms für das Schreiben der Werte in Dateien (vgl. Tabelle 7-4). Verhältnismäßig lang

mit 1,2s braucht die Berechnung von COLORHISTBORDER, wobei der überwiegende Anteil an der Rechenzeit die Extraktion einnimmt. Anders verhält es sich bei den lokalen Features. Bei den meisten nehmen das Schreiben und das für die Distanzberechnung anschließende Lesen der XML-Dokumente den Großteil der Rechenzeit ein. Aber auch die reine Zeit zum Berechnen der Features ist höher als bei den globalen Verfahren. Der entscheidende Faktor für die Rechenzeit ist die Anzahl der extrahierten Merkmalswerte. Während bei den meisten globalen Extraktionsverfahren meist nur wenige Features detektiert und beschrieben werden müssen, sind es bei fast allen lokalen Algorithmen einige Hundert bis Tausend Werte, wobei jedes Feature je nach verwendeten Deskriptor 32 bis 64 Dimensionen besitzt (vgl. Tabelle 7-4). Eine Ausnahme bildet dabei STAR. Wie in Kapitel 3 beschrieben wird beim STAR-Detektor eine sehr kompakte und relativ schnell zu berechnende Darstellung der lokalen Bildeigenschaften ermittelt. Vor allem der geringen Anzahl der detektierten Features verdank STAR, dass es bei der Extraktionszeit auf dem Niveau der globalen Features liegt. Ebenfalls eine sehr gute Rechenzeit besitzt der MSER-Detektor. Die Regionen-basierten Detektionsverfahren SIFT und SURF besitzen mit bis zu 10s eine mittlere Rechenzeit, wobei das Schreiben der Dokumente den Hauptanteil daran trägt. Mit Abstand die meiste Zeit für die Extraktion benötigt FAST. Wird der FAST-Detektor in Pythia mit einem BRIEF-Deskriptor kombiniert, braucht die Extraktion nur rund 4s. Der Einsatz zusammen mit einem SURF-Deskriptor erhöht die Rechenzeit dagegen auf über 28s pro Bild. Die Hauptursache für die extremen Laufzeiten bei FAST ist die hohe Anzahl der detektierten Merkmalswerte, welche sich auch bei der Distanzberechnung negativ auf die benötigte Rechenzeit auswirkt.

Bei der Analyse der Distanzberechnung soll ebenfalls das Lesen der gespeicherten Features getrennt von der Berechnung der Distanzwerte untersucht werden. Genau wie bei der Extraktion werden zur besseren Anschauung wieder die globalen von den lokalen Verfahren getrennt betrachtet, da bei Ersteren die Rechenzeit im Millisekunden-Bereich liegt und die lokalen Features dagegen mehrere Sekunden benötigen. Die Angaben beziehen sich jeweils auf die Berechnung einer Anfrage, also das Lesen der einhundert Dokumente mit anschließender Ermittlung der Distanzen. Die folgende Abbildung zeigt die in Pythia verwendete Rechenzeit zur Distanzermittlung bei den globalen Features. Die Messwerte sind in Tabelle 7-5 zusammengefasst.

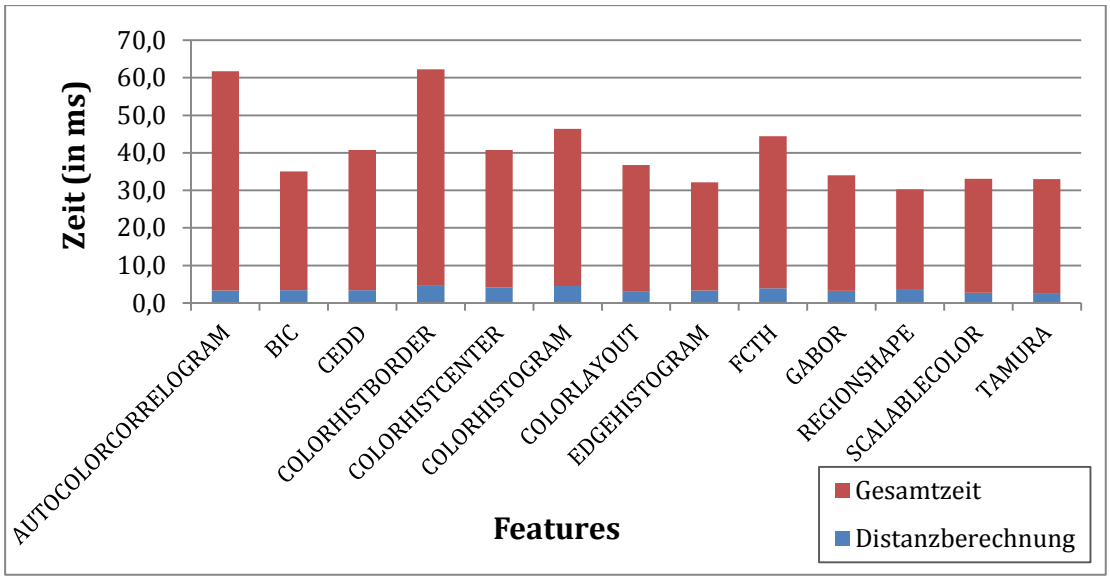

Abbildung 5-12: Rechenzeit zur Distanzberechnung bei globalen Features in Pythia

Ähnlich der Extraktion existieren keine signifikanten Unterschiede zwischen den einzelnen globalen Verfahren. Eine Anfrageauswertung benötigt je nach verwendetem Feature zwischen 30ms und 60ms, wobei die Berechnung der Distanzwerte bei allen Verfahren den geringsten Anteil an der Rechenzeit hat. Die Abweichungen beruhen auf der unterschiedlichen Anzahl an extrahierten Merkmalswerten, wodurch beim Einlesen der Dokumente mehr oder weniger Zeit benötigt wird. Die beiden Features mit der höchsten Gesamtrechenzeit, AUTOCOLORCORRELOGRAM und COLORHISTBORDER, besitzen unter den globalen Features ebenfalls die höchste Zahl extrahierter Features, wie in der Tabelle 7-4 zu sehen ist. Die Dimensionalität der Features spielt dagegen bei den globalen Verfahren keine große Rolle. COLORHISTOGRAM benötigt mit seinen 512-dimensionalen Features nur eine mittlere Rechenzeit. Der Einfluss der Werteanzahl auf die Zeit zur Distanzberechnung ist dagegen gering, wie der Abbildung 5-12 zu entnehmen ist. Die Rechenzeit einer Anfrage ist bei lokalen Features um ein Vielfaches höher, was in Abbildung 5-13 dargestellt ist und in Tabelle 7-6 nachvollzogen werden kann.

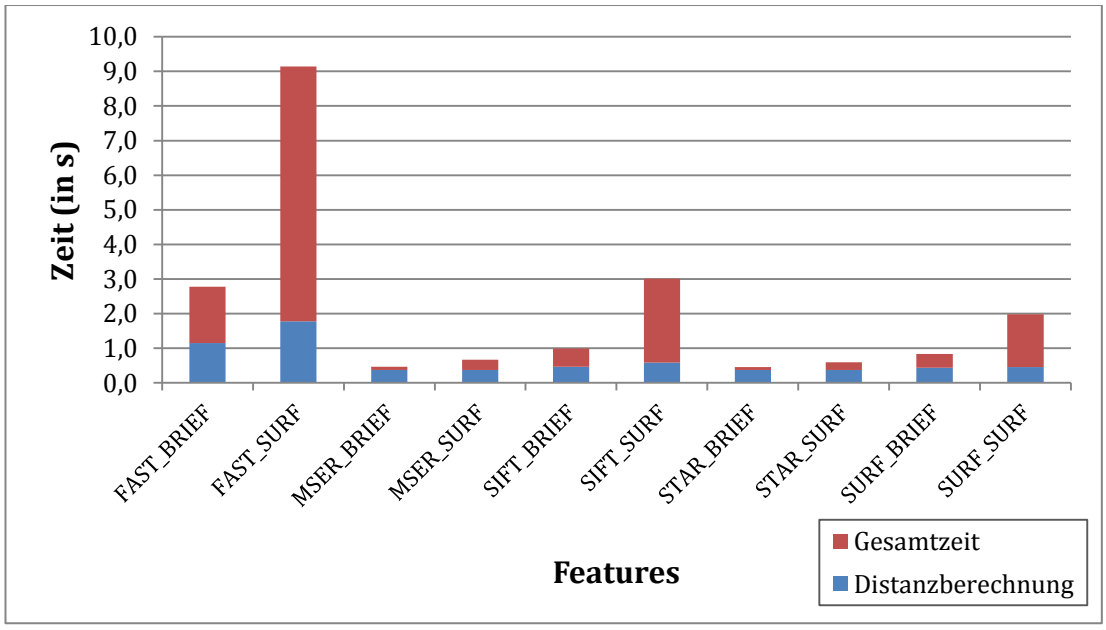

Abbildung 5-13: Rechenzeit zur Distanzberechnung bei lokalen Features in Pythia

Bei der Berechnung von Distanzwerten bestätigen sich die bei der Extraktion getroffenen Thesen. Generell benötigt jedes Feature, welches mit einem SURF-Deskriptor kombiniert wird, deutlich mehr Zeit für die Distanzberechnung als beim Einsatz des BRIEF-Deskriptors. Eine Ursache dafür ist die verdoppelte Anzahl der Dimensionswerte pro detektiertes Feature bei SURF. Dies wird durch den Umstand verstärkt, dass bei der Distanzberechnung einer BRIEF-Deskription eine Hamming-Distanz auf einfachen Binärzeichenketten angewandt werden kann (vgl. Kapitel 3.4). Somit ist auch die Berechnung der Distanzwerte deutlich schneller als die euklidische Distanzberechnung beim SURF-Deskriptor. Am meisten Zeit aufgrund der vielen Merkmalswerte benötigt FAST. Interessant sind zudem die Rechenzeiten bei MSER und STAR. Wie bereits erläutert zeichnen sich beide Methoden durch sehr wenige ermittelte Features aus, wodurch die Zeit zum Berechnen der Distanzen im Mittel höher als die Zeit zum Einlesen der Werte ist. Voraussetzung hierfür ist, dass, wie bei diesem Versuch geschehen, möglichst viele Dokumente vor der Berechnung eingelesen werden.

Zusammenfassend kann festgestellt werden, dass der BRIEF-Deskriptor durch die kompaktere Beschreibung in Form von Binärzeichenketten weniger Zeit bei der Dis-

tanzberechnung von lokalen Werten beansprucht. MSER und STAR sind damit annähernd in dem Bereich der globalen Verfahren. Der FAST-Detektor benötigt dagegen bei der Extraktion und der Distanzberechnung, vor allem in Verbindung mit einem SURF-Deskriptor, mit großem Abstand am meisten Zeit. Allgemein brauchen die lokalen Features aufgrund der höheren Zahl von ermittelten Werten sowohl bei der Extraktion, als auch bei der Distanzberechnung, mehr Rechenzeit. Um diese Kluft zu minimieren, wurden am Lehrstuhl Datenbank- und Informationssysteme bereits einige Entwicklungen getätigt. Beispielsweise erfolgt die Berechnung der lokalen Features nicht wie bei den globalen Verfahren auf der CPU, sondern ist GPU-basiert, was nach ersten Analysen etwa um den Faktor 7 schneller ist. Zurzeit wird am Lehrstuhl DBIS noch an mehreren Aspekten geforscht. Zum einen wird an einer XML-unabhängigen Speichermöglichkeit der Features gearbeitet, wodurch die Schreib- und Lesezeiten der extrahierten Merkmale minimiert werden. Außerdem soll die Distanzberechnung der lokalen Verfahren durch einen speziellen Clusteransatz aus dem Text-Retrieval-Bereich optimiert werden. Durch eine globale Zerlegung des Feature-Raums und einer Zusammenfassung von mehreren Werten zu einem Feature-Block (vgl. Kapitel 6.2) ist in Zukunft eine Verkürzung der Distanzermittlung möglich.

5.4 Korrelationsanalyse und Clustering

In diesem Abschnitt soll auf Basis der durch Pythia getätigten Relevanzbewertungen eine Aussage über den statistischen Zusammenhang der verschiedenen Retrieval-Verfahren getroffen werden. In den vorherigen Kapiteln werden hierfür verschiedene Analysemethoden vorgestellt und deren Implementierung veranschaulicht. In Kapitel 5.1.2 werden zudem einige Vorteile von minimalen Feature-Mengen erörtert. Die Ergebnisse der durchgeführten Korrelationsanalysen, sowie eines distanzbasierten Clusterings, sind im Folgenden beschrieben und durch verschiedene Diagramme visualisiert. Begonnen wird dabei mit dem Clustering der Features auf Basis der Abweichung der getätigten Relevanzbewertungen.

Wie in Kapitel 5.1.2 erläutert wurde, wird bei diesem Ansatz der Zusammenhang zwischen zwei Features über eine Manhattan-Distanz bestimmt. Das Ergebnis dieser Analyse ist eine n x n-Distanzmatrix für die n untersuchten Verfahren. Auf der Basis dieser Matrix wird das Clustering durchgeführt, wobei für die hier durchgeführte Analyse Complete-Link als Clustering-Methode gewählt wird. Durch das im vierten Kapitel vorgestellte Analysewerkzeug wird im Anschluss folgendes Dendogramm erstellt, welches die durch den Analyseprozess aufgedeckten Zusammenhänge der einzelnen Features darstellt.

Die Abbildung 5-14 schlüsselt den Clusterbildungsprozess in *n-1* Vereinigungsschritte auf. Dabei nimmt die Distanz der Cluster bei der Verschmelzung bei jedem Schritt zu. Die geringste Distanz zu einander und damit einen sehr starken Zusammenhang besitzen GABOR, TAMURA und SIFT_BRIEF. Die Cluster dieser Verfahren werden im Dendogramm zuerst vereinigt. Die größte Distanz zu den übrigen Features haben FCTH, EDGEHISTOGRAM und SIFT_SURF. Somit ist der Zusammenhang der Verfahren zu den anderen Extraktionsmethoden vergleichsweise gering. Sie sollten daher aus der Sichtweise der Zusammenhangsbetrachtung unter den gegebenen Testbedingungen alle eingesetzt werden.

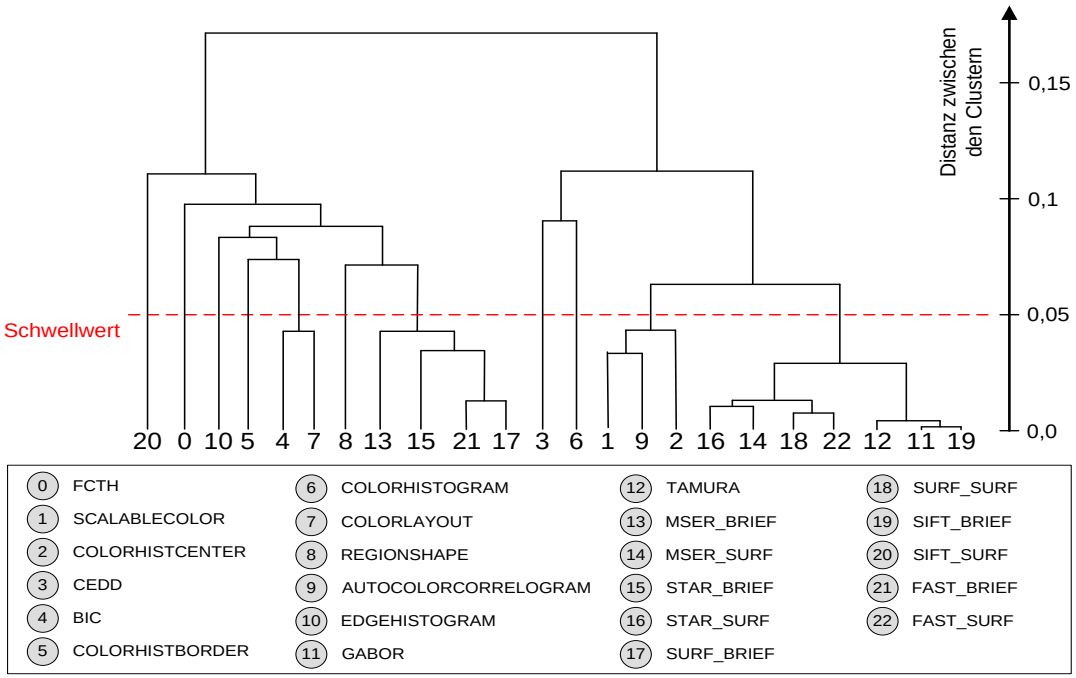

Abbildung 5-14: Dendogramm eines hierarchischen Clusterings der Features in Pythia

Die Distanzen zwischen allen Verfahren ist durch die Kalibrierung und der damit einhergehende Annäherung der Features aneinander allgemein relativ gering, was bei der Grenzwertbetrachtung mit berücksichtigt werden muss. Durch das Setzen des maximalen Schwellwertes für die Vereinigung auf 0,05 werden hier elf Cluster definiert (vgl. Abbildung 5-14), welche die dreiundzwanzig Verfahren enthalten. Dabei gilt, aus Clustern mit mehreren Elemente genügt die Wahl eines Vertreters für die Analyse. Verfahren, die einen eigenen Cluster bilden, sollten nach Möglichkeit alle verwendet werden. Die folgende Abbildung visualisiert die entstandenen Cluster in anderer Art und Weise. Für diese Darstellung wird die Graphvisualisierungssoftware aiSee3 verwendet (vgl. Kapitel 4), wobei die erstellte Grafik zur Hervorhebung der ermittelten Cluster zusätzlich noch nachträglich bearbeitet wurde.

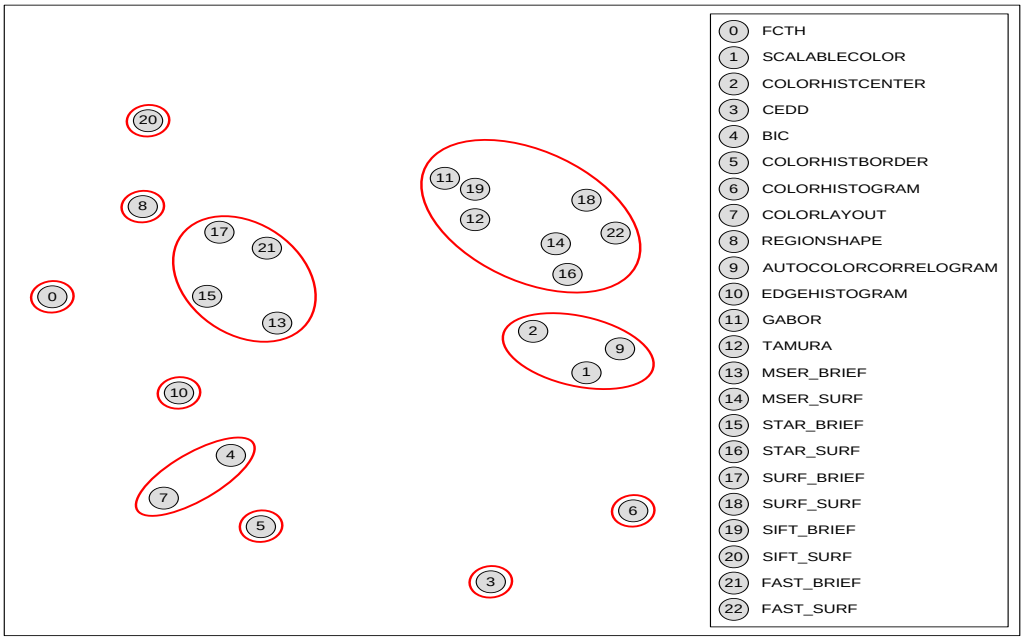

Abbildung 5-15: Visualisierung Cluster der Features in Pythia

Die durch das Clustering ermittelten Gruppen sollen nun durch eine Korrelationsanalyse überprüft werden. Dabei wird der prinzipielle Bewertungsverlauf zweier Verfahren auf einen Zusammenhang hin untersucht. Dieser wird anschließend in Form eines Korrelationskoeffizienten angegeben. Das Ergebnis dieser Analyse ist eine *n* x *n* Korrelationsmatrix, welche die Korrelation der *n* Features wiedergibt. Features mit geringen Korrelationskoeffizienten besitzen dabei keinen signifikanten Zusammenhang. Ein Beispiel für zwei nicht-korrelierte Features ist EDGEHISTOGRAM und GABOR mit einem Korrelationskoeffizienten von -0,01. Dieser geringe Zusammenhang lässt sich auch im folgenden Streudiagramm ablesen.

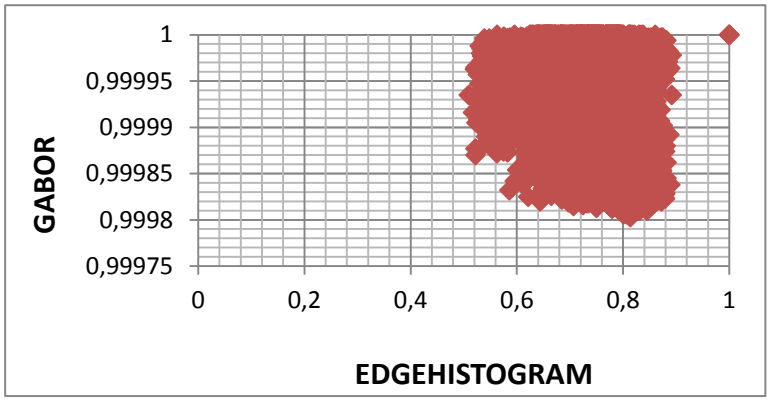

Abbildung 5-16: Beispielstreudiagramm für Features mit geringer Korrelation

Im Gegensatz zum distanzbasierten Clustering wird bei der Korrelationsanalyse der Werteverlauf anstatt des Abstands zwischen den Werten untersucht. Aus diesem Grund ist es auch nicht notwendig, dass die Verteilung der Werte angenähert wird. Eine Kalibrierung würde zudem das Korrelationsergebnis beeinflussen, weshalb bei dieser Analyse auf die vorherige Kalibrierung der Ähnlichkeitsbewertungen verzichtet wird. Da die Bildung von Clustern mit dem entwickelten Analysewerkzeug auf Distanzen basiert, Korrelationskoeffizienten hingegen Ähnlichkeiten ausdrücken, müssen die Werte vor der Clusterbildung in Distanzwerte transformiert werden. Die Forderung nach gleichem Werteverlauf ist zudem stärker als die zuvor durchgeführte Abstandsbetrachtung, weshalb die Korrelationsanalyse im Allgemeinen weniger korrelierende Feature liefert. Bei diesem Test werden insgesamt zwanzig Gruppen für die dreiundzwanzig Features erkannt. Demnach besteht ein starker Zusammenhang zwischen SURF_BRIEF und FAST_BRIEF, sowie zwischen CEDD, BIC und COLOR_LAYOUT. Dieser ist in nachfolgender Abbildung dargestellt und kann mit dem geringen Zusammenhang der in Abbildung 5-16 untersuchten Features verglichen werden.

Bei einem durchgeführten t-Test mit einer Irrtumswahrscheinlichkeit von $\alpha=0{,}01$ wird bestätigt, dass der evaluierte Zusammenhang der Korrelationsergebnisses signifikant ist.

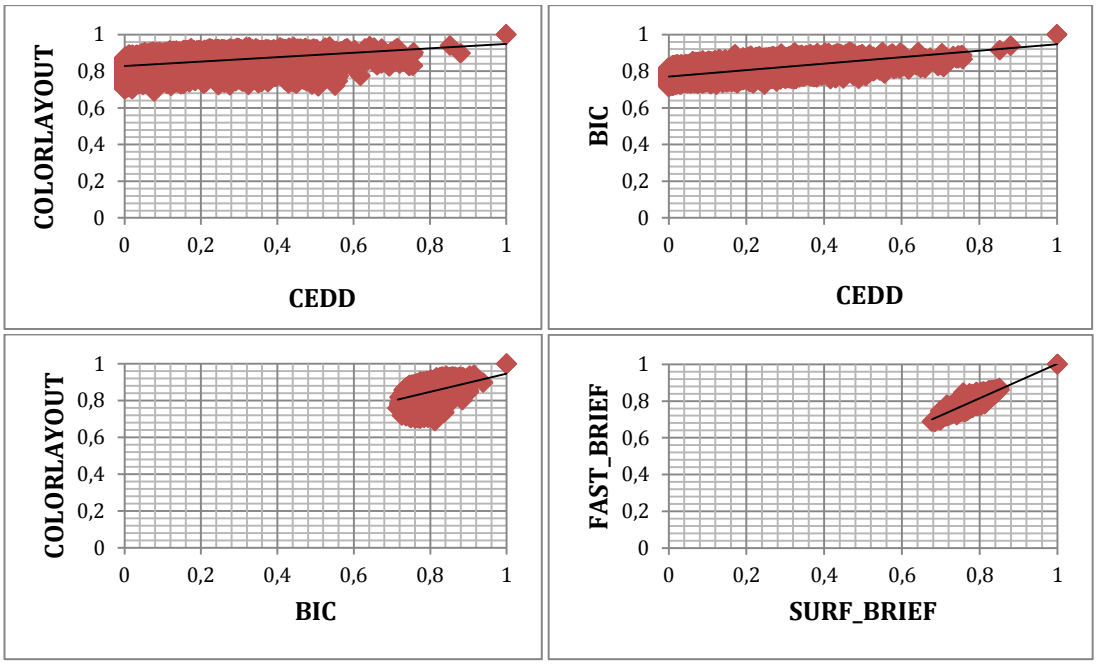

Abbildung 5-17: Beispielstreudiagramm für Features mit starker Korrelation

Bei beiden Untersuchungsverfahren wurde ein Zusammenhang zwischen einzelnen Features festgestellt. Dabei konnten durch das distanzbasierte Clustering wesentlich mehr bzw. größere Gruppen identifiziert werden. Da bei den durchgeführten Analyseverfahren grundsätzlich verschiedene Aspekte untersucht werden, ist eine Überdeckung der Ergebnisse allgemein nicht gegeben. Eine Überschneidung gibt es lediglich im Fall von COLORLAYOUT und BIC, bei denen sowohl beim Clustering, als auch bei der Korrelationsanalyse ein signifikanter Zusammenhang festgestellt wird.

5.5 Einfluss von Distanzfunktionen

Wie bereits bei der Vorstellung des Retrieval-Systems Pythia erläutert wurde, sind für jedes Feature zum Teil mehrere Distanzfunktionen implementiert. Der Grund hierfür ist, dass das System so an unterschiedliche Anforderungen angepasst werden kann. Bei vorangegangenen Untersuchungen am Lehrstuhl Datenbank- und Informationssysteme hat sich gezeigt, dass die Verwendung einer anderen Distanzfunktion das Suchergebnis beeinflussen kann. In diesem Abschnitt soll die Einflussnahme der gewählten Distanzfunktion evaluiert werden. Dabei sind folgende drei Kriterien von Bedeutung. Zuerst werden die Schwankungen der Ergebnisqualität bei unterschiedlichen Distanzen anhand der eingeführten IR-Maße gemessen. Nachfolgend wird der Einfluss auf die notwendige Rechenzeit bei der Distanzberechnung bestimmt. Zuletzt wird geprüft, ob sich durch eine andere Distanzfunktion der Zusammenhang der Extraktionsverfahren ändert.

Analog zur Gegenüberstellung der einzelnen Feature wird bei der Evaluierung der Retrieval-Qualität unterschiedlicher Distanzfunktionen die Precision an markanten Punkten bestimmt. Da die Untersuchung, wie in Abschnitt 5.3 beschrieben, auf einer Teilmenge von hundert Bildern der Kollektion 101Categories durchgeführt wird, werden hier nur der P@5-, P@10-, P@15- und der P@20-Wert bestimmt. Die folgenden beiden Abbildungen zeigen den Vergleich der Retrieval-Bewertungen.

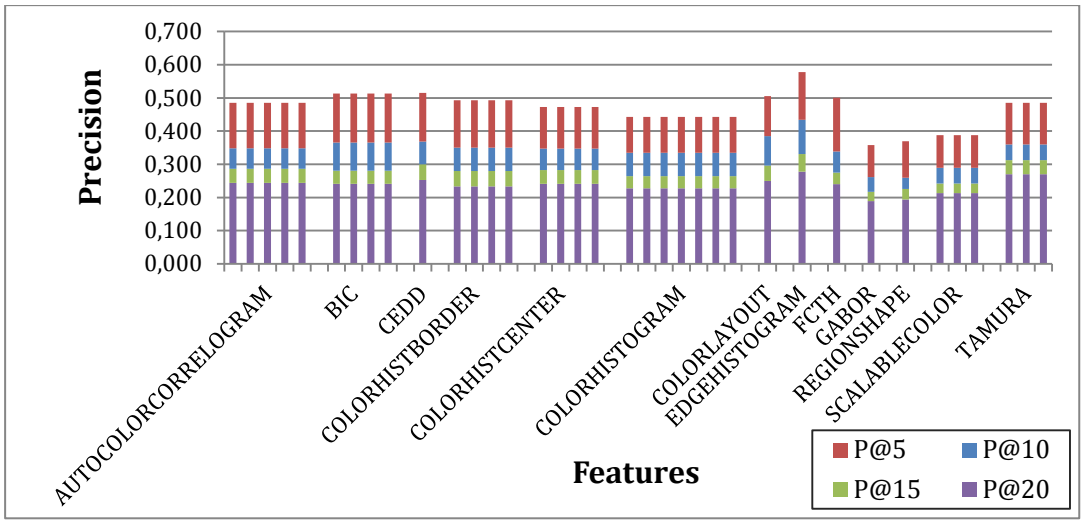

Abbildung 5-18: Gegenüberstellung der Precision@X globaler Features in Pythia unter Verwendung verschiedener Distanzfunktionen

Aus Abbildung 5-18 und Abbildung 5-19 wird deutlich, dass bei diesem Test die Distanzfunktion nahezu keinen Einfluss auf das Suchergebnis hat. Bei allen getesteten Features sind die Precision@X bei jeder Distanzfunktion gleich hoch. Zudem sind die Werte höher als bei den in Kapitel 5.2 getätigten Analysen. Der Grund hierfür kann die reduzierte Bildmenge bei diesem Test sein. Da der Anteil der relevanten Bilder der Kategorien an der Gesamtanzahl der Bilder durch die Reduktion im Vergleich zu Kapitel 5.2 höher ist, steigt auch die Wahrscheinlichkeit eines dieser Bilder zu wählen. Des Weiteren sind nur geringe Unterschiede zwischen den untersuchten lokalen und globalen Features feststellbar. Bei Beiden sinkt die Precision zwischen dem fünften und zehnten Treffer stärker und bleibt danach annähernd auf einem Niveau. Trotzdem ist unabhängig von der Distanzfunktion bei allen Features eine stetige Verschlechterung des Suchergebnisses nach dem fünften Dokument der Ergebnisliste zu erkennen.

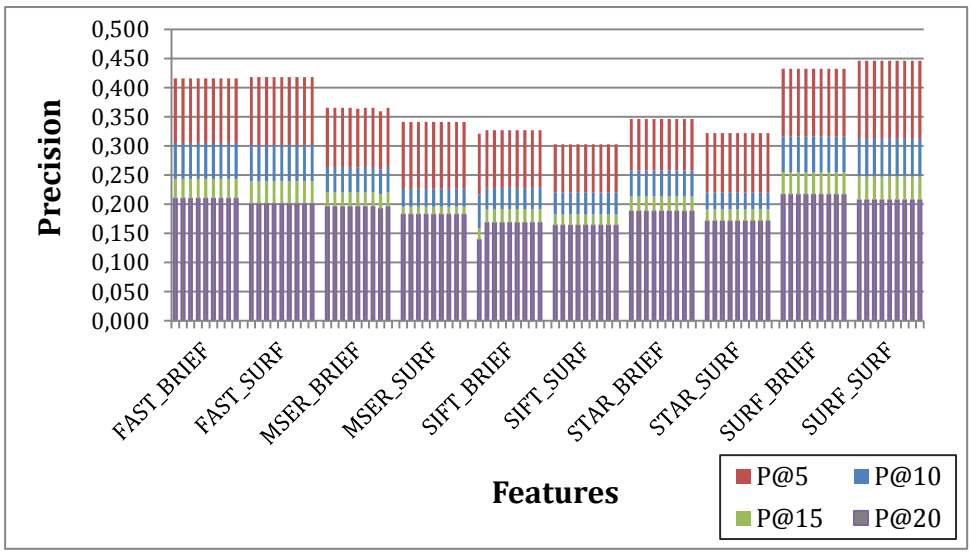

Abbildung 5-19: Gegenüberstellung der Precision@X lokaler Features in Pythia unter Verwendung verschiedener Distanzfunktionen

Als Nächstes soll der Einfluss der Distanzwahl auf die Rechenzeit gemessen werden. Dazu wird, wie in Kapitel 5.3, die Gesamtrechenzeit einer Anfrage und die Zeit zum Berechnung der Distanzwerte betrachtet. Dies geschieht sowohl für die lokalen, als auch für die globalen Features, um etwaige Unterschiede durch die Distanzwahl zu un-

tersuchen. Die folgende Abbildung zeigt eine Laufzeitübersicht der globalen Features. Gleiche Features mit unterschiedlichen Distanzfunktionen sind dabei gruppiert angeordnet.

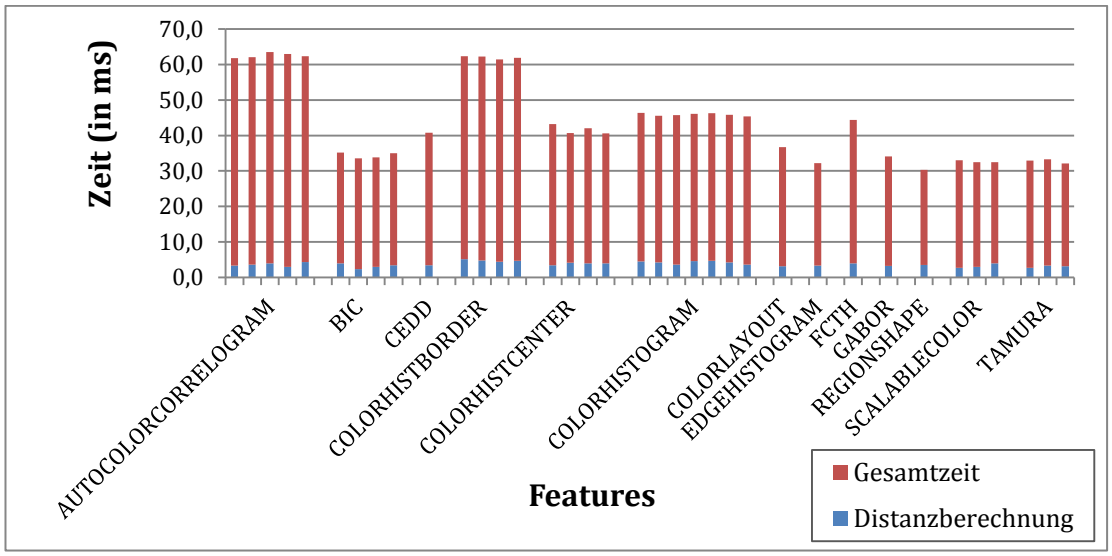

Abbildung 5-20: Rechenzeit zur Distanzberechnung bei globalen Features mit unterschiedlichen Distanzen in Pythia

In der Darstellung ist zu erkennen, dass bei verschiedenen Distanzen zu einem Feature nur sehr geringe Unterschiede in der Rechenzeit zu messen sind. Die Ursache hierfür ist die in Kapitel 5.3 beschriebene Abhängigkeit der Laufzeit von der Anzahl der extrahierten Features, welche bei der Verwendung verschiedener Distanzfunktionen gleich bleibt. Verglichen mit der Zeit zum Einlesen der Features, welche überwiegend von der Anzahl der Werte abhängt, hat die Distanzberechnung einen sehr geringen Einfluss. Somit bleibt festzuhalten, dass die Wahl der Distanzfunktion die Bearbeitungszeit einer Anfrage nicht signifikant ändert. Ähnlich ist der Zusammenhang auch bei den lokalen Features, was in Abbildung 5-21 verdeutlicht wird. Hier ist der Einfluss der Distanzberechnung aufgrund der höheren Zahl der Merkmalswerte noch geringer.

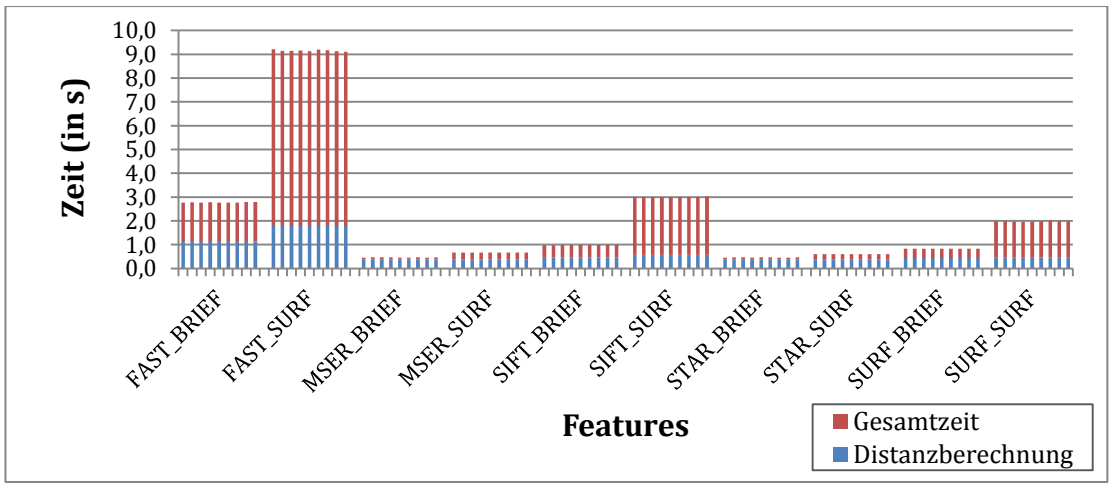

Abbildung 5-21: Rechenzeit zur Distanzberechnung bei lokalen Features mit unterschiedlichen Distanzen in Pythia

Aus den bisherigen Ergebnissen ist ersichtlich, dass im durchgeführten Test die Wahl der Distanzfunktion keinen Einfluss auf die Berechnung der Ähnlichkeitsbewertungen nimmt. Daraus folgt, dass sich auch die Korrelation der untersuchten Features

durch die Distanzwahl nicht beeinflussen lässt. Die nachfolgende Tabelle zeigt eine Übersicht über die gefundenen Cluster, welche stark zusammenhängende Features beinhalten. Diese gilt aus den erwähnten Gründen für alle Distanzverfahren. Zur Untersuchung wurde die bereits beschriebene 100-elementige Teilmenge der Bildkollektion 101Categories verwendet.

Cluster	Features
1	COLORHISTCENTER, COLORHISTOGRAM, FAST_SURF, GABOR, MSER_SURF, SCALABLECOLOR, SIFT_BRIEF, SIFT_SURF, STAR_SURF, SURF_SURF
2	FAST_BRIEF, SURF_BRIEF
3	COLORHISTBORDER, TAMURA
4	BIC, COLORLAYOUT
5	AUTOCOLORCORRELOGRAM
6	CEDD
7	EDGEHISTOGRAM
8	FCTH
9	MSER_BRIEF
10	REGIONSHAPE
11	STAR_BRIEF

Tabelle 5-5: Feature-Cluster bei einem distanzbasierten Complete-Link-Clustering mit einem Schwellwert von 0,05

Bei der durchgeführten Analyse hat die Wahl der Distanzfunktion keine Auswirkung auf die Berechnung der Ähnlichkeiten bzw. auf die daraus folgende Retrieval-Qualität. Das bedeutet allerdings nicht, dass sie generell keinen Einfluss auf die Suche hat. Bei anderen Suchparametern oder einer geänderten Bildkollektion ist ein anderes Untersuchungsergebnis denkbar. Auch auf die Laufzeit kann die Distanzfunktion im Test keinen Einfluss nehmen. Der Grund hierfür ist die in Kapitel 5.3 beobachtete Aufteilung der Rechenzeit auf die verschiedenen Abläufe. Zurzeit nimmt das Lesen der Feature-Dokumente weit mehr Zeit als das eigentliche Berechnen von Distanzwerten ein. Da die Komplexität der verschiedenen Distanzen sich nicht signifikant unterscheidet, ist auch bei der Zeitnahme keine Beeinflussung ermittelbar. Für den statistischen Zusammenhang der Features gilt ebenfalls, dass er in den Untersuchungen nicht von der Distanzwahl abhängt. Insgesamt konnte in allen durchgeführten Tests kein messbarer Unterschied der betrachteten Systemeigenschaften, der auf die Wahl der Distanzfunktion zurückzuführen ist, beobachtet werden. Im Folgenden soll evaluiert werden, ob über Suchparameter eine Einflussnahme möglich ist.

5.6 Einfluss von Suchparametern

Ein Ergebnis dieser Arbeit ist, dass die Verteilung der Distanzwerte Einfluss auf verschiedene Aspekte, wie die Qualität des Suchergebnisses und den messbaren Zusammenhang verschiedener Features hat. Die Werteverteilung kann wiederum durch durch das Setzen von Parameter bei der Suche beeinflusst werden. Ein solcher Suchparameter ist der Threshold, welcher im Folgenden näher betrachtet werden soll. Der Threshold ist ein Schwellwert, der bei den Matching-Distanzen lokaler Features in Pythia (Dice-Koeffizient u.a.) die akzeptierte prozentuale Abweichung der Feature-Vektoren steuert. Er liegt dementsprechend im Intervall [0.0,…,1.0]. Ein Wert nah null bewirkt, dass bereits minimale Abweichungen als unähnlich betrachtet werden. Daraus folgt, dass bis auf das zur Anfrage identische Bild alle Objekte mit einer Ähnlichkeit

dicht an null bewertet werden. Auf der anderen Seite bewirkt ein Threshold nah an eins, dass sich alle Bewertungen um eins bewegen. In beiden Fällen ist kein akzeptables Suchergebnis zu erwarten, da alle Bilder ähnlich bewertet werden und bereits geringe Unterschiede mehrere Plätze im Ranking ausmachen können.

Aber nicht nur die Retrieval-Qualität, sondern auch weitere Untersuchungsergebnisses sind von dem gesetzten Schwellwert abhängig. In Kapitel 5.4 werden die Ähnlichkeitswerte im Nachhinein kalibriet, um die Voraussetzungen für das Clustering zu schaffen. Dies ist allerdings nur eine Annäherung der Werteverteilungen verschiedener Verfahren aneinander. Wesentlich genauer wären die Analyseergebnisses, wenn bereits zuvor alle Features gleichverteilt sind. Die folgende Abbildung zeigt, wie sich die Verteilung der Werte durch Änderung des Thresholds beeinflussen lässt. Bei diesem Test wird beispielhaft das Feature MSER_BRIEF in Kombination mit einer *kHamming* | *kCosineSet*-Distanz untersucht.

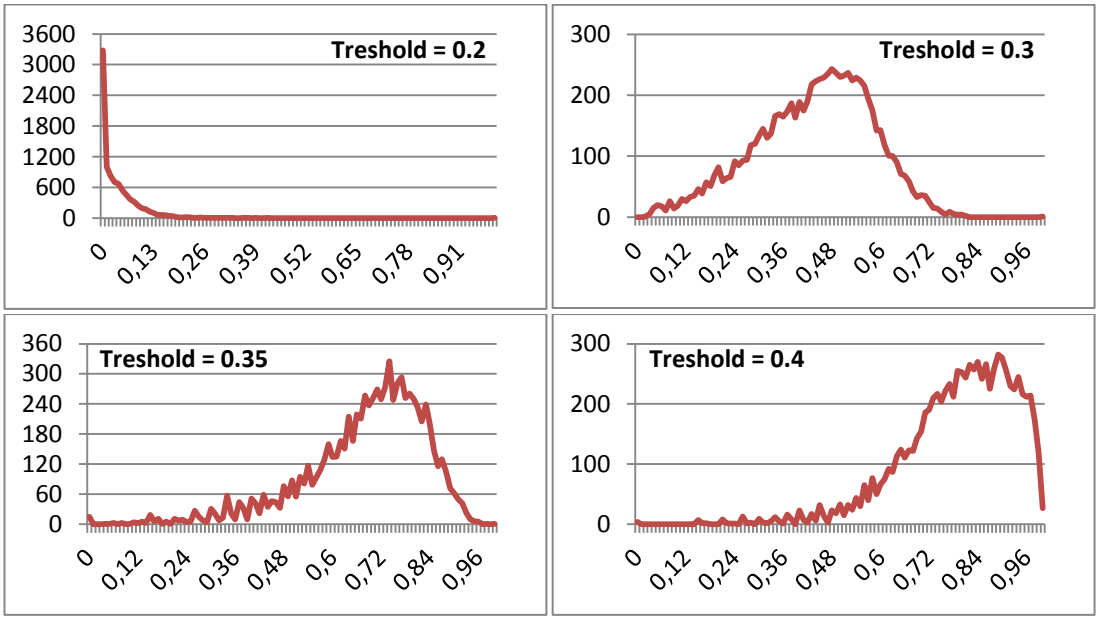

Abbildung 5-22: Einfluss des Thresholds auf die Werteverteilung bei MSER_BRIEF

Das Ergebnis der Schwellwertbetrachtung ist, dass um möglichst gleichmäßig über den gesamten Wertebereich verteilte Bewertungen zu erhalten, die Verteilung einen Mittelwert nah 0,5 und eine maximale Varianz besitzen sollte. Da die Varianz der Ähnlichkeitswerte aufgrund des vergrößerten Wertebereichs mit zunehmenden Threshold ebenfalls steigt, kann der hierfür ideale Schwellwert allein über den Erwartungswert der Ähnlichkeiten bestimmt werden. Im betrachteten Beispiel ergibt ein Schwellwert zwischen 0,3 und 0,35 den gewünschten Mittelwert von 0,5. Zudem fällt auf, dass bereits geringe Abweichungen die Werteverteilung stark beeinflussen. Eine Änderung des Thresholds von 0,1 bewirkt eine starke Verschiebung Richtung null bzw. eins (vgl. Abbildung 5-22). Eine ähnliche Betrachtung wurde für alle lokalen Features und alle Matching-basierten Distanzfunktionen durchgeführt. Das Ergebnis dieser Untersuchung ist in folgender Tabelle aufgeführt, welche die aus Verteilungssicht optimalen Thresholds in Abhängigkeit des gewählten Features und Distanz zeigt. Der ermittelte Threshold bezieht sich auf die bereits erwähnte 100-elemtige Teilmenge der 101Categories. Als Anfragebild wurde im Test das Bild 0001.png der Bildkategorie *accordion* gewählt. Da der Threshold auch von der gewählten Bildkollektion abhängt, sind Abweichungen bei anderen Bildsammlungen oder Anfragebildern möglich.

Features	kJaccard	kMinOverlap	kMaxOverlap	kDice	kCosineSet
FAST_BRIEF	0,2900	0,1900	0,2800	0,2350	0,2250
FAST_SURF	0,0250	0,0170	0,0240	0,0200	0,0190
MSER_BRIEF	0,3450	0,2900	0,3350	0,3150	0,3100
MSER_SURF	0,0350	0,0275	0,0340	0,0300	0,0300
SIFT_BRIEF	0,3000	0,2350	0,2850	0,2650	0,2600
SIFT_SURF	0,0265	0,0205	0,0250	0,0230	0,0225
STAR_BRIEF	0,3550	0,2750	0,3450	0,3150	0,3050
STAR_SURF	0,0370	0,0265	0,0360	0,0305	0,0295
SURF_BRIEF	0,3000	0,2150	0,2900	0,2500	0,2450
SURF_SURF	0,0280	0,0200	0,0265	0,0225	0,0220

Tabelle 5-6: Übersicht Thresholds für eine Gleichverteilung der verschiedenen Feature-Distanzfunktions-Kombinationen

Bei der Betrachtung der ermittelten Thresholds fällt auf, dass der Wert bei Verwendung des SURF-Deskriptors deutlich niedriger gewählt werden kann. Er beträgt rund ein Zehntel des Schwellwertes bei einer BRIEF-Beschreibung des gleichen Features. Soll eine Gleichverteilung der Ähnlichkeitswerte realisiert werden, ist er zudem distanzabhängig. KJaccard und kMaxOverlag zeigen im Test einen deutlich höheren Threshold. Bei der Verwendung von kMinOverlap ist dagegen ein niedrigerer Schwellwert zu nutzen. Allgemein sollte immer ein möglichst niedriger Schwellwert angesetzt werden, da in diesem Fall nur sehr ähnliche Bilder als relevant bewertet werden, was wiederum das Suchergebnis verbessert.

Als nächstes soll evaluiert werden, bei welchem Threshold die besten Retrieval-Ergebnisse zu erwarten sind und ob dieser Wert mit den ermittelten Schwellwerten aus Tabelle 5-6 korreliert. Da die Ermittlung des Schwellwertes für das beste Suchergebnis sehr aufwändig ist und hierfür Informationen über die Relevanz der gefundenen Dokumente zur Anfrage (ground truth) notwendig sind, sollte der Schwellwert nach Möglichkeit aus anderen Quellen abgeleitet werden können. Möglich wären hierbei die Ermittlung eines festen Schwellwertes, der unabhängig von der Suchanfrage gut performt, oder das Aufdecken eines Zusammenhangs zwischen der Ergebnisqualität und anderen Daten, welche beispielsweise aus der Verteilung abgeleitet werden können. Auch dies wird nachfolgend untersucht. Die in folgender Tabelle dargestellten Thresholds werden auf der gleichen Teilmenge der 101Categories ermittelt. Zur Erhebung wurden die ersten zehn Bilder der Kategorie *accordion* als Anfragebild verwendet und die Retrieval-Ergebnisse gemittelt. Der Threshold, welcher bei den zehn Anfragen die besten Ergebnisse liefert, wird als ideal betrachtet.

Features	kJaccard	kMinOverlap	kMaxOverlap	kDice	kCosineSet
FAST_BRIEF	0,2400	0,0900	0,2300	0,2350	0,2500
FAST_SURF	0,0275	0,0045	0,0240	0,0275	0,0190
MSER_BRIEF	0,2700	0,2150	0,2350	0,2150	0,2350
MSER_SURF	0,0150	0,0150	0,0140	0,0150	0,0150
SIFT_BRIEF	0,1750	0,1350	0,1350	0,1650	0,1600
SIFT_SURF	0,0165	0,0155	0,0150	0,0155	0,0150
STAR_BRIEF	0,2550	0,2500	0,2450	0,2650	0,2550
STAR_SURF	0,0195	0,0250	0,0235	0,0230	0,0195
SURF_BRIEF	0,2250	0,1900	0,1900	0,2250	0,2200
SURF_SURF	0,0180	0,0175	0,0190	0,0175	0,0170

Tabelle 5-7: Übersicht Thresholds für eine maximale Ergebnisqualität der verschiedenen Feature-Distanzfunktions-Kombinationen

Auch bei den ermittelten Thresholds für ein optimales Retrieval-Ergebnis liegt der Schwellwert bei der SURF-Deskription wieder deutlich unter dem der BRIEF-Beschreibung. Daraus lässt sich schlussfolgern, dass eine Kombination von mehreren Features, wobei einige mit einem BRIEF- und andere mit einem SURF-Deskriptor verwendet werden, mit einem festen Threshold nicht optimal ist. Anstatt eines einzigen Wertes für alle lokalen Features wäre daher die Verwendung mindestens zweier Schwellwerte für beide Gruppen empfehlenswert. Weiterhin fällt auf, dass sich der Threshold einer Detektor-Deskriptor-Kombination bei unterschiedlichen Distanzfunktionen oft nur gering unterscheidet. Bei größeren Abweichungen des Suchparameters handelt es sich meist um Features, bei denen sich auch durch Änderung dieses Parameters nur geringfügig bessere Ergebnisse erzielen lassen. Der Threshold kann somit nahezu unabhängig von der genutzten Distanzfunktion gewählt werden.

Die Anpassung des Schwellwertes führt allerdings nicht in jedem Fall zu einer signifikanten Verbesserung des Suchergebnisses. Wie der Tabelle 7-7 zu entnehmen ist, liegt die Precision nach fünf Treffern bei SIFT_BRIEF beispielsweise zwischen 0,2 und 0,3. Auch eine andere Wahl des Thresholds kann dieses schlechte Suchergebnis nicht verbessern. Bei SIFT_SURF hat der Schwellwert dagegen einen höheren Einfluss. Je nach Distanzfunktion liegt der P@5-Wert bei ideal angepasstem Schwellwert zwischen 0,5 und 0,6, was ein recht gutes Ergebnis ist. Allerdings nimmt dieser Wert bereits bei geringer Änderung des Thresholds stark ab. Häufig ist dabei die Wahl eines zu niedrigen Schwellwertes aus Sicht der Retrieval-Qualität auf den ersten Blick besser als diesen zu hoch anzusetzen (Tabelle 7-7). Da für alle hier untersuchten Verfahren die Selbstidentität gilt, wodurch das zur Anfrage identische Bild immer mit 1 bewertet ist, wird auch bei einem zu niedrigen Threshold eine P@5 von 0,2 erreicht. Dies ist nicht gegeben, wenn der Wert zu hoch angesetzt wird. Dann wird neben dem Anfragebild auch die Ähnlichkeit weiterer Bilder mit 1 bewertet, wodurch nicht mehr garantiert ist, dass das Anfragebild zuerst gefunden wird. Aber auch ein zu geringer Schwellwert liefert im Allgemeinen kein gutes Suchergebnis. Die nachfolgende Abbildung demonstriert die Abhängigkeit der Retrieval-Qualität vom gewählten Suchparameter. Hierfür wurde das Feature SIFT_SURF mit der Distanzfunktion kDice und einem Threshold zwischen 0,0080 und 0,0230 untersucht.

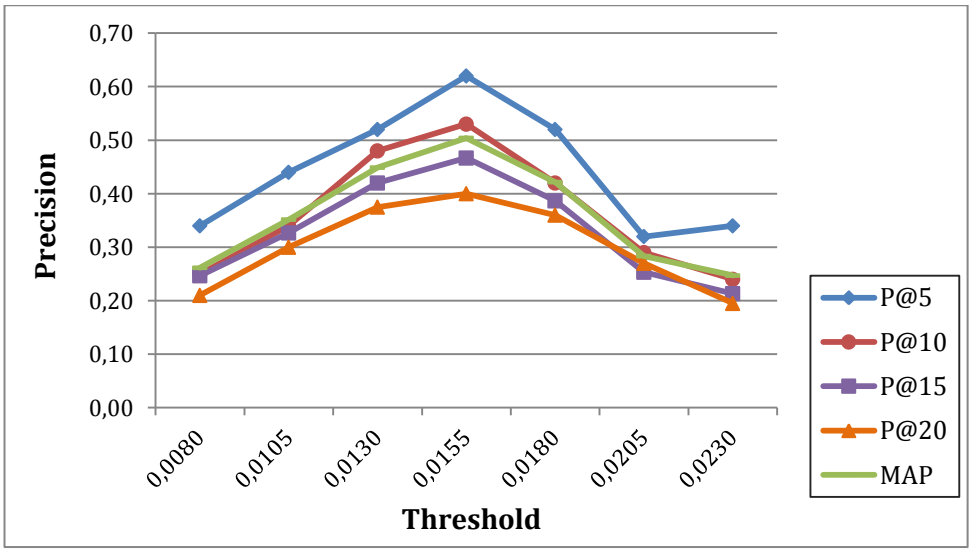

Abbildung 5-23: Abhängigkeit der Retrieval-Qualität vom gewählten Threshold am Beispiel SIFT_SURF kDice

Trotz diesem schnellen Abfall der Precision-Werte bei geringfügiger Änderung des Thresholds gilt als Faustregel: Der Schwellwert sollte bei einem BRIEF-beschriebenen Feature im Intervall [0.200,…,0.250] und bei einer SURF-Deskription im Intervall [0.015,…,0.025] liegen. Dies gilt für aller Features und alle Distanzfunktionen auf der getesteten Bildkollektion, wobei die Anpassung des Schwellwertes nicht in jedem Fall eine Verbesserung der Qualität des Suchergebnisses bedeutet (vgl. Tabelle 7-7). Eine Abhängigkeit des optimalen Schwellwertes von statistischen Maßen, welche über die Verteilung der Ähnlichkeitsbewertungen erhoben werden können, konnte dagegen nicht festgestellt werden. Sowohl der Mittelwert und die Varianz, als auch der Wertebereich und somit der Ähnlichkeitswert des n-ten Treffers im Ranking unterscheiden sich bei verschiedenen Features stark voneinander. Die Hypothese, dass der Threshold aus der Verteilung der Bewertungen abgeleitet werden kann, muss somit verworfen werden.

Als nächstes soll nun untersucht werden, wie die Threshold-Wahl den Zusammenhang der Features beeinflusst. Hierfür wird für die zehn untersuchten Detektor-Deskriptor-Kombinationen, auf welche sich die Wahl des Schwellwertes auswirkt, der Zusammenhang bei verschiedenen Werten betrachtet. Dabei wird neben dem angepassten Threshold eine Abweichung dieses Wertes um 0,025 bei BRIEF bzw. um 0,0025 bei SURF in beide Richtungen untersucht. Der Schwellwert wird hierfür bei allen Features jeweils um den gleichen Wert geändert. Die Abbildung 5-24 vergleicht den ermittelbaren Zusammenhang bei angepasstem, sowie verringertem und erhöhtem Threshold.

Das Ergebnis des Tests ist, dass die Wahl des Thresholds auf der verwendeten Bildmenge einen sehr großen Einfluss auf den Zusammenhang der Features ausübt. Bereits geringe Abweichungen ändern die gefundenen Feature-Gruppen. Der Grund hierfür ist der erwähnte Einfluss des Thresholds auf die Verteilung der Ähnlichkeitsbewertungen. Diese wird bei Modifizierung des Schwellwertes verschoben, wodurch sich ebenfalls die bei der Analyse berechneten Clusterdistanzen verändern. Es ist hierbei zu beobachten, dass sich eine Erhöhung stärker auf den Zusammenhang auswirkt als eine Verringerung. Eine Möglichkeit einer distanzunabhängigen Zusammenhangsbetrachtung ist die Korrelationsanalyse. Da bei der kleinen Testkollektion allerdings die Korrelation der Features zu stark ist und somit die überwiegende Zahl der Verfahren in einer Gruppe liegen, müsste die Untersuchung für eine derartige Aussage mit einer größeren Kollektion wiederholt werden.

Insgesamt übt der Threshold, wie zu erwarten war, einen großen Einfluss auf die Qualität des Suchergebnisses und den Zusammenhang zwischen den Features aus. Bereits geringe Abweichungen führen zu einem schlechteren Retrieval-Ergebnis. Die Resultate dieser Untersuchungen müssen aber noch auf anderen und vor allem größeren Bildsammlungen verifiziert werden. Auf der kleinen Testkollektion, bei der alle Ähnlichkeitsbewertungen stark korrespondieren, genügen bereits geringe Schwankungen der berechneten Clusterdistanz, damit Features in verschiedenen Clustern liegen, welche zuvor einen Cluster teilten. Auch eine höhere Stabilität der Retrieval-Qualität ist bei größeren Bildsammlungen möglich. Die Qualität des Suchergebnisses wird hingegen durch die höhere Anzahl der Bilder im Allgemeinen geringer ausfallen als bei diesem Test. Bereits durch die Mittelung der Precision-Werte der zehn Anfragebilder einer Kategorie ist eine Verschlechterung verglichen mit den Werten eines Anfragebildes, für welches der Threshold erhoben wurde, erkennbar.

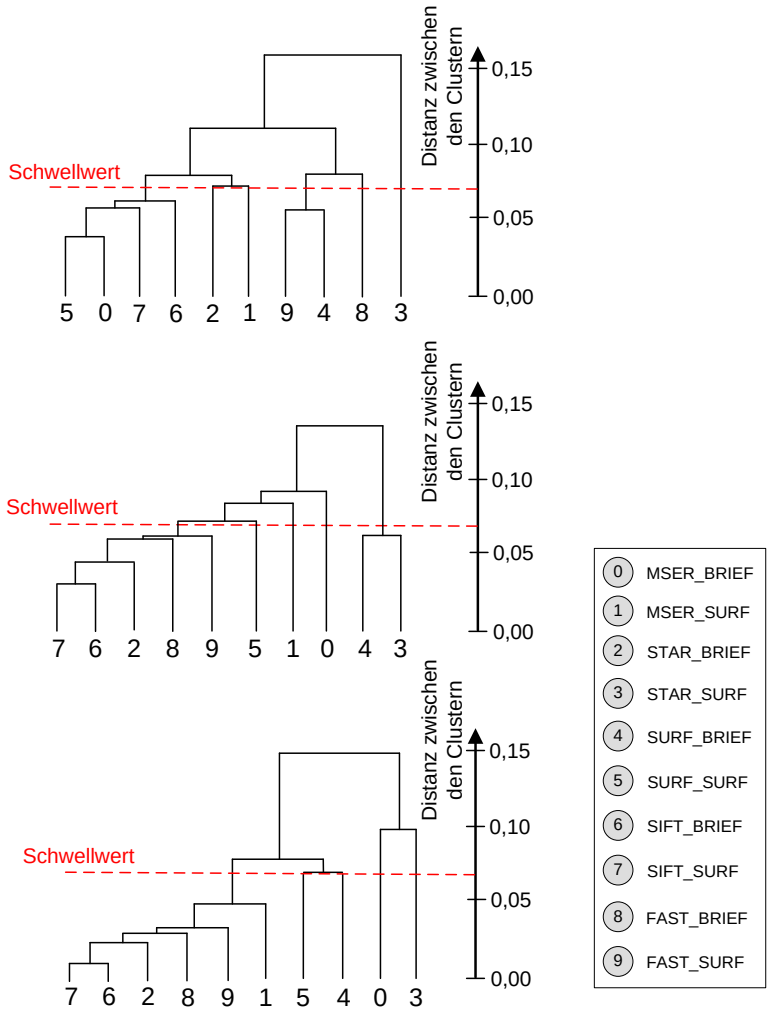

Abbildung 5-24: Änderung des statistisch ermittelbaren Zusammenhangs von lokalen Features bei leicht verringertem (unten), angepasstem (Mitte) und leicht erhöhtem Schwellwert (oben)

5.7 Zusammenfassung

In diesem Kapitel werden sowohl globale, als auch lokale Features aus Pythia nach verschiedenen Gesichtspunkten hin untersucht. Zum einen wird dabei die Retrieval-Qualität der verschiedenen Verfahren evaluiert. Dabei zeigt sich, dass hierbei auf dem verwendeten Datenbestand nur geringe Unterschiede zwischen den einzelnen Extraktionsmethoden zu erkennen sind. Bei der Untersuchung fällt auch auf, dass die Retrieval-Verfahren generell relativ schlecht abschneiden. Es wird hierbei erwähnt, dass verschiedene Systemeinstellungen Einfluss auf das Suchergebnis nehmen können, wodurch die Retrieval-Qualität in einzelnen Fällen gesteigert werden kann. Im weiteren Verlauf des Kapitels wird zu einigen dieser Einstellungsmöglichkeiten Bezug genommen. Dabei wird festgestellt, dass die gewählte Distanzfunktion bei der verwendeten Bildkollektion keinen Einfluss auf die Qualität des Ergebnisses hat. Da allein bei Verwendung einer anderen Bildsammlung die Retrieval-Qualität sich ändern kann, wäre auch eine Einflussnahme der Distanz bei anderen Untersuchungen denkbar. Weiterhin wird der Threshold in Kapitel 5.6 betrachtet. Dieser bestimmt hingegen den Anteil der relevanten Treffer im Suchergebnis und sollte daher stets möglichst gut angepasst werden.

Neben der Retrieval-Qualität ist auch die benötigte Rechenzeit zum Ermitteln der Relevanz Gegenstand er Untersuchung. Dabei stellt sich heraus, dass die meisten lokalen Features sowohl bei Extraktion der Merkmalswerte, als auch bei der anschließenden Distanzberechnung deutlich langsamer als die globalen Varianten sind. Lediglich MSER und STAR lassen sich aufgrund ihrer geringen Zahl an extrahierten Features noch relativ schnell berechnen. Am meisten Zeit benötigt FAST, welches aber auch die größte Zahl von Keypoints extrahiert. Zwischen den betrachteten globalen Features kann kein gravierender Unterschied in der Berechnungszeit festgestellt werden. Bei den lokalen Verfahren besteht dieser hingegen schon. Dabei spielt neben der Wahl des Detektors, welcher die Zahl der ermittelten Keypoints bestimmt, auch der verwendete Deskriptor eine große Rolle. SURF-beschriebene Features benötigten im Test deutlich mehr Zeit bei Extraktion und Distanzberechnung. Neben der verdoppelten Anzahl der Merkmalswerte pro Keypoint bei SURF, wodurch zum Parsen der Dokumente zum Teil fast fünf Mal so viel Zeit benötigt wird, spielt hier auch die Art der Beschreibung eine wichtige Rolle. Wie in Kapitel 3.3 beschrieben erfolgt bei BRIEF eine Deskription mittels Binärzeichenketten. Auf diese kann bei der Distanzberechnung eine Hamming-Distanz angewendet werden, welche schneller zu berechnen ist als die euklidische Distanz bei SURF. Insgesamt nimmt die Zeit zum Lesen und Schreiben der Features auf dem XML-Speicherformat bei fast allen Verfahren den größten Anteil der Gesamtrechenzeit ein.

Des Weiteren wurde in dieser Arbeit der statistische Zusammenhang zwischen den verschiedenen Features untersucht. Bei den Tests kann zwischen einzelnen Features eine zum Teil starke Korrespondenz festgestellt werden, wobei das distanzbasierte Clustering zu größeren Feature-Gruppen tendiert. Der durch beide hier betrachteten Analyseverfahren ermittelte Zusammenhang besteht lediglich unter den im Test verwendeten Bedingungen. Das bedeutet, bei einer anderen Bildsammlung, anderen Anfragebildern oder auf andere Weise geänderten Testbedingungen könnte sich ein anderer statistischer Zusammenhang ergeben und muss daher bei größeren Änderungen am Testsetting erneut berechnet werden. Die Wahl der Distanzfunktion hatte dagegen auf den ermittelbaren Feature-Zusammenhang, wie auch auf die Retrieval-Qualität, keinen Einfluss. Durch den Threshold kann wiederum die Verteilung der Ähnlichkeitsbewertungen beeinflusst werden, welche auch den berechneten Zusammenhang der Features verändert. Allerding korrelieren die ermittelten optimalen Schwellwerte für eine ideale Verteilung und ein sehr gutes Retrieval-Ergebnis nicht, wodurch auch weiterhin eine Kalibrierung der Features für die Analyse notwendig ist. Auf der getesteten Kollektion ist in Hinblick auf das Suchergebnis ein Threshold im Intervall [0.200,…,0.250] für BRIEF-Deskriptoren und bei einer SURF-Beschreibungen im Intervall [0.015,…,0.025] zu wählen. Da sich diese Werte stark unterscheiden, sollten für BRIEF und SURF verschiedene Schwellwerte verwendet werden.

Insgesamt existieren bei dem Retrieval-System Pythia viele Möglichkeiten auf das Suchergebnis Einfluss zu nehmen. Dazu zählen neben den verschiedenen Detektoren, Deskriptoren und Distanzfunktionen auch Suchparameter wie der Threshold. Eine optimale Abstimmung kann, wie in diesem Kapitel gezeigt, die Suche beeinflussen und das Ergebnis so verbessern. Deshalb sollte in vorherigen Tests stets eine Annäherung der Suchparameter erfolgen, um das System möglichst gut an die jeweiligen Gegebenheiten anpassen zu können. Die Kalibrierung des Suchsystems an die Anfrage ist allerding aufwändig und bedarf zum Teil ground truth-Informationen.

6. Ergebnisse und Ausblick

Zum Abschluss soll an dieser Stelle noch ein kurzes Fazit zu den gemachten Untersuchungen gezogen werden. Nach Zusammenfassung der Analyseergebnisse und deren Bedeutung für den weiteren Einsatz von Pythia schließt ein Ausblick auf bevorstehende Entwicklungen dieses Retrieval-Systems die Arbeit ab.

6.1 Ergebnisse

Zu Beginn dieser Abschlussarbeit wird zuerst das Gebiet der multimedialen Suche vorgestellt. Nach einigen Definitionen und grundlegenden Prinzipien des CBIR wird der Schwerpunkt auf die lokalen Detektions- und Deskriptionsalgorithmen gelegt. Dabei zeichnet sich der STAR-Detektor durch seine hohe Genauigkeit und Robustheit bei zudem schneller Berechnung der Eigenschaftswerte aus. Auch bei den im fünften Kapitel durchgeführten Analysen zeigt STAR eine vergleichsweise gute Retrieval-Performanz und benötigt neben MSER für die Feature-Extraktion und die Distanzberechnung bei den lokalen Features die geringste Rechenzeit. Außerdem ließ sich die Ergebnisqualität bei diesem Feature gut über eine Anpassung des Schwellwertes verbessern. Alles in allem schneidet STAR deshalb am besten unter den lokalen Features ab, was auf die geringe Zahl der Features und deren effiziente Berechnung bei gleichzeitiger hoher Deskriptivität zurückzuführen ist. Anders sieht es bei FAST aus. In der Literatur (vgl. Kapitel 3) gilt es als sehr schnell zu berechnen, was zu Lasten der Retrieval-Qualität geht. Da im Test bei FAST allerdings sehr viele Keypoints detektiert werden, benötigt das Verfahren bei allen Berechnungen zum Teil erheblich mehr Zeit. Bei der Qualität des Suchergebnisses liegt FAST hingegen im mittleren bis vorderen Bereich. Allerdings kann diese bei den Tests nicht signifikant über eine Schwellwertanpassung verbessert werden. Vor allem durch die langen Schreib- und Lesezeiten der vielen extrahierten Merkmalswerte sollte der Einsatz dieses Features gegenwärtig überdacht werden.

Bei den Deskriptoren kann hingegen keine definitive Antwort auf die Frage, welcher Deskriptor für den Einsatz besser geeignet ist, gegeben werden. Wie bei vielen in dieser Arbeit untersuchten Aspekten gilt, dass es von der jeweiligen Anwendung abhängt. In Kapitel 5.2 zeigen die BRIEF-basierten Beschreibungen trotz weniger Merkmalswerten die besseren Ergebnisse. In Kapitel 5.6 stellte sich dagegen heraus, dass dies stark vom verwendeten Schwellwert abhängt. Zurzeit wird lediglich ein Threshold für alle Matching-Distanzen lokaler Features verwendet. Wenn dieser für die SURF-beschriebenen Features angepasst wird können diese ähnlich gute Resultate erreichen. Bei den Betrachtungen wird ebenfalls herausgestellt, dass sich der optimale Schwellwert je nach verwendetem Deskriptor stark unterscheidet. Aus diesem Grund sollten verschiedene Thresholds beim Einsatz mehrerer Deskriptoren verwendet werden.

Die verwendete Distanzfunktion hatte hingegen bei den Betrachtungen weder Einfluss auf die Qualität des Ergebnisses, noch auf die Rechenzeit, den Zusammenhang der Features oder den idealen Threshold. Ein Grund hierfür könnte bei den lokalen Verfahren die hohe Zahl der extrahierten Features liefern. Verglichen mit der Gesamtanzahl der Merkmalswerte ist die Zahl derer, die eine semantisch bedeutsame Information über das dargestellte Bildobjekt enthalten, sehr gering. Diese gehen bei der Distanzberechnung in der Menge unter. Würden durch geeignete Verfahren lediglich wenige Keypoints des Hauptbildobjektes berechnet werden, könnte so die Retrieval-Qualität verbessert werden und der Einfluss der Distanzwahl zunehmen. Zudem würde die Re-

chenzeit beim Matching der Features stark abnehmen. Diese Hypothese muss allerding in späteren Untersuchungen noch überprüft werden und ist nicht Gegenstand dieser Arbeit.

Für die meisten hier getätigten Untersuchungen gilt weiterhin, dass diese lediglich auf kleinen Stichproben erfolgten. Um eine zuverlässige Aussage über die verschiedenen Aspekte zu erhalten, sollten daher die durchgeführten Tests in größeren Erhebungen und auf anderen Bildsammlungen wiederholt werden, da besonders die verwendete Bildkollektion die Evaluierung stark beeinflusst. Deshalb kann nie eine allgemeingültige Aussage darüber getroffen werden, welches Verfahren für die Bildsuche am besten geeignet ist. Dies hängt stets von den jeweiligen Untersuchungsbedingungen ab. Darin liegt auch ein großer Vorteil des Pythia-Systems. Durch die vielfältigen Anpassungsmöglichkeiten des Suchsystems an die Anwendungen ist stets eine optimale Suche möglich. Wie sich während der durchgeführten Tests allerdings zeigte, ist eine ideale Anpassung oft sehr zeitaufwändig.

Des Weiteren werden alle Features während dieser Arbeit stets separat untersucht. Durch eine Kombination verschiedener Verfahren lässt sich die Retrieval-Qualität mitunter auch steigern. Da die Berechnung aller Features hingegen zu aufwändig und meistens unnötig ist, wird während der Arbeit auch evaluiert, inwieweit die einzelnen Methoden zusammenhängen. Zu diesem Zweck werden zwei Verfahren zur Ermittlung des statistisch messbaren Zusammenhangs der einzelnen Features vorgestellt. Dieser ist hingegen wieder von den Testbedingungen abhängig und muss ebenfalls stichprobenartig für jede Bildkollektion bzw. Anfrage ermittelt werden. Unter den Testbedingungen konnte ein starker Zusammenhang zwischen einzelnen Verfahren bestimmt werden. Ob mehrere Features eine allgemeingültige Beziehung aufweisen, muss in größeren Studien mit verschiedenen Bildsammlungen unter Variation der Suchbedingungen noch evaluiert werden. Vom jetzigen Standpunkt ist dies allerdings unwahrscheinlich, da bereits geringe Änderungen der Testbedingungen einen anderen messbaren Zusammenhang ergeben. Im Fall des distanzbasierten Clustering ist die berechnete Clusterdistanz zwischen zwei Features beispielsweise von der Datenverteilung abhängig, welche wiederum vom verwendeten Threshold beeinflusst wird. Um die Verteilung anzunähern wird in Kapitel 5 ein Kalibrierungsverfahren vorgestellt. Weiterhin bleibt in größeren Tests noch zu überprüfen, ob unter Beachtung des ermittelten Zusammenhangs bei der Feature-Auswahl bessere Ergebnisse erzielt werden können oder bei gleichbleibender Qualität die Suchzeit verkürzt werden kann.

Insgesamt wird in dieser Arbeit die Vielschichtigkeit eines Retrieval-Systems demonstriert. Durch Kalibrierung des IR-Systems an die Anforderungen lassen sich in den meisten Fällen bessere Ergebnisse erzielen. Hierfür werden im Laufe der getätigten Untersuchungen einige Möglichkeiten aufgezeigt. Für alle hier diskutierten Analysen müssen jedoch noch größere Datensätze herangezogen werden, um eine zuverlässige Aussage über das Systemverhalten in realen Laufzeitumgebungen zu erhalten. Neben der Suchqualität spiel aber auch die Rechenzeit zur Ermittlung des Ergebnisrankings eine große Rolle. Im folgenden Ausblick werden hierzu einige Möglichkeiten, an welchen derzeit am Lehrstuhl DBIS gearbeitet wird, aufgezeigt.

6.2 Ausblick

Zur Steigerung der Qualität wurden in dieser Arbeit zahlreiche mögliche Einstellungen diskutiert. Dabei existiert keine Systemeinstellung, die unter allen Anwendungsbedingungen sehr gute Ergebnisse liefert. Vielmehr muss diese durch vorherige Tests bestimmt werden. Nur bei einem optimal eingestellten System sind auch gute Suchergebnisse zu erwarten. Neben der schwankenden Retrieval-Qualität ist die Rechenzeit, welche für die Ermittlung der Features und Distanzen notwendig ist, eines der größten momentan bestehenden Probleme bei der Arbeit mit CBIR-Verfahren. Vor allem die lokalen Features, welche potentiell bessere Ergebnisse liefern (vgl. Kapitel 5.6), brauchen in den durchgeführten Untersuchungen erheblich mehr Zeit für die Berechnungen. Wie bereits während dieser Arbeit anklang, gibt es verschiedene Möglichkeiten dieses Zeitproblem zu lösen oder zumindest zu minimieren. Zum einen haben sämtliche Schreib- und Lesevorgänger einen mitunter erheblichen Anteil an der Gesamtrechenzeit. Da die Daten zur Speicherung der Features bereits bei relativ kleinen Bildkollektionen zu groß werden, um permanent im Arbeitsspeicher gehalten zu werden, wird derzeit am Lehrstuhl Datenbank- und Informationssysteme an effizienteren Speichermöglichkeiten gearbeitet. Die Features werden zukünftig wahrscheinlich nicht mehr wie jetzt in Form von XML-Dokumenten abgelegt, sondern in einer eigens konzipierten Datenbank gespeichert. Dadurch wird viel Zeit zum Parsen von Dokumenten eingespart und der Datenoverhead zur Speicherung der Features reduziert.

Des Weiteren hat sich in dieser Arbeit herausgestellt, dass die Rechenzeit stark von der Anzahl der extrahierten Merkmalswerte abhängt. Eine Möglichkeit um die benötigte Zeit zu reduzieren, ist es die Zahl der Features zu minimieren. Neben verschiedenen Verfahren zur Reduktion redundanter Dimensionen, welche auf den beschreibenden Daten operieren und diese gruppieren, existieren auch Methoden, die auf dem Feature-Raum arbeiten. Die Idee dabei ist, häufig gemeinsam auftretende Features zu lokalisieren und durch Clustering-Methoden zusammenzufassen. Dadurch erfolgt eine lokale Gruppierung des Feature-Raums in verschiedene Cluster, wobei jede Gruppe beispielsweise durch einen Index beschrieben werden kann. Ein Verfahren, welches sich für derartige Analysen eignet, ist das *bag of words*-Model. Dieses stammt ursprünglich aus dem Text-Retrieval. Jeder Text wird dabei als „Sack voller Wörter“ betrachtet, da es egal ist an welcher Stelle die einzelnen Wörter in dieser Wortmenge stehen. Mehrere Wörter sind dabei immer einem Themengebiet zugeordnet. Durch Analyse welche Wortmenge ein Dokument enthält kann somit bestimmt werden, über welche Themen ein Text berichtet. Diese Analogie ist nach neusten Erkenntnissen im Bereich des CBIR auch auf den Bildraum übertragbar, wobei „visuelle Wörter“ genutzt werden. Ähnlich den in dieser Arbeit betrachteten Verfahren wird das Bild zuerst nach bedeutsamen Bildregionen durchsucht und ein hochdimensionaler Feature-Vektor für jede Region berechnet. Dieser Deskriptor wird anschließend quantisiert oder mithilfe der visuellen Wörter geclustert, sodass jeder markanten Region ein visuelles Indexwort, welches zu ihr am ähnlichsten ist, zugeordnet wird. Das Bild kann nun durch einen „Sack virtueller Wörter“ repräsentiert werden. Die Menge der identifizierten Wörter bestimmt den Index eines Bildes, welcher beispielsweise über ein Histogramm dargestellt und für sehr effiziente Suchanfragen genutzt werden kann. Am Lehrstuhl DBIS wird bereits an Distanzmaßen gearbeitet, welche auf diesen Techniken basieren. [Phi07S. 1]

7. Anhang

A. Aufgabenstellung

”Diskriminanzanalyse und Evaluierung von Detektor-Deskriptor-Kombinationen in einem Multimedia-Retrieval-System“

Discriminant analysis and evaluation of detector-descriptor combinations in a multimedia retrieval system

Datum der Themenausgabe: 01.06.2011

Multimedia-Retrieval beschäftigt sich mit der Suche, Erschließung und dem Auffinden von Informationen in einer Vielzahl von Formen, wie atomaren Medienobjekten (Bild, Text, Ton) und multimedialen Medienobjekten (Video, Animationen, Rich-Media-Dokumente). Die Suche in diesen Datenbeständen geschieht auf Grundlage von Features, welche die zu durchsuchenden Medienobjekte charakterisieren. Dabei werden lokale und globale Features unterschieden. Globale Features beziehen sich auf das Medienobjekt als Ganzes, während lokale Features Teilbereiche zur Beschreibung der Multimedia-Objekte nutzen. Bei der Extraktion von Features wird weiterhin zwischen Detektion und Deskription unterschieden: Detektoren ermitteln Feature-charakterisierende Werte, während Deskriptoren die Detektions-Ergebnisse in ein vergleichbares Format überführen. OpenCV, als freie Sammlung von Algorithmen des computergestützten Sehens, widmet sich der Extraktion von Features für visuelle Medienobjekte unter Wahrung der Trennung von Detektion und Deskription. Dabei wird ein Zwischenfromat gewählt, das verschiedene Kombinationen von Detektoren und Deskriptoren zulässt. Ziel dieser Arbeit ist es, die in OpenCV vorhandenen sowie ggf. weitere Detektoren und Deskriptoren zu untersuchen. Dabei sollen sie jeweils einer Diskriminanzanalyse unterzogen werden. Unter den Deskriptoren werden besonders schnelle Verfahren ohne Informationsverlust gesucht. Bei den Detektoren sind möglichst perfomante Algorithmen mit gering korrelierenden Ergebnismengen zu finden. Weiterhin steht die Kombinierbarkeit der Detektoren und Deskiptoren aus OpenCV im Mittelpunkt. Das Ergebnis der Untersuchung soll eine Matrix sein, die Aussagen über Möglichkeit und Qualität von Kombinationen zulässt. Die Ergebnisse sollen anhand aktueller IR-Maße evaluiert werden. Die Verfahren sind in ein bestehendes System des Lehrstuhls DBIS zu integrieren.

Die Arbeit kann dabei in folgende Schritte unterteilt werden:

1. Einarbeitung in Multimedia-Retrieval, insbesondere Detektion und Deskription bei lokalen Verfahren
2. Literaturrecherche zu den Detektions- und Deskriptions-Verfahren aus OpenCV
3. Analyse von Evaluierungsverfahren des Information Retrieval sowie Diskriminanzanalyse-Verfahren
4. Einarbeitung in das bestehende Sotware-System des Lehrstuhls DBIS sowie Qt und OpenCV
5. Einbindung noch fehlender Verfahren in das Software-System
6. Analyse und Evaluierung: Kombinierbarkeit, Geschwindigkeit, Korrelation und Evaluation mittels IR-Maßen
7. Ggf. Untersuchung des Einflusses von Parameter-Variationen auf die Ergebnisse

B. Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Masterarbeit selbständig und ohne unerlaubte Hilfe angefertigt habe, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Weiterhin habe ich die Masterarbeit nicht bereits in derselben oder einer ähnlichen Fassung an einer anderen Fakultät oder einem anderen Fachbereich zur Erlangung eines akademischen Grades eingereicht.

Cottbus, den 28. November 2011

Dominik Müller

C. Ergänzende Tabellen

Features	MAP	P@5	P@10	P@15	P@20	P@30
AUTOCOLORCORRELOGRAM	0,14	0,28	0,17	0,13	0,11	0,09
BIC	0,18	0,31	0,21	0,17	0,15	0,12
CEDD	0,17	0,30	0,20	0,16	0,14	0,11
COLORHISTBORDER	0,17	0,31	0,20	0,16	0,14	0,11
COLORHISTCENTER	0,16	0,30	0,19	0,15	0,12	0,10
COLORHISTOGRAM	0,09	0,28	0,18	0,14	0,12	0,10
COLORLAYOUT	0,21	0,36	0,24	0,19	0,17	0,14
EDGEHISTOGRAM	0,27	0,42	0,31	0,26	0,23	0,20
FCTH	0,16	0,29	0,18	0,14	0,12	0,10
GABOR	0,04	0,04	0,04	0,04	0,04	0,03
REGIONSHAPE	0,14	0,27	0,16	0,12	0,10	0,08
SCALABLECOLOR	0,11	0,23	0,13	0,10	0,08	0,07
TAMURA	0,19	0,33	0,22	0,18	0,16	0,14
FAST_BRIEF	0,19	0,33	0,22	0,18	0,15	0,13
FAST_SURF	0,14	0,26	0,16	0,13	0,11	0,09
MSER_BRIEF	0,16	0,30	0,19	0,15	0,13	0,10
MSER_SURF	0,14	0,27	0,16	0,13	0,11	0,09
SIFT_BRIEF	0,13	0,26	0,15	0,11	0,09	0,07
SIFT_SURF	0,10	0,22	0,12	0,09	0,07	0,06
STAR_BRIEF	0,19	0,34	0,23	0,18	0,16	0,13
STAR_SURF	0,15	0,28	0,18	0,14	0,12	0,10
SURF_BRIEF	0,19	0,33	0,23	0,18	0,16	0,13
SURF_SURF	0,19	0,32	0,22	0,17	0,15	0,13

Tabelle 7-1: Retrieval-Kennzahlen zu verschieden Features in Pythia

Feature	Distanzfunktion	P@5	P@10	MAP
AUTOCOLORCORRELOGRAM	kManhattan	0,485	0,348	0,417
	kEuclidean	0,485	0,348	0,417
	kCosineCoeff	0,485	0,348	0,417
	kJSD	0,485	0,348	0,417
	kTanimoto	0,485	0,348	0,417
BIC	kManhattan | kEMDSerratosa	0,513	0,366	0,439
	kManhattan | kEMD	0,513	0,366	0,439
	kManhattan	0,513	0,366	0,439
	kEuclidean	0,513	0,366	0,439
CEDD	kTanimoto	0,515	0,369	0,442
COLORHISTBORDER	kManhattan	0,493	0,351	0,422
	kEuclidean	0,493	0,351	0,422
	kManhattan | kEMD	0,493	0,351	0,422
	kManhattan | kEMDSerratosa	0,493	0,351	0,422
COLORHISTCENTER	kManhattan	0,473	0,347	0,410
	kEuclidean	0,473	0,347	0,410
	kManhattan | kEMD	0,473	0,347	0,410
	kManhattan | kEMDSerratosa	0,473	0,347	0,410

COLORHISTOGRAM	kManhattan	0,442	0,334	0,388
	kEuclidean	0,442	0,334	0,388
	kJSD	0,442	0,334	0,388
	kTanimoto	0,442	0,334	0,388
	kEMD | kManhattan	0,442	0,334	0,388
	kEMD | kJSD	0,442	0,334	0,388
	kEMDSerratosa | kManhattan	0,442	0,334	0,388
COLORLAYOUT	kEuclideanWeighted	0,505	0,385	0,445
EDGEHISTOGRAM	kManhattanEdge HistogramWeighted	0,578	0,434	0,506
FCTH	kTanimoto	0,501	0,338	0,420
GABOR	kEuclideanGabor	0,358	0,262	0,310
REGIONSHAPE	kManhattan	0,370	0,260	0,315
SCALABLECOLOR	kManhattan	0,388	0,290	0,339
	kEMD | kManhattan	0,388	0,290	0,339
	kEMDSerratosa | kManhattan	0,388	0,290	0,339
TAMURA	kEuclidean	0,485	0,360	0,422
	kManhattan | kEMD	0,485	0,360	0,422
	kManhattan | kEMDSerratosa	0,485	0,360	0,422

Tabelle 7-2: Retrieval-Kennzahlen globaler Features in Pythia unter Verwendung verschiedener Distanzfunktionen

Feature	Distanzfunktion	P@5	P@10	MAP
FAST_BRIEF	kHamming | kJaccardIndex	0,416	0,305	0,361
	kHamming | kMaxMinimum	0,416	0,305	0,361
	kHamming | kManhattanMinimum	0,416	0,305	0,361
	kHamming | kMinOverlap	0,416	0,305	0,361
	kHamming | kMaxOverlap	0,416	0,305	0,361
	kHamming | kDice	0,416	0,305	0,361
	kHamming | kCosineSet	0,416	0,305	0,361
	kHamming | kEuclideanMinimum	0,416	0,305	0,361
	kHamming | kMinMinimum	0,416	0,305	0,361
FAST_SURF	kEuclidean | kJaccardIndex	0,418	0,302	0,360
	kEuclidean | kMaxMinimum	0,418	0,302	0,360
	kEuclidean | kManhattanMinimum	0,418	0,302	0,360
	kEuclidean | kMinOverlap	0,418	0,302	0,360
	kEuclidean | kMaxOverlap	0,418	0,302	0,360
	kEuclidean | kDice	0,418	0,302	0,360
	kEuclidean | kCosineSet	0,418	0,302	0,360
	kEuclidean | kEuclideanMinimum	0,418	0,302	0,360
	kEuclidean | kMinMinimum	0,418	0,302	0,360
MSER_BRIEF	kHamming | kJaccardIndex	0,366	0,264	0,315
	kHamming | kMaxMinimum	0,366	0,264	0,315
	kHamming | kManhattanMinimum	0,366	0,264	0,315
	kHamming | kMinOverlap	0,366	0,264	0,315
	kHamming | kMaxOverlap	0,364	0,263	0,313
	kHamming | kDice	0,366	0,264	0,315

	kHamming | kCosineSet	0,366	0,264	0,315
	kHamming | kEuclideanMinimum	0,360	0,260	0,310
	kHamming | kMinMinimum	0,366	0,264	0,315
MSER_SURF	kEuclidean | kJaccardIndex	0,341	0,226	0,284
	kEuclidean | kMaxMinimum	0,341	0,226	0,284
	kEuclidean | kManhattanMinimum	0,341	0,226	0,284
	kEuclidean | kMinOverlap	0,341	0,226	0,284
	kEuclidean | kMaxOverlap	0,341	0,226	0,284
	kEuclidean | kDice	0,341	0,226	0,284
	kEuclidean | kCosineSet	0,341	0,226	0,284
	kEuclidean | kEuclideanMinimum	0,341	0,226	0,284
	kEuclidean | kMinMinimum	0,341	0,226	0,284
SIFT_BRIEF	kHamming | kJaccardIndex	0,321	0,217	0,269
	kHamming | kMaxMinimum	0,327	0,228	0,278
	kHamming | kManhattanMinimum	0,327	0,228	0,278
	kHamming | kMinOverlap	0,327	0,228	0,278
	kHamming | kMaxOverlap	0,327	0,228	0,278
	kHamming | kDice	0,327	0,228	0,278
	kHamming | kCosineSet	0,327	0,228	0,278
	kHamming | kEuclideanMinimum	0,327	0,228	0,278
	kHamming | kMinMinimum	0,327	0,228	0,278
SIFT_SURF	kEuclidean | kJaccardIndex	0,303	0,220	0,262
	kEuclidean | kMaxMinimum	0,303	0,220	0,262
	kEuclidean | kManhattanMinimum	0,303	0,220	0,262
	kEuclidean | kMinOverlap	0,303	0,220	0,262
	kEuclidean | kMaxOverlap	0,303	0,220	0,262
	kEuclidean | kDice	0,303	0,220	0,262
	kEuclidean | kCosineSet	0,303	0,220	0,262
	kEuclidean | kEuclideanMinimum	0,303	0,220	0,262
	kEuclidean | kMinMinimum	0,303	0,220	0,262
STAR_BRIEF	kHamming | kJaccardIndex	0,346	0,258	0,302
	kHamming | kMaxMinimum	0,346	0,258	0,302
	kHamming | kManhattanMinimum	0,346	0,258	0,302
	kHamming | kMinOverlap	0,346	0,258	0,302
	kHamming | kMaxOverlap	0,346	0,258	0,302
	kHamming | kDice	0,346	0,258	0,302
	kHamming | kCosineSet	0,346	0,258	0,302
	kHamming | kEuclideanMinimum	0,346	0,258	0,302
	kHamming | kMinMinimum	0,346	0,258	0,302
STAR_SURF	kEuclidean | kJaccardIndex	0,322	0,220	0,271
	kEuclidean | kMaxMinimum	0,322	0,220	0,271
	kEuclidean | kManhattanMinimum	0,322	0,220	0,271
	kEuclidean | kMinOverlap	0,322	0,220	0,271
	kEuclidean | kMaxOverlap	0,322	0,220	0,271
	kEuclidean | kDice	0,322	0,220	0,271
	kEuclidean | kCosineSet	0,322	0,220	0,271
	kEuclidean | kEuclideanMinimum	0,322	0,220	0,271
	kEuclidean | kMinMinimum	0,322	0,220	0,271
SURF_BRIEF	kHamming | kJaccardIndex	0,432	0,316	0,374
	kHamming | kMaxMinimum	0,432	0,316	0,374
	kHamming | kManhattanMinimum	0,432	0,316	0,374

	kHamming | kMinOverlap	0,432	0,316	0,374
	kHamming | kMaxOverlap	0,432	0,316	0,374
	kHamming | kDice	0,432	0,316	0,374
	kHamming | kCosineSet	0,432	0,316	0,374
	kHamming | kEuclideanMinimum	0,432	0,316	0,374
	kHamming | kMinMinimum	0,432	0,316	0,374
SURF_SURF	kEuclidean | kJaccardIndex	0,446	0,312	0,379
	kEuclidean | kMaxMinimum	0,446	0,312	0,379
	kEuclidean | kManhattanMinimum	0,446	0,312	0,379
	kEuclidean | kMinOverlap	0,446	0,312	0,379
	kEuclidean | kMaxOverlap	0,446	0,312	0,379
	kEuclidean | kDice	0,446	0,312	0,379
	kEuclidean | kCosineSet	0,446	0,312	0,379
	kEuclidean | kEuclideanMinimum	0,446	0,312	0,379
	kEuclidean | kMinMinimum	0,446	0,312	0,379

Tabelle 7-3: Retrieval-Kennzahlen lokaler Features in Pythia unter Verwendung verschiedener Distanzfunktionen

Feature	Extraktion (in s)	Gesamtzeit (in s)	Anzahl Features	Anzahl Dimensionen pro Feature
AUTOCOLORCORRELOGRAM	0,312	0,452	64	4
BIC	0,218	0,608	2	64
CEDD	0,234	0,717	6	24
COLORHISTBORDER	0,889	1,155	12	32
COLORHISTCENTER	0,453	0,562	3	64
COLORHISTOGRAM	0,218	0,327	1	512
COLORLAYOUT	0,219	0,312	3	40
EDGEHISTOGRAM	0,266	0,609	1	80
FCTH	0,296	0,795	8	24
GABOR	0,296	0,749	1	60
REGIONSHAPE	0,296	0,359	1	35
SCALABLECOLOR	0,218	0,359	1	64
TAMURA	0,234	0,312	3	16
FAST_BRIEF	0,25	4,056	1415	32
FAST_SURF	1,607	28,595	1863	64
MSER_BRIEF	0,811	1,046	48	32
MSER_SURF	0,952	1,95	66	64
SIFT_BRIEF	1,607	2,886	441	32
SIFT_SURF	2,559	9,688	615	64
STAR_BRIEF	0,234	0,483	49	32
STAR_SURF	0,234	1,014	49	64
SURF_BRIEF	0,749	1,794	334	32
SURF_SURF	1,482	5,897	382	64

Tabelle 7-4: Übersicht der Rechenzeiten für die Feature-Extraktion und der Anzahl der durchschnittlich extrahierten Features

Feature	Distanzfunktion	SD	Distanz-berechnung (in ms)	Gesamtzeit (in ms)
AUTOCOLORCORRELOGRAM	kManhattan	x	3,3	61,8
	kEuclidean		3,6	62,1
	kCosineCoeff		3,9	63,5
	kJSD		3,0	63,0
	kTanimoto		4,3	62,3
BIC	kManhattan | kEMDSerratosa		3,9	35,2
	kManhattan | kEMD		2,4	33,6
	kManhattan		3,0	33,8
	kEuclidean	x	3,4	35,0
CEDD	kTanimoto	x	3,4	40,7
COLORHISTBORDER	kManhattan		5,1	62,3
	kEuclidean	x	4,7	62,3
	kManhattan | kEMD		4,4	61,5
	kManhattan | kEMDSerratosa		4,7	61,9
COLORHISTCENTER	kManhattan		3,4	43,2
	kEuclidean	x	4,1	40,7
	kManhattan | kEMD		3,9	42,0
	kManhattan | kEMDSerratosa		3,9	40,6
COLORHISTOGRAM	kManhattan	x	4,5	46,4
	kEuclidean		4,2	45,6
	kJSD		3,6	45,7
	kTanimoto		4,6	46,1
	kEMD | kManhattan		4,7	46,3
	kEMD | kJSD		4,3	45,9
	kEMDSerratosa | kManhattan		3,6	45,4
COLORLAYOUT	kEuclideanWeighted	x	3,2	36,7
EDGEHISTOGRAM	kManhattanEdge HistogramWeighted	x	3,3	32,2
FCTH	kTanimoto	x	3,9	44,4
GABOR	kEuclideanGabor	X	3,3	34,1
REGIONSHAPE	kManhattan	x	3,5	30,3
SCALABLECOLOR	kManhattan	x	2,7	33,1
	kEMD | kManhattan		3,0	32,5
	kEMDSerratosa | kManhattan		3,9	32,5
TAMURA	kEuclidean	x	2,7	33,0
	kManhattan | kEMD		3,3	33,3
	kManhattan | kEMDSerratosa		3,2	32,1

Tabelle 7-5: Übersicht der Rechenzeiten für die Distanzberechnung im Verhältnis an der Gesamtrechenzeit (inkl. Lesen der XML-Dateien) bei globalen Features

Feature	Distanzfunktion	SD	Distanz-berechnung (in s)	Gesamtzeit (in s)
FAST_BRIEF	kHamming | kJaccardIndex		1,152	2,775
	kHamming | kMaxMinimum		1,154	2,777
	kHamming | kManhattanMinimum		1,155	2,773
	kHamming | kMinOverlap	x	1,153	2,776
	kHamming | kMaxOverlap		1,150	2,769
	kHamming | kDice		1,153	2,771
	kHamming | kCosineSet		1,151	2,768
	kHamming | kEuclideanMinimum		1,151	2,794
	kHamming | kMinMinimum		1,147	2,789
FAST_SURF	kEuclidean | kJaccardIndex		1,782	9,210
	kEuclidean | kMaxMinimum	x	1,783	9,145
	kEuclidean | kManhattanMinimum		1,783	9,149
	kEuclidean | kMinOverlap		1,782	9,162
	kEuclidean | kMaxOverlap		1,784	9,137
	kEuclidean | kDice		1,782	9,191
	kEuclidean | kCosineSet		1,784	9,170
	kEuclidean | kEuclideanMinimum		1,782	9,130
	kEuclidean | kMinMinimum		1,781	9,101
MSER_BRIEF	kHamming | kJaccardIndex		0,374	0,461
	kHamming | kMaxMinimum		0,375	0,462
	kHamming | kManhattanMinimum		0,375	0,462
	kHamming | kMinOverlap	x	0,377	0,462
	kHamming | kMaxOverlap		0,376	0,460
	kHamming | kDice		0,376	0,460
	kHamming | kCosineSet		0,376	0,462
	kHamming | kEuclideanMinimum		0,376	0,460
	kHamming | kMinMinimum		0,377	0,463
MSER_SURF	kEuclidean | kJaccardIndex		0,380	0,669
	kEuclidean | kMaxMinimum	x	0,378	0,667
	kEuclidean | kManhattanMinimum		0,379	0,669
	kEuclidean | kMinOverlap		0,380	0,667
	kEuclidean | kMaxOverlap		0,380	0,667
	kEuclidean | kDice		0,380	0,667
	kEuclidean | kCosineSet		0,382	0,668
	kEuclidean | kEuclideanMinimum		0,380	0,669
	kEuclidean | kMinMinimum		0,380	0,668
SIFT_BRIEF	kHamming | kJaccardIndex		0,455	0,984
	kHamming | kMaxMinimum		0,453	0,980
	kHamming | kManhattanMinimum		0,468	0,994
	kHamming | kMinOverlap	x	0,470	0,997
	kHamming | kMaxOverlap		0,472	0,997
	kHamming | kDice		0,467	0,996
	kHamming | kCosineSet		0,471	0,998
	kHamming | kEuclideanMinimum		0,471	0,995
	kHamming | kMinMinimum		0,471	0,996
SIFT_SURF	kEuclidean | kJaccardIndex		0,579	3,011
	kEuclidean | kMaxMinimum	x	0,582	3,014
	kEuclidean | kManhattanMinimum		0,579	3,009
	kEuclidean | kMinOverlap		0,580	3,013
	kEuclidean | kMaxOverlap		0,581	3,016

	kEuclidean | kDice		0,580	3,011
	kEuclidean | kCosineSet		0,579	3,013
	kEuclidean | kEuclideanMinimum		0,580	3,013
	kEuclidean | kMinMinimum		0,579	3,030
STAR_BRIEF	kHamming | kJaccardIndex		0,374	0,460
	kHamming | kMaxMinimum		0,376	0,462
	kHamming | kManhattanMinimum		0,378	0,462
	kHamming | kMinOverlap	x	0,378	0,461
	kHamming | kMaxOverlap		0,379	0,462
	kHamming | kDice		0,378	0,462
	kHamming | kCosineSet		0,376	0,460
	kHamming | kEuclideanMinimum		0,378	0,461
	kHamming | kMinMinimum		0,378	0,462
STAR_SURF	kEuclidean | kJaccardIndex		0,377	0,601
	kEuclidean | kMaxMinimum	x	0,376	0,597
	kEuclidean | kManhattanMinimum		0,376	0,598
	kEuclidean | kMinOverlap		0,375	0,598
	kEuclidean | kMaxOverlap		0,376	0,598
	kEuclidean | kDice		0,377	0,599
	kEuclidean | kCosineSet		0,376	0,601
	kEuclidean | kEuclideanMinimum		0,374	0,598
	kEuclidean | kMinMinimum		0,376	0,600
SURF_BRIEF	kHamming | kJaccardIndex		0,437	0,827
	kHamming | kMaxMinimum		0,437	0,835
	kHamming | kManhattanMinimum		0,438	0,834
	kHamming | kMinOverlap	x	0,437	0,838
	kHamming | kMaxOverlap		0,439	0,835
	kHamming | kDice		0,438	0,836
	kHamming | kCosineSet		0,437	0,836
	kHamming | kEuclideanMinimum		0,438	0,835
	kHamming | kMinMinimum		0,438	0,836
SURF_SURF	kEuclidean | kJaccardIndex		0,462	1,991
	kEuclidean | kMaxMinimum	x	0,461	1,982
	kEuclidean | kManhattanMinimum		0,460	1,977
	kEuclidean | kMinOverlap		0,461	1,978
	kEuclidean | kMaxOverlap		0,461	1,979
	kEuclidean | kDice		0,463	1,989
	kEuclidean | kCosineSet		0,470	2,034
	kEuclidean | kEuclideanMinimum		0,461	1,987
	kEuclidean | kMinMinimum		0,460	1,987

Tabelle 7-6: Übersicht der Rechenzeiten für die Distanzberechnung im Verhältnis an der Gesamtrechenzeit (inkl. Lesen der XML-Dateien) bei lokalen Features

	FAST									
	BRIEF					**SURF**				
Cosine Set	**0.2000**	**0.2250**	**0.2500**	**0.2750**	**0.3000**	**0.0140**	**0.0165**	**0.0190**	**0.0215**	**0.0240**
	0,22	0,22	0,22	0,20	0,10	0,24	0,24	0,82	0,24	0,34
	0,13	0,15	0,16	0,13	0,11	0,16	0,16	0,58	0,22	0,25
	0,09	0,12	0,12	0,12	0,10	0,13	0,15	0,46	0,21	0,19
	0,09	0,13	0,12	0,12	0,11	0,13	0,12	0,39	0,18	0,18
	0,13	0,15	0,15	0,14	0,10	0,16	0,17	0,56	0,21	0,24

Dice	**0.1850**	**0.2100**	**0.2350**	**0.2600**	**0.2850**	**0.0225**	**0.0250**	**0.0275**	**0.0300**	**0.0325**
	0,20	0,22	0,22	0,20	0,20	0,26	0,32	0,34	0,26	0,04
	0,11	0,14	0,15	0,15	0,13	0,24	0,24	0,22	0,21	0,10
	0,09	0,12	0,13	0,12	0,13	0,21	0,19	0,17	0,15	0,15
	0,09	0,10	0,12	0,12	0,11	0,18	0,16	0,14	0,12	0,15
	0,12	0,15	0,15	0,15	0,14	0,22	0,23	0,22	0,19	0,11
Jaccard	**0.1900**	**0.2150**	**0.2400**	**0.2650**	**0.2900**	**0.0225**	**0.0250**	**0.0275**	**0.0300**	**0.0325**
	0,20	0,20	0,22	0,20	0,16	0,26	0,32	0,34	0,26	0,04
	0,11	0,14	0,15	0,15	0,11	0,24	0,24	0,22	0,21	0,10
	0,09	0,13	0,12	0,13	0,10	0,21	0,19	0,17	0,15	0,15
	0,08	0,12	0,12	0,12	0,11	0,18	0,16	0,14	0,12	0,15
	0,12	0,15	0,15	0,15	0,12	0,22	0,23	0,22	0,19	0,11
Max Overlap	**0.1800**	**0.2050**	**0.2300**	**0.2550**	**0.2800**	**0.0190**	**0.0215**	**0.0240**	**0.0265**	**0.0290**
	0,20	0,22	0,22	0,20	0,20	0,22	0,24	0,34	0,34	0,34
	0,10	0,13	0,15	0,16	0,13	0,16	0,22	0,25	0,23	0,20
	0,08	0,09	0,12	0,12	0,12	0,15	0,21	0,19	0,18	0,15
	0,09	0,09	0,12	0,12	0,11	0,13	0,18	0,18	0,16	0,14
	0,12	0,13	0,15	0,15	0,14	0,17	0,21	0,24	0,23	0,21
Min Overlap	**0.0400**	**0.0650**	**0.0900**	**0.1150**	**0.1400**	**0.0010**	**0.0020**	**0.0045**	**0.0070**	**0.0095**
	0,20	0,20	0,22	0,20	0,20	0,32	0,56	0,66	0,34	0,26
	0,10	0,11	0,21	0,14	0,11	0,16	0,32	0,50	0,36	0,15
	0,09	0,09	0,19	0,15	0,12	0,11	0,21	0,36	0,37	0,17
	0,10	0,08	0,21	0,15	0,12	0,08	0,16	0,27	0,33	0,24
	0,12	0,12	0,21	0,16	0,14	0,17	0,31	0,45	0,35	0,20
	MSER									
	BRIEF					**BRIEF**				
Cosine Set	**0.1850**	**0.2100**	**0.2350**	**0.2600**	**0.2850**	**0.0100**	**0.0125**	**0.0150**	**0.0175**	**0.0200**
	0,48	0,40	0,50	0,48	0,44	0,24	0,22	0,30	0,24	0,22
	0,38	0,32	0,41	0,36	0,35	0,14	0,17	0,18	0,20	0,19
	0,37	0,29	0,36	0,35	0,32	0,11	0,16	0,15	0,18	0,17
	0,33	0,25	0,33	0,32	0,29	0,08	0,14	0,15	0,16	0,17
	0,39	0,31	0,40	0,38	0,35	0,14	0,17	0,19	0,19	0,19
Dice	**0.1650**	**0.1900**	**0.2150**	**0.2400**	**0.2650**	**0.0100**	**0.0125**	**0.0150**	**0.0175**	**0.0200**
	0,40	0,48	0,48	0,48	0,42	0,24	0,22	0,30	0,24	0,22
	0,34	0,40	0,41	0,41	0,33	0,14	0,17	0,18	0,20	0,19
	0,35	0,34	0,37	0,35	0,32	0,11	0,16	0,15	0,18	0,17
	0,30	0,34	0,34	0,35	0,30	0,08	0,14	0,15	0,16	0,17
	0,35	0,39	0,40	0,40	0,34	0,14	0,17	0,19	0,19	0,19
Jaccard	**0.2200**	**0.2450**	**0.2700**	**0.2950**	**0.3200**	**0.0100**	**0.0125**	**0.0150**	**0.0175**	**0.0200**
	0,50	0,44	0,52	0,40	0,24	0,20	0,22	0,30	0,24	0,22
	0,41	0,38	0,36	0,36	0,27	0,13	0,17	0,18	0,20	0,19
	0,36	0,37	0,32	0,31	0,25	0,13	0,16	0,15	0,18	0,17
	0,35	0,34	0,30	0,29	0,28	0,12	0,14	0,15	0,16	0,17
	0,40	0,38	0,38	0,34	0,26	0,15	0,17	0,19	0,19	0,19
Max Overlap	**0.1850**	**0.2100**	**0.2350**	**0.2600**	**0.2850**	**0.0090**	**0.0115**	**0.0140**	**0.0165**	**0.0190**
	0,48	0,44	0,50	0,48	0,44	0,20	0,22	0,28	0,24	0,24
	0,38	0,38	0,41	0,36	0,35	0,15	0,17	0,18	0,19	0,19
	0,37	0,37	0,36	0,35	0,32	0,11	0,13	0,16	0,16	0,16
	0,33	0,34	0,33	0,32	0,29	0,10	0,11	0,16	0,16	0,16
	0,39	0,38	0,40	0,38	0,35	0,14	0,16	0,19	0,19	0,19
Min Overlap	**0.1650**	**0.1900**	**0.2150**	**0.2400**	**0.2650**	**0.0100**	**0.0125**	**0.0150**	**0.0175**	**0.0200**

	0,40	0,48	0,48	0,48	0,42	0,24	0,22	0,30	0,24	0,22
	0,34	0,40	0,41	0,41	0,33	0,14	0,17	0,18	0,20	0,19
	0,35	0,34	0,37	0,35	0,32	0,11	0,16	0,15	0,18	0,17
	0,30	0,34	0,34	0,35	0,30	0,08	0,14	0,15	0,16	0,17
	0,35	0,39	0,40	0,40	0,34	0,14	0,17	0,19	0,19	0,19
	SIFT									
	BRIEF					**BRIEF**				
Cosine Set	**0.1100**	**0.1350**	**0.1600**	**0.1850**	**0.2100**	**0.0100**	**0.0125**	**0.0150**	**0.0175**	**0.0200**
	0,26	0,26	0,28	0,26	0,24	0,44	0,50	0,54	0,48	0,34
	0,18	0,17	0,21	0,18	0,15	0,35	0,49	0,52	0,47	0,28
	0,16	0,17	0,17	0,15	0,13	0,33	0,40	0,48	0,41	0,28
	0,15	0,16	0,16	0,16	0,13	0,28	0,35	0,40	0,40	0,27
	0,19	0,19	0,20	0,19	0,16	0,35	0,43	0,48	0,44	0,29
Dice	**0.1150**	**0.1400**	**0.1650**	**0.1900**	**0.2150**	**0.0105**	**0.0130**	**0.0155**	**0.0180**	**0.0205**
	0,22	0,26	0,30	0,28	0,24	0,44	0,52	0,62	0,52	0,32
	0,20	0,18	0,21	0,16	0,13	0,34	0,48	0,53	0,42	0,29
	0,16	0,17	0,17	0,15	0,13	0,33	0,42	0,47	0,39	0,25
	0,17	0,16	0,17	0,16	0,13	0,30	0,38	0,40	0,36	0,27
	0,10	0,19	0,19	0,19	0,16	0,35	0,45	0,50	0,42	0,28
Jaccard	**0.1250**	**0.1500**	**0.1750**	**0.2000**	**0.2250**	**0.0115**	**0.0140**	**0.0165**	**0.0190**	**0.0215**
	0,24	0,26	0,26	0,24	0,22	0,48	0,54	0,58	0,34	0,28
	0,23	0,20	0,19	0,16	0,13	0,41	0,51	0,51	0,37	0,28
	0,19	0,16	0,16	0,13	0,12	0,33	0,47	0,46	0,31	0,24
	0,18	0,15	0,17	0,14	0,11	0,32	0,39	0,41	0,32	0,21
	0,21	0,19	0,19	0,17	0,14	0,38	0,48	0,49	0,34	0,25
Max Overlap	**0.0850**	**0.1100**	**0.1350**	**0.1600**	**0.1850**	**0.0100**	**0.0125**	**0.0150**	**0.0175**	**0.0200**
	0,24	0,26	0,28	0,26	0,26	0,44	0,50	0,54	0,48	0,34
	0,15	0,18	0,21	0,17	0,18	0,35	0,49	0,52	0,47	0,28
	0,13	0,16	0,17	0,17	0,15	0,33	0,40	0,48	0,41	0,28
	0,13	0,15	0,16	0,16	0,16	0,28	0,35	0,40	0,40	0,27
	0,16	0,19	0,20	0,19	0,19	0,35	0,43	0,48	0,44	0,29
Min Overlap	**0.0850**	**0.1100**	**0.1350**	**0.1600**	**0.1850**	**0.0105**	**0.0130**	**0.0155**	**0.0180**	**0.0205**
	0,24	0,26	0,28	0,26	0,26	0,44	0,52	0,52	0,52	0,32
	0,15	0,18	0,21	0,17	0,18	0,34	0,48	0,48	0,42	0,29
	0,13	0,16	0,17	0,17	0,15	0,33	0,42	0,42	0,39	0,25
	0,13	0,15	0,16	0,16	0,16	0,30	0,38	0,38	0,36	0,27
	0,16	0,19	0,20	0,19	0,19	0,35	0,45	0,45	0,42	0,28
	STAR									
	BRIEF					**BRIEF**				
Cosine Set	**0.2050**	**0.2300**	**0.2550**	**0.2800**	**0.3050**	**0.0145**	**0.0170**	**0.0195**	**0.0220**	**0.0245**
	0,46	0,48	0,52	0,40	0,24	0,28	0,22	0,36	0,32	0,30
	0,37	0,39	0,42	0,31	0,27	0,16	0,14	0,24	0,25	0,23
	0,31	0,30	0,34	0,31	0,23	0,13	0,13	0,19	0,21	0,19
	0,28	0,27	0,31	0,31	0,25	0,13	0,13	0,16	0,18	0,16
	0,35	0,36	0,40	0,33	0,25	0,18	0,16	0,24	0,24	0,22
Dice	**0.2150**	**0.2400**	**0.2650**	**0.2900**	**0.3150**	**0.0180**	**0.0205**	**0.0230**	**0.0255**	**0.0280**
	0,46	0,48	0,50	0,34	0,24	0,24	0,36	0,36	0,26	0,16
	0,36	0,39	0,39	0,28	0,20	0,20	0,23	0,23	0,20	0,17
	0,29	0,31	0,35	0,31	0,21	0,17	0,19	0,21	0,18	0,17
	0,26	0,25	0,31	0,26	0,23	0,13	0,17	0,19	0,17	0,17
	0,34	0,36	0,39	0,30	0,22	0,18	0,24	0,25	0,20	0,17
Jaccard	**0.2050**	**0.2300**	**0.2550**	**0.2800**	**0.3050**	**0.0145**	**0.0170**	**0.0195**	**0.0220**	**0.0245**

	0,46	0,48	0,52	0,40	0,24	0,22	0,26	0,36	0,32	0,30
	0,37	0,39	0,42	0,31	0,27	0,14	0,20	0,24	0,25	0,23
	0,31	0,30	0,34	0,31	0,23	0,13	0,21	0,19	0,21	0,19
	0,28	0,27	0,31	0,31	0,25	0,13	0,18	0,16	0,18	0,16
	0,35	0,36	0,40	0,33	0,25	0,16	0,21	0,24	0,24	0,22
Max Overlap	**0.1950**	**0.2200**	**0.2450**	**0.2700**	**0.2950**	**0.0185**	**0.0210**	**0.0235**	**0.0260**	**0.0285**
	0,52	0,44	0,50	0,48	0,32	0,30	0,34	0,36	0,26	0,16
	0,36	0,35	0,41	0,35	0,28	0,22	0,23	0,25	0,24	0,16
	0,29	0,30	0,32	0,33	0,29	0,19	0,21	0,19	0,19	0,17
	0,27	0,26	0,28	0,31	0,27	0,16	0,18	0,19	0,18	0,18
	0,36	0,34	0,38	0,37	0,29	0,22	0,24	0,25	0,22	0,17
Min Overlap	**0.2000**	**0.2250**	**0.2500**	**0.2750**	**0.3000**	**0.0140**	**0.0165**	**0.0190**	**0.0215**	**0.0240**
	0,46	0,46	0,52	0,44	0,32	0,24	0,20	0,36	0,34	0,34
	0,36	0,34	0,42	0,34	0,26	0,17	0,16	0,25	0,24	0,24
	0,31	0,30	0,35	0,33	0,29	0,15	0,14	0,21	0,21	0,19
	0,28	0,28	0,29	0,30	0,27	0,12	0,11	0,16	0,17	0,16
	0,35	0,35	0,40	0,35	0,28	0,17	0,15	0,24	0,24	0,23
	SURF									
	BRIEF					**BRIEF**				
Cosine Set	**0.1700**	**0.1950**	**0.2200**	**0.2450**	**0.2700**	**0.0120**	**0.0145**	**0.0170**	**0.0195**	**0.0220**
	0,28	0,30	0,30	0,28	0,24	0,26	0,30	0,36	0,32	0,36
	0,20	0,23	0,23	0,23	0,17	0,19	0,21	0,31	0,34	0,27
	0,19	0,19	0,20	0,19	0,16	0,17	0,19	0,27	0,30	0,23
	0,18	0,19	0,18	0,18	0,15	0,17	0,19	0,23	0,26	0,22
	0,21	0,23	0,23	0,22	0,18	0,20	0,22	0,29	0,30	0,27
Dice	**0.1750**	**0.2000**	**0.2250**	**0.2500**	**0.2750**	**0.0125**	**0.0150**	**0.0175**	**0.0200**	**0.0225**
	0,30	0,30	0,30	0,26	0,22	0,24	0,30	0,40	0,36	0,36
	0,21	0,22	0,23	0,19	0,17	0,19	0,21	0,32	0,35	0,28
	0,19	0,18	0,20	0,17	0,13	0,17	0,21	0,29	0,29	0,22
	0,18	0,19	0,18	0,17	0,15	0,17	0,21	0,25	0,25	0,21
	0,22	0,22	0,23	0,20	0,17	0,19	0,23	0,31	0,31	0,27
Jaccard	**0.1750**	**0.2000**	**0.2250**	**0.2500**	**0.2750**	**0.0130**	**0.0155**	**0.0180**	**0.0205**	**0.0230**
	0,30	0,30	0,30	0,26	0,22	0,24	0,30	0,44	0,36	0,36
	0,21	0,22	0,23	0,19	0,17	0,19	0,22	0,34	0,34	0,28
	0,19	0,18	0,20	0,17	0,13	0,16	0,21	0,29	0,27	0,23
	0,18	0,19	0,18	0,17	0,15	0,17	0,22	0,26	0,25	0,23
	0,22	0,22	0,23	0,20	0,17	0,19	0,24	0,33	0,30	0,28
Max Overlap	**0.1400**	**0.1650**	**0.1900**	**0.2150**	**0.2400**	**0.0140**	**0.0165**	**0.0190**	**0.0215**	**0.0240**
	0,28	0,28	0,30	0,30	0,28	0,26	0,34	0,40	0,36	0,36
	0,20	0,22	0,22	0,21	0,20	0,20	0,29	0,36	0,28	0,26
	0,18	0,19	0,19	0,21	0,19	0,16	0,25	0,29	0,25	0,22
	0,17	0,19	0,18	0,18	0,18	0,16	0,22	0,26	0,22	0,22
	0,21	0,22	0,22	0,22	0,21	0,20	0,27	0,33	0,28	0,27
Min Overlap	**0.1400**	**0.1650**	**0.1900**	**0.2150**	**0.2400**	**0.0125**	**0.0150**	**0.0175**	**0.0200**	**0.0225**
	0,28	0,28	0,30	0,30	0,28	0,24	0,30	0,40	0,36	0,36
	0,20	0,22	0,22	0,21	0,20	0,19	0,21	0,32	0,35	0,28
	0,18	0,19	0,19	0,21	0,19	0,17	0,21	0,29	0,29	0,22
	0,17	0,19	0,18	0,18	0,18	0,17	0,21	0,25	0,25	0,21
	0,21	0,22	0,22	0,22	0,21	0,19	0,23	0,31	0,31	0,27

Tabelle 7-7: Übersicht der Retrieval-Qualität von verschiedenen lokalen Features bei veränderlichen Threshold

8. Verzeichnis

Glossar

AP	Average precision
ASIFT	Affin Scale Invariant Feature Transform
BRIEF	Binary Robust Independent Elementary Features
CBIR	Content Based Image Retrieval
CEDD	Color and Edge Directivity Descriptor
CenSurE	Center Surround Extremas
CFO	Config-File-Objekt (Struktur des Analysewerkzeugs)
CIE	Commission Internationale de l'Éclairage (Internationale Beleuchtungskommission)
CIE L*a*b*	CIE Luminanz Farbartebene a, b (Farbmodell)
CIE L*u*v*	CIE Luminanz Farbartebene u, v (Farbmodell)
CMY	Cyan, Magenta, Yellow (Farmodell)
CSS	Curvature Scale Space
CV	Computer Vision
CVV	Color Coherence Vector
DBIS	Datenbank- und Informationssysteme
DBS	Datenbanksystem
DCT	Diskrete Cosinus Transformation
DFD	Datenflussdiagramm/data flow diagram
DFT	Diskrete Fourier-Transformation
DoB	Difference of Boxes
DoG	Difference of Gaussian
DoH	Difference of Hessian
DWT	Diskrete Wavelet-Transformationen
EBR	Edge-Based Regions
FAST	Features from Accelerated Segment Test
FCTH	Fuzzy Color and Texture Histogram
FFT	Fast Fourier-Transformation
ForMaT	Forschung für den Markt im Team

GFTT	Good Features to Track
HSB	Hue, Saturation, Brightness (Farbmodell)
HSV	Hue, Saturation, Value (Farbmodell)
IBR	Intentsity-Based Regions
IR	Information Retrieval
LoG	Laplacian of Gaussian
MAP	Mean average precision
MIR	Multimedia-Information Retrieval
MSER	Maximally Stable Extremal Regions
OBRIEF	Orientation sensitive Binary Robust Independent Elementary Features
OpenCV	Open Source Computer Vision
PCA	Principal component analysis (diskrete Karhunen-Loève Transformation)
PCA-SIFT	Principal component analysis Scale Invariant Feature Transform
POI	Point of Interest
PSURF	Phase-space based Speeded Up Robust Features
QBE	Query-by-example
QBS	Query-by-sketch
RGB	Red, Green, Blue (Farbmodell)
SBRIEF	Scaled Binary Robust Independent Elementary Features
SIFT	Scale Invariant Feature Transform
SPCA	Simple Principal Component Analysis
SUSAN	Smallest Univalue Segment Assimilating Nucleus
SURF	Speeded Up Robust Features
TREC	Text Retrieval Conference
UBRIEF	Upright Binary Robust Independent Elementary Features
UML	Unified Modeling Language
USURF	Upright Speeded Up Robust Features
XML	Extensible Markup Language

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[Agr08] M. Agrawal, K. Konolige, M. Blas, *CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching*, 2008

[Akm09] AKM-Group, *Geschichte der Datenbanken*, 2009, `http://www.amk-group.net/Datenbanken.9.0.html`, Zugriff: 04.06.11

[Ame10] S. Ameling, *Improving the Quality of Endoscopic Images and Videos*, 2010

[Bae99] R. Baeza-Yates, *Modern Information Retrieval*, ACM Press, 1999, ISBN 0-201-39829-X

[Bay06] H. Bay, T. Tuytelaars, L. Van Gool, *SURF: Speeded Up Robust Features*, 2006

[Ber11] M. Bertram, *Dokumentation für das System Pythia am Lehrstuhl DBIS*, 2011

[Bla05] U. Blazey, R. Dumke, *Ein Metriken basierter Ansatz für das Informationsmanagement von eLearning Projekten*, 2005

[Bla07] B. Blair, C. Murphy, *Difference of Gaussian Scale-Space Pyramids for SIFT Feature Detection*, 2007

[Bla08] C. Blab, *Semantische Bildannotation*, 2008

[Bra11] G. Bradski, *Willow Garage, OpenCV, ROS And Object Recognition*, 2011

[Bra98] D. Brauer, R. Raible-Besten, M. Weigert, *Multimedia-Lexikon*, Oldenbourg Verlag ,1998, ISBN 3-486-24445-0

[Bue08] W. Büschel, *Ähnlichkeitsmaße zur Objekterkennung*, 2008

[Bur06] W. Burger, M. Burge, *Digitale Bildverarbeitung*, Springer-Verlag, 2006, ISBN 3-540-30940-3

[Cal03] Caltech, 101C*ategories*, `http://www.vision.caltech.edu/Image_Datasets/Caltech101`, Zugriff: 13.07.11

[Cal10] M. Calonder, V. Lepetit, C. Strecha, P. Fua, *BRIEF: Binary Robust Independent Elementary Features*, 2010

[Cal11] M. Calonder, V. Lepetit, M. Özuysal, T. Trzcinski, C. Strecha, P. Fua *BRIEF: Computing a local binary descriptor very fast*, 2011

[Cas02] V. Castelli, L. Bergman, *Image Databases: Search and Retrieval of Digital Imagery*, Wiley Online Library, 2002

[Che93] Chaur-Chin Chen, Improved Moment Invariants for shape discrimination, 1993

[Col02] Peter Collin, *Dictionary of Multimedia (3rd edition)*, Peter Collin Publishing, 2002, ISBN 1-901659-51-8

[Des03] T. Deselaers, *Features for Image Retrieval*, 2003

[Ebl11] Michael Eblinger, *Kantendetektion*, 2011

[Fel04] J. Feldkamp, *Image Retrieval aus Bilddatenbanken*, 2004

[Fen03] D. Feng, W.C. Sui, H. Zhang, *Multimedia Information Retrieval and Management*, Springer-Verlag, 2003,
ISBN 3-540-00244-8

[Fis03] R.Fisher, S.Perkins, A.Walker, E.Wolfart, *Feature Detectors*,
`http://homepages.inf.ed.ac.uk/rbf/HIPR2/featops.htm`,
Zugriff: 13.07.11

[Fuh06] N. Fuhr, *Information Retrieval Skriptum SS06 Universität Duisburg*, 2006

[FZI11] Forschungszentrum Informatik Karlsruhe, *Visuelle Methoden zur semantischen Bildannotation und –suche,*
`http://www.fzi.de/index.php/de/forschung/forschungsbereiche/ipe/222`
Zugriff: 14.06.11

[Gal09] Andrei Galea, *Entwurf und Implementierung eines Frameworks zur Extraktion von Features aus Bildern,* 2009

[Gim06] G. Gimel'farb, *Content-Based Video Information Search and Retrieval*,
`http://www.cs.auckland.ac.nz/compsci708s1c/lectures/Glect-html/top708-2006.html`
Zugriff: 23.06.11

[Har06] J. Hare, P. Lewisa, P. Enserb, und C. Sandomb, *Mind the gap: another look at the problem of the semantic gap in image retrieval*, 2006

[Hin09] S. Hinterstoisser, O. Kutter, N. Navab, P. Fua, V. Lepetit, *Real-Time Learning of Accurate Patch Rectification*, 2009

[Hol11] Holger Meier (Bildquelle), `www.holger-meier.net`, Zugriff: 07.07.11

[Jäh05] B. Jähne, *Digitale Bildverarbeitung (Auflage 6)*, Axel Springer Verlag, 2005, ISBN 978-3540249993

[Kel11] C. Kellner, *Seminar Angewandtes Information Retrieval WS1011 Universität Basel*, http://pages.unibas.ch/LIlab/studies/IR-FS2011/Content-Based%20Image%20retrieval.pdf Zugriff: 02.06.11

[Ker11] Kerana (Bildquelle), www.kerana.de, Zugriff: 07.07.11

[Kie07] W. Kienzle, F. Wichmann, B. Schölkopf, M. Franz, *Center-surround filters emerge from optimizing predictivity in a free-viewing task*, 2007

[Kir08] G. Kirchhoff, *Bildverarbeitung in der Medizin*, 2008

[Kub05] A. Kubias, *OpenCV – Open Source Computer Vision Library,* http://www.uni-koblenz.de/~kubias/FolienOpenCV.pdf Zugriff: 17.06.11

[Kun11] Stonegate (Bildquelle), www.kunststeine.eu, Zugriff: 07.07.11

[Lew07] D. Lewandowski, *Qualitätsmessung bei Suchmaschinen*, 2007

[Los02] D. Loss, *Data Mining: Klassifikations- und Clusteringverfahren*, 2002

[Man11] MangaCarta (Bildquelle), www.mangacarta.de, Zugriff: 07.07.11

[Mat02] J. Matas, O. Chum, M. Urban,T. Pajdla, *RobustWide Baseline Stereo from Maximally Stable Extremal Regions*, 2002

[McG03] McGraw-Hill, *McGraw-Hill Dictionary of Scientific & Technical Terms* (6th edition), McGraw-Hill Companies, 2003, ISBN 978-0-070-42313-8

[Mes10] MesosWorld, *Korrelationsanalyse,* http://www.mesosworld.ch/lerninhalte/Biv_Korrelation/de/html/index.html, Zugriff: 28.08.11

[Mik05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Mantas, F. Schaffalitzky, T. Kadir, L. Van Gool, *A Comparison of Affine Region Detectors*, 2005

[Mue02] H. Müller, S. Marchand-Maillet, T. Pun, *The truth about Corel - evaluation in image retrieval*, 2002

[Nob11] Nobsta's Foto-Blog (Bildquelle), nobsta.wordpress.com, Zugriff: 07.07.11

[Ope11] Willow Garage, *OpenCV Wiki*, http://opencv.willowgarage.com/wiki/, Zugriff: 17.06.11

[Phi07] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, *Object retrieval with large vocabularies and fast spatial matching*, 2007

[Piq11] piqs.de (Bildquelle), `www.piqs.de`, Zugriff: 07.07.11

[Rao99] A. Rao, R. Srihari, Z. Zhang, *Spatial Color Histograms for Content-Based Image Retrieval*, Journal IEEE International Conference on Tools with Artificial Intelligence, 1999

[Rij79] C. J. van Rijsbergen, *INFORMATION RETRIEVAL*, `http://www.dcs.glasgow.ac.uk/Keith/Preface.html`, Zugriff: 10.06.11

[Ros05] E. Rosten, T. Drummond, *Fusing Points and Lines for High Performance Tracking*, 2005

[Ros06] E. Rosten, T. Drummond, *Machine learning for high-speed corner detection*, 2006

[Sal83] G. Salton, M. McGill, *Introduction to Modern Information Retrieval,* 1983

[San01] S. Santini, *Exploratory Image Databases*, Academic Press, 2001, ISBN 0-12-619261-8

[Sch06] I. Schmitt, *Ähnlichkeitssuche in Multimedia-Datenbanken*, Oldenbourg Wissenschaftsverlag GmbH, 2006, ISBN 978-3-486-57907-9

[Shi93] J. Shi, C. Tomasi, *Good Features to Track*, 1993

[Sme00] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, *Content Based Image Retrieval at the End of the Early Years*, Journal IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000

[Som04] C. Sommer, *Seminar Inhaltsbasierte Bildsuche*, 2004, `http://132.230.167.110/lectures/bild_seminar/data/03-col.pdf` Zugriff: 10.06.11

[Sou11] Sourceforge, *Open Computer Vision Library*, `http://sourceforge.net/projects/opencvlibrary/files/opencv-win/stats/timeline?dates=2001-03-15+to+2011-05-17` Zugriff: 10.06.11

[Sti06] S. Stiene, *Konturbasierte Objekterkennung aus Tiefenbildern eines 3D-Laserscanners*, 2006

[Sto07] W. Stock, *Information Retrieval: Informationen suchen und finden*, Oldenbourg Wissenschaftsverlag GmbH, 2007, ISBN 3-12486-58172-4

[Str03] C. Straßer, *Kantendetektion in der Bildverarbeitung*, 2003

[Tha11] Thaimassage Jatt (Bildquelle), `www.thaimassage-jatt.de`, Zugriff: 07.07.11

[Tis11] Artipics (Bildquelle), `www.tischkunst.de`, Zugriff: 07.07.11

[Tuy08] T. Tuytelaars, K. Mikolajczyk, *Local Invariant Feature Detectors: A Survey*, 2008

[Ulg06] Adrian Ulges, *Recognizing Objects in Still Images and Video Streams*, 2006

[Vie11] View Fotocommunity (Bildquelle), `http://view.stern.de/files/img/basic/blank.gif` Zugriff: 07.07.11

[Vis07] Vision & Control GmbH, *Tutorial Filter*, 2007

[Wag06] C. Wagner, *Kantenextraktion*, 2006

[Wie08] J. Wienke, *Bildverarbeitung mit OpenCV*, 2008

[Zha01] D. Zhang, G. Lu, *Content-Based Shape Retrieval Using Different Shape Descriptors: A Comparative Study*, 2001

[Zha03] D. Zhang, G. Lu, *A Comparative Study of Curvature Scale Space and Fourier Descriptors for Shape-based Image Retrieval*, 2003

[Zho04] X. Zhou, *Henry Small and His Sciences Mapping*, 2004

[Zib01] C. Zibreira und F. Pereira, *A Study of Similarity Measures for a Turning Angles-based Shape Descriptor*, 2001