Kontextmodelle für lokale Merkmale zur inhaltsbasierten Bildsuche in großen Bilddatenbanken

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Daniel Manger

Tag der mündlichen Prüfung:	28.05.2018
Erster Referent:	Prof. Dr.-Ing. Jürgen Beyerer
Zweiter Referent:	Prof. Dr.-Ing. Rainer Stiefelhagen

Kurzfassung

Vor allem seit Smartphones für viele zum ständigen Begleiter geworden sind, wächst die Menge der aufgenommenen Bilder rasant an. Oft werden die Bilder schon unmittelbar nach der Aufnahme über soziale Netzwerke mit anderen geteilt. Zur späteren Verwendung der Aufnahmen hingegen wird es zunehmend wichtiger, die für den jeweiligen Zweck relevanten Bilder in der Masse wiederzufinden. Für viele bekannte Objektklassen ist die automatische Verschlagwortung mit entsprechenden Detektionsverfahren bereits eine große Hilfe. Anhand der Metadaten können außerdem häufig Ort oder Zeit der gesuchten Aufnahmen eingegrenzt werden. Dennoch führt in bestimmten Fällen nur eine inhaltsbasierte Bildsuche zum Ziel, da dort explizit mit einem Anfragebild nach individuellen Objekten oder Szenen gesucht werden kann.

Obwohl die Forschung im Bereich der inhaltsbasierten Bildsuche im letzten Jahrzehnt bereits zu vielen Anwendungen geführt hat, ist die Skalierbarkeit der sehr genauen Varianten noch eingeschränkt. Das bedeutet, dass die existierenden Verfahren, mit denen ein Bildpaar robust auf lokal ähnliche Teilinhalte untersucht werden kann, nicht ohne weiteres auf die Suche in vielen Millionen von Bildern ausgeweitet werden können.

Diese Dissertation widmet sich dieser Art der inhaltsbasierten Bildsuche, die Bilder anhand ihrer lokalen Bildmerkmale indexiert, und adressiert zwei wesentliche Einschränkungen des populären Bag-of-Words-Modells. Zum einen sind die Quantisierung und Komprimierung der lokalen Merkmale, die für die Suchgeschwindigkeit in großen Bildmengen essentiell sind, mit einem gewissen Verlust von Detailinformation verbunden. Zum anderen müssen die indexierten Merkmale aller Bilder immer im Arbeitsspeicher vor-

liegen, da jede Suchanfrage den schnellen Zugriff auf einen beträchtlichen Teil des Index erfordert. Konkret beschäftigt sich die Arbeit mit Repräsentationen, die im Index nicht nur die quantisierten Merkmale, sondern auch ihren Kontext einbeziehen. Abweichend zu den bisher üblichen Ansätzen, wird der Kontext, also die größere Umgebung eines lokalen Merkmals, als eigenständiges Merkmal erfasst und ebenfalls quantisiert, was den Index um eine Dimension erweitert. Zunächst wird dafür ein Framework für die Evaluation solcher Umgebungsrepräsentationen entworfen. Anschließend werden zwei Repräsentationen vorgeschlagen: einerseits basierend auf den benachbarten lokalen Merkmalen, die mittels des Fisher Vektors aggregiert werden, andererseits auf Basis der Ergebnisse von Faltungsschichten von künstlichen neuronalen Netzen. Nach einem Vergleich der beiden Repräsentationen sowie Kombinationen davon im Rahmen des Evaluationsframeworks, werden die Vorteile für ein Gesamtsystem der inhaltsbasierten Bildsuche anhand von vier öffentlichen Datensätzen bewertet. Für die Suche in einer Million Bildern verbessern die vorgeschlagenen Repräsentationen auf Basis der neuronalen Netze die Suchergebnisse des Bag-of-Words-Modells deutlich.

Da die zusätzliche Indexdimension einen effektiveren Zugriff auf die indexierten Merkmale ermöglicht, wird darüber hinaus eine neue Realisierung des Gesamtsystems vorgeschlagen. Das System ist bezüglich des Index nicht mehr auf den Arbeitsspeicher angewiesen, sondern kann von aktuellen nichtflüchtigen Speichermedien profitieren, etwa von Solid-State-Disk (SSD)-Laufwerken. Von der Kombination der vorgeschlagenen Umgebungsrepräsentation der lokalen Merkmale und der Realisierung mit großen und günstigen SSD-Laufwerken können bereits heutige Systeme profitieren, denn sie können dadurch noch größere Bilddatenbanken für die inhaltsbasierte Bildsuche zugänglich machen.

Abstract

The number of digital images is growing rapidly, especially since smartphones have become part of our daily lives. Often, immediately after being taken, the images are shared with others via social networks. In order to use the images at a later stage, however, it is becoming increasingly important to find relevant images within large collections for the respective purpose. Methods that automatically assign keywords using appropriate detection methods are already of great help for many well-known object classes. Moreover, metadata can often be exploited to narrow down the location or time of the images to be searched for. Nevertheless, the number of remaining images is often very large and a further step is required to achieve meaningful results. In these cases, Content-Based Image Retrieval (CBIR) can be beneficial, since individual objects or scenes can be searched for using a specific query image. Although research in the last decade has already led to many applications for CBIR, the scalability of the very precise variants is still limited. In other words, the existing approaches to robustly compare a pair of images for locally similar contents cannot easily be extended to searches in many millions of images.

This thesis focuses on those CBIR techniques that index images via their local features, and addresses two major limitations of the established bag-of-words model. On the one hand, the quantization and compression of local features, which allow for fast search in large databases, lead to a certain loss of information. On the other hand, the indexed features of all images must be kept in the main memory, since during a query operation, fast access to a considerable part of the index is required. More concretely, this thesis deals with representations that integrate not only the quantized local features but

also their larger context into the index. In contrast to previous approaches, the context is represented as a separate feature to be quantized as well, thus adding one more dimension to the index. First, a framework for the evaluation of such context representations is designed. Subsequently, two representations are proposed: the first being based on the aggregation of neighboring local features with the Fisher vector, and the second extracting information from convolutional layers of artificial neural networks. After comparing the two representations and combinations of them within the evaluation framework, the advantages for an overall CBIR system are assessed using four public datasets. For searching in one million images, the proposed representations based on neural networks significantly improve the results of the bag-of-words model.

Since the additional dimension of the index allows for a much more discriminative access to the indexed features, this thesis further proposes a new variant of a CBIR system, in which the index is not required to be kept in the main memory anymore. Instead, the presented approach can benefit from the most advanced storage technology such as solid-state drives. This combination of the proposed context representation and an implementation utilizing large and inexpensive solid-state drives adds value for today's systems enabling content-based image retrieval in even larger image collections.

Inhaltsverzeichnis

1

Einleitung

1.1. Motivation

Galt ein Bild in Zeiten der analogen Fotografie noch als Versuch, einen besonderen Moment für die Ewigkeit festzuhalten, so ist das Fotografieren heute ein nicht mehr wegzudenkender Bestandteil unseres Alltags. Möglich wurde dies durch die Digitalisierung. Bereits im Jahr 2003 wurden erstmals mehr Digitalkameras als analoge Fotoapparate verkauft [Dol12] und ab 2010 wiederum mehr Smartphones als Digitalkameras [Bit17]. Durch die ständige Begleitung des Smartphones und die Tatsache, dass ein Schnappschuss keine unmittelbaren Kosten mehr verursacht, liegt der Finger lockerer denn je auf dem Auslöser. In der Folge steigen die Bild- und Videomassen rasant an. Laut einer Schätzung von InfoTrends wurden im Jahr 2017 weltweit etwa 1,2 Billionen Bilder aufgenommen – etwa 85 Prozent davon mit Smartphones [Bit17].

Die bloße Aufnahme eines Bildes steht dabei nur am Anfang der digitalen Informationsverarbeitung. Die Bilder werden anschließend häufig mit Cloud-Diensten synchronisiert, bearbeitet und in sozialen Netzwerken mit anderen Nutzern geteilt. Allein zu den verschiedenen Diensten von Facebook wurden 2016 täglich etwa 2 Milliarden Bilder hochgeladen [LeC16]. Um angesichts der stetig wachsenden Bilddaten nicht den Überblick zu verlieren und sie für die jeweiligen Zwecke wiederfinden zu können, sind neben den

Metadaten wie Aufnahmezeitpunkt und GPS-Position auch automatische Bildanalyse- und Suchverfahren für viele Anwendungsbereiche unerlässlich geworden. Eine große Unterstützung stellt hier bereits die automatische Verschlagwortung mit Hilfe entsprechender Detektionsverfahren für zuvor festgelegte und trainierte Objekttypen dar (Personen, Gesichter, Fahrzeuge, etc.). Einen Schritt weiter geht die inhaltsbasierte Bildsuche (engl. *content-based image retrieval*). Deren Ziel ist es, einen großen Bilddatenbestand vorab so zu indexieren, dass er später innerhalb kürzester Zeit mit einem beliebigen Anfragebild durchsucht werden kann. Das Ergebnis einer solchen Suche stellt eine Trefferliste dar, die nur diejenigen Bilder des Datenbestands sortiert auflistet, die im gesamten Bild oder auch nur in Teilbereichen eine gewisse Ähnlichkeit zum Anfragebild aufweisen. Abbildung 1.1 zeigt ein Beispiel eines solchen Systems anhand einer Suchanfrage. Links unten in der Oberfläche ist das ausgewählte Anfragebild zu sehen und im rechten Teil sind die Ergebnisse der Suchanfrage in Form der gefundenen neun ähnlichsten Bilder in einem Datenbestand von ca. 1 Million Bilder aufgelistet. Die Suche basiert hier auf lokalen Merkmalen, sodass die im Anfragebild abgebildete Golden Gate Bridge auch bei abweichendem Vordergrund gefunden wird.

Bezüglich der *Ähnlichkeit* zweier Bilder existieren im allgemeinen Sprachgebrauch viele unterschiedliche Interpretationen, die auch in der Fachliteratur zu uneinheitlich verwendeten Begriffen führen. Zwei unterschiedliche Aspekte der Ähnlichkeit sollen an dieser Stelle getrennt behandelt werden:

1. Der kausale Aspekt: Aus welchem Grund können sich die Ähnlichkeiten ergeben?

2. Die bildliche Ausprägung der Ähnlichkeit: Was sind die Indizien, anhand derer für zwei gegebene beliebige Bilder eine Ähnlichkeit festgestellt werden kann?

Für den kausalen Aspekt sind die am häufigsten anzutreffenden Interpretationen

- Ähnlichkeit bei Fast-Duplikaten, d. h. in beiden Bildern sind Teile enthalten, die ursprünglich von derselben digitalen Vorlage stammen, aber anschließend unterschiedlich modifiziert wurden.

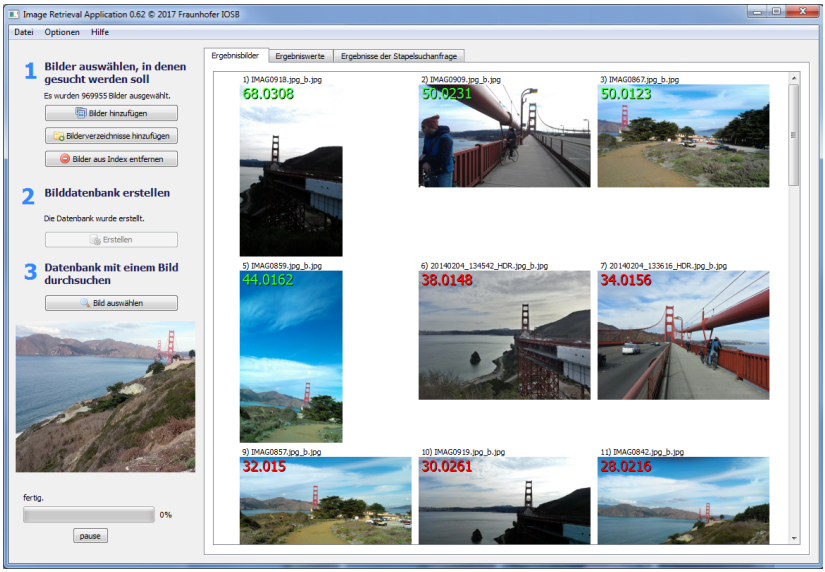

Abbildung 1.1.: Oberfläche einer Software für die inhaltsbasierte Bildsuche, die im Rahmen dieser Arbeit entstanden ist, und überwiegend im forensischen Bereich eingesetzt wird (`https://s.fhg.de/cbir`).

- Ähnlichkeit basierend auf Objektinstanzen, beispielsweise zwei unterschiedliche Aufnahmen desselben Motorrads. In der Regel schließt diese Interpretation auch zwei Instanzen desselben Typs ein, sofern diese in den Bildern nicht unterscheidbar sind, etwa zwei Tafeln Schokolade derselben Sorte desselben Herstellers.
- Ähnlichkeit basierend auf semantischen Klassen, beispielsweise zwei unterschiedliche Bäume.

Diese Dissertation fokussiert sich auf die Objektinstanz-basierte Ähnlichkeit und geht von einer bildlichen Ausprägung aus, die keinerlei Hintergrundwissen über die Objekte und deren semantische Klassen voraussetzt. Die Ähnlichkeit soll allein auf lokal ähnlichen charakteristischen Bereichen der Bilder basieren, etwa auf Bildgradienten in einer ähnlicher Anordnung – in Abbildung 1.1 trifft dies auf die Pylone der Golden Gate Bridge zu.

Bezüglich der möglichen Anwendungen der inhaltsbasierten Bildsuche liegt der Fokus dieser Arbeit im forensischen Bereich. Ein Anfragebild soll beispielsweise mit allen Datenbankbildern verglichen werden um festzustellen, ob Vorder- oder Hintergrund gemeinsam abgebildete Objekte, Objektteile oder Szenen enthalten. Dabei ist weder in den Datenbankbildern noch im Anfragebild vorab bekannt, welche Objekte für die Suche relevant sein werden. Im Gegensatz zu Systemen zur Wiedererkennung einer endlichen Menge von Sehenswürdigkeiten kann daher im Vorfeld kein domänenspezifisches Vorwissen eingebracht werden.

Die Forschung der letzten Dekade auf dem Gebiet der inhaltsbasierten Bildsuche mündete zwar bereits in viele erfolgreiche Anwendungen, allerdings ist deren Skalierbarkeit noch immer eingeschränkt. Das heißt, dass in der Regel nur wenige Millionen Bilder indexiert werden können, wenn eine praxisrelevante Suchgenauigkeit erhalten bleiben soll. Für viele Anwendungen im forensischen Kontext oder für lokale Mediendatenbanken, ist dies ausreichend; für die Bildermengen im Web-Maßstab ist diese Art der inhaltsbasierten Bildsuche aber weiterhin nicht verfügbar. Zwar bieten die gängigen Suchmaschinen Google[1], Bing[2], Baidu[3] oder TinEye[4] neben der textbasierten Suche auch die Suche mit einem Anfragebild an. Mit dem Anfragebild aus Abbildung 1.1 konfrontiert, liefern sie allerdings nur Ergebnisbilder wie die in Abbildung 1.2(b)-(e) dargestellten mit ähnlichen Szenen (Küste, Strand oder Berge) zurück, in denen die markante Golden Gate Bridge jedoch nicht vorkommt. Korrekte Suchtreffer *mit* der Sehenswürdigkeit werden hingegen nur gefunden, wenn das interessierende Objekt im Anfragebild den überwiegenden Teil des Bildes darstellt, oder wenn die Aufnahmeperspektive sehr ähnlich zum Anfragebild ist.

1 https://www.google.de/imghp
2 https://www.bing.com/?scope=images
3 https://image.baidu.com
4 https://www.tineye.com

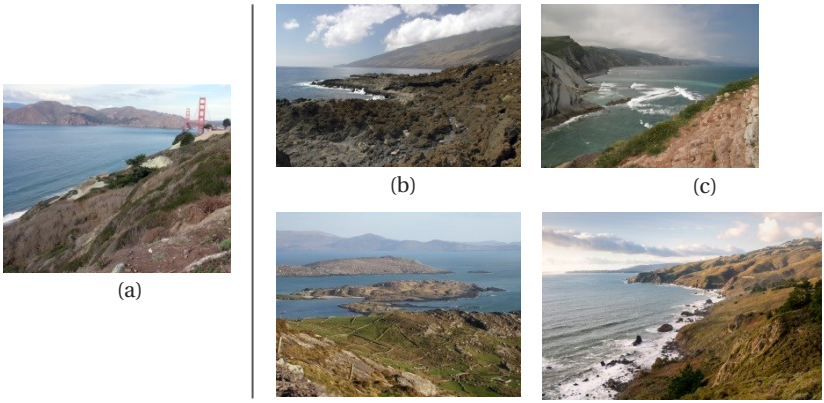

Abbildung 1.2.: Ergebnisse einer inhaltsbasierten Bildsuche im Internet. (a): Anfragebild, (b)-(e): Ergebnisbilder, die mit dem Anfragebild im Rahmen der Google Bildsuche ermittelt wurden, wobei aus lizenzrechtlichen Gründen nur eine Auswahl der gefundenen Ergebnisse aufgeführt ist. Quellen: siehe Quellenverzeichnis.

1.2. Herausforderungen

Robuste Verfahren, mit denen zwei Bilder bezüglich ähnlicher Bildstrukturen untersucht werden können, existieren seit vielen Jahren. Wie in vielen anderen Aufgabenstellungen der Bildverarbeitung ergab sich durch die Verwendung von lokalen Bildmerkmalen, wie beispielsweise Scale-Invariant Feature Transform (SIFT) [Low99], auch hier ein bemerkenswerter Durchbruch. Dabei werden lokale Bildbereiche in Merkmale überführt, die sich hinsichtlich Ähnlichkeit mit den Merkmalen eines anderen Bildes vergleichen lassen. Dass heute – annähernd 20 Jahre nach dem Beginn der Ära der lokalen Bildmerkmale – diese Möglichkeit, Bildpaare anhand lokaler Strukturen robust zu vergleichen, immer noch nicht auf aktuelle Datenbankgrößen übertragen werden konnte, liegt im Wesentlichen an der Quantisierung und Komprimierung der lokalen Merkmale. Sie sind essentiell für die Geschwindigkeit der Suche, denn ein paarweiser Vergleich des Anfragebildes mit jedem der Datenbankbilder ist basierend auf den originalen lokalen Merkmalen aus Laufzeitgründen in der Praxis unmöglich. Bei der Erstellung

des *Index*, also der für die Suche komprimierten Form der Merkmale aller Datenbankbilder, müssen eine Reihe von Herausforderungen berücksichtigt werden:

- **Kompakte Repräsentation:** Da der Index im Allgemeinen im Hauptspeicher gehalten wird, ist der erforderliche Speicherbedarf pro Bild entscheidend für die maximale Anzahl der Bilder, die in der Datenbank berücksichtigt werden können.
- **Robuste Repräsentation:** Objekte im Anfragebild sollen auch dann in den Datenbankbildern gefunden werden, wenn sie nur einen geringen Teil des Bildes ausmachen. Außerdem soll die Suche möglichst robust sein im Hinblick auf bildliche Variationen, die durch Rauschen, Unschärfe, Verdeckungen und Unterschieden in Beleuchtung, Blickwinkel, Aufnahmesensorik etc. verursacht werden. Diese Leistungsfähigkeit der Bildsuche wird in der Regel mit zwei Charakteristiken beurteilt:
  - Die **Trefferquote** (engl. *recall*) misst, wie viele der relevanten Bilder gefunden werden und zielt somit auf die Vollständigkeit der Suchergebnisse, während die
  - **Genauigkeit** (engl. *precision*) angibt, wie viele der gefundenen Bilder relevant sind. Damit wird also die Fähigkeit des Systems beurteilt, sich von nichtrelevanten Bildern möglichst wenig beeinflussen zu lassen.
- **Effiziente Suchanfrage:** Pro Datenbankbild sollte die Laufzeit für eine Suchanfrage möglichst gering sein. Für internetbasierte Suchmaschinen gilt dies aus zweierlei Gründen: es soll nicht nur in Milliarden von Datenbankbildern gesucht werden können, sondern es gibt in der Regel auch Milliarden von Anwendern, die neue Suchmöglichkeiten ohne Verzögerungen[1] nutzen möchten.

1 Bei der Bildsuche von Bing wird bislang beispielsweise die Möglichkeit, ein Bild als Anfrage zu verwenden, erst aktiviert, nachdem eine textbasierte Bildsuche durchgeführt wurde. Da die Resultate einer inhaltsbasierten Bildsuche aber nicht vom vorherigen textuellen Suchbegriff abzuhängen scheinen, könnte diese Hürde auf eine Begrenzung der Anzahl der Suchanfragen zielen.

1.3. Eigene Beiträge

Da mit der erforderlichen Quantisierung und Komprimierung der lokalen Bildmerkmale immer auch ein gewisser Verlust von Information verbunden ist, zielt ein Schwerpunkt der Forschung auf verschiedene Ansätze, diesen Informationsverlust auszugleichen. Im Rahmen dieser Dissertation werden dabei diejenigen Verfahren betrachtet, die den Index durch zusätzliche Informationen erweitern, um die lokalen Merkmale nicht unabhängig voneinander zu repräsentieren, sondern ihren Kontext mit einzubeziehen. Verglichen mit den bisherigen Strategien in diesem Bereich, fokussiert sich diese Arbeit darauf, die größere Bildumgebung der lokalen Merkmale wiederum als eigenständiges Merkmal zu erfassen und somit im Index als eine weitere Dimension zu verwenden. Für diese Repräsentation des Index einer inhaltsbasierten Suche werden in dieser Dissertation im Einzelnen die folgenden neuen Beiträge eingebracht:

1. Entwurf und Realisierung eines Evaluationsframeworks, welches die relevante Sicht der inhaltsbasierten Bildsuche auf die lokalen Merkmale darstellt und dabei die spezifische Aufgabe modelliert, die die neue Repräsentation erfüllen soll [Man17b].

2. Definition von zwei unterschiedlichen Repräsentationen, die die größere Bildumgebung eines Merkmals erfassen und dabei die unterliegenden Invarianzen erhalten: einerseits basierend auf den jeweils benachbarten Merkmalen [Man16a] und andererseits auf Grundlage von künstlichen neuronalen Netzen [Man17b].

3. Evaluation der vorgeschlagenen Repräsentationen und ihrer Kombinationen im Evaluationsframework sowie abschließend im Rahmen eines Gesamtsystems zur inhaltsbasierten Bildsuche mit vier gängigen öffentlichen Datensätzen in diesem Bereich [Man17a].

4. Entwicklung und Untersuchung neuer Möglichkeiten für die Speicherauslegung, die sich durch die neue Repräsentation des Index eröffnen [Man18]. In Kombination mit den schnellen Zugriffszeiten für zufällig verteilte Leseoperationen aktueller SSD-Laufwerke wird es erstmals möglich, den Index im nichtflüchtigen Speicher zu belassen, statt ihn wie bisher aus Laufzeitgründen zwingend auch im Arbeitsspeicher

bereitzuhalten. Damit fällt eine der Haupteinschränkungen hinsichtlich der Skalierungsfähigkeit der Systeme in Bezug auf die Größe der Bilddatenbank weg.

1.4. Gliederung

Diese Arbeit gliedert sich wie folgt: Zunächst wird der *Stand der Forschung* beschrieben, beginnend mit den Verfahren, die auf lokalen Merkmalen basieren und diese einzeln indexieren, gefolgt von Varianten, die ein Bild mit einer globalen Beschreibung repräsentieren. Im darauffolgenden Kapitel *Konzept* wird die exakte Stelle in einem System zur inhaltsbasierten Bildsuche herausgearbeitet, an der die Beiträge dieser Arbeit ansetzen, und die einzelnen Beiträge werden motiviert. Das Kapitel *Framework zur Evaluation von Umgebungsmerkmalen* leitet dann den ersten Beitrag ein und beschreibt das entworfene Evaluationsframework. Als nächstes folgt die Vorstellung der beiden Varianten der *Umgebungsmerkmale*, die danach im Rahmen des Evaluationsframeworks miteinander verglichen werden. Die letztlichen Auswirkungen der Umgebungsrepräsentation auf die Genauigkeit der Suchergebnisse eines kompletten Systems zur inhaltsbasierten Bildsuche werden im Kapitel *Evaluation im Rahmen der inhaltsbasierten Bildsuche* untersucht. Der letzte Beitrag – die neuen Möglichkeiten für die Speicherauslegung – wird im Kapitel *Speicherauslegung des Index* dargestellt und mit eigenen Auswertungen untermauert bevor *Zusammenfassung und Ausblick* die Arbeit abschließen.

2

Stand der Forschung

In der Fachliteratur wurde bislang eine Vielzahl unterschiedlicher Systeme vorgeschlagen, um bestimmte Bilder in einer großen Datenbank zu suchen. Dies liegt zum einen an den vielfältigen Zielen der Nutzer, zum anderen werden Bilder im digitalen Zeitalter in den seltensten Fällen komplett isoliert erfasst. So speichern heutige mobile Geräte bereits bei der Aufnahme unterschiedliche Metadaten (Zeitstempel, GPS-Position, Aufnahmeeinstellungen etc.) und auch im Internet sind Bilder oft mit relevanten Schlagwörtern oder Dateinamen versehen und in der Regel in eine textuelle Beschreibung eingebettet, die einen semantischen Kontext des Bildes darstellt.

Für die eigentliche Anfrage an das Suchsystem wurden aber neben textuellen Suchbegriffen und Anfragebildern auch einige weitere Modalitäten untersucht, basierend z. B. auf der Farbanordnung des gesuchten Bildes [Wan11a], auf Skizzen oder Zeichnungen [Lia08, Fon09, Cao10b, Sou10, Cao11, Xia15], oder auf der Anordnung bestimmter Objektklassen [Xu10b, Xu10a, Lan12].

Der Fokus dieser Arbeit liegt auf der reinen Bildrepräsentation, d. h. die Suchanfrage stellt ein Bild dar und es werden hinsichtlich Semantik oder Metadaten keine weiteren Anforderungen an die zu durchsuchenden Bilder in der Datenbank gestellt. Mit Bildsuche ist im weiteren Verlauf daher stets die inhaltsbasierte Bildsuche gemeint, die keinerlei Metadaten erfordert.

Dieses Kapitel zum Stand der Forschung ist zweigeteilt: zunächst werden Verfahren vorgestellt, die auf lokalen Bildmerkmalen basieren. Sie indexieren entsprechende Bildinformationen für jedes lokale Merkmal von jedem

Bild. Solche Systeme erlauben es einerseits, auch kleine Objekte in Bildern zu finden, sind aber andererseits hinsichtlich der Datenbankgröße auf wenige Millionen Bilder beschränkt, denn damit eine schnelle Suche möglich ist, muss der Index im Arbeitsspeicher gehalten werden. Der zweite Teil des Kapitels beschreibt daher Verfahren, die eine globale und wesentlich kompaktere Darstellung für jedes Bild erzeugen, sodass zwar wesentlich mehr Bilder im Arbeitsspeicher Platz finden, aber Ähnlichkeiten nur noch gefunden werden können, wenn ein beträchtlicher Teil des Bildinhalts übereinstimmt.

2.1. Lokale Merkmale

Lokale Merkmale werden nicht aus dem gesamten Bild abgeleitet, sondern beschreiben nur einen örtlich begrenzten Bereich eines Bildes. Typischerweise werden lokale Merkmale für die Bildsuche in zwei Schritten berechnet:

1. **Finden von markanten Punkten** (*Keypoints*) im Bild: durch bestimmte Filterschritte werden charakteristische Positionen im Bild ermittelt, die in anderen Bildern mit hoher Wahrscheinlichkeit wiedergefunden werden können.

2. **Berechnung eines Deskriptors**: in diesem Schritt wird die lokale Bildumgebung jedes gefundenen markanten Punktes durch einen Deskriptorvektor in einem üblicherweise hochdimensionalen Vektorraum $\mathbb{R}^z$ beschrieben. Anhand der Abstände von Deskriptorvektoren im $\mathbb{R}^z$ lassen sich die jeweiligen lokalen Merkmale und damit die verschiedenen lokalen Bildstrukturen miteinander vergleichen.

Insbesondere die Einführung der SIFT Merkmale [Low99] sorgte für einen großen Durchbruch in der inhaltsbasierten Bildsuche, weshalb die Grundlagen im Folgenden kurz zusammengefasst werden.

Zur Berechnung von SIFT Merkmalen wird für den ersten Schritt (markante Punkte finden) zunächst der sog. Skalenraum erstellt, der sich durch Faltung des Bildes mit Gauß'schen Filterkernen unterschiedlicher Größen ergibt. Aus der Differenz zweier benachbarter Bilder im Skalenraum (Difference-of-Gaussian (DoG)) wird die skalennormierte Laplacian-of-Gaussian Funktion approximiert. Minima oder Maxima in den DoG-Bildern werden dann

als Kandidaten für markante Punkte ermittelt, wobei bei der Suche nicht nur die Achternachbarschaft im Pixelraster derselben Skala verglichen wird, sondern auch die jeweils neun Nachbarn der darüber und darunter liegenden Ebene im Skalenraum. Dies ermöglicht es, zu jedem markanten Punkt eine Skalierung zu erfassen, die angibt, wie grob die lokale Struktur ist, denn durch die wiederholten Faltungen verschwinden feine Bildstrukturen in den höheren Ebenen des Skalenraums zunehmend. Die Kandidaten durchlaufen anschließend noch weitere Filterschritte, um etwa Punkte mit zu geringem Kontrast oder Punkte, die auf Kanten liegen, zu verwerfen, da sich diese in anderen Bildern schlecht wiederfinden bzw. unzureichend exakt lokalisieren lassen.

Für jeden gefundenen markanten Punkt wird im zweiten Schritt ein Deskriptor berechnet. Dazu wird mittels eines Gradientenhistogramms die vorliegende lokale Hauptorientierung des Gradienten ermittelt und dann relativ zu dieser Hauptorientierung die Umgebung erfasst, deren Größe von der zuvor ermittelten Skalierung abhängt. Die Umgebung wird in 16 quadratische Bereiche unterteilt (4×4 Gitter) und in jedem Bereich wird ein Gradientenhistogramm mit acht Richtungs-Bins berechnet.

In die Bins werden die jeweiligen Gradienten eingetragen, gewichtet mit dem Gradientenbetrag sowie Gauß-gewichtet bezüglich des Abstandes zum markanten Punkt, also dem Mittelpunkt des Gitters. Durch Konkatenation der 16 einzelnen Histogramme mit jeweils 8 Bins ergibt sich der 128-dimensionale Deskriptor, der dann zwei Normierungsschritten unterzogen wird: um Beleuchtungseinflüsse zu minimieren wird er zunächst auf Einheitslänge normiert, danach wird jeder Wert im Deskriptor, der größer als 0,2 ist, auf 0,2 begrenzt und abschließend erfolgt eine erneute Normierung des Deskriptors auf die Länge 1.

Das Ergebnis der Berechnung der SIFT Merkmale eines Bildes stellt also eine Menge $\mathcal{F} = \{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n\}$ von lokalen Merkmalen dar, wobei jedes Merkmal $\boldsymbol{f}_i$ ein Tupel darstellt, bestehend aus der Position $\boldsymbol{x}_i = (x_i, y_i)^\top$, der zugehörigen Skala σ_i, der Hauptorientierung θ_i, sowie dem 128-dimensionalen Deskriptorvektor $\boldsymbol{d}_i \in \mathbb{R}^{128}$ besteht:

$$\boldsymbol{f}_i = (\boldsymbol{x}_i, \sigma_i, \theta_i, \boldsymbol{d}_i), \quad \boldsymbol{x}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}. \tag{2.1}$$

Die Menge $\mathcal{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n\}$, $\boldsymbol{d}_i \in \mathbb{R}^z$ bezeichnet im Folgenden die Deskriptoren der Merkmale $\mathcal{F}$ eines Bildes und z die Dimensionalität der Merkmalsdeskriptoren. Mittels SIFT Merkmalen ($z := 128$) lässt sich somit ein einfacher und für viele Anwendungen sehr effektiver Vergleich von zwei Bildern in Bezug auf gemeinsame ähnliche Objekte oder lokale Strukturen wie folgt realisieren:

1. Berechnung der SIFT Merkmale in jedem Bild: $\mathcal{F}^{(1)}, \mathcal{F}^{(2)}$
2. Suche nach *Korrespondenzen* $\mathcal{M}_\varepsilon$ (ähnlichen Merkmalen) indem die jeweiligen Deskriptoren beider Merkmalsmengen, $\mathcal{D}^{(1)}$ und $\mathcal{D}^{(2)}$, miteinander anhand des euklidischen Abstands verglichen werden. Da die Ähnlichkeit von Deskriptoren aber nicht nur vom Abstand im $\mathbb{R}^z, z = 128$, sondern auch von der „Belegungsdichte" im Merkmalsraum abhängt, hat sich anstelle eines globalen Schwellwerts ein Schwellwert ε bezüglich des Abstandsverhältnisses vom nächsten zum zweitnächsten Nachbar bewährt [Low04]:

$$\mathcal{M}_\varepsilon\left(\mathcal{F}^{(1)}, \mathcal{F}^{(2)}\right) = \left\{ \left(\boldsymbol{d}_i \in \mathcal{D}^{(1)}, \boldsymbol{d}_j, \boldsymbol{d}_{j'} \in \mathcal{D}^{(2)}\right) \middle| \frac{\left\|\boldsymbol{d}_i - \boldsymbol{d}_{j'}\right\|_2}{\left\|\boldsymbol{d}_i - \boldsymbol{d}_j\right\|_2} > \varepsilon \right\}, \quad (2.2)$$

   wobei $\boldsymbol{d}_j$ denjenigen Deskriptor aus $\mathcal{D}^{(2)}$ bezeichnet, der den kleinsten Abstand zu $\boldsymbol{d}_i$ aufweist und $\boldsymbol{d}_{j'} \in \mathcal{D}^{(2)}$ entsprechend denjenigen mit dem zweitkleinsten Abstand. Obwohl dieser Vergleich der Deskriptoren anhand des Abstandsverhältnisses auf die Einzigartigkeit von Merkmalen abzielt und somit in Bildern mit sich wiederholenden Strukturen nur eingeschränkt funktioniert, liefert er in der Praxis oft ausreichend gute Ergebnisse. Falls nicht abweichend angegeben, wird in dieser Arbeit $\varepsilon = 1{,}5$ verwendet. Abbildung 2.1a auf Seite 24 zeigt ein Beispiel, in dem die Korrespondenzen allein anhand der Deskriptoren ermittelt wurden.

Eine vorteilhafte Nachverarbeitung der SIFT Deskriptoren wurde in [Ara12] vorgeschlagen. Anstelle der euklidischen Distanz zwischen Deskriptorvektoren wird empfohlen, eine Distanz basierend auf dem Hellinger Kernel zu verwenden. Für zwei ℓ_1-normierte Vektoren $\boldsymbol{d}_i, \boldsymbol{d}_j \in \mathbb{R}^z, z = 128$ ist der

Hellinger Kernel $K_h(\boldsymbol{d}_i,\boldsymbol{d}_j) = \sum_{l=1}^{z} \sqrt{d_i^{(l)} d_j^{(l)}}$, wobei $d_i^{(l)}$ den Wert in der l-ten Dimension des Vektors $\boldsymbol{d}_i$ bezeichnet. Als Motivation wird angeführt, dass die SIFT Deskriptoren letztlich aus Histogrammen entstehen, und Histogramme üblicherweise mit dem Hellinger Kernel verglichen werden. Diese Umstellung wird in zwei einfachen Schritten erreicht: die bislang ℓ_2-normierten Vektoren werden ℓ_1-normiert und anschließend einem elementweisen Wurzelziehen unterzogen, was letztendlich den Einfluss von sehr großen Elementen im Verhältnis zu den übrigen Werten reduziert. Der resultierende Vektor ist wieder ℓ_2-normiert und wird als *RootSIFT* Deskriptor bezeichnet. Entscheidend dabei ist, dass die euklidische Distanz von RootSIFT Deskriptoren äquivalent zu den Hellinger Kernel basierten Distanzen der ursprünglichen Deskriptoren ist. Somit können alle bisherigen Verfahren, die intern die euklidische Distanz verwenden, jetzt – ohne Anpassung der Verfahren, sondern allein durch die RootSIFT Normalisierung der Daten – implizit den Hellinger Kernel verwenden. Im Bereich der Bildsuche führt diese Normalisierung zu signifikanten Verbesserungen und hat sich deshalb zum De-facto-Standard entwickelt.

Seit Einführung der SIFT Merkmale wurden außerdem unzählige Erweiterungen zur Integration von Farbinformationen vorgestellt (z. B. Color-SIFT [VDS10], HSV-SIFT [Bos08], HueSIFT [VdW06], OpponentSIFT [VDS10], CSIFT [AH06]). Für bestimmte Systeme und Daten mit definierten Aufnahmebedingungen lassen sich damit auch durchaus bessere Suchergebnisse erzielen. Aufgrund der oft extremen Beleuchtungsänderungen konnte sich aber keines der Verfahren für die inhaltsbasierte Bildsuche anwendungsübergreifend durchsetzen.

Ähnliches gilt für alternative lokale Merkmale wie SURF [Bay06], DAISY [Tol10], BRIEF [Cal10], CONGAS [Zho11a], ORB [Rub11], BRISK [Leu11], FREAK [Ala12], ALOHA [Sah12] oder LATCH [Lev16], die meist binäre Deskriptoren erzeugen und erhebliche Geschwindigkeitsvorteile bei der Merkmalsberechnung oder dem Vergleich von Desktriptoren[1] ermöglichen. Eine aktuelle Evaluation von binären Deskriptoren vor dem Hintergrund mobiler Anwendungen ist [Mad17] zu entnehmen. Hinsichtlich der Suchgenauigkeit

1 Die binären Deskriptoren werden dabei mit der Hamming Distanz verglichen, für die sowohl in Central Processing Unit (CPU) als auch in Graphics Processing Unit (GPU) Prozessoren spezielle Funktionen existieren (z. B. binäres XOR und bit count).

im Gesamtsystem konnte bislang jedoch keine der Alternativen eine signifikante Verbesserung erzielen und sich somit durchsetzen. Auch in dieser Arbeit sind daher im weiteren Verlauf mit lokalen Merkmalen stets SIFT Merkmale gemeint, die der RootSIFT Normalisierung unterzogen wurden.

2.2. Inhaltsbasierte Bildsuche mit lokalen Merkmalen

Prinzipiell lassen sich beliebige Bilder auf lokal ähnliche Bildinhalte untersuchen, indem zunächst die lokalen Merkmale berechnet und anschließend die Korrespondenzen $\mathcal{M}_\varepsilon$ ermittelt werden. Die typischerweise wenige tausend lokalen Merkmale eines Anfragebildes mit denen aller Datenbankbilder direkt zu vergleichen, ist aber per exakter Nächste-Nachbar-Suche im 128-dimensionalen Deskriptorraum aus Laufzeitgründen nur für wenige Hundert Bilder realisierbar. Außerdem sind durch die Größe des Arbeitsspeichers gewisse Grenzen gesetzt. Bei angenommenen 3 000 Merkmalen je Bild und 128 Byte pro Deskriptor bieten die aktuell üblichen 16 GiB Arbeitsspeicher eines PCs Platz für die Merkmale von ca. 50 000 Bildern.

Um den Bildvergleich mit lokalen Merkmalen auch für Bildermengen im Millionenbereich zu ermöglichen, sind daher unterschiedliche Verfahren entstanden, basierend auf Hashing, Quantisierung und Aggregierung, die im Folgenden vorgestellt werden.

2.2.1. Hashing

Im Bereich der approximativen Nächste-Nachbar-Suche wurden unzählige Hashing-Verfahren für die Bildsuche vorgestellt. Die Verfahren können grob in zwei Kategorien eingeteilt werden:

- **Datenunabhängige Hashing-Verfahren** wie z. B. Locality Sensitive Hashing (LSH) [Dat04], das einen Vektor mit einer Zufallsprojektion auf eine Dimension projiziert. Der resultierende Skalarwert wird – quantisiert gemäß einer Anzahl von Feldern – in eine Hashtabelle eingetragen mit dem Ziel, dass ähnliche Vektoren im Ursprungsraum nach der Projektion mit hoher Wahrscheinlichkeit in dieselben Feldern quantisiert werden und sich somit die gewünschte Hashkolli-

sion ergibt. Obwohl sich in der Theorie bei Nutzung mehrerer Zufallsprojektionen die gewünschte Kollisionswahrscheinlichkeit mit den Verfahrensparametern beliebig einstellen lässt, werden in der Praxis hunderte oder tausende Hashtabellen benötigt, um die für die Bildsuche erforderliche Trefferquote (d. h. wie viele der wirklichen nächsten Nachbarn werden gefunden) zu erhalten. Zum damit verbundenen Speicherplatzbedarf kommt erschwerend hinzu, dass in der Regel auch die originalen Vektoren im Arbeitsspeicher verfügbar sein müssen. Diese werden benötigt, um die exakten Distanzen aller bei der Kollision ermittelten Nächste-Nachbar-Kandidaten zu berechnen.

- **Datenabhängige Hashverfahren**, (auch Learning-to-hash Verfahren genannt) wie z. B. Spectral Hashing [Wei09] oder Linear Discriminant Analysis (LDA) Hashing [Str12] hingegen berücksichtigen die Verteilung der Daten im Ursprungsraum und übertreffen dadurch die LSH Verfahren. Eine aktuelle und umfassende Übersicht über die Learning-to-hash Verfahren bietet [Wan17]. Obwohl Speicherbedarf und Laufzeit immer weiter verbessert wurden, liefern auch diese Verfahren trotz beeindruckender Genauigkeit (d. h. wie oft stimmt der approximativ ermittelte nächste Nachbar mit dem echten nächsten Nachbar überein) für die Praxis immer noch eine eingeschränkte Trefferquote. In der Bildsuche kommen sie daher nur für die Fast-Duplikat-Suche zum Einsatz, also wenn davon ausgegangen werden kann, dass der überwiegende Teil der lokalen Merkmale zwischen den ähnlichen Bildern nahezu übereinstimmt und es daher zu verschmerzen ist, einen signifikanten Anteil dieser Korrespondenzen mit dem Hashverfahren nicht zu finden. Ein weiterer Einsatzbereich sind Systeme zur Gruppierung von Bildermengen, in denen es im ersten Schritt darauf ankommt, für die Gruppen jeweils in kürzester Zeit wenige initiale Bilder zu finden, deren Gruppen dann mit weiteren Verfahren vergrößert werden [Chu09, Gon15, Avr15].

2.2.2. Quantisierung und Bag-of-Words-Modell

Um ausgehend von den lokalen Merkmalen eine kompaktere Darstellung eines Bildes zu erreichen, stellt [Siv03] in Anlehnung an die Textsuche im Jahr 2003 das sogenannte *Bag-of-(visual)-Words (BoW)*-Modell vor, das als

wichtiger Meilenstein der inhaltsbasierte Bildsuche gilt. Hierbei wird mit einem Quantisierer q

$$\begin{aligned} q : \mathbb{R}^z &\to \{1,\dots,k\} \\ \boldsymbol{d}_i &\mapsto q(\boldsymbol{d}_i) \end{aligned} \tag{2.3}$$

jeder Merkmalsdeskriptor $\boldsymbol{d}_i \in \mathbb{R}^z$ quantisiert und ein Bild somit in eine textähnliche Darstellung umgewandelt, denn aus der Menge der lokalen Merkmale $\mathcal{F} = \{\boldsymbol{f}_1,\dots,\boldsymbol{f}_n\}$, $\boldsymbol{f}_i = (\boldsymbol{x}_i, \sigma_i, \theta_i, \boldsymbol{d}_i)$ wird nach der Quantisierung ihrer Deskriptoren $\mathcal{D} = \{\boldsymbol{d}_1,\dots,\boldsymbol{d}_n\}$ eine Ansammlung diskreter visueller Wörter $\{q(\boldsymbol{d}_1),\dots,q(\boldsymbol{d}_n)\}$. Das endliche Vokabular der visuellen Wörter $\mathcal{C} = \{\boldsymbol{c}_1,\dots,\boldsymbol{c}_k\}$, $\boldsymbol{c}_i \in \mathbb{R}^z$ wird dabei (visuelles) Codebook genannt und modelliert die in Bildern typischerweise vorkommenden lokalen Merkmale. Ein Codebook der Größe $k = |\mathcal{C}|$ wird in der Regel gelernt durch das Clustern einer großen Menge unabhängiger Merkmalsdeskriptoren in k Cluster. Nach dem Clustern ist der Deskriptorraum somit in k nichtüberlappende Bereiche (Voronoi-Zellen) unterteilt und in jedem Cluster wird ein Repräsentant – nämlich das zugehörige visuelle Wort – als Mittelwert der darin liegenden Deskriptoren berechnet. Für die Quantisierung der Merkmalsdeskriptoren eines Bildes wird dann jeder Deskriptor $\boldsymbol{d}_i$ mit dem Codebook abgeglichen, das heißt dem nächstgelegenen visuellen Wort zugeordnet:

$$q(\boldsymbol{d}_i) = \underset{j \in \{1,\dots,k\}}{\arg\min} \left\| \boldsymbol{d}_i - \boldsymbol{c}_j \right\|_2 . \tag{2.4}$$

Für zwei Deskriptoren $\boldsymbol{d}_i$ und $\boldsymbol{d}_j$, die im Deskriptorraum nahe beieinander liegen, soll dabei mit hoher Wahrscheinlichkeit $q(\boldsymbol{d}_i) = q(\boldsymbol{d}_j)$ gelten, d. h. beide sollen möglichst dem gleichen visuellen Wort zugeordnet werden.

Analog zu den Korrespondenzen $\mathcal{M}_\varepsilon$, die anhand der Abstände im Deskriptorraum berechnet wurden, werden im Folgenden mit $\mathcal{M}_\mathcal{C}$ aus Gleichung 2.2 die auf einem Codebook $\mathcal{C}$ basierenden Bag-of-Words-Korrespondenzen bezeichnet:

$$\mathcal{M}_\mathcal{C}\left(\mathcal{F}^{(1)}, \mathcal{F}^{(2)}\right) = \left\{ \left(\boldsymbol{d}_i \in \mathcal{D}^{(1)}, \boldsymbol{d}_j \in \mathcal{D}^{(2)}\right) \middle| q(\boldsymbol{d}_i) = q\left(\boldsymbol{d}_j\right) \right\} . \tag{2.5}$$

Der Bag-of-Words-Vektor $\boldsymbol{w} \in \mathbb{N}_0^k$,

$$\boldsymbol{w} = \left(w^{(1)}, \ldots, w^{(k)}\right)^\top, \quad w^{(i)} = \left|\left\{\boldsymbol{d}_j \in \mathcal{D} \,\middle|\, q\left(\boldsymbol{d}_j\right) = i\right\}\right| \tag{2.6}$$

eines Bildes mit den Deskriptoren $\mathcal{D}$, erfasst schließlich für jedes visuelle Wort die Anzahl der Vorkommen im Bild. Seine Länge ist somit nicht mehr von der Anzahl der lokalen Merkmale eines Bildes abhängig, sondern fest und entspricht der Größe k des Codebooks.

Drei wichtige Aspekte der BoW-basierten Quantisierung sind die Codebookgröße, das Clusteringverfahren sowie das Quantisierungsverfahren:

- **Größe des Codebooks** k: Obwohl die Anzahl der unterscheidbaren SIFT-Deskriptoren in der Theorie unvorstellbar groß ist[1], wiederholen sich die lokalen visuellen Strukturen in realen Bildern, sodass mit einem vergleichsweise kleinen Codebook die meisten der in der Praxis auftretenden Bilder hinreichend genau modelliert werden können. Ein in Anhang A.1 beschriebenes Experiment veranschaulicht diesen Sachverhalt. Gleichzeitig ist die sinnvolle Größe des Codebooks anwendungsspezifisch, denn für die Fast-Duplikat-Suche liegen die korrespondierenden Merkmale näher beieinander, sodass größere Codebooks möglich und sinnvoll sind. Bei der üblichen inhaltsbasierten Bildsuche hingegen liegen die Deskriptoren von korrespondierenden Merkmalen aufgrund der vielfältigen Transformationen, Beleuchtungsunterschiede etc. in der Regel im Deskriptorraum sehr viel weiter auseinander. Ein zu großes Codebook führt in diesem Fall dazu, dass die Deskriptoren korrespondierender Merkmale durch die Quantisierung nicht mehr in dieselben Voronoi-Zellen fallen und somit auf unterschiedliche visuelle Wörter abgebildet werden. In der Praxis werden daher je nach Anwendung typischerweise Codebookgrößen von 20 000 [Laz06, Per10b, Liu12] bis zu einer Million [Phi07, Jég08, Far13]

1 Üblicherweise werden die 128 Dimensionen mit je 8 Bit erfasst, sodass sich etwa 10^{308} unterscheidbare Deskriptoren ergeben. Zum Vergleich: bereits ein 8-Bit Graustufenbild der Größe 6 × 6 Pixel kann kombinatorisch $2^{8 \cdot 6 \cdot 6} \approx 10^{87}$ unterschiedliche Ausprägungen annehmen – also mehr als die von Astronomen auf 10^{80} geschätzte Anzahl der Atome im sichtbaren Universum.

verwendet und vereinzelt mehr (drei Million in [Zha13d] und 16 Million in [Mik10]).

- **Generierung des Codebooks**

  Bei kleinen Codebooks kann das Clustering mit dem k-Means Verfahren erfolgen, das für eine gewünschte Anzahl Cluster k alternierend die Clustermittelpunkte basierend auf den zu clusternden Daten $\check{\mathcal{D}}$ errechnet und anschließend die Daten wieder den neuen Clustern zuordnet [Llo82]. Da die Komplexität von k-Means $\mathcal{O}\left(k|\check{\mathcal{D}}|\right)$ beträgt, ist dieser Prozess in Bereichen von 1 Million Clustern und mehr als 10 Millionen 128-dimensionalen Datenpunkten allerdings zu rechenintensiv, weshalb eine hierarchische Variante vorgeschlagen wurde – Hierarchisches k-Means (HKM) – bei der die Daten zunächst in wenige Cluster unterteilt werden und jede Partition dann wiederum rekursiv in weitere Cluster unterteilt wird [Nis06]. Alternativ wurde ein Approximatives k-Means (AKM) vorgestellt, bei dem nur der Zuordnungsschritt durch approximative Verfahren wie randomisierte k-d-Bäume ausgetauscht wird [Phi07] und sich ebenso für HKM eine Komplexität von $\mathcal{O}\left(k\log(|\check{\mathcal{D}}|)\right)$ ergibt.

- **Deskriptorquantisierung** Für ein gegebenes Codebook wird bei der Deskriptorquantisierung dem Deskriptor jeden lokalen Merkmals eines Bildes das ähnlichste visuelle Wort des Codebooks zugeordnet. Bei einem Codebook, das durch hierarchisches Clustering [Nis06] erzeugt wurde, kann ein Deskriptor hierarchisch von der Baumwurzel beginnend entlang der jeweils ähnlichsten Knoten propagiert werden, sodass sich die Komplexität nicht als linear (Vergleich mit allen Blättern des Baumes), sondern als logarithmisch bezüglich der Codebookgröße ergibt. Bei großen Codebooks basierend auf AKM wird für die Quantisierung ebenfalls auf approximative Verfahren für die Nächste-Nachbar-Suche gesetzt. In [Low04] wird dabei auf einen modifizierten k-d-Baum zurückgegriffen. Seit [Muj14] haben sich jedoch randomisierte k-d-Bäume durchgesetzt für die hochdimensionale approximative Nächste-Nachbar-Suche. Der Spannungsbereich zwischen Speicherbedarf und Laufzeit, die im Allgemeinen unterschiedlich für den Offline-Schritt (Codebookgenerierung) und Online-Schritt (Quantisierung) sind, sowie Genauigkeit und Hardwarearchitekturen wird

aber noch durch viele weitere Verfahren ausgefüllt. In den letzten Jahren wurden außerdem bemerkenswerte Fortschritte bezüglich der Parallelisierung mit GPUs erzielt [Wak14, Tan15b, Wie16, Joh17].

Um Bilder basierend auf BoW-Vektoren zu vergleichen, werden diese in Anlehnung an die textbasierte Suche zunächst der sogenannten *Term Frequency - Inverse Document Frequency (TF-IDF)*-Gewichtung [Sal88] unterzogen, die den gewichteten BoW-Vektor $\boldsymbol{\omega} \in \mathbb{R}^k_{\geq 0}$ ergibt :

$$\boldsymbol{\omega} = \left(\omega^{(1)}, \ldots, \omega^{(k)}\right)^\top, \quad \omega^{(i)} = \text{TF-IDF}\left(w^{(i)}\right) := \underbrace{\frac{w^{(i)}}{n}}_{\text{TF}} \underbrace{\log \frac{N}{N^{(i)}+1}}_{\text{IDF}(i)}, \tag{2.7}$$

wobei w_i die Anzahl der Vorkommen des visuellen Wortes i im Bild, n die Anzahl aller Merkmale im Bild, N die Gesamtanzahl der Bilder in der Datenbank und $N^{(i)}$ die Anzahl derjenigen Datenbankbilder, die mindestens ein Vorkommen des visuellen Wortes i aufweisen, bezeichnen.

Die Gewichtung setzt sich somit aus dem Produkt der folgenden beiden Komponenten zusammen:

- Term Frequency (TF): misst die relative Häufigkeit der visuellen Wörter,
- Inverse Document Frequency (IDF): reduziert den Einfluss von visuellen Wörtern, die häufig in der Datenbank vorkommen, und daher für die Unterscheidung der Bilder weniger hilfreich erscheinen, als seltene visuelle Wörter. Die IDF stellt somit eine bildübergreifende Normalisierung dar. Die Addition von eins im Nenner verhindert die Division durch Null, falls ein visuelles Wort in keinem der Datenbankbilder vorkommt.

Für die Suche mit einem Anfragebild I_q wird die Ähnlichkeit $s(I_q, I_d)$ zu einem Datenbankbild I_d anhand des Skalarproduktes der zugehörigen ℓ_2-normierten gewichteten Bag-of-Words-Vektoren berechnet:

$$s\left(I_q, I_d\right) := \frac{\boldsymbol{\omega}_q^\top \boldsymbol{\omega}_d}{\|\boldsymbol{\omega}_q\| \|\boldsymbol{\omega}_d\|}. \tag{2.8}$$

Dies entspricht der Kosinus-Ähnlichkeit, die den Kosinus des von den beiden Vektoren eingeschlossenen Winkels angibt. Da die ℓ_2-Normierung für alle Datenbankbilder im Vorfeld durchgeführt werden kann und die für das Anfragebild ad-hoc, vereinfacht sich Gleichung 2.8 zum Skalarprodukt. Darüber hinaus ist $\|\boldsymbol{\omega}_q\|$ für alle Datenbankbilder identisch und somit für die Ähnlichkeiten lediglich ein konstanter, vom Anfragebild anhängiger Faktor, der bei der Berechnung vernachlässigt werden kann, sofern es nur auf die Reihenfolge der Ergebnisliste ankommt. Das ist oft schon allein deshalb der Fall, weil die vordersten Bilder der Trefferliste in einem zweiten Schritt noch einem genaueren Vergleich mit dem Anfragebild unterzogen werden, und die finale Ähnlichkeit dann ohnehin auf anderen Metriken basiert – etwa der Anzahl lokaler Merkmalskorrespondenzen, die unter einer gefundenen geometrischen Transformation plausibel erscheinen, siehe „Re-Ranking“ im Abschnitt 2.2.4.

In der Regel enthalten Bilder nur wenige tausend lokale Merkmale, gleichzeitig wird aber mit Codebookgrößen im Bereich von 10^6 gearbeitet, d. h. es gilt $k \gg n$. Die hochdimensionalen gewichteten BoW-Vektoren $\boldsymbol{\omega}$ sind daher sehr dünn besetzt, sodass nur sehr wenige Dimensionen im Skalarprodukt $\boldsymbol{\omega}_q^\top \boldsymbol{\omega}_d$ überhaupt einen von 0 abweichenden Beitrag liefern – nämlich die, deren zugehörige visuelle Wörter jeweils in beiden zu vergleichenden Bildern enthalten sind. Diese Tatsache lässt sich mit einem Index, basierend auf dem sogenannten *Inverted File Prinzip* [Wit99], ausnutzen. Statt für jedes Bild die darin vorkommenden visuellen Wörter zu speichern, wird bei dieser Indexart umgekehrt verfahren: für jedes visuelle Wort $\boldsymbol{c}_i$ des Codebooks wird eine Liste $\mathcal{Q}_i$ erstellt, in der diejenigen Datenbankbilder erfasst werden, die mindestens ein Vorkommen des visuellen Wortes aufweisen:

$$i \mapsto \mathcal{Q}_i = \left\{ \left(j, \omega_j^{(i)} \right) \,\middle|\, j \in \{1, \ldots, N\}, \omega_j^{(i)} > 0 \right\}, \tag{2.9}$$

wobei $\omega_j^{(i)}$ das i-te Element des BoW-Vektors $\boldsymbol{\omega}_j$ von Bild j der Datenbank bezeichnet. Im Index werden also Tupel $(j, \omega_j^{(i)})$ gespeichert, bestehend aus der Bildnummer j und dem jeweiligen, das visuelle Wort $\boldsymbol{c}_i$ betreffenden Teil $\omega_j^{(i)}$ des BoW-Vektors. Die Durchführung einer Suchanfrage kann damit als Abstimmungsprozess mit einem Akkumulator $A \in \mathbb{R}^N$ interpretiert werden, bei dem für jedes Datenbankbild $I_j, j \in \{1, \ldots, N\}$ alle Stimmen in Form der relevanten Beiträge $\omega_q^{(i)} \omega_j^{(i)}$ für sein Skalarprodukt in $A_q(j)$ aufsummiert

werden. Dazu wird für jedes, im Anfragebild vorkommende visuelle Wort i die zugehörige inverse Liste $\mathcal{Q}_i$ traversiert und jedes darin enthaltene Tupel erhöht den Akkumulator des entsprechenden Bildes gemäß

$$A_q(j) = \sum_{i=1}^{k} \sum_{l=1}^{|\mathcal{Q}_i|} \delta_{jj'} \omega_q^{(i)} \omega_{j'}^{(i)} \quad , \tag{2.10}$$

wobei $(j', \omega_{j'}^{(i)})$ das l-te Tupel aus der Liste $\mathcal{Q}_i$ bezeichnet und $\delta_{jj'}$ die Kronecker-Delta Notation bezeichnet, die den Wert eins annimmt, falls $j = j'$ gilt und null anderenfalls.

In der bisherigen Formulierung tragen mehrere, demselben visuellen Wort zugeordnete Merkmale eines Bildes jeweils mit identischen Beiträgen zur Ähnlichkeit bei. In der Regel möchte man aber die einzelnen Merkmale eines visuellen Wortes unterschiedlich stark gewichten – beispielsweise in Abhängigkeit der approximierten Deskriptordistanz wie bei der in Abschnitt 2.2.3 beschriebenen Technik des Hamming Embeddings. Der Abstimmungsprozess und die inversen Listen im Index können dazu entsprechend umgestaltet werden [Jég08], sodass die Merkmale eines Datenbankbildes *einzeln* indexiert werden, und nicht mehr aggregiert durch die visuellen Wörter. Der Abstimmungsprozess, also das Sammeln der Beiträge für das Skalarprodukt $\boldsymbol{\omega}_q^\top \boldsymbol{\omega}_d$, erfolgt dann nicht mehr durch Iteration über die relevanten Dimensionen des BoW-Vektors des Anfragebildes, sondern über seine lokalen Merkmale. Für die Bildähnlichkeit aus Gleichung 2.8 ergibt sich unter Verwendung der Gleichungen 2.6 und 2.7 :

$$s^\star(I_q, I_d) := \frac{\boldsymbol{\omega}_q^\top \boldsymbol{\omega}_d}{\|\boldsymbol{\omega}_q\| \|\boldsymbol{\omega}_d\|} = \frac{\sum_{i=1}^{k} \frac{w_q^{(i)}}{n_q} \mathrm{IDF}(i) \frac{w_d^{(i)}}{n_d} \mathrm{IDF}(i)}{\|\boldsymbol{\omega}_q\| \|\boldsymbol{\omega}_d\|} \tag{2.11}$$

$$= \frac{\frac{1}{n_q n_d} \sum_{j=1}^{n_q} \sum_{j'=1}^{n_d} \mathrm{IDF}\left(q(\boldsymbol{d}_j)\right)^2 \delta_{q(\boldsymbol{d}_j) q(\boldsymbol{d}_{j'})}}{\|\boldsymbol{\omega}_q\| \|\boldsymbol{\omega}_d\|} , \tag{2.12}$$

wobei $\mathrm{IDF}(i)$ für den IDF-Term aus Gleichung 2.7 steht und * im Weiteren die Symbole für die Indexierung auf Basis der Merkmale (anstelle der visuellen Wörter) kennzeichnet. Mit n_q und n_d sind die Anzahl der Merkmale und mit $\boldsymbol{d}_j$ und $\boldsymbol{d}_{j'}$ die jeweiligen Deskriptoren im Anfragebild I_q bzw. im Datenbankbild I_d gemeint. Die effiziente Ermittlung derjenigen Daten-

bankmerkmale, für die $q(\boldsymbol{d}_j) = q(\boldsymbol{d}_{j'})$ gilt, wird auch hier wieder mit dem Inverted File Prinzip realisiert. In den Listen $\mathcal{Q}_i$ aus Gleichung 2.9 müssen für die einzelnen Merkmale nunmehr lediglich die Bildnummern gespeichert werden – zuzüglich etwaiger weiterer Informationen, die pro Merkmal im Index gespeichert werden sollen:

$$i \mapsto \mathcal{Q}_i^\star = \left\{ j \;\middle|\; \left(j, j'\right) \in \{1,\ldots,N\} \times \left\{1,\ldots,n_j\right\}, q\left(\boldsymbol{d}_{j'}\right) = i \right\} . \tag{2.13}$$

Dabei bezeichnet n_j die Anzahl der Merkmale des Datenbankbildes I_j und $\boldsymbol{d}_{j'}$, $j' \in \{1,\ldots,n_j\}$ seine Merkmalsdeskriptoren. Beim Traversieren der Listen[1] $\mathcal{Q}_i^\star$ im Rahmen einer Suchanfrage ergibt sich der Akkumulator des jeweiligen Datenbankbildes I_j verglichen mit Gleichung 2.10 nun folgendermaßen:

$$A_q^\star(j) = \sum_{i=1}^{n_q} \sum_{l=1}^{\left|\mathcal{Q}_{q(\boldsymbol{d}_i)}^\star\right|} \delta_{jj'} \mathrm{IDF}\left(q\left(\boldsymbol{d}_i\right)\right)^2 , \tag{2.14}$$

wobei j' den Wert des l-ten Eintrages aus der Liste $\mathcal{Q}_{q(\boldsymbol{d}_i)}^\star$ bezeichnet. Abschließend erfolgt die Normalisierung gemäß Gleichung 2.12, wobei auch hier die Komponenten für das Anfragebild, also $\frac{1}{n_q}$ und $\frac{1}{\|\boldsymbol{\omega}_q\|}$, wieder vernachlässigt werden können, wenn allein die Reihenfolge der gefundenen Datenbankbilder von Interesse ist.

2.2.3. Erweiterungen bezüglich der Quantisierung

Die Quantisierung der Merkmalsdeskriptoren im Bag-of-Words-Modell ermöglicht es einerseits, die Ähnlichkeit zweier quantisierter Deskriptoren mit einer simplen Vergleichsoperation zu bewerten. Auf der anderen Seite entstehen durch die Codebook-basierte Einteilung des Deskriptorraums in die Voronoi-Zellen immer dann Fehler, wenn Deskriptoren nahe an den Zellengrenzen liegen und gleichzeitig die korrespondierenden Deskriptoren durch verschiedene Einflüsse wie Beleuchtungsänderungen, perspektivi-

1 Gemäß obiger Beschreibungen sind $\mathcal{Q}_i^\star$ Multimengen, d. h. Elemente können mehrfach auftreten und die Reihenfolge der Elemente ist beliebig. Prinzipiell ist aber auch eine Sortierung der Elemente denkbar, sodass in dieser Arbeit weiterhin von Listen gesprochen wird.

sche Verzerrungen etc. einer gewissen Streuung unterliegen, sodass manche davon in benachbarte Voronoi-Zellen fallen obwohl sie dieselbe reale Struktur im Bild beschreiben. Abbildung 2.1 veranschaulicht dies anhand eines Bildpaares, das dasselbe Objekt aus einer ähnlichen Ansicht zeigt, allerdings in sehr unterschiedlichen und herausfordernden[1] Beleuchtungssituationen. Im linken Bild wurden 2 386 und im rechten Bild 1 820 lokale Merkmale extrahiert. In Abbildung 2.1a wird der Vergleich mittels der unquantisierten Originaldeskriptoren im $\mathbb{R}^{128}$ durchgeführt und die Korrespondenzen $\mathcal{M}_\varepsilon$ ergeben sich anhand des Abstandsverhältnisses von ähnlichstem und zweit-ähnlichstem Deskriptor (Gleichung 2.2). Nach der Deskriptorquantisierung basieren die Korrespondenzen $\mathcal{M}_\mathcal{C}$ in Abbildung 2.1b bzw. 2.1c allein auf den visuellen Wörtern eines Codebooks der Größe 10^5 bzw. 10^6. In Grün sind dabei diejenigen Korrespondenzen eingefärbt, die auch gemäß Abbildung 2.1a als Korrespondenzen betrachtet werden (d. h. $\mathcal{M}_\mathcal{C} \cap \mathcal{M}_\varepsilon$). Gemeinsam mit den rot eingefärbten Korrespondenzen wird deutlich, dass zum einen durch die Quantisierung der Merkmalsdeskriptoren viele falsche Korrespondenzen auf Basis der visuellen Wörter entstehen und andererseits viele der ursprünglich korrekten Korrespondenzen im Zuge der Quantisierung „verloren gehen“. Die Schwierigkeit der Bildsuche liegt daher in der Anzahl der Korrespondenzen, die in Abbildung 2.1 jeweils rechts aufgeführt ist: Wenn lediglich 1% der visuellen Wörter eines Anfragebildes mit denen eines relevanten Datenbankbildes übereinstimmen, steigt mit zunehmender Datenbankgröße das Risiko, dass in der Datenbank ein nichtrelevantes Bild existiert, welches durch die Quantisierungsverluste mehr übereinstimmende visuelle Wörter aufweist und dadurch die Suchergebnisse beeinträchtigt.

1 Die oft gelesene Aussage, SIFT Merkmale seinen invariant gegenüber Beleuchtungsänderungen, ist mit Vorsicht zu genießen: zwar sind die Deskriptoren invariant gegenüber additiven Helligkeitsänderungen (da sie auf Gradienten basieren) und linearen Kontraständerungen (da die Deskriptoren normiert werden), nicht jedoch gegenüber nichtlinearer Kontraständerungen. Eine weitaus größere Herausforderung beruht allerdings auf einem anderen Aspekt der Beleuchtung: Wenn das abgebildete Objekt keine flache Oberfläche, sondern dreidimensionale Strukturen aufweist, entstehen auf dem Objekt in der Regel Schattenwürfe, die zu Gradienten im Bild führen. Bei Beleuchtungsänderungen kann sich deren Stärke und Lage verschieben, was in den Gradientenhistogrammen schnell zu signifikanten Änderungen im hochdimensionalen Deskriptorraum führt. In der Folge können dann wie in Abbildung 2.1 viele der – für den Betrachter einfach erscheinenden – lokalen Ähnlichkeiten nicht mehr als Korrespondenzen im Deskriptorraum ermittelt werden.

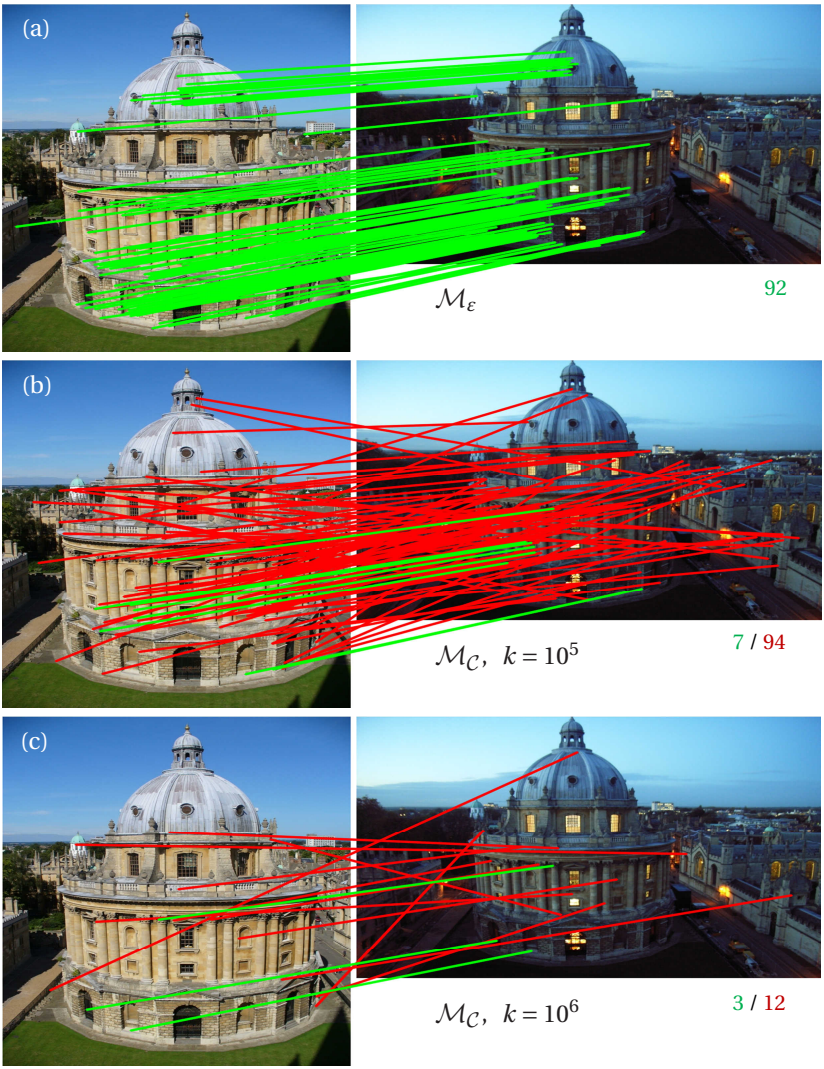

Abbildung 2.1.: Vergleich eines Bildpaares desselben Objekts anhand von Korrespondenzen ihrer originalen (a) und quantisierten lokalen Merkmale (b+c). Beide Bilder stammen aus dem OXFORD5K Datensatz [Phi07].

Die Größe des Codebooks stellt diesbezüglich einen relevanten Designparameter dar, der aber in beiden Richtungen (bezüglich Deskriptorvariationen und Quantisierungseffekten) gleichzeitig gewissen Einschränkungen unterliegt:

- **Kleinere Codebooks**: Für kleine Codebooks liegen korrespondierende Deskriptoren zwar auch dann in derselben Voronoi-Zelle, wenn sie starken Transformationen unterliegen. Weil die einzelnen Voronoi-Zellen größer sind, liegen aber auch vermehrt Deskriptoren darin, die sich nicht auf dieselbe reale Struktur in den Bildern beziehen, sodass die Genauigkeit der Suche darunter leidet. Im BoW-Modell schwindet bei kleiner werdenden Codebooks zudem der Effizienzvorteil des Inverted File Prinzips, denn die einzelnen Listen $\mathcal{Q}_i$ bezüglich der visuellen Wörter werden länger, sodass während einer Suchanfrage mehr Merkmale verarbeitet werden müssen.

- **Größere Codebooks**: Die kleineren Voronoi-Zellen sorgen dafür, dass die Deskriptoren näher an den Grenzen liegen und nur noch Deskriptoren sehr ähnlicher Korrespondenzen in dieselbe Voronoi-Zelle fallen. Dadurch können in anspruchsvollen Ähnlichkeitsszenarien nicht mehr genügend Korrespondenzen mit dem BoW-Modell gefunden werden, wodurch die Trefferquote der Suche leidet. Die Suche ist somit nur noch zum Finden von Fast-Duplikaten geeignet.

Um dem Dilemma der besten Codebookgröße etwas entgegenzusetzen, haben sich – auf Kosten des Speicherbedarfs und der Laufzeit – zwei grundlegende Techniken in der BoW-basierten Bildsuche etabliert:

- **Soft Quantization (SQ)** : Um für große Codebooks nach der Quantisierung auch diejenigen korrespondierenden Deskriptoren im Index zu erreichen, die in benachbarten Voronoi-Zellen liegen, ist die Idee von Soft Quantization [Phi08a], dass jeder Deskriptor nicht nur dem ähnlichsten visuellen Wort zugeordnet wird, sondern den ähnlichsten ξ visuellen Wörtern. Oft wird $\xi = 3$ gewählt und die Zuordnung eines Deskriptors zum visuellen Wort wird außerdem einer Gauß'schen Gewichtung bezüglich des Abstandes des Deskriptors zum Clustermittelpunkt des visuellen Wortes unterzogen. Der Speicherbedarf im Index erhöht sich somit um den Faktor ξ. Falls SQ für Datenbank- und

Anfragebilder angewandt wird, steigt die Laufzeit um den Faktor ξ^2, denn die durchschnittliche Länge der zu traversierenden Listen $\mathcal{Q}_i$ steigt um den Faktor ξ und gleichzeitig müssen für jeden Deskriptor im Anfragebild seine ξ zugeordneten visuellen Wörter berücksichtigt werden. Oft wird SQ deshalb nur für das Anfragebild verwendet, sodass sich nur moderate Auswirkungen (Faktor ξ) auf die Laufzeit ergeben.

Ein verwandter Ansatz, der visuelle Wörter nicht nur anhand von Abständen im Deskriptorraum zuordnet, wird in [Mik10] vorgestellt. Ausgehend von einem sehr großen Codebook (16 Mio. visuelle Wörter) wird in über 5 Mio. Bildern nach geometrisch verifizierten Korrespondenzpfaden über jeweils mehrere Bilder desselben Objekts hinweg gesucht, aus denen dann die probabilistischen Beziehungen der visuellen Wörter untereinander abgeleitet werden. In ähnlicher Weise sammelt [Mak10] Merkmal-Tracks aus bezüglich der Aufnahmeposition benachbarten Bilder von Google Street View. Als sogenannte alternative Wörter werden dann jeweils diejenigen visuellen Wörter zusammengefasst, die sich im Trainingsbildmaterial auf dieselbe gemeinsame reale Struktur beziehen. Dieses aufwändig gelernte Vokabular der alternativen Wörter ist somit diskriminanter, da sich die Ähnlichkeit eher semantisch ergibt als durch bloße Entfernungen im Deskriptorraum. Noch einen Schritt weiter gehen [Phi10, Sim12], die bereits vor der Generierung eines Codebooks versuchen, einen Deskriptor zu lernen, der sich besser für den Vergleich per euklidischer Distanz eignet.

- **Hamming Embedding (HE)** : Diese Technik [Jég08] widmet sich kleinen Codebooks mit typischerweise wenigen zehn- oder hunderttausend visuellen Wörtern und folglich größeren Voronoi-Zellen. Das Ziel von HE ist, einen quantisierten Deskriptor $q(\boldsymbol{d})$ mit einer Binärsignatur $\boldsymbol{b} \in \mathbb{B}^{\hat{z}}$ zu ergänzen, die die genauere Lage des ursprünglichen Deskriptors $\boldsymbol{d}$ innerhalb der Voronoi-Zelle angibt. Zwei Deskriptoren werden anschließend nur dann als ähnlich betrachtet, wenn sie nicht nur auf dasselbe visuelle Wort quantisiert wurden, sondern ein Vergleich ihrer Signaturen ergibt, dass sie auch innerhalb der gemeinsamen Voronoi-Zelle in unmittelbarer Nähe zueinander liegen. Die

binäre Signatur der Länge $\grave{z}$ ist daher so entworfen, dass für zwei Signaturen $\boldsymbol{b}_i = (b_i^{(1)},\ldots,b_i^{(\grave{z})})$ und $\boldsymbol{b}_{i'} = (b_{i'}^{(1)},\ldots,b_{i'}^{(\grave{z})})$ ihre Hamming-Distanz

$$h(\boldsymbol{b}_i,\boldsymbol{b}_{i'}) = \sum_{l=1}^{\grave{z}} \left| b_i^{(l)} - b_{i'}^{(l)} \right|, \qquad h(\cdot) \in \{0,\ldots,\grave{z}\} \tag{2.15}$$

mit hoher Wahrscheinlichkeit klein ist, wenn sie im ursprünglichen Deskriptorraum nahe beieinander liegen. Dies wird durch ein Modell erreicht, das zunächst in einem Offline-Schritt mit einer unabhängigen Menge von repräsentativen Deskriptoren $\check{\mathcal{D}}$ wie folgt gelernt wird [Jég10a]:

1. Eine orthogonale Projektionsmatrix $P \in \mathbb{R}^{z\times\grave{z}}$ wird erzeugt, indem eine Matrix mit Zufallswerten[1] einer Gauß'schen Normalverteilung erstellt wird, anschließend einer QR-Zerlegung unterzogen wird und von der Ergebnismatrix die ersten $\grave{z}$ Zeilen extrahiert werden.
2. Alle Deskriptoren $\boldsymbol{d}_i \in \check{\mathcal{D}}$ werden mittels P in den $\grave{z}$-dimensionalen Raum projiziert: $\grave{\boldsymbol{d}}_i = P\boldsymbol{d}_i$
3. Für jedes visuelle Wort $\boldsymbol{c}_j$ wird bezüglich jeder Dimension $l = (1,\ldots,\grave{z})$ der Median $m_j^{(l)}$ der Menge $\left\{ \grave{d}_i^{(l)} \middle| q(\boldsymbol{d}_i) = j \right\}$ bestimmt, also von allen projizierten Deskriptoren, deren ursprünglicher Deskriptor auf das visuelle Wort $\boldsymbol{c}_j$ abgebildet wurde. Neben der Projektionsmatrix P ergibt sich als Modell somit pro visuellem Wort ein $\grave{z}$-dimensionaler Vektor $\boldsymbol{m}_j$, der die Medianwerte bezüglich jeder Dimension enthält und auf eine sehr grobe Art die typische Lage der projizierten Deskriptoren in dieser Voronoi-Zelle repräsentiert.

1 Wie [Jég10a] bereits anmerkt, stellt sich die Frage, warum anstelle der Zufallsprojektion nicht eine Transformation auf Basis einer Principle Component Analysis (PCA) gewählt wird. Ein Test der Autoren ergab, dass solch eine globale PCA Projektionsmatrix keine nennenswerten Vorteile bringt. Statt einer globalen Projektionsmatrix hingegen für jedes visuelle Wort eine separate Projektionsmatrix zu erstellen, führt zu einem sehr hohem Speicherbedarf und würde vor allem ein Vielfaches der Trainingsdatenmenge benötigen (im Vergleich zur vorgeschlagenen Bestimmung der Medianwerte, die bereits mit wenigen (< 50) Deskriptoren pro Voronoi-Zelle möglich ist).

Mit dem gelernten Modell $(P, \{\boldsymbol{m}_j\}_{j=1}^{c})$ wird die binäre Signatur $\boldsymbol{b}$ eines Deskriptors $\boldsymbol{d}$ folgendermaßen bestimmt:

1. $\boldsymbol{d}$ wird mittels P von $\mathbb{R}^z$ nach $\mathbb{R}^{\grave{z}}$ projiziert: $\grave{\boldsymbol{d}} = P\boldsymbol{d}$
2. $\grave{\boldsymbol{d}}$ wird bezüglich jeder Dimension $l = (1, \ldots, \grave{z})$ mit dem zum visuellen Wort $q(\boldsymbol{d})$ gehörenden Medianvektor verglichen, wobei der jeweilige gelernte Medianwert $m^{(l)}_{q(\boldsymbol{d})}$ als Schwellwert für die Binarisierung in jeder Dimension dient:

$$b^{(l)} = \begin{cases} 1, & \text{falls } \grave{d}^{(l)} > m^{(l)}_{(q(\boldsymbol{d}))} \\ 0, & \text{sonst.} \end{cases}$$

Üblicherweise wird $\grave{z} = 128$ gewählt, sodass jeder Deskriptor mit einer 128 Bit Signatur erweitert wird. Da diese auch im Index gespeichert werden muss, erhöht sich der Speicherbedarf also signifikant um zusätzliche 16 Byte pro indexiertem Merkmal. Während einer Suchanfrage werden dann nur solche Einträge aus den inversen Listen beim Abstimmungsprozess berücksichtigt, deren Signatur $\boldsymbol{b}_i$ eine gewisse Hamming Distanz τ zur Signatur des jeweiligen Deskriptors im Anfragebild $\boldsymbol{b}_j$ nicht überschreitet $(h(\boldsymbol{b}_i, \boldsymbol{b}_j) < \tau)$. Bewährt hat sich außerdem eine zusätzliche Gewichtung der Einträge in den Akkumulator, die von der Hamming Distanz abhängt [Jég10a]. Da diese nur $\grave{z} + 1$ verschiedene Werte annehmen kann, ist neben der Distanzberechung auch die Gewichtung mit einer Lookup-Tabelle effizient zu implementieren.

2.2.4. Ergänzende Module für die Bildsuche

Alternativ zu den genannten Methoden, die direkt den Informationsverlust des BoW-Modells durch dessen Quantisierung adressieren, haben sich einige grundlegende Bausteine in der inhaltsbasierten Bildsuche bewährt, die im Folgenden kurz vorgestellt werden. Sie können die Suchergebnisse für gewisse Anwendungsszenarien verbessern, sind aber prinzipiell unabhängig vom BoW-Modell als Bildrepräsentation.

- **Re-Ranking** zielt darauf ab, die Ergebnisse einer Suchanfrage zu verbessern, indem die vorderen η Einträge der Ergebnisliste, also die

gefundenen η ähnlichsten Bilder, in einem zweiten Schritt noch einmal mit aufwändigeren und exakteren Verfahren mit dem Anfragebild verglichen werden. Aus Laufzeitgründen wird der Vergleich in der Regel nur für wenige hundert Bilder durchgeführt. Die meisten Ansätze versuchen, basierend auf den Merkmalskorrespondenzen, eine geometrische Transformation des gemeinsam abgebildeten Objekts oder der Szene zwischen den Bildern zu schätzen. Die untersuchten Bilder werden dann neu sortiert gemäß der Anzahl der plausiblen Korrespondenzen (Inlier), die sich für die jeweils ermittelte Transformation ergeben, sodass der Einfluss von Ausreißern begrenzt wird. Die Transformation kann beispielsweise durch das RANSAC Verfahren [Fis81] gesucht werden [Phi07, Chu05, Xie11b, Per09]. Dabei werden aus zufällig ausgewählten Korrespondenzen entsprechende Transformationshypothesen erstellt und diese anschließend anhand der Anzahl an Inliern bewertet, die sich unter der jeweiligen Hypothese ergeben. Falls die optimale Transformation – bezogen auf die Anzahl der Korrespondenzen – allerdings nur wenige Inlier aufweist, müssen oft sehr viele Hypothesen aufgestellt und verifiziert werden, um eine ausreichend gute Transformation zu finden. Um die quadratische Laufzeit von RANSAC bezogen auf die Anzahl der Korrespondenzen zu vermeiden, schränkt [Phi07] die Hypothesen daher auf fünf Freiheitsgrade ein und erzeugt approximative Hypothesen [Chu04] aus jeweils einer einzigen Merkmalskorrespondenz. Dadurch können sämtliche Hypothesen eines Bildpaares verifiziert werden und das Re-Ranking wird dadurch deterministisch. [Lin10] wählt, inspiriert durch die Branch-and-Bound-Verfahren in der Objektdetektion [Lam09], einen lokalen Ansatz für das Re-Ranking und versucht, in jedem der ersten η ähnlichsten Datenbankbilder diejenige rechteckige Teilregion zu ermitteln, deren BoW-Darstellung die größte Ähnlichkeit mit dem Anfragebild aufweist. [She12] führt mit jedem der η gefundenen Datenbankbilder dagegen eine neue Suchanfrage durch und argumentiert, dass die tatsächlich ähnlichen Datenbankbilder in vielen der η Ergebnislisten in den vorderen Rängen auftauchen müssten, nicht so aber die im ersten Suchschritt gefundenen nicht-relevanten Bilder. Diese Variante des Re-Rankings kann allerdings nur eingesetzt werden, wenn davon ausgegangen werden kann, dass es mehrere relevante Da-

tenbankbilder gibt. Und ganz allgemein kann ein Datenbankbild im Suchprozess vom Re-Ranking nur dann profitieren, wenn es im ersten Schritt der Suche bereits als eines der ähnlichsten η Ergebnisbilder gefunden wurde.

- **Query Expansion:** Inspiriert durch die Textsuche werden hierbei in einem Nachverarbeitungsschritt die ähnlichsten gefundenen und durch Re-Ranking geometrisch verifizierten Bilder verwendet, um daraus eine erneute Suchanfrage zu erstellen. [Chu07b, Chu11] schlägt dazu vor, die BoW-Vektoren der geometrisch verifizierten Bilder zusammen mit dem BoW-Vektor des Anfragebildes zu mitteln und durch erneute Suchanfragen inkrementell ein Modell der Umgebung des Anfrageobjekts zu lernen. [Ara12] hingegen trainiert jeweils eine Support Vektor Maschine (SVM) mit den ersten (geometrisch verifizierten) und letzten Bildern der initialen Ergebnisliste und sortiert alle Ergebnisbilder anschließend basierend auf dem jeweiligen Abstand des BoW-Vektors zur Trennhyperebene der SVM. Obwohl die Strategien, um aus den Ergebnissen der ersten Suchanfrage weitere Suchanfragen zu erstellen, immer weiter verfeinert wurden [Qin11, She12, Tol14a], ist Query Expansion systembedingt nur anwendbar, wenn für ein Anfragebild mehrere Trefferbilder in der Datenbank existieren. Die Ähnlichkeit der Datenbankbilder untereinander kann aber nicht nur während einer Anfrage genutzt werden, sondern auch bereits zum Zeitpunkt der Indexierung, indem etwa ein Ähnlichkeitsgraph erstellt wird [Tur09, Phi08b] oder die Geoposition berücksichtigt wird [Sch07, Tor11].

- **Burstiness:** Als *visual burstiness* wird das Phänomen bezeichnet, dass in manchen Bildern bestimmte lokale Strukturen öfter auftreten, als es die im BoW-Modell angenommene statistische Unabhängigkeit der visuellen Wörter vermuten lässt. Dadurch werden die BoW-Vektoren von den sehr häufigen visuellen Wörtern (beispielsweise bei den Fenstern eines Hochhauses) dominiert und letztendlich die Ergebnisse der Bildsuche beeinträchtigt. In [Jég09a, Shi15] werden dazu verschiedene Strategien vorgestellt, die den Einfluss der betreffenden visuellen Wörter innerhalb eines Bildes und über mehrere Bilder hinweg begrenzen. [Zhe13b] hingegen schlägt mit der $\mathcal{L}_p$-Norm eine Anpassung der IDF vor, die nicht nur die Anzahl der Datenbankbilder für jedes visuelle

Wort, sondern auch das Burstiness Phänomen berücksichtigt. [Tor13] und [Dou10] zeigen wiederum, dass lokal repititive Bildstrukturen keineswegs nur als Störung im BoW-Modell betrachtet werden können, sondern sich auch als wichtiges Unterscheidungsmerkmal für Gebäudefassaden und andere, sich wiederholende bildliche Strukturen eignen. Die TF-IDF-Gewichtung wird allerdings nicht nur von sehr häufig vorkommenden visuellen Wörtern beeinflusst, sondern auch durch häufig gemeinsam auftretende visuelle Wörter, was [Chu10] und [Cum08] in die Gewichtung einfließen lassen. [Jég12a] modelliert sogar den Informationsgehalt von gemeinsam *nicht* auftretenden visuellen Wörtern, indem vom BoW-Vektor ein kleiner Teil des über alle Bilder gemittelten BoW-Vektors abgezogen wird, sodass gemeinsam fehlende visuelle Wörter im Skalarprodukt dennoch einen Beitrag zur Ähnlichkeit liefern.

Abgesehen von den Erweiterungen zur Quantisierung und den beschriebenen ergänzenden Modulen, ist außerdem versucht worden, mehr Informationen über die lokalen Merkmale im Index zu berücksichtigen. Die verschiedenen Strategien dazu werden als nächstes vorgestellt.

2.2.5. Erweiterung des Index

Hinsichtlich der Skalierungsfähigkeit des BoW-Modells, also der Frage, wie sehr diese Art der Komprimierung von Bildern im Index deren Auffindbarkeit in großen Datenbanken beeinträchtigt, sind zwei Aspekte entscheidend:

1. **Ressourcenbedarf**: Die heute übliche PC-basierte Hardware ermöglicht auf einem Einzelsystem Datenbankgrößen von wenigen Millionen Bildern, wobei sowohl die Anfragedauer als auch die Speicheranforderungen linear mit der Anzahl der Merkmale aller Datenbankbilder wachsen. Auch wenn verschiedene Verfahren existieren, einen auf dem Inverted File Prinzip beruhenden Index zu komprimieren [Zha08, Zob06] und den Speicherbedarf um etwa den Faktor vier zu reduzieren [Jég09b, Per09], stellt der verfügbare Arbeitsspeicher die wichtigste Einschränkung dar.

2. **Diskriminanz**: Da die Merkmale im BoW-Modell nur anhand ihres quantisierten Deskriptors indexiert werden, steigt mit jedem, zur Da-

tenbank hinzukommenden Bild die Gefahr der Verwechslung. Das bedeutet, dass in sehr großen Datenbanken für ein Anfragebild oft entsprechende Datenbankbilder existieren, die zwar einen ähnlichen BoW-Vektor aufweisen, aber die ähnlichen Merkmale in einer gänzlich anderen geometrischen Anordnung vorkommen.

Während beim ersten Punkt der technische Fortschritt regelmäßig etwas Entgegenkommen verspricht[1], ist bezüglich des zweiten Punktes auf verschiedene Arten versucht worden, mehr Informationen über die Merkmale in den Index zu integrieren. Diese Ansätze können in die drei grundlegenden Strategien *Akkumulatorerweiterung*, *Filterung* und *Dimensionserweiterung* eingeteilt werden, die im Folgenden anhand der relevantesten Verfahren beschrieben werden.

2.2.5.1. Akkumulatorerweiterung

Bei dieser Strategie zur Indexerweiterung werden für jedes Merkmal zusätzliche Informationen im Index mitgespeichert, und während einer Suchanfrage wird mit einem erweiterten Akkumulator gearbeitet, der mehr als nur *ein* Feld für jedes Bild aufweist. Die Idee ist, dass diejenigen Stimmen, die von plausiblen Merkmalen stammen, sich dann in einem Teil der Felder konzentrieren und dadurch für einen großen Ähnlichkeitswert sorgen. Unplausible Merkmale hingegen sollen über alle Felder des Akkumulator eines Bildes verteilt streuen, sodass die Ähnlichkeit bezogen auf den maximalen Wert im Akkumulator abgeschwächt wird.

Als grundsteinlegend für die Integration von Merkmalsgeometrie in den Index gelten die Arbeiten bezüglich *Weak Geometric Consistency (WGC)* [Jég08, Jég10a]. Das Verfahren beruht auf der Idee, dass die meisten korrespondierenden Merkmale $(\boldsymbol{f}_i, \boldsymbol{f}_j)$ eines dasselbe Objekt zeigenden Bildpaares jeweils eine ähnliche Orientierungsdifferenz $\theta_i - \theta_j$ aufweisen müssten, die in etwa dem globalen Winkel entspricht, um den das Objekt zwischen den beiden Bildern gedreht erscheint. Gleiches gilt für die Skalierungsun-

1 Gemeint ist hier die übliche Größe des Arbeitsspeichers eines einzelnen Rechners; in dieser Arbeit nicht näher betrachtet werden dagegen Systeme, die aus mehreren Rechnern bestehen wie z. B. [Aly11] mit 2 000 Computern für 100 Millionen Bilder oder [Ste12] mit einer nicht genannten Anzahl an Rechnerknoten für 94 Milliarden äußerst grob indexierte Bilder.

terschiede $\log(\frac{\sigma_i}{\sigma_j})$ der Merkmale – auch hier ist zu erwarten, dass die korrespondierenden Merkmale eines doppelt so groß abgebildeten Objektes in etwa die doppelte Skalierung aufweisen. Bei nicht korrespondierenden Bildern hingegen ist davon auszugehen, dass sich die Orientierungsdifferenzen der inkorrekten Korrespondenzen nicht um einen Wert häufen, sondern alle möglichen Werte $[0, 2\pi)$ annehmen, da sie größtenteils auf den Quantisierungseffekten und nicht auf einem gemeinsamen Objekt beruhen. Diese „schwache Konsistenz der Geometrie“ des WGC Ansatzes kann im Abstimmungsprozess dadurch berücksichtigt werden, dass der Akkumulator $A \in \mathbb{R}^N$ nicht mehr nur *ein* Feld für jedes der N Datenbankbilder bereithält, sondern mehrere – nämlich eines für jede mögliche, in a_θ Intervalle eingeteilte Winkeldifferenz, sodass sich die plausiblen Korrespondenzen in möglichst wenigen der a_θ Intervalle häufen. Gleiches gilt auch hier wieder für die Skalierungsunterschiede, die in a_σ Intervalle eingeteilt werden. Da die Unterschiede bezüglich Orientierungen und Skalierungen aber als annähernd statistisch unabhängig voneinander betrachtet werden können[1], muss der Akkumulator nicht pro Bild zweidimensional modelliert werden ($A \in \mathbb{R}^{N \times a_\theta \times a_\sigma}$), sondern man arbeitet mit zwei Akkumulatoren $A_\theta \in \mathbb{R}^{N \times a_\theta}$ und $A_\sigma \in \mathbb{R}^{N \times a_\sigma}$, die separat befüllt und ausgewertet werden. Nachdem alle Stimmen eingetragen sind, wird eine gleitende Mittelung in beiden Akkumulatoren durchgeführt, um die Quantisierungseffekte zu minimieren. Anschließend werden in beiden Akkumulatoren die Maxima bestimmt, und pro Bild wird jeweils der kleinere der beiden Werte für die weitere Ähnlichkeitsbewertung verwendet. Im Index muss für die Nutzung von WGC zu jedem Merkmal seine Orientierung und Skalierung mitgespeichert werden. In [Jég10a] werden die Orientierungen mit 6 Bit und die Skalierungen mit 5 Bit quantisiert und $a_\theta = a_\sigma = 128$ gewählt.

Um im Vergleich mit WGC eine striktere Anordnung der visuellen Wörter zu verarbeiten, schlägt [Zha11b] die sogenannten *Geometry-Preserving Visual Phrase (GVP)* vor. Eine GVP der Länge k_{gvp} wird dabei definiert als eine visuelle Phrase bestehend aus k_{gvp} visuellen Wörtern, die in einer bestimm-

1 Das heißt veranschaulicht etwa, dass Objekte, die aus größerer Distanz fotografiert werden, nicht automatisch mit einer anderen Kameraorientierung aufgenommen werden – auch wenn dies für manche Objekte vereinzelt zutreffen dürfte, etwa für Kirchtürme.

ten räumlichen Anordnung auftreten. Unterschiedliche beteiligte visuelle Wörter oder unterschiedliche Anordnungen resultieren in unterschiedlichen GVPs. Als Anordnung wird dabei allein die relative Lage der visuellen Wörter zueinander bezüglich beider Bildkoordinaten betrachtet. Um eine gewisse Toleranz gegenüber lokalen Deformationen zu ermöglichen, wird die Lage in jeweils zehn Intervalle in X- und Y-Richtung quantisiert. Mittels des Verfahrens von [Zha09c] kann für ein Bildpaar die Menge der in beiden Bildern gemeinsam auftretenden GVPs ermittelt werden und für jedes Bild kann prinzipiell – ähnlich dem BoW-Modell – eine Vektordarstellung im Raum aller möglichen GVPs erstellt werden. Dass dieser Vektor kombinatorisch allerdings nie direkt aufgestellt werden kann, ist dabei unproblematisch, denn ein Skalarprodukt zweier GVP Vektoren ergibt sich als die Gesamtanzahl der in beiden Bildern gemeinsam auftretenden GVPs [Zha11b]. Um die Beziehungen der visuellen Wörter untereinander im Rahmen der GVPs in der Suche zu berücksichtigen, werden zum einen die Merkmale im Index um die jeweilige Position im Bild (7 Bit für die Koordinate im 10×10 Raster) ergänzt. Zum anderen wird der Akkumulator für die Suchabfrage erweitert, sodass jedes Datenbankbild nicht mehr nur durch ein Feld repräsentiert wird, sondern durch den sogenannten Offset-Space bestehend aus 100 Feldern – also ein Feld für jede mögliche relative Verschiebung der visuellen Wörter der GVPs. Für jedes visuelle Wort im Anfragebild werden anschließend anhand des Index die weiteren Vorkommen in den Datenbankbildern ermittelt und gemäß der Verschiebung zwischen der Position im Anfragebild und der im jeweiligen Datenbankbild wird das zugehörige Feld des Bildes inkrementiert. Die Gesamtanzahl der gemeinsam auftretenden GVPs für ein Datenbankbild wird anschließend berechnet, indem alle Felder des Bildes traversiert werden und die verschiedenen Möglichkeiten aufaddiert werden. Ein Feld, das m_{gvp} mal inkrementiert wurde, resultiert dabei kombinatorisch in $\binom{m_{gvp}}{k_{gvp}}$ gemeinsam auftretenden GVPs der Länge k_{gvp}. Aus der Gesamtanzahl ergibt sich nach einem Normierungsschritt und mit an das GVP-Konzept angepassten IDF Gewichtungen schließlich der Ähnlichkeitswert jedes Bildes. Für GVPs der Länge $k_{gvp} = 2$ wurden dabei die besten Werte erzielt, wobei die größte Einschränkung darin besteht, dass der Offset-Space in dieser Form lediglich translationsinvariant ist.

[She12] erweitert den Gedanken daher und erzielt durch weitere zwei Dimensionen im Akkumulator für Skalierung und Rotation die entspre-

chende Invarianzeigenschaften. Die gewählten Intervalle ($16 \times X$, $16 \times Y$, 8×Skalierung, 6×Orientierung) führen allerdings zu 16384 Felder pro Datenbankbild, die für jede Suchanfrage und pro Datenbankbild initialisiert, befüllt und ausgewertet werden müssen. Es wird zwar ein Trade-off vorgeschlagen, der z. B. für jede der acht vorgesehenen möglichen Skalierungen einmal den Index traversiert und dadurch bei achtfacher Laufzeit nur 2048 Feldern pro Datenbankbild benötigt, aber für große Datenbanken stößt auch dieser Kompromiss schnell an Speicher- und/oder Laufzeitgrenzen.

2.2.5.2. Filterung von Merkmalen

Bei dieser zweiten Strategie zur Indexerweiterung werden ebenfalls für jedes Merkmal zusätzliche Informationen im Index mitgespeichert. Während einer Suchanfrage wird aber nicht mit einem erweiterten Akkumulator gearbeitet, sondern weiterhin mit nur einem Feld pro Datenbankbild. Stattdessen werden die zusätzlichen Informationen der jeweiligen Merkmale explizit auf Plausibilität überprüft und nur im Erfolgsfall resultiert dies in einer Erhöhung des Akkumulatorwertes.

[Wu09a] bündelt beispielsweise mehrere SIFT Merkmale in Gruppen, und argumentiert, dass diese diskriminanter sind als die einzelnen Merkmale. Die Gruppen werden gebildet, indem zunächst markante Regionen im Bild mit dem Maximally Stable Extremal Regions (MSER) Detektor [Mat04] berechnet werden. Verglichen mit dem Detektor von SIFT liefert der MSER Detektor typischerweise wenige, elliptische Regionen. Alle SIFT Merkmale, die innerhalb einer elliptischen MSER Region liegen, werden in einer Gruppe zusammengefasst. SIFT Merkmale können dabei zu mehreren Gruppen gehören, da sich die MSER Regionen überlagern können, oder auch in gar keine Gruppe fallen, wobei Gruppen verworfen werden, deren elliptische Region mehr als die halbe Bildbreite oder -höhe einnimmt. Der Vergleich von Bildern erfolgt anhand der ermittelten Gruppen. Die Ähnlichkeit von zwei Gruppen basiert wiederum auf zwei Komponenten: der Anzahl der gemeinsamen visuellen Wörter und der geometrischen Plausibilität der beiden SIFT Merkmalsmengen. Sie beruht auf einer relativen Anordnung der BoW Korrespondenzen bezüglich der Bildkoordinatenachsen. Um diese gruppenbasierte Ähnlichkeit mit dem Inverted File Prinzip zu realisieren, werden für jedes indexierte Merkmal die Zugehörigkeit zu den Gruppen sowie seine re-

lative Anordnung innerhalb der jeweiligen Gruppe im Index mitgespeichert. Während einer Suchanfrage müssen diese Informationen aufwändig verarbeitet werden, um die jeweiligen Gruppenähnlichkeiten zu berechnen. Für Fast-Duplikat-Datensätze konnte mit diesen gebündelten Merkmalen eine signifikante Verbesserung erzielt werden, aber für anspruchsvollere Ähnlichkeitsszenarien ergeben sich mehrere Probleme: bei kleinen Gruppen existieren in den enthaltenen Merkmalsmengen zu wenige BoW-Korrespondenzen, da die benachbarten Merkmale eines korrespondierenden Merkmals nur selten auch Korrespondenzen aufweisen. Bei großen Gruppen hingegen steigen Speicherbedarf und Laufzeit signifikant. Die MSER Regionen korrespondieren außerdem nicht zwangsweise auch mit den Objektkonturen, und Merkmale, die in keiner MSER Region liegen, können keinen Beitrag zur Ähnlichkeit liefern.

Neben dieser speziellen Bündelung von Merkmalen mittels MSER Regionen lässt sich die Filterungsstrategie aber auch für beliebige Merkmalskombinationen umsetzen, etwa für die oben genannten visuellen Phrasen [Zha11b]. Zwischen visuellen Phrasen in der obigen Definition entsteht aber nur dann eine Korrespondenz, wenn sowohl die Anzahl der beteiligten visuellen Wörter identisch ist, als auch alle visuellen Wörter übereinstimmen. In der Praxis ergeben sich daher oft nur sehr wenige Korrespondenzen von höherwertigen Phrasen, denn mit jedem zusätzlichen visuellen Wort akkumulieren sich die Quantisierungsfehler. [Zha13b] stellt daher die *Multi-Order Visual Phrase (MVP)* vor. Für jedes lokale Merkmal $\boldsymbol{f}_i$ werden dabei die vier räumlich nächsten Merkmale mit ähnlicher Skalierung ermittelt. Für jedes dieser vier Merkmale werden zwei Charakteristiken berechnet und als Umgebungsinformation von $\boldsymbol{f}_i$ im Index mitgespeichert: einerseits die grobe relative Lage zu $\boldsymbol{f}_i$ anhand der relativen Distanz und des Unterschieds der Merkmalsorientierungen und andererseits der grob quantisierter Deskriptor. Die grobe Quantisierung der vier Merkmalsdeskriptoren erfolgt mit einem Codebook der Größe 256 und die relative Entfernung und die Orientierungsunterschiede werden in jeweils 16 Stufen quantisiert, sodass pro Merkmal – neben den 32 Bit für die Bildnummer – insgesamt weitere $4 \cdot (8 + 4 + 4) = 64$ Bit für seine Umgebungsinformation im Index benötigt werden. Während der Suche wird für jedes Merkmalspaar, das demselben visuellen Wort zugeordnet wurde, zusätzlich überprüft, wie viele der vier

Nachbarmerkmale übereinstimmen. Falls keines davon übereinstimmt, handelt es sich um eine gewöhnliche BoW-Korrespondenz, falls $k_{mvp} = 1, \ldots, 4$ der Nachbarmerkmale übereinstimmen, um eine MVP der Ordnung k_{mvp}. Die übliche TF-Gewichtung wird schließlich durch den Term $(1 + \alpha_{mvp})^{k_{mvp}}$ mit $\alpha_{mvp} = 0{,}4$ ersetzt, um höherwertige MVPs stärker zu gewichten.

Abgesehen von der Verdreifachung des Speicherbedarfs für den Index bringen die MVPs eine signifikant höhere Rechenkomplexität mit sich. Um die jeweils vier Nachbarn zu vergleichen, sind bis zu 16 Vergleiche notwendig, und für eine einzelne Übereinstimmung müssen sowohl die visuellen Wörter bezüglich des groben Codebooks übereinstimmen, als auch die quantisierten Distanzen und Orientierungen bezüglich eines Schwellwerts überprüft werden. Außerdem sind in der Praxis, bedingt durch verschiedene Transformationen, selbst die räumlich vier nächsten Nachbarn eines Merkmals nur selten auch alle im korrespondierenden Bild vorhanden.

Ein während der Suche effizienterer Vergleich von Merkmalsumgebungen wird in [Liu14b] vorgestellt. Die Umgebung wird dort in drei Winkelbereiche eingeteilt. In jedem der drei 120-Grad-Fächer werden die enthaltenen Merkmale dadurch repräsentiert, dass deren Merkmalsdeskriptoren aufsummiert werden. Dabei wird eine entfernungsabhängige Gewichtung bei der Summierung verwendet, um die Deskriptoren räumlich naher (in Bildkoordinaten) Merkmale stärker zu gewichten als die von entfernten Merkmalen. Die gewichteten Deskriptormittelwerte der drei Fächer werden anschließend konkateniert, ℓ_2-normiert, auf einen 64-dimensionalen Binärcode komprimiert und für jedes Merkmal im Index mitgespeichert. Während der Suche werden die Binärcodes der Merkmale per Hamming Distanz verglichen.

[Cao10a] stellt eine *spatial-bag-of-features* genannte Bildrepräsentation vor, bei der ein Bild als eine Menge von Histogrammen betrachtet wird, wobei jedes Histogramm die räumliche Reihenfolge der visuellen Wörter unter bestimmten linearen oder kreisförmigen Projektionen angibt. Durch Zerlegung, Normalisierung und Umsortieren der Histogrammbins wird versucht, die Invarianz zu erreichen. Aus 420 parametrisierten Histogrammvarianten werden per RankBoost [Fre03] durch überwachtes Lernen auf einem Datensatz die acht relevantesten Projektionen ermittelt und per Inverse File

Prinzip indexiert.

2.2.5.3. Dimensionserweiterung

Im Gegensatz zu den bisherigen zwei Strategien zur Indexerweiterung speichert diese dritte Strategie die zusätzlichen Informationen für die Merkmale nicht zusammen mit den quantisierten Merkmalen – also als Teil der Tupel in den inversen Listen $\mathcal{Q}_i^\star$ aus Gleichung 2.13 – ab. Die zusätzlichen Informationen werden vielmehr als eigenständiges Merkmal aufgefasst und mittels eines separaten Codebooks der Größe $\tilde{k}$ quantisiert. Dies induziert im Index eine zweite Dimension, denn alle Merkmale werden nicht mehr nur anhand ihres visuellen Wortes (BoW-Dimension) in den inversen Listen $\mathcal{Q}_i^\star$ organisiert, sondern bezüglich beider Dimensionen:

$$\begin{aligned}\left(i,\tilde{i}\right) \mapsto \mathcal{Q}_{i,\tilde{i}}^\star = \\ \left\{ j \;\middle|\; \left(j,j'\right) \in \{1,\dots,N\} \times \left\{1,\dots,n_j\right\}, \right. \\ \left. q\left(\boldsymbol{d}_{j'}\right) = i \wedge \tilde{q}\left(\boldsymbol{d}_{j'}\right) = \tilde{i} \right\},\end{aligned} \tag{2.16}$$

wobei sich alle Symbole mit Tilde auf die zusätzliche Indexdimension beziehen. Da die einzelnen inversen Listen $\mathcal{Q}_{i,\tilde{i}}^\star$ somit – verglichen mit den inversen Listen $\mathcal{Q}_i^\star$ des BoW-Modells – im Mittel um den Faktor $\tilde{k}$ kürzer sind, werden entsprechend weniger Speicherzugriffe auf den 2D-Index benötigt.

[Zhe14a] integriert auf diese Weise Farbinformationen in den Index und verwendet dafür den ColorName (CN) Deskriptor [Kha12]. Dieser codiert den Farbwert eines Pixels durch einen 11-dimensionalen Vektor $\boldsymbol{s} \in \mathbb{R}^{11}$, dessen einzelne Dimensionen die Farbanteile für Schwarz, Blau, Braun, Grau, Grün, Orange, Pink, Violett, Rot, Weiß und Gelb repräsentieren sollen. Aus der Umgebung eines lokalen Merkmals, deren Größe proportional zu seiner Skalierung ist, werden die CN Deskriptoren für alle Pixel berechnet und durch Mittelwertbildung zusammengefasst. Die so erhaltene Farbrepräsentation der Merkmalsumgebung wird als eigenständiges Merkmal aufgefasst und mittels eines separaten Codebooks in „Farbworte“ quantisiert. [Zhe14a] wählt allerdings eine Codebookgröße von lediglich $\tilde{k} = 200$ Farbwörtern, um eine ausreichende Robustheit gegenüber Beleuchtungsunterschieden

zu erreichen und setzt außerdem sowohl Hamming Embedding als auch Soft Quantization (siehe Kapitel 2.2.3) ein, um Trefferquote und Genauigkeit auszutarieren. Für HE wird dazu aus dem 11-dimensionalen CN-Vektor mit einem in [Zho12] vorgestellten Binarisierungsschema ein 22-Bit-Vektor erzeugt, der für jedes lokale Merkmal im Index mitgespeichert und während der Suche per Hamming Distanz verglichen wird. Bezüglich SQ wird jedes Merkmal im Anfragebild nicht nur seinem nächsten der möglichen 200 Farbwörtern zugeordnet, sondern seinen 100 ähnlichsten Farbwörtern, sodass für jedes Merkmal sozusagen die Hälfte des Farb-Codebooks benötigt wird. Diese exzessive Soft Quantization führt den Vorteil einer zweiten diskriminanten Dimension im Index ein Stück weit ad absurdum, denn dadurch kann lediglich die Hälfte aller Merkmale des Index direkt ausgeschlossen werden. Auch die Tatsache, dass die andere Hälfte anschließend noch mittels der HE Signaturen gefiltert werden muss, macht deutlich, dass die in [Zhe14a] vorgestellte Weise, die Merkmalsumgebung durch CN Deskriptoren zu repräsentieren, nur eine sehr eingeschränkte Diskriminanz aufweist und vermutlich deshalb auch vorwiegend auf Fast-Duplikat-Datensätzen evaluiert wurde.

Diese Dissertation untersucht deshalb alternative, diskriminativere Repräsentationen, um die Merkmalsumgebung im Rahmen eines 2D-Index zu verwenden. Statt des CN Deskriptors wird dabei auf die leistungsfähigen Verfahren der globalen Bildbeschreibung zurückgegriffen, die im folgenden zweiten Teil dieses Kapitels vorgestellt werden.

2.3. Inhaltsbasierte Bildsuche mit globalen Merkmalen

Auf globalen Merkmalen basierende Verfahren zur Bildsuche versuchen, ein Bild in eine möglichst kompakte Repräsentation von oft nur wenigen hundert Byte zu verdichten, damit die Ähnlichkeiten des Anfragebildes zu jedem der Datenbankbilder möglichst schnell berechnet werden können.

Die ersten Verfahren in den frühen Neunzigern orientierten sich an der globalen Beschreibung von Bildern mittels Texturen, Farben oder Kanteninformationen, wobei der GIST Deskriptor [Oli01] lange Zeit die gebräuch-

lichste globale Darstellung von Bildern war. Eine umfassende Darstellung der Suche mit globalen Merkmalen bieten fünf Übersichtsarbeiten [Rui99, Sme00, Lew06, Liu07, Dat08]. Globale Merkmale führen zwar zu sehr kompakten Bildrepräsentationen, die sehr große Bilddatenbanken ermöglichen. Allerdings sorgt die fehlende Lokalität dafür, dass Objekte in relevanten Bildern nicht mehr gefunden werden können, falls sie nur einen kleinen Teil des Bildes einnehmen oder starken Beleuchtungsunterschieden, Verdeckungen oder einem deutlich abweichenden Hintergrund ausgesetzt sind. Damit eignen sie sich hauptsächlich zur Suche von Fast-Duplikaten bzw. sehr ähnlichen Bildern. Da es andererseits mit zunehmender Datenbankgröße auch wahrscheinlicher wird, zu einem Anfragebild ähnliche Bilder zu finden[1], werden globale Merkmale oft auch nur für Teilsysteme verwendet. [Li06] sucht etwa in einem ersten Schritt in 2,4 Millionen Bildern aus dem Internet nach ähnlichen Bildern und ermittelt anschließend aus den jeweiligen ursprünglichen Textumgebungen der Bilder eine möglichst repräsentative textuelle Annotierung durch Schlüsselworte. [Tor08a] nutzt 80 Millionen automatisch annotierte Bilder der Größe 32×32 Pixel, um für ein beliebiges Anfragebild die k nächsten Nachbarn zu bestimmen und daraus unter anderem Objekte im Anfragebild zu prognostizieren und zu klassifizieren. [Wan10b] schließlich nutzt zwei Milliarden Bilder, um Schlagworte für ein Bild zu generieren.

Obwohl solche Systeme mit einfachen globalen Merkmalen zur Suche von Fast-Duplikaten nur wenige Anwendungsbereiche abdecken können, stellen sie derzeit immer noch die einzige Möglichkeit dar, derart immense Bildmengen überhaupt durchsuchbar zu machen.

In der allgemeinen inhaltsbasierten Bildsuche haben sich dagegen in den letzten Jahren zwei komplexere globale Repräsentationen bewährt, die im Folgenden vorgestellt werden: Fisher Vektoren, die lokale Merkmale aggregieren sowie Repräsentationen basierend auf neuronalen Netzen.

1 Die Untersuchungen von [Wan10b] mit zwei Milliarden Bildern aus dem Internet ergaben beispielsweise, dass ca. 22% der Bilder Fast-Duplikate aufweisen und 8% sogar mehr als zehn Fast-Duplikate.

2.3.1. Aggregation lokaler Merkmale durch FV und VLAD

Während der BoW-Vektor die lokalen Merkmale eines Bildes lediglich anhand der Häufigkeiten der visuellen Wörter in einen Vektor konstanter Größe überführt, wurde – zunächst für die Bildklassifizierung – von [Per07] mit dem **Fisher Vektor** eine umfassendere Aggregierung vorgestellt, die mehr Details zur Verteilung der Merkmale im Deskriptorraum berücksichtigt und auf dem Fisher Kernel [Jaa99] basiert. Die Deskriptoren $\mathcal{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n\}$, $\boldsymbol{d}_i \in \mathbb{R}^z$ werden dabei als Stichprobe betrachtet, die durch ein generatives probabilistisches Modell erzeugt wurde, das wiederum durch eine Wahrscheinlichkeitsdichtefunktion $p(\boldsymbol{d}|\boldsymbol{\beta})$ mit den Parametern $\boldsymbol{\beta} \in \mathbb{R}^B$ gegeben ist. Die Stichprobe $\mathcal{D}$ kann dann durch den Gradienten der Log-Likelihood-Funktion der Stichprobe bezüglich der Parameter betrachtet werden:

$$G_{\boldsymbol{\beta}}^{\mathcal{D}} = \nabla_{\boldsymbol{\beta}} \log p\left(\mathcal{D}|\boldsymbol{\beta}\right) . \tag{2.17}$$

Anschaulich wird die Stichprobe somit anhand der Richtung beschrieben, in die der Parametervektor verändert werden müsste, um besser auf die Stichprobe zu passen. Da $G_{\boldsymbol{\beta}}^{\mathcal{D}} \in \mathbb{R}^B$, ist diese Beschreibung der Stichprobe nur noch von der Anzahl B der Parameter abhängig und nicht mehr von der Größe n der Stichprobe. Um basierend auf dieser Beschreibung zwei Stichproben $\mathcal{D}_1$ und $\mathcal{D}_2$ miteinander zu vergleichen, schlägt [Jaa99] den Fisher Kernel vor:

$$\Upsilon(\mathcal{D}_1, \mathcal{D}_2) = G_{\boldsymbol{\beta}}^{\mathcal{D}_1} \Psi_{\boldsymbol{\beta}}^{-1} G_{\boldsymbol{\beta}}^{\mathcal{D}_2}, \tag{2.18}$$

wobei $\Psi_{\boldsymbol{\beta}} \in \mathbb{R}^{B \times B}$ die sog. Fisher Informationsmatrix darstellt:

$$\Psi_{\boldsymbol{\beta}} = \mathbb{E}_{\boldsymbol{d} \sim p}\left[G_{\boldsymbol{\beta}}^{\boldsymbol{d}} G_{\boldsymbol{\beta}}^{\boldsymbol{d}\top} \right]. \tag{2.19}$$

Da durch die positiv semi-definite Matrix $\Psi_{\boldsymbol{\beta}}$ eine Cholesky-Zerlegung von $\Psi_{\boldsymbol{\beta}}^{-1} = L_{\boldsymbol{\beta}}^{\top} L_{\boldsymbol{\beta}}$ ermöglicht wird, kann der Fisher Kernel als Skalarprodukt der normalisierten Gradientenvektoren dargestellt werden:

$$\Upsilon(\mathcal{D}_1, \mathcal{D}_2) = \boldsymbol{\Phi}(\mathcal{D}_1)^{\top} \boldsymbol{\Phi}(\mathcal{D}_2) \tag{2.20}$$

mit

$$\mathbf{\Phi}(\mathcal{D}) = L_{\boldsymbol{\beta}} G_{\boldsymbol{\beta}}^{\mathcal{D}} = L_{\boldsymbol{\beta}} \nabla_{\boldsymbol{\beta}} \log p\left(\mathcal{D}|\boldsymbol{\beta}\right) . \tag{2.21}$$

Die *Fisher Vektor* genannte Repräsentation $\mathbf{\Phi}(\mathcal{D})$ besitzt die Dimensionalität B und aggregiert somit eine beliebige Anzahl an Merkmalen in einen Vektor einer fester Länge, die der Anzahl der Parameter des generativen Modells entspricht.

Nachdem im Kontext der Bildrepräsentation die Menge der Deskriptoren der lokalen Merkmale eines Bildes die Stichprobe darstellt, kann das zugehörige generative Modell als probabilistisches visuelles Codebook interpretiert werden. Üblicherweise wird als Modell ein Gauß'sches Mischmodell (GMM) verwendet

$$p\left(\boldsymbol{d}|\boldsymbol{\beta}\right) = \sum_{\check{k}=1}^{K} \alpha_{\check{k}} p_{\check{k}}\left(\boldsymbol{d}|\boldsymbol{\beta}\right) = \sum_{\check{k}=1}^{K} \alpha_{\check{k}} \mathcal{N}\left(\boldsymbol{d}|\, \boldsymbol{\mu}_{\check{k}}, \Sigma_{\check{k}}\right), \qquad \boldsymbol{d}, \boldsymbol{\mu}_{\check{k}} \in \mathbb{R}^z, \tag{2.22}$$

mit den Parametern $\boldsymbol{\beta} = \left\{\alpha_{\check{k}}, \boldsymbol{\mu}_{\check{k}}, \Sigma_{\check{k}}\right\}_{\check{k}=1,\ldots,K}$. Diese beinhalten die Gewichtung $\alpha_{\check{k}}$, den Mittelwert $\boldsymbol{\mu}_{\check{k}}$ und eine Kovarianzmatrix $\Sigma_{\check{k}}$ für jede der K Komponenten. Alle Gewichte sind positiv und summieren sich zu 1. Jede Komponente der Mischverteilung ist eine multivariate Gaußfunktion

$$\mathcal{N}\left(\boldsymbol{d}|\boldsymbol{\mu}_{\check{k}}, \Sigma_{\check{k}}\right) = \frac{1}{(2\pi)^{z/2}|\Sigma_{\check{k}}|^{1/2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{d}-\boldsymbol{\mu}_{\check{k}}\right)^{\top} \Sigma_{\check{k}}^{-1}\left(\boldsymbol{d}-\boldsymbol{\mu}_{\check{k}}\right)\right) . \tag{2.23}$$

Die Kovarianzmatrix $\Sigma_{\check{k}}$, deren Determinante mit $|\Sigma_{\check{k}}|$ bezeichnet ist, wird typischerweise als diagonal modelliert und mit $\boldsymbol{\sigma}_{\check{k}}^2$ ist im Folgenden der Varianzvektor für die Komponente $\check{k}$, also die Diagonale von $\Sigma_{\check{k}}$ gemeint. Die Parameter $\boldsymbol{\beta}$ werden mit repräsentativen Trainingsdaten mittels Maximum-Likelihood-Schätzung mit dem Expectation-Maximization (EM)-Algorithmus [Dem77] bestimmt.

Die Wahrscheinlichkeit, dass der Deskriptor $\boldsymbol{d}_i$ der Gaußkomponente $\check{k}$ zugeordnet ist, sei im Folgenden mit $\gamma_{\check{k}}(\boldsymbol{d}_i)$ bezeichnet und ergibt sich

mittels des Satzes von Bayes zu

$$\gamma_{\check{k}}(\boldsymbol{d}_i) = \frac{\alpha_{\check{k}} p_{\check{k}}\left(\boldsymbol{d}_i|\boldsymbol{\beta}\right)}{\sum_{j=1}^{K} \alpha_j p_j\left(\boldsymbol{d}_i|\boldsymbol{\beta}\right)} \,. \tag{2.24}$$

Zur Bestimmung des Fisher Vektors $\boldsymbol{\Phi}(\mathcal{D})$ einer Menge von Deskriptoren $\mathcal{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n\}$, $\boldsymbol{d}_i \in \mathbb{R}^z$ ergibt sich Gleichung 2.21 unter der Annahme, dass die Deskriptoren statistisch unabhängig sind, also $p\left(\mathcal{D}|\boldsymbol{\beta}\right) = \prod_{i=1}^{n} p\left(\boldsymbol{d}_i|\boldsymbol{\beta}\right)$ gilt, zu

$$\boldsymbol{\Phi}(\mathcal{D}) = \sum_{i=1}^{n} L_{\boldsymbol{\beta}} \nabla_{\boldsymbol{\beta}} \log p\left(\boldsymbol{d}_i|\boldsymbol{\beta}\right) \,. \tag{2.25}$$

Bezüglich $L_{\boldsymbol{\beta}} = \Psi_{\boldsymbol{\beta}}^{-1/2}$ schlägt [Per07] eine Approximation der Fisher Informationsmatrix in Form einer Diagonalmatrix vor, die auf der Annahme basiert, dass die Zuordnungen der Deskriptoren zu den einzelnen Gaußkomponenten (Gleichung 2.24) annähernd exklusiv erfolgt. Die Normalisierung der Gradientenvektoren mittels $L_{\boldsymbol{\beta}}$ entspricht dann letztlich einem dimensionsweisen Whitening. Für die normalisierten Gradientenvektoren bezüglich Mittelwert und Varianz ergeben sich schließlich für jede Gaußkomponente $\check{k}$:

$$\boldsymbol{\Phi}_{\boldsymbol{\mu}_{\check{k}}}(\mathcal{D}) = \frac{1}{n\sqrt{\alpha_{\check{k}}}} \sum_{i=1}^{n} \gamma_{\check{k}}(\boldsymbol{d}_i) \left(\frac{\boldsymbol{d}_i - \boldsymbol{\mu}_{\check{k}}}{\boldsymbol{\sigma}_{\check{k}}} \right) \quad \text{und} \tag{2.26}$$

$$\boldsymbol{\Phi}_{\boldsymbol{\sigma}_{\check{k}}}(\mathcal{D}) = \frac{1}{n\sqrt{2\alpha_{\check{k}}}} \sum_{i=1}^{n} \gamma_{\check{k}}(\boldsymbol{d}_i) \left[\left(\frac{\boldsymbol{d}_i - \boldsymbol{\mu}_{\check{k}}}{\boldsymbol{\sigma}_{\check{k}}} \right)^2 - 1 \right], \tag{2.27}$$

wobei die Division und Potenzierung hier elementweise zu verstehen sind. Der Gradient bezogen auf die Gewichte $\alpha_{\check{k}}$ liefert hingegen keinen nennenswerten Beitrag [Per10b], sodass er typischerweise nicht für den Fisher Vektor verwendet wird. Der finale Fisher Vektor konkateniert schließlich für jede

der K GMM-Komponenten die einzelnen z-dimensionalen Vektoren:

$$\mathbf{\Phi}(\mathcal{D}) = \begin{bmatrix} \vdots \\ \mathbf{\Phi}_{\boldsymbol{\mu}_k}(\mathcal{D}) \\ \vdots \\ \mathbf{\Phi}_{\boldsymbol{\sigma}_k}(\mathcal{D}) \\ \vdots \end{bmatrix}, \tag{2.28}$$

sodass sich für den Fisher Vektor der Deskriptoren eine Dimensionalität von $2 \cdot K \cdot z$ ergibt wobei z die Dimensionalität der Deskriptoren darstellt.

[Per10a] schlägt vor, den Fisher Vektor einer sog. *Power Normalization* zu unterziehen, was als vorzeichenerhaltendes Wurzelziehen beschrieben werden kann, denn jedes Element ϕ_i des Vektors wird durch $\text{sign}(\phi_i)\sqrt{|\phi_i|}$ normalisiert. [Per10b] beobachtet außerdem, dass eine anschließende ℓ_2-Normierung vorteilhaft ist. Über die Ursachen und Hintergründe, warum diese Schritte die Ergebnisse deutlich verbessern, existieren inzwischen diverse Interpretationen, die in [Sán13] ausführlich dargestellt und wiederum ergänzt werden.

Eine weitere, vom Fisher Vektor inspirierte Möglichkeit der Aggregierung von Deskriptoren stellt [Jég10b] unter dem Namen ***Vector of Locally Aggregated Descriptors (VLAD)*** vor. Sie kann als eine vereinfachte Variante des Fisher Vektors interpretiert werden: anstelle des GMM wird das visuelle Codebook verwendet, die Deskriptoren werden nur einem visuellen Wort zugeordnet (nicht mehr mittels Gewichten α_j zu allen Komponenten), und es werden nur noch die Statistiken erster Ordnung erfasst. Die Abweichung bezüglich der Varianz fällt also weg und es werden allein die Residuen betrachtet, also die Differenzvektoren der Deskriptoren zu den Clusterzentren ihrer jeweiligen visuellen Wörter. Diese Residuen werden für jedes visuelle Wort aufsummiert. Die Dimensionalität halbiert sich dadurch verglichen mit dem Fisher Vektor auf $K \cdot z$. Die resultierende einfachere und effizientere Berechnung verhalf VLAD zu großer Beliebtheit, sodass anschließend diverse Verbesserungen vorgeschlagen wurden:

- Power Normalization der VLAD-Vektoren anstelle der ℓ_2-Normierung [Jég12a, Jég12b],
- ℓ_2-Normierung der Residuen *vor* der Summierung, damit auch die sehr nahe an den Clusterzentren liegenden Deskriptoren einen Beitrag liefern [Del13],
- Intra-Normalization: ℓ_2-Normierung der Residuen *nach* deren Summierung – für jedes visuelle Wort separat, damit sich Burstiness-Effekte (siehe Seite 30) nur lokal in dieser Voronoi-Zelle und nicht global auf den VLAD-Vektor auswirken [Ara13, Tol13],
- Verwerfen derjenigen Deskriptoren, die nahe an den Voronoi-Grenzen ihres visuellen Wortes liegen, da diese in anderen Bildern ohnehin meist auf das benachbarte visuelle Wort quantisiert werden [Che11],
- PCA-basierte Rotation der Residuen innerhalb jedes visuellen Wortes mit einer individuellen PCA, die mit zu diesem visuellen Wort zugeordneten Deskriptoren berechnet wurde [Del13].

Verglichen mit dem BoW-Modell kommen sowohl bei VLAD als auch beim Fisher Vektor verhältnismäßig kleine Codebooks bzw. wenige GMM Komponenten zum Einsatz, typischerweise 16 bis 256. Die resultierenden Vektoren sind dadurch nicht dünn besetzt, sodass eine Indexierung nach dem Inverted File Prinzip nicht angewendet werden kann. In der Regel wird daher eine Dimensionsreduktion mittels PCA durchgeführt und der Vergleich von Bildern anschließend per erschöpfender Suche oder durch Verfahren der approximativen Nächste-Nachbar-Suche wie etwa randomisierte k-d-Bäume [Muj14] oder Produkt Quantisierung [Jég11] durchgeführt.

2.3.2. Faltende neuronale Netze

Im Bereich der Bildverarbeitung finden neuronale Netze bereits seit geraumer Zeit Anwendung [LeC10], wobei in den meisten Fällen ein *Multi-Layer Perzeptron* genannter Aufbau verwendet wird. Ein MLP ist ein vorwärtsgerichtetes Netz von Perzeptronen [Ros58], die in Schichten angeordnet sind. Ein Perzeptron stellt ein künstliches Neuron dar, das im Vergleich zum biologischen Vorbild im Gehirn eine stark abstrahierte Form der Informationsverarbeitung realisiert: Es führt für eine Anzahl n_{cnn} von skalaren

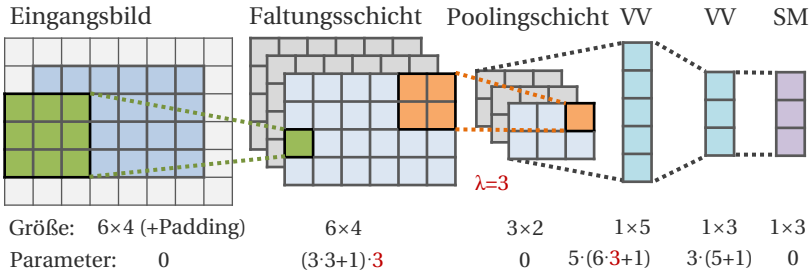

Abbildung 2.2.: Skizze eines faltenden neuronalen Netzes mit einer Faltungsschicht mit $\lambda = 3$ Kanälen, einer Poolingschicht, zwei vollvernetzen Schichten (VV) und einer SoftMax-Schicht (SM) zur Klassifikation eines Bildes in drei Objektklassen. Die weißen Randpixel des Eingangsbildes deuten das Padding der Größe 1 Pixel an. Typischerweise greift die Faltung auf alle Kanäle der vorherigen Schicht zu, was aus Gründen der Übersichtlichkeit jedoch nicht veranschaulicht ist, denn das Eingangsbild besitzt im Beispiel nur einen Kanal.

Eingangsgrößen (in Vektordarstellung zusammengefasst als $\boldsymbol{x}_{cnn} \in \mathbb{R}^{n_{cnn}}$) eine mittels $\boldsymbol{w}_{cnn}$ gewichtete Summierung durch, addiert einen Offset-Wert b_{cnn} und wendet eine nichtlineare Funktion φ darauf an:

$$y_{cnn} = \varphi\left(\boldsymbol{w}_{cnn}^T \boldsymbol{x}_{cnn} + b_{cnn}\right) . \tag{2.29}$$

Die Eingangsgrößen $\boldsymbol{x}_{cnn}$ stellen dabei die Ausgänge der Perzeptronen der vorherigen Schicht dar, und das Ergebnis y_{cnn} wird wiederum an die Perzeptronen der nächsten Schicht weitergeleitet. Als nichtlineare Funktion φ, die auch als Aktivierungsfunktion bezeichnet wird, kommt beispielsweise die Sigmoid-Funktion $\varphi(x) = \frac{1}{1+e^{-x}}$, der Tangens Hyperbolicus $\varphi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ oder die sogenannte *Rectified Linear Unit Aktivierungsfunktion* (ReLU) $\varphi(x) = \max(0,x)$ zum Einsatz. Die Menge der Gewichtsvektoren und Offset-Werte aller Perzeptronen bilden die Parameter des Netzes, die in einem iterativen Trainingsvorgang durch den Backpropagation Algorithmus [Rum85] gelernt werden.

Der durchschlagende und anhaltende Erfolg der neuronalen Netze begann 2012 mit tiefen faltenden neuronalen Netzen (Convolutional Neural Network (CNN)) [Kri12] für den Bereich der bildbasierten Objektklassi-

fikation, als sowohl Trainingsdaten als auch Rechenkapazität in ausreichend großem Maße verfügbar wurden. Diese Netze sind dabei im Wesentlichen aus vielen, sich abwechselnden Faltungs- und Poolingschichten aufgebaut, gefolgt von vollvernetzten Schichten und einer finalen SoftMax-Schicht [LeC98], die unter anderem sicherstellt, dass als Ergebnis der Klassifikation ein ℓ_1-normierter Vektor entsteht, dessen Elemente als Wahrscheinlichkeiten für die einzelnen Klassen interpretiert werden können. Die Netze werden mit großen Bildmengen trainiert – üblicherweise mit den 1,2 Millionen Bilder des *ImageNet* Datensatzes [Den09], die in 1 000 semantische Klassen eingeteilt sind. Während des Trainings lernen die Netze in den unteren Schichten lokale Bildstrukturen, was prinzipiell vergleichbar ist mit der Merkmalsextraktion von bislang händisch entworfenen Merkmalen wie SIFT. Mit jeder höheren Schicht abstrahieren sie zunehmend, sodass die letzten Schichten bereits Objektklassen-spezifische Muster modellieren [Zei14] und die darauffolgenden vollvernetzten Schichten, die typischerweise die weitaus meisten der gelernten Parameter beinhalten, als Klassifikator interpretiert werden können. Abbildung 2.2 zeigt beispielhaft den typischen Aufbau eines faltenden neuronalen Netzes.

Diese, durch das Training der Netze erlernten Strukturen der Trainingsbilder können als globale Beschreibung eines Bildes für die Bildsuche genutzt werden. Dazu wird ein Bild durch ein vortrainiertes Netz wie z. B. *AlexNet* [Kri12], *VGGNet* [Sim14], *GoogleNet* [Sze15] oder *ResNet* [He16] propagiert, und anschließend werden in bestimmten Schichten die dort berechneten Ergebnisse extrahiert. Dabei hat sich generell gezeigt, dass die Informationen aus den letzten Faltungsschichten [Bab15, Kal15, Tol15] bessere Ergebnisse liefern als die Informationen der nachfolgenden vollvernetzten Schichten [Bab14, Gon14], was auch daran liegt, dass die Faltungsschichten eine noch größere Robustheit gegenüber Transformationen wie Verschiebung, Verdeckung etc. aufweisen und lokale Informationen enthalten.

Die Ergebnisse einer Faltungsschicht werden dabei als *Feature Maps* bezeichnet und jede Feature Map ist eine Matrix F der Größe $W \times H$. Jeder Wert einer Feature Map ergibt sich aus der Faltung eines Teilbereichs des Eingangs der Schicht mit einem Faltungskern der Größe $\kappa \times \kappa$, d. h. jedes Element der Feature Map bezieht sich nur auf einen lokalen Teil des Eingangs, der als rezeptives Feld bezeichnet wird und die Größe des Faltungskerns aufweist. Diese Lokalität und die Tatsache, dass jedes Element der Feature Map mit

demselben Faltungskern arbeitet, der während des Trainings gelernt wird, verhindern, dass die Anzahl der Parameter des Netzes bei zunehmender Schichtanzahl zu sehr ansteigt. Die Lokalität erscheint einerseits plausibel vor dem Hintergrund, dass in natürlichen Bildern die lokalen Strukturen (Kanten, Muster, etc.) in der Regel auch an anderer Stelle im Bild in ähnlicher Weise auftreten können. Andererseits genügt ein einziger Faltungskern nicht, um die vielfältigen Strukturen in Bildern zu erfassen. Daher werden in einer Faltungsschicht mehrere Faltungskerne parallel verwendet und das Ergebnis einer Faltungsschicht besteht somit aus λ einzelnen Feature Maps $\left\{F_i \in \mathbb{R}^{W \times H}\right\}_{i=1,\ldots,\lambda}$. Bezüglich der unterschiedlichen Faltungskerne wird auch von *Kanälen* gesprochen. Das Beispiel in Abbildung 2.2 zeigt eine Faltungsschicht mit $\kappa = 3, W = 6, H = 4$ und $\lambda = 3$. Die Faltungen werden außerdem mit einem Padding der Größe eins ausgeführt, das heißt, das Eingangsbild wird am Rand um jeweils ein Pixel erweitert, damit der Ausgang der Faltung wieder dieselbe Größe aufweist wie der ursprüngliche Eingang. Um die Ergebnisse einer Faltungsschicht für die Bildsuche in einen globalen Vektor zu aggregieren, werden pro Kanal die Einträge der Feature Map aufsummiert [Bab15] oder durch ihr Maximum repräsentiert [Azi15], sodass sich ein λ-dimensionaler Vektor ergibt. Um daraus robuste und kompakte Deskriptoren für den Bildvergleich zu erhalten, wird dieser anschließend diversen Nachverarbeitungsschritten unterzogen, etwa ℓ_2-Normierung, elementweises Quadrieren oder Wurzelziehen, Dimensionsreduktion per PCA oder durch Lernen einer linearen Projektionsmatrix [Sim13] oder anschließendem Whitening [Jég12a]. [Kal15] verwendet außerdem, ähnlich dem Konzept der IDF, eine räumliche und kanalabhängige Gewichtung der Einträge in den Feature Maps vor der Aggregierung. [Tol15] hingegen extrahiert zunächst räumlich einzelne quadratische überlappende Regionen in den Feature Maps über alle Kanäle hinweg, normalisiert diese Regionen einzeln, aggregiert sie anschließend durch Summierung und normalisiert den resultierenden Vektor erneut. Die Motivation dahinter ist, die Suchergebnisse für Objekte zu verbessern, die nicht mittig im Bild abgebildet sind, wie dies in den Trainingsbildern üblicherweise der Fall ist. Die inhaltsbasierte Bildsuche schließlich wird dann durchgeführt, indem die Bilder anhand der euklidischen Distanz oder der Kosinusdistanz dieser globalen Bildrepräsentationen verglichen werden.

Da die gängigen vortrainierten Netze für die Objektklassifikation entwi-

ckelt wurden, zielen auch die darin enthaltenen und durch das Training ermittelten Parameter darauf ab, die zuvor festgelegten semantischen Klassen zu unterscheiden. Die dabei zwangsläufig entstehende und durchaus gewünschte Robustheit gegenüber der Intraklassen-Variabilität ist für die Bildsuche jedoch unvorteilhaft, denn dort gilt es, die unterschiedlichen Instanzen einer Objektklasse zu unterscheiden. In [Bab14] wird daher ein vortrainiertes Netz mit dem LANDMARKS Datensatz [Bab14] nachtrainiert, der über 200 000 Bilder von 672 unterschiedlichen Sehenswürdigkeiten enthält. Die letzte vollvernetzte Schicht wurde dafür von den ursprünglichen 1 000 auf 672 Knoten angepasst. Da die Datensätze zur Evaluation ähnliche visuelle Eigenschaften aufweisen wie der für das Training verwendete LANDMARKS Datensatz, nämlich überwiegend Gebäudeansichten, konnte dadurch eine signifikante Verbesserung erzielt werden.

[Gor16] geht bezüglich der Netzarchitektur noch einen Schritt weiter und verwendet ein Ranking-Framework, das mit dem Triplet Loss – basierend jeweils auf einem Bild, einem relevanten ähnlichen Bild sowie einem sogenannten *hard negative*, also einem nichtrelevanten, aber „zum Verwechseln ähnlichen“ Bild – arbeitet und ein Region-Proposal-Network (RPN) [Ren15] integriert. Durch das Mitlernen des RPN verbessern sich zwar die Suchergebnisse auf den einschlägigen Datensätzen und eine Lokalisierung wird ermöglicht, allerdings ist bei der inhaltsbasierten Bildsuche – anders als bei der Objektklassifikation – zum Zeitpunkt des Trainings der Netze nicht notwendigerweise bekannt, nach welchen Objekten oder Szenen später einmal im zu indexierenden Bildmaterial gesucht werden soll. Es besteht also die Gefahr, dass insbesondere kleinere Objekte im RPN-Teil des Netzes nicht als Hypothesen vorgeschlagen werden und somit nicht mit dem Gesamtnetz wiedergefunden werden können. Falls jedoch zur Trainingszeit bereits bekannt ist, nach welchen Objekttypen oder gar Instanzen später gesucht werden soll, sind solche auf neuronalen Netzen basierenden Systeme inzwischen das Mittel der Wahl, da die vielen Teilbereiche (Detektion und anschließende Repräsentation der Hypothesen, Komprimierung auf kompakte Codes, Distanzmetriken) gemeinsam in einem Netz integriert gelernt werden können.

Ausgehend von diesem Stand der Forschung wird im nun folgenden Kapitel die Position in einem System zur inhaltsbasierten Bildsuche herausgear-

beitet, an der die Beiträge dieser Arbeit ansetzen, und die einzelnen Beiträge werden motiviert

3

Konzept

Angesichts der stetig anwachsenden Menge an Bilddaten liegt ein Schwerpunkt der Forschung zur inhaltsbasierten Bildsuche seit jeher auf der Skalierungsfähigkeit der vorgeschlagenen Modelle. Für Systeme, die auf lokalen Merkmalen und dem BoW-Modell [Siv03] aufbauen, wurden dafür im letzten Jahrzehnt zum einen viele ergänzende Techniken wie Re-Ranking [Per09], Query Expansion [Chu07b] oder Burstiness [Jég09a] vorgeschlagen und stetig verfeinert. Gleichzeitig wurde auch versucht, den hinsichtlich der Skalierungsfähigkeiten relevantesten Teil des BoW-Modells zu verbessern: die Quantisierung der Merkmalsdeskriptoren durch ein Codebook. Mit entsprechenden Erweiterungen wie Soft Quantization [Phi08a] oder Hamming Embedding [Jég08] sowie verschiedenen Quantisierungsverfahren lassen sich vielfältige Kompromisse hinsichtlich der relevantesten Randbedingungen realisieren:

- **Trefferquote**: Wie viele der relevanten Bilder werden gefunden?
- **Genauigkeit**: Wie viele der gefundenen Bilder sind relevant?
- **Speicheranforderungen bezüglich des Index**: Je weniger Byte pro indexiertem Merkmal benötigt werden, desto mehr Datenbankbilder finden im begrenzten Arbeitsspeicher Platz.
- **Speicheranforderungen bezüglich des Akkumulators**: Er muss für jede Suchanfrage initialisiert, anschließend anhand der im Index er-

mittelten Merkmale mit Stimmen befüllt, und schließlich ausgewertet werden, um die Ähnlichkeit der Datenbankbilder zum Anfragebild zu bestimmen.

- **Laufzeit und Speicheranforderungen für die Modellerstellung (offline)**: Das Modell (Codebook, ggf. HE-Parameter, etc.) wird typischerweise nur einmal erstellt, sodass dieser Aspekt in der Regel eine untergeordnete Rolle einnimmt.
- **Laufzeit für die Indexierung (offline)**: Auch die Indexierung wird pro Datenbankbild zwar nur einmal durchgeführt, bei Datenbankgrößen jenseits der Millionen-Grenze sind aber effiziente Verfahren erforderlich.
- **Laufzeit für die Suchanfrage**: Je nach Implementierung der Parallelisierung muss hier noch zwischen der Latenz, die für die meisten Anwendungsfälle wenige Sekunden nicht überschreiten sollte, und dem Durchsatz unterschieden werden, der in webbasierten Anwendungen aufgrund der hohen Nutzerzahlen oft ebenso relevant ist.

Dennoch haben Phiblin *et al.* bei der Einführung der Merkmalsanordnung im Re-Ranking Schritt bereits angemerkt, dass man die Skalierungsfähigkeit des BoW-Modells nur dann grundlegend steigern können wird, wenn es gelingt, die lokalen Merkmale nicht nur als lose Ansammlung ihrer Deskriptoren zu erfassen, sondern deren Anordnung im Index zu berücksichtigen [Phi07]:

> »However, even if the ranking function is adequate, we believe that retrieval performance will not scale unless we find efficient ways to include spatial information in the index, and move some of the burden of spatial matching from the ranking stage to the filtering stage.«

Dies wurde daraufhin mit den vielfältigsten Modellen versucht, die sich praktisch alle in zwei grundlegende Klassen einordnen lassen:

1. Verfahren, die den **Akkumulator** eines Bildes **erweitern**, sodass sich beim Abstimmungsprozess die Stimmen derjenigen Merkmale in einem Teilbereich des Akkumulators eines Datenbankbildes häufen, die für eine konsistente Anordnung sprechen, und

2. auf **Filterung** basierende Verfahren, die den Akkumulator nicht erweitern, sondern noch vor der Stimmabgabe die Konsistenz der Merkmalsanordnung überprüfen und dadurch nichtkonsistente Merkmale vom Abstimmungsprozess ausschließen.

Eine dritte Möglichkeit hat dagegen in der Fachliteratur bislang nur wenig Aufmerksamkeit auf sich gezogen: die **Anordnung als eigenständiges Merkmal** zu erfassen, das in dieser Arbeit als *Umgebungsmerkmal* bezeichnet wird, und dieses zu quantisieren. Analog zu den visuellen Wörtern des Codebooks für die Deskriptoren der lokalen SIFT Merkmale werden die quantisierten Umgebungsmerkmale als *Umgebungswörter* bezeichnet.

Der große Vorteil dieser dritten Variante ist, dass weder der Akkumulator erweitert werden muss, noch aufwändige Filterschritte vor der Stimmabgabe erforderlich sind. Denn durch die Quantisierung der Umgebungsinformationen muss von vornherein nur auf diejenigen Merkmale im Index zugegriffen werden, die nicht nur bezüglich ihres visuellen Wortes, sondern auch bezüglich ihrer jeweiligen größeren Umgebung übereinstimmen. Abbildung 3.1b veranschaulicht die sich dadurch ergebende weitere Dimension im Index. Alle indexierten Merkmale werden nun anhand beider Dimensionen adressiert: die zum BoW gehörende Dimension, und die für die Merkmalsumgebung U. Falls die Umgebung noch mit einem weiteren Verfahren in entsprechenden Merkmalen erfasst und quantisiert wird, kann die Umgebung sogar mit zwei Dimensionen U_1 und U_2 charakterisiert werden. Wie in Abbildung 3.1c) veranschaulicht, können dann entweder beide Umgebungswörter übereinstimmen (orangefarbene Zelle), oder mindestens eines (gelbe Zellen).

Als Alternativen zum BoW-Modell wurden mit dem Fisher Vektor [Per07] und VLAD [Jég10b] weitere Verfahren der Aggregierung von lokalen Merkmalen vorgestellt, die die Skalierungsfähigkeiten auf eine wiederum andere Art adressieren. Das Ziel ist hierbei, jedes Bild in einen äußerst kompakten Vektor zu kodieren und zu einem Anfragebild den nächstgelegenen Vektor der Datenbankbilder zu ermitteln - per erschöpfender oder approximativer Suche. In eine ähnliche Richtung gehen auch die Ansätze, die auf künstlichen neuronalen Netzen basieren [Bab14, Gon14, Bab15, Kal15, Tol15]. Auch hier wird jedes Bild mit einem möglichst kompakten Vektor beschrieben, damit die Suche in großen Datenbanken effizient durchgeführt werden kann. Zwar

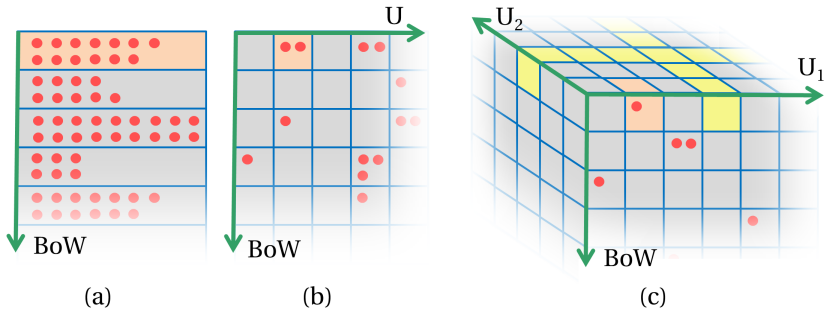

Abbildung 3.1.: Erweiterung des BoW-Index um zusätzliche Dimensionen, die die Merkmalsumgebung beschreiben. Die roten Punkte stellen die einzelnen quantisierten lokalen Merkmale der Datenbankbilder dar. Während einer Suchanfrage muss für jedes Merkmal im Anfragebild die entsprechende inverse Liste (orangefarbene Zelle) ausgewertet werden, die mit jeder zusätzlichen Dimension weniger Merkmale enthält.

lassen sich damit für viele Datensätze beeindruckende Ergebnisse erzielen, die Fähigkeit, kleine Objekte in Bildern zu finden, leidet unter diesen kompakten Bildrepräsentationen aber deutlich.

Im Rahmen dieser Dissertation werden daher die bisher genannten Ansätze auf eine neue Art kombiniert. Ausgehend vom BoW-Modell wird systematisch die Frage untersucht, wie die Umgebung eines lokalen Merkmals vorteilhaft repräsentiert und anschließend quantisiert werden kann, um wie in Abbildung 3.1b oder 3.1c veranschaulicht, als separate Dimension im Index zu fungieren. Bezogen auf das oben genannte Zitat von Phiblin *et al.* geht diese Arbeit damit noch einen Schritt weiter als die Ansätze des dort erwähnten „filtering stages“, denn der Vergleich von Merkmalsumgebungen wird komplett in den Offline-Schritt - also die Indexierung - verschoben. Nicht korrespondierende Merkmalsumgebungen müssen während einer Suchanfrage gar nicht erst verarbeitet werden.

Abbildung 3.2 zeigt die thematischen Teilbereiche der inhaltsbasierten Bildsuche, die in dieser Arbeit hauptsächlich adressiert werden. Die Vorgehensweise zur Beschreibung und Evaluation der Umgebungsmerkmale sowie der sich daraus ergebenden neuen Möglichkeiten werden in den folgenden drei Abschnitten beschrieben.

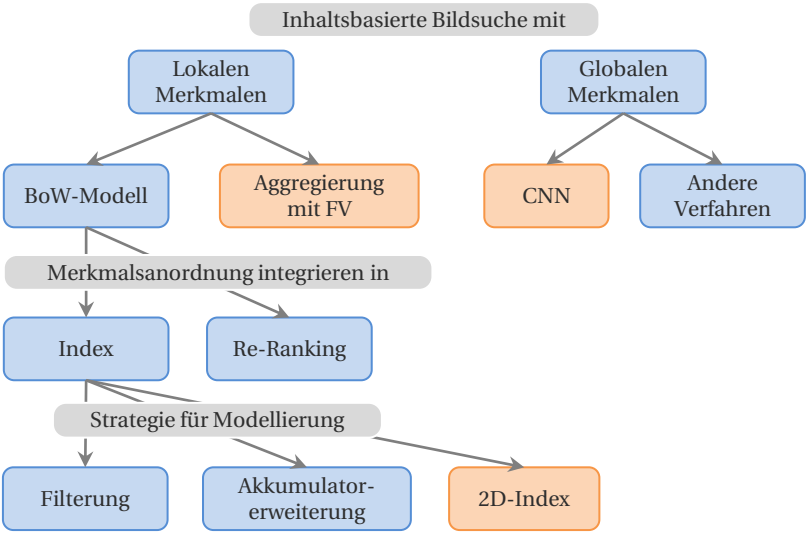

Abbildung 3.2.: Grobe Einordnung der Arbeit bezüglich verschiedener Verfahren in der inhaltsbasierten Bildsuche. Diese Dissertation fokussiert sich auf die drei orangefarben dargestellten Ansätze.

3.1. Umgebungsmerkmale

Um die größere Umgebung eines lokalen Merkmals zu erfassen, werden in Kapitel 5 zwei Repräsentationen analysiert. Zum einen werden alle in der Umgebung liegenden lokalen Merkmale mittels des **Fisher Vektors** aggregiert und somit in eine Vektordarstellung konstanter Größe überführt [Man16a]. Zum anderen werden Repräsentationen untersucht [Man17b], die auf Ergebnissen von faltenden **neuronalen Netzen** basieren. Dazu wird jedes Bild durch ein neuronales Netz verarbeitet und in bestimmten Schichten des Netzes werden diejenigen Informationen extrahiert, die aus den Bildregionen der Umgebung des lokalen Merkmals entstanden sind. Diese Repräsentation basiert also nicht auf den händisch entworfenen lokalen Merkmalen, sondern versucht, das in einem neuronalen Netz vorgelernte

und durch die Netzparameter kodierte Wissen über Objekte[1] in Bildern zu verwenden.

Wie auch bei den lokalen Merkmalen üblich, steht in dieser Arbeit der Begriff *Umgebungsmerkmal* sowohl für die neuen vorgeschlagenen Repräsentationen, als auch für deren konkret sich ergebende Merkmalsvektoren. Wenn dagegen die einzelnen lokalen SIFT Merkmale innerhalb einer bestimmten Umgebung gemeint sind, etwa bei Aggregierung durch den Fisher Vektor, wird zur Abgrenzung immer von *lokalen Merkmalen in der Umgebung* gesprochen.

3.2. Evaluationsframework

Um den Nutzen der Umgebungsmerkmale zu analysieren, und um die beiden Repräsentationen zu vergleichen, oder, um jeweils verschiedene Designparameter zu evaluieren, sind prinzipiell immer die Auswirkungen auf die Suchergebnisse eines kompletten Systems entscheidend. Andererseits kann bei komplexen Umgebungsmodellen die Indexierung großer Bildmengen zu aufwändig sein, um sie für jede Evaluation der Verfahren oder der Parameter separat durchzuführen. Im Rahmen dieser Dissertation wurde daher ein Framework entwickelt, das schnellere Aussagen für die Evaluation der Umgebungsmerkmale dadurch ermöglicht, dass derjenige Aspekt der Suche getrennt modelliert und erfasst wird, der im späteren Gesamtsystem entscheidend ist [Man17b]. Denn die vorgeschlagene Erfassung der Umgebung eines lokalen Merkmals zielt darauf ab, die im BoW-Modell bislang allein auf visuellen Worten basierenden Korrespondenzen zu verfeinern. Nur solche Merkmale sollen in den Akkumulator eingetragen werden, die – verglichen mit dem jeweiligen Merkmal im Anfragebild – einen plausiblen Kontext aufweisen. Die anderen sollen dagegen verworfen werden. Dafür muss das Umgebungsmerkmal korrekte BoW-Korrespondenzen von inkor-

1 Das mag ein gewisser Widerspruch sein zur oben formulierten Anforderung, dass vorab nicht bekannt ist, nach welchen Objekten gesucht werden soll – schließlich sind die gängigen neuronalen Netze für die Klassifikation von üblicherweise 1 000 unterschiedlichen Objektklassen trainiert worden. Es hat sich allerdings gezeigt, dass solche Netze durch die Millionen von Trainingsbilder auch eine sehr allgemeine Kenntnis über die typischen lokalen Strukturen in Bildern erlangt haben und dass diese vortrainierten Modelle dadurch für viele andere Anwendungsbereiche aus dem Stand heraus gute Ergebnisse liefern [SR14].

rekten BoW-Korrespondenzen unterscheiden. Weil darüber hinaus das Umgebungsmerkmal quantisiert weiterverarbeitet wird, um die zusätzliche(n) Dimension(en) im Index zu ermöglichen, kann diese Unterscheidung entweder gelingen (falls die Umgebungsmerkmale auf denselben Wert quantisiert werden) oder nicht (falls unterschiedliche quantisierte Werte resultieren). Genau dieser Sachverhalt wird im Framework modelliert, indem für bekannte Datensätze unter Nutzung ihrer Annotierungen zwei Mengen von Korrespondenzen zusammengestellt werden. Jede Korrespondenz bezieht sich dabei auf ein Bildpaar und besteht aus je einem Merkmal des einen Bildes und einem des anderen Bildes:

1. **Korrekte BoW-Korrespondenzen** $\mathcal{K}_k$: Die Deskriptoren ihrer Merkmale wurden auf dasselbe visuelle Wort abgebildet und sie beziehen sich auch auf dieselbe reale Struktur in beiden Bildern (siehe grün eingefärbte Korrespondenzen in Abbildung 2.1 b). Auch durch die hinzukommenden Umgebungsinformation sollen diese Korrespondenzen möglichst erhalten bleiben, d. h. ihre Umgebungsmerkmale sollen ebenfalls auf einen übereinstimmenden Wert quantisiert werden.

2. **Inkorrekte BoW-Korrespondenzen** $\mathcal{K}_i$: Die Deskriptoren ihrer Merkmale wurden zwar auf dasselbe visuelle Wort abgebildet, aber sie beziehen sich auf *verschiedene* reale Strukturen in beiden Bildern (siehe rot eingefärbte Korrespondenzen in Abbildung 2.1 b). Diese Korrespondenzen entstehen entweder durch die Quantisierungsverluste des BoW-Modells oder weil die beteiligten Merkmale tatsächlich eine gewisse Ähnlichkeit aufweisen, d. h. im Deskriptorraum räumlich nahe beieinander liegen. Anhand ihrer Umgebungsinformation soll es nun gelingen, diese inkorrekten BoW-Korrespondenzen zu verwerfen, indem die Umgebungsmerkmale auf unterschiedliche Werte quantisiert werden.

Zur Evaluation einer Umgebungsrepräsentation kann mit diesen beiden Korrespondenzmengen direkt ermittelt werden, inwiefern die jeweiligen Umgebungswörter der korrekten Korrespondenzen übereinstimmen und sich die Umgebungswörter der inkorrekten Korrespondenzen als verschieden ergeben.

Nachdem eine geeignete Umgebungsrepräsentation gefunden wurde, kann der Frage nach dem Gewinn für das Gesamtsystem nachgegangen werden [Man17a]. Kapitel 6 widmet sich dieser Evaluation mit den gängigen öffentlichen Datensätzen.

3.3. Speicherauslegung des Index

Wenn die diskriminante Quantisierung der Umgebung eines lokalen Merkmals gelingt, dann wird während einer Suchanfrage ein Großteil der inkorrekten BoW-Korrespondenzen direkt (d. h. ohne weitere Filterungsschritte, die bei den Methoden in Abschnitt 2.2.5.2 erforderlich wären) verworfen, sodass diese gar nicht erst im Speicher gelesen werden müssen. Abbildung 3.1b und 3.1c deuten an, dass die einzelnen inversen Listen bzw. Zellen dann weniger Merkmale beinhalten, da sich alle, einem visuellen Wort zugeordneten Merkmale entlang der neuen Dimensionen verteilen. Die Effizienz des inversen Index profitiert davon außerordentlich, da sich die effektive Codebookgröße somit als Produkt der Größe des Codebooks der visuellen Wörter und der des Codebooks der Umgebungswörter ergibt.

Ein limitierender Aspekt von BoW-basierten Systemen hinsichtlich der maximalen Datenbankgröße war bislang stets, dass der Index im Arbeitsspeicher gehalten werden muss. Der Grund dafür liegt in der Tatsache, dass beispielsweise bei einer Codebookgröße von 100 000 visuellen Wörtern und 3 000 Merkmalen im Anfragebild für jede Suche etwa 3 000 inverse Listen traversiert, und somit etwa 3% aller Merkmale im Index gelesen werden müssen. Die Datenmenge dieser Lesezugriffe und die Anzahl der unterschiedlichen Speicheradressen (im Beispiel 3 000) ermöglichten es bislang nicht, den Index dauerhaft auf den wesentlich größeren und preisgünstigeren, aber langsameren Magnetfestplattenspeichern zu belassen.

Mit den in den letzten Jahren aufkommenden SSDs-Laufwerken, also großen, nichtflüchtigen Speichermedien auf Halbleiterbasis, ändert sich diese Limitierung, da diese sowohl schnellere Übertragungsraten, als auch immens mehr Leseoperationen pro Sekunde auf zufällig verteilte Speicheradressen bieten. Die Kombination des in dieser Dissertation vorgestellten diskriminanten Zugriffs auf die quantisierten Merkmale im 2D/3D-Index mit dieser Speichertechnologie ermöglicht es daher erstmals, den Index auf einem SSD-Laufwerk zu belassen [Man18]. Kapitel 7 untersucht dazu

die Details vor dem Hintergrund, dass die Merkmale im Index anhand von zwei oder drei Dimensionen adressiert werden, der Speicher aber nur eine eindimensionale Adressierung anbietet.

4

Framework zur Evaluation von Umgebungsmerkmalen

Dieses Kapitel widmet sich dem in Abschnitt 3.2 motivierten Framework, mit dem unterschiedliche Varianten von Umgebungsmerkmalen oder verschiedene Parameter ein und desselben Umgebungsmerkmals verglichen werden können, ohne jedes Mal ein komplettes System zur inhaltsbasierten Suche zu realisieren [Man17b].

Zunächst werden die in dieser Arbeit verwendeten Datensätze vorgestellt. Anschließend wird beschrieben, wie daraus die Informationen extrahiert werden, die für Modellerstellung und für die Evaluierung erforderlich sind.

4.1. Datensätze

Für das Lernen der Modelle und für die Evaluation werden die im Bereich der inhaltsbasierten Bildsuche etablierten öffentlichen Datensätze OXFORD5K [Phi07], PARIS6K [Phi08a], HOLIDAYS [Jég08], LANDMARKS [Bab14] und MIRFLICKR1M [MJH10] herangezogen. Eine wichtige Charakteristik der Datensätze ist in Abbildung 4.1 verdeutlicht: der Anteil der Merkmale, die in den jeweiligen, als ähnlich annotierten Bildpaaren Korrespondenzen aufweisen, wobei hier Korrespondenzen basierend auf den Originaldeskrip-

toren gemeint sind, d. h. $\mathcal{M}_\varepsilon$, ermittelt mit dem Abstandsverhältnis aus Gleichung 2.2.

Die Objekte in den Bildern sind oft nur in einem gewissen Teilausschnitt zu sehen, und häufig unterscheiden sich Beleuchtung, Betrachtungswinkel, Aufnahmesensorik etc. deutlich für die Bilder desselben Objektes oder derselben Szene. Daher ergeben sich in den verwendeten Datensätzen nur für einen geringen Teil der lokalen Merkmale überhaupt Korrespondenzen in den jeweils ähnlichen Bildern. Für den PARIS6K Datensatz gilt beispielsweise, dass in der „schwierigeren" Hälfte aller Bilder weniger als 1,7% der Merkmale überhaupt Korrespondenzen zum jeweils annotierten ähnlichen Bild aufweisen. Für OXFORD5K ergibt dieser Medianwert 2,4%, für LANDMARKS 1,6% und für HOLIDAYS 5,3%. Diese sehr geringe Anzahl an Korrespondenzen wird durch die Quantisierung der Merkmalsdeskriptoren für das BoW-Modell noch weiter reduziert. Viele andere Datensätze, die im Bereich der inhaltsbasierten Bildsuche verwendet werden, bieten hingegen nicht genügend Ablenkung für diese Suche der Nadel im Heuhaufen, weshalb sie in dieser Arbeit nicht betrachtet werden. Als Beispiel sei der *Ukbench* Datensatz [Nis06] angeführt, für den sich der oben beschriebene Medianwert zu 13,6% ergibt.

Die verwendeten Datensätze weisen die folgenden Eigenschaften auf:

- OXFORD5K [Phi07]: Dieser Datensatz beinhaltet 5 062 Bilder und deckt elf verschiedene Sehenswürdigkeiten in Oxford ab. Für jede Sehenswürdigkeit existiert eine Menge unterschiedlich vieler korrespondierender Bilder (insgesamt 567 Bilder), die im Rahmen der Evaluation gefunden werden sollen, wobei die restlichen 4 495 Bilder als Ablenkung dienen und keine Gemeinsamkeiten mit den Sehenswürdigkeiten aufweisen. Als Anfragebilder dienen jeweils 5 ausgewählte Bilder pro Sehenswürdigkeit.

- PARIS6K [Phi08a] beinhaltet 6 392 Bilder und deckt analog zu OXFORD5K elf verschiedene Sehenswürdigkeiten in Paris ab, auch mit jeweils fünf Anfragebildern sowie unterschiedlich vielen korrespondierenden Bildern (insgesamt 1 791 korrespondierende Bilder sowie 4 601 Bilder zur Ablenkung). Der Datensatz wurde hauptsächlich ergänzend zu OXFORD5K eingeführt, um Repräsentationen (Codebook, Fisher Vektoren, HE, etc.) auf einem von der Evaluation unabhängigen Datensatz trainieren zu können. Dies führt zwar zu schlechteren

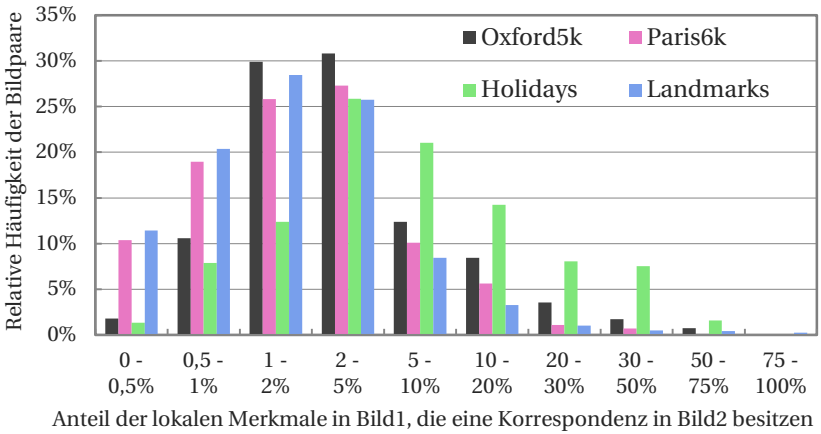

Abbildung 4.1.: „Schwierigkeit" der Datensätze in Bezug auf den Anteil der korrespondierenden lokalen Merkmale: pro Datensatz wurde für 2 000 zufällig ausgewählte, gemäß Annotierung korrespondierende (d. h. dasselbe Objekt beinhaltende) Bildpaare ermittelt, wie viele der Merkmale des einen Bildes ein korrespondierendes Merkmal im anderen Bild aufweisen. Der MIRFLICKR1MDatensatz ist nicht aufgeführt, da dafür keine Annotierungen hinsichtlich korrespondierender Bilder existieren.

Ergebnissen [Nis06, Sch07, Phi08a], ist aber in der Praxis häufig unumgänglich, da die Bilddatenbanken in der Regel mit der Zeit wachsen und die Bilder aus Zeitgründen nicht ständig erneut indexiert werden können.

- HOLIDAYS [Jég08]: beinhaltet 1 491 Bilder von 500 unterschiedlichen Szenen, die typische Urlaubsbilder repräsentieren. Die meisten Szenen sind allerdings zur gleichen Zeit mit derselben Kamera aufgenommen, sodass, wie in Abbildung 4.1 erkennbar, viele Bildpaare mit vergleichsweise vielen Korrespondenzen existieren. Andererseits variieren manche Bilder hinsichtlich der Aufnahmeperspektive und des gemeinsamen Bildinhaltes deutlich.

- LANDMARKS [Bab14]: Im Gegensatz zu den obigen Datensätzen wurden die Bilder dieses Datensatzes nicht manuell, sondern durch einen semi-automatischen Prozess zusammentragen. Ausgehend von Wikipedias 10 000 meistgelesenen Seiten über Sehenswürdigkeiten wurde

mittels der Seitentitel die Yandex Bildsuche[1] bemüht, um jeweils bis zu 1 000 potentielle Bilder einer Sehenswürdigkeit herunterzuladen. Durch grobe manuelle Inspektion der jeweils ersten 100 Bilder wurden die Sehenswürdigkeiten in unterschiedlich präzise Teilmengen untergliedert. In dieser Dissertation wird die von den Autoren als „clean subset“ bezeichnete Partition verwendet. Da allerdings – wie in solchen Fällen aus rechtlichen Gründen üblich – nur die Downloadadressen zur Verfügung gestellt werden, sind inzwischen viele Bilder unter den Adressen nicht mehr erreichbar, sodass 35 224 Bilder von 586 verschiedenen Sehenswürdigkeiten verwendet werden. Die Bilder innerhalb einer Sehenswürdigkeit (im Folgenden *Gruppe* genannt) bilden dabei oft unterschiedliche Teilbereiche ab - etwa die Außenansicht und das Gebäudeinnere, sodass mit einem Anfragebild einer Gruppe grundsätzlich nicht immer sämtliche Bilder der Gruppe auch gefunden werden können. Durch einen automatischen Abgleich mit den übrigen drei Datensätzen wurde weitgehend sichergestellt, dass sich keine überlappenden Sehenswürdigkeiten im Datensatz befinden.

- MIRFLICKR1M [MJH10]: Dieser Datensatz wird für die Experimente hinsichtlich der Skalierungsfähigkeit in Kapitel 6 genutzt. Er besteht aus etwa einer Million Bilder von Flickr[2], die anhand der „Interestingness“ ausgewählt wurden – ein Flickr-internes Maß, das aus diversen Einflussfaktoren die Beliebtheit der Bilder bei den Nutzern einschätzt.

Die farbliche Kennzeichnung der vier Datensätze in Abbildung 4.1 wird im Sinne einer besseren Orientierung in den weiteren Diagrammen dieser Dissertation einheitlich verwendet.

1 http://images.yandex.ru
2 http://www.flickr.com

4.2. Ermittlung der Korrespondenz- und Merkmalsmengen

Um den Nutzen eines Umgebungsmerkmals für ein Gesamtsystem der Bildsuche zu quantifizieren, wird mit dem in diesem Kapitel vorgestellten Framework derjenige Teil des Systems betrachtet, der von den Umgebungsmerkmalen profitieren soll: die Verarbeitung der BoW-Korrespondenzen. Dazu werden basierend auf den Datensätzen und ihren Annotierungen die in Abschnitt 3.2 beschriebenen Mengen der korrekten und inkorrekten BoW-Korrespondenzen definiert, mit dem Ziel, später die jeweiligen Umgebungen der beiden Merkmale in jeder Korrespondenz zu erfassen und zu vergleichen. Die Umgebungen der korrekten BoW-Korrespondenzen sollten bei diesem Vergleich Ähnlichkeiten aufweisen, wohingegen inkorrekte BoW-Korrespondenzen anhand ihrer nicht übereinstimmenden Umgebungen der jeweiligen beiden Merkmale erkannt werden können.

Für die Quantisierung der Merkmalsdeskriptoren der Bilder wird in dieser Arbeit ein Codebook $\mathcal{C}$ der Größe $k = 18^4 = 104\,976$ verwendet, das mit den Deskriptoren aller Bilder des OXFORD5K Datensatzes durch hierarchisches k-Means-Clustering (HKM) erzeugt wurde. Bei der Wahl der Basis und des Exponenten für das hierarchische Clustering ergeben sich typischerweise geringfügig bessere Ergebnisse für breitere Clusterings, daher wird 18^4 gegenüber der vergleichbaren Größe 10^5 bevorzugt. Die gewählte Größe des Codebooks liegt etwa im Mittelfeld der üblichen Codebookgrößen, da die Untersuchungen unabhängig von den in Abschnitt 2.2.3 genannten Erweiterungen Soft Quantization (für noch größere Codebooks) und Hamming Embedding (für kleinere Codebooks) durchgeführt werden sollen. Die Quantisierung selbst wird mittels randomisierter k-d-Bäume [Muj14] realisiert, also mit einem Verfahren für die approximative Nächste-Nachbar-Suche. Der Kompromiss zwischen Laufzeit und Genauigkeit wird dabei so gewählt, dass im Mittel für etwa 90% der Deskriptoren der korrekte nächste Nachbar – also das tatsächlich ähnlichste visuelle Wort – gefunden wird.

4.2.1. Korrekte BoW-Korrespondenzen $\mathcal{K}_k^*$

Um diejenigen lokalen Merkmale zu ermitteln, die demselben visuellen Wort zugeordnet wurden, und gleichzeitig auch dieselbe reale Struktur in

der Welt kennzeichnen, werden für jeden Datensatz zunächst die mitgelieferten Annotierungen herangezogen. Da die Bilder in Gruppen eingeteilt sind (je 11 für OXFORD5K und PARIS6K, 500 für HOLIDAYS und 586 für LANDMARKS), ergibt sich bei j Bildern in einer Gruppe kombinatorisch eine Menge von $\frac{j(j-1)}{2}$ Bildpaaren, die dasselbe Objekt oder dieselbe Szene zeigen. Da im LANDMARKS Datensatz einige Gruppen mit sehr vielen Bildern existieren, wurden dort je Gruppe maximal 2 000 aller kombinatorisch möglichen Bildpaare per Zufall ausgewählt. In jedem der Bildpaare werden anschließend die BoW-Korrespondenzen $\mathcal{M}_\mathcal{C}$ ermittelt, also diejenigen Merkmalspaare ausgewählt, die mittels des Codebooks demselben visuellen Wort zugeordnet wurden. Da durch die Quantisierungsfehler auch BoW-Korrespondenzen entstehen, die nicht dieselbe reale Struktur in der Szene betreffen, müssen die BoW-Korrespondenzen noch einer detaillierteren Überprüfung unterzogen werden. Dazu werden mit den originalen (unquantisierten) Deskriptoren gemäß dem Abstandsverhältnis im Deskriptorraum aus Gleichung 2.2 die wesentlich genaueren Korrespondenzen $\mathcal{M}_\varepsilon$ ermittelt. Da selbst diese aber noch vereinzelt fehlerhaft sind, werden sie anschließend noch einer weiteren Plausibilitätsüberprüfung unterzogen, in der mindestens drei Korrespondenzen der Nachbarschaft ähnlich angeordnet sein müssen ($\mathcal{M}_g$) [Man15a]. Eine korrekte BoW-Korrespondenz in einem Bildpaar, bestehend aus je einem Merkmal in beiden Bildern, entsteht mit anderen Worten also nur dann, wenn drei Bedingungen erfüllt sind: (1) beide Merkmale wurden demselben visuellen Wort des Codebooks zugeordnet, (2) beide Merkmale werden außerdem gemäß dem Abstandsverhältnis aus Gleichung 2.2 als korrespondierend betrachtet, und (3) in der näheren Umgebung existieren mindestens drei weitere Korrespondenzen, die zwischen beiden Bildern geometrisch plausibel angeordnet sind:

$$\begin{aligned} \mathcal{K}_k = \Big\{ \Big(\boldsymbol{f}_i \in \mathcal{F}^{(1)}, \boldsymbol{f}_j \in \mathcal{F}^{(2)} \Big) \Big| \\ \big(\boldsymbol{d}_i, \boldsymbol{d}_j\big) \in \mathcal{M}_\mathcal{C}\left(\mathcal{F}^{(1)}, \mathcal{F}^{(2)}\right) \wedge \\ \big(\boldsymbol{d}_i, \boldsymbol{d}_j\big) \in \mathcal{M}_\varepsilon\left(\mathcal{F}^{(1)}, \mathcal{F}^{(2)}\right) \wedge \\ \Big(\boldsymbol{f}_i, \boldsymbol{f}_j\Big) \in \mathcal{M}_g\left(\mathcal{F}^{(1)}, \mathcal{F}^{(2)}\right) \Big\}, \end{aligned} \tag{4.1}$$

wobei $\mathcal{F}^{(1)}$ und $\mathcal{F}^{(2)}$ die lokalen Merkmalsmengen von zwei Bildern bezeichnen, die dasselbe Objekt oder dieselbe Szene zeigen.

In den Datensätzen – insbesondere in HOLIDAYS – existieren vereinzelt sehr ähnliche Bildpaare, die Fast-Duplikat-Charakter aufweisen, da sie kurz hintereinander mit derselben Kamera in etwa derselben Ausrichtung aufgenommen wurden. Da in diesen Bildpaaren nach obiger Vorgehensweise mitunter tausende korrekte BoW-Korrespondenzen ermittelt würden, wird die Anzahl pro Bildpaar per Zufallsauswahl auf 100 begrenzt. Denn bezüglich dieser Korrespondenzen werden später die jeweiligen Umgebungen analysiert und bei zu vielen Korrespondenzen in einem Bildpaar würden sich viele der Umgebungen räumlich überlappen, was die Aussagekraft der Evaluation einschränken würde.

Im OXFORD5K Datensatz ergaben sich auf diese Weise in 18 002 Bildpaaren insgesamt 612 295 korrekte BoW-Korrespondenzen. Tabelle 4.1 gibt die Anzahl der korrekten BoW-Korrespondenzen für alle vier Datensätze an, die im Weiteren mit $\mathcal{K}_k^{Ox}, \mathcal{K}_k^{Pa}, \mathcal{K}_k^{Ho}$ und $\mathcal{K}_k^{La}$ bezeichnet werden. In Abschnitten, die für alle diese vier Mengen gelten, steht $\mathcal{K}_k^*$ jeweils stellvertretend für jede der vier Mengen.

4.2.2. Inkorrekte BoW-Korrespondenzen $\mathcal{K}_i^*$

Bezüglich der inkorrekten BoW-Korrespondenzen, also Paaren von Merkmalen, die zwar jeweils demselben visuellen Wort zugeordnet wurden, die aber nicht vom selben Objekt stammen, werden zwei Mengen zusammengestellt:

- $\mathcal{K}_i^{OxPa}$: Es wird jeweils zufällig ein Bild aus OXFORD5K und eines aus PARIS6K ausgewählt. Die beiden zugehörigen Merkmalsmengen werden auf gemeinsame visuelle Wörter überprüft und von den sich so ergebenden inkorrekten BoW-Korrespondenzen $\mathcal{M}_C$ werden 30% per Zufallsauswahl zur Menge $\mathcal{K}_i^{OxPa}$ hinzugefügt. Alle Korrespondenzen werden deshalb nicht verwendet, da es sonst bei den Umgebungen häufig zu Überlappungen käme. Dadurch, dass die Bilder in unterschiedlichen Städten bzw. Ländern aufgenommen wurden, ist prinzipiell davon auszugehen, dass die auf diese Weise ermittelten BoW-Korrespondenzen nicht vom selben Objekt stammen[1].

1 Für kleinere Objekte im Hintergrund (etwa Fahrzeuge oder Verkehrsschilder) kann dies nicht gänzlich ausgeschlossen werden. Ein automatischer Abgleich der beiden Datensätze ergab aber keine nennenswerte Übereinstimmung, weshalb hier auf weitere Filterschritte verzichtet wird.

Tabelle 4.1.: Anzahl der verwendeten Bilder, Bildpaare und Korrespondenzen für die zusammengestellten Mengen $\mathcal{K}_k^*$ der korrekten BoW-Korrespondenzen, die zur Berechnung der Evaluationsmaße verwendet werden.

Datensatz	Bilder	Szenen	Verwendete Bildpaare	$\lvert\mathcal{K}_k^*\rvert$
OXFORD5K	5 062	11	18 002	612 295
PARIS6K	6 392	11	26 441	693 128
HOLIDAYS	1 491	500	1 399	79 660
LANDMARKS	35 224	586	35 447	524 014

- $\mathcal{K}_i^{La}$: Um bezüglich der inkorrekten BoW-Korrespondenzen auch die Verwechslungseffekte innerhalb eines Datensatzes analysieren zu können, wird zusätzlich diese Menge erstellt, indem jeweils ein Bildpaar aus unterschiedlichen Sehenswürdigkeiten des LANDMARKS Datensatzes ausgewählt wird und wie oben jeweils 30% der BoW-Korrespondenzen zu $\mathcal{K}_i^{La}$ hinzugefügt werden.

Für die Erstellung beider Korrespondenzmengen wurden so viele Bildpaare ausgewählt, dass sich ein vergleichbarer Umfang zu $\mathcal{K}_k^{Ox}$ ergibt, also etwa 600 000 Korrespondenzen. In den weiteren Ausführungen steht $\mathcal{K}_i^*$ auch hier jeweils stellvertretend für eine der beiden Mengen $\mathcal{K}_i^{OxPa}$ und $\mathcal{K}_i^{La}$.

4.2.3. Merkmalsmenge für die Modellerstellung $\mathcal{F}_M^{Ox}$

Bisher wurden die Korrespondenz-Mengen zur Evaluation von Umgebungsmerkmalen beschrieben. Da die Evaluation anhand der quantisierten Umgebungsmerkmale erfolgt, ist außerdem eine Menge von Umgebungsmerkmalen für das Lernen der Quantisierungsfunktion erforderlich. Wie auch beim Clustern von lokalen Merkmalsdeskriptoren zu visuellen Wörtern, die dann ein Codebook definieren, wird hier eine Menge von Umgebungsmerkmalen zu Umgebungswörten geclustert, die somit ein Umgebungscodebook ergeben. Für diese Modellbildung wird die folgende Menge an lokalen Merkmalen definiert:

- $\mathcal{F}_M^{Ox}$: Aus allen Bildern des OXFORD5K Datensatzes werden etwa 1,2 Millionen Merkmale zufällig ausgewählt, was etwa 5% aller Merkmale entspricht.

4.3. Evaluationsmethodik und -maß

Die Evaluation von Umgebungsmerkmalen innerhalb des Frameworks zielt auf die für die Bildsuche entscheidende Fähigkeit, korrekte von unkorrekten BoW-Korrespondenzen unterscheiden zu können. Dazu werden für die vorgestellten Korrespondenz-Mengen $\mathcal{K}_k^*$ und $\mathcal{K}_i^*$ zunächst die Umgebungsmerkmale berechnet. Für die jeweils zwei lokalen Merkmale $\boldsymbol{f}_i \in \mathcal{F}^{(1)}$ und $\boldsymbol{f}_j \in \mathcal{F}^{(2)}$ von jeder Korrespondenz seien die zugehörigen Umgebungsmerkmale mit $\boldsymbol{u}(\boldsymbol{f}_i)$ und $\boldsymbol{u}(\boldsymbol{f}_j)$ bezeichnet. Nach der Quantisierung der Umgebungsmerkmale mittels eines Umgebungscodebooks sollte idealerweise

$$q\left(\boldsymbol{u}\left(\boldsymbol{f}_i\right)\right) = q\left(\boldsymbol{u}\left(\boldsymbol{f}_j\right)\right) \Leftrightarrow \left(\boldsymbol{f}_i, \boldsymbol{f}_j\right) \in \mathcal{K}_k^* \tag{4.2}$$

gelten. Zur Evaluation der Fehler, also der Fälle, in denen dies nicht gelingt, wird auf die Bezeichnungen und Maße zur Bewertung eines binären Detektors zurückgegriffen, indem die Quantisierung der Umgebungsmerkmale als Detektionsaufgabe interpretiert wird: Als Detektion gilt, wenn die beiden quantisierten Werte übereinstimmen, und die zu detektierenden Objekte sind durch die korrekten Korrespondenzen vorgegeben, während die inkorrekten Korrespondenzen sozusagen den Hintergrund darstellen. Basierend auf dieser Veranschaulichung werden die beiden Fehlermaße herangezogen:

- **Falsch-Negativ-Rate (FNR)** bezogen auf $\mathcal{K}_k^*$, d. h. der Anteil der korrekten Korrespondenzen, deren Umgebungsmerkmale jedoch auf unterschiedliche Werte quantisiert werden:

$$\mathrm{FNR} = \frac{\left|\left\{\left(\boldsymbol{f}_i, \boldsymbol{f}_j\right) \in \mathcal{K}_k^* \middle| q\left(\boldsymbol{u}\left(\boldsymbol{f}_i\right)\right) \neq q\left(\boldsymbol{u}\left(\boldsymbol{f}_j\right)\right)\right\}\right|}{\left|\mathcal{K}_k^*\right|}. \tag{4.3}$$

- **Falsch-Positiv-Rate (FPR)** bezogen auf $\mathcal{K}_i^*$, d. h. der Anteil der inkorrekten Korrespondenzen, deren Umgebungsmerkmale jedoch auf denselben Wert quantisiert werden:

$$\mathrm{FPR} = \frac{\left|\left\{\left(\boldsymbol{f}_i, \boldsymbol{f}_j\right) \in \mathcal{K}_i^* \,\middle|\, q\left(\boldsymbol{u}\left(\boldsymbol{f}_i\right)\right) = q\left(\boldsymbol{u}\left(\boldsymbol{f}_j\right)\right)\right\}\right|}{\left|\mathcal{K}_i^*\right|}. \tag{4.4}$$

Für jede der vier Korrespondenzmengen $\mathcal{K}_k^{Ox}, \mathcal{K}_k^{Pa}, \mathcal{K}_k^{Ho}$ und $\mathcal{K}_k^{La}$ ergibt sich somit eine individuelle FNR und für $\mathcal{K}_i^{OxPa}$ und $\mathcal{K}_i^{La}$ jeweils eine FPR. Aus Sicht des Gesamtsystems zielt eine geringe FNR auf die Trefferquote (siehe Seite 6), da dann möglichst viele der korrekten Korrespondenzen auch nach der Analyse der jeweiligen Umgebungen erhalten bleiben. Eine möglichst geringe FPR hingegen zielt auf die Genauigkeit, da dann die meisten inkorrekten Korrespondenzen anhand der Umgebungsinformationen aus dem Suchprozess ausgefiltert werden und somit mehr relevante Bilder in den vorderen Rängen der Ergebnisliste erscheinen.

Bezüglich der Abwägung zwischen den beiden Evaluationsmaßen kann zwar keine exakte Umrechnung angegeben werden und die Auswirkungen auf die Suchergebnisse lassen sich ebenfalls nicht direkt aus FNR und FPR ermitteln. Angesichts der in Abbildung 4.1 dargestellten Datensatzcharakteristiken dürfte einer möglichst geringen FNR allerdings die größere Bedeutung zukommen, denn in den verwendeten Datensätzen existieren viele Bildpaare, bei denen sehr wenige Korrespondenzen über den Erfolg oder Misserfolg einer Suche entscheiden.

Nach dieser Einführung der Daten, der daraus extrahierten Korrespondenzmengen sowie der Evaluationsmaße werden nun im folgenden Kapitel die Umgebungsrepräsentationen vorgestellt und damit bewertet.

5

Umgebungsmerkmale

Dieses Kapitel beschreibt, wie die Umgebung eines lokalen Merkmals repräsentiert wird. Dazu werden zunächst die Ziele und Randbedingungen einer solchen Umgebungsrepräsentation dargestellt und anschließend zwei unterschiedliche Repräsentationen definiert [Man16a, Man17b]. Im letzten Abschnitt werden beide Repräsentationen verglichen und Kombinationen davon evaluiert [Man17a].

5.1. Designziele

Beim Entwurf einer Repräsentation, die die Umgebung eines lokalen Merkmals erfasst, um damit die Bildsuche in einem BoW-Modell-basierten System zu unterstützen, ist es von entscheidender Bedeutung, die Invarianzeigenschaften des unterliegenden Systems zu berücksichtigen. Bei den verwendeten SIFT Merkmalen sind dies die Invarianzen gegenüber Translation, Skalierung sowie Rotation in der Bildebene. Nur wenn diese Transformationen bei der Invarianz des Umgebungsmerkmals berücksichtigt werden, bleiben die Invarianzeigenschaften des Gesamtsystems erhalten und die Objekte und Szenen können auch weiterhin in entsprechend transformierter Form in der Datenbank gefunden werden.

Da die Umgebungsmerkmale in dieser Arbeit aber nie isoliert, sondern immer für ein gegebenes lokales Merkmal $\boldsymbol{f}_\circ = (\boldsymbol{x}_\circ, \sigma_\circ, \theta_\circ, \boldsymbol{d}_\circ)$ berechnet

werden, können dessen Informationen (Position im Bild $\boldsymbol{x}_\circ$, Skalierung $\sigma_\circ$, Orientierung $\theta_\circ$) ausgenutzt werden. Dadurch muss das Umgebungsmerkmal per se gar nicht invariant sein, sondern seine Berechnung kann einfach relativ zur Anordnung seines lokalen Merkmals erfolgen, um die Invarianzeigenschaften des Gesamtsystems nicht zu gefährden. Von diesen Invarianzen abgesehen ist eine Umgebungsrepräsentation anzustreben, die möglichst diskriminant ist und gleichzeitig robust gegenüber den vielfältigen Artefakten wie Variation der Beleuchtung, Blickwinkel, Aufnahmesensor etc.

Aus diesem Grund werden für die Umgebungsrepräsentation die beiden Verfahren untersucht, die im Bereich der globalen Bildrepräsentationen in den letzten Jahren am erfolgreichsten waren: Fisher Vektoren und faltende neuronale Netze (CNN). Die beiden Repräsentationen unterscheiden sich in wesentlichen Aspekten:

- Bei der Fisher Vektor (FV)-Repräsentation werden die umliegenden lokalen Merkmale aggregiert. Damit basiert die Repräsentation auf denselben – händisch entworfenen – lokalen SIFT Merkmalen wie die übergeordnete BoW-basierte Bildsuche. Die Repräsentation auf CNN-Basis dagegen arbeitet mit einem vortrainierten faltenden neuronalen Netz. Dies stellt eine Art automatisch gelernte Merkmalsextraktion dar, denn im Rahmen des Trainings lernt das CNN in den ersten Schichten eine Vielzahl von Faltungskernen, die die objektrelevante Strukturen im Bild ermitteln.

- Bei der CNN-Repräsentation fließt externes Datenmaterial ein, denn das Netztraining erfolgte mit den 1,2 Millionen Bilder des *ImageNet* Datensatzes [Den09], die in 1 000 semantische Klassen eingeteilt sind. Das den Fisher Vektoren zugrundeliegende GMM andererseits, wird mit den Daten des OXFORD5K Datensatzes trainiert, sodass diese Repräsentation stärker an die Zieldomäne angepasst ist.

In den folgenden Abschnitten werden die beiden Repräsentationen vorgestellt und anschließend mit dem Evaluationsframework miteinander verglichen.

5.2. Fisher Vektor-basierte Umgebungsrepräsentation

Um in einem Bild mit den lokalen Merkmalen

$$\mathcal{F} = \{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n\}, \ \boldsymbol{f}_i = (\boldsymbol{x}_i, \sigma_i, \theta_i, \boldsymbol{d}_i) \tag{5.1}$$

die Umgebung eines bestimmten Merkmals $\boldsymbol{f}_\circ = (\boldsymbol{x}_\circ, \sigma_\circ, \theta_\circ, \boldsymbol{d}_\circ)$ mittels seiner Nachbarmerkmale zu beschreiben, werden diese zunächst bestimmt. Ausgehend von der Merkmalsposition $\boldsymbol{x}_\circ$ und abhängig von der Skalierung $\sigma_\circ$ wird eine um den Faktor ζ_{FV} größere Umgebung im Bildkoordinatensystem definiert. In dieser Umgebung werden alle Merkmale berücksichtigt, deren Skalierung σ_i nicht zu stark von $\sigma_\circ$ abweicht:

$$\mathcal{U}_\circ := \left\{ \boldsymbol{f}_i \in \mathcal{F} \,\middle|\, \left(\|\boldsymbol{x}_i - \boldsymbol{x}_\circ\|_2 < \zeta_{FV}\sigma_\circ\right) \wedge \left(\frac{1}{\rho} < \frac{\sigma_i}{\sigma_\circ} < \rho\right) \right\}. \tag{5.2}$$

Die Filterung basierend auf dem relativen Skalierungsunterschied $\frac{\sigma_i}{\sigma_\circ}$ trägt der Tatsache Rechnung, dass etwa bei groben Strukturen die benachbarten sehr feinen Strukturen in korrespondierenden Bildern nur selten wiedergefunden werden. Anhand einiger korrespondierender Bilder aus OXFORD5K wurde ρ für diese Arbeit empirisch auf 2,5 festgelegt. Abbildung 5.2 veranschaulicht für die korrekten BoW-Korrespondenzen $\mathcal{K}_k^{Pa}$ den Anteil der Merkmale, die für unterschiedliche Umgebungsgrößen ζ_{FV} aggregiert werden, also $\frac{|\mathcal{U}_\circ|}{|\mathcal{F}|}$. Dass selbst für $\zeta_{FV} = 50$ nie alle Merkmale eines Bildes in der Umgebung $\mathcal{U}_\circ$ erfasst werden, ist vor allem durch die Filterung basierend auf dem relativen Skalenunterschied aus Gleichung 5.2 begründet.

Die ausgewählten Nachbarmerkmale werden anschließend, wie in Abbildung 5.1 veranschaulicht, durch die in Kapitel 2.3.1 beschriebene Fisher Vektor Repräsentation in einen Vektor konstanter Länge aggregiert. Die Dimensionalität der Deskriptoren der Nachbarmerkmale wird dazu zunächst mittels PCA von 128 auf 64 Dimensionen reduziert. Abgesehen davon, dass durch die Dimensionsreduktion auch die späteren Fisher Vektoren kompakter sind, zielt dieser Schritt vor allem auf die Dekorrelierung der Deskriptoren. Denn dadurch, dass die SIFT Deskriptoren durch Konkatenation von 16 räumlich benachbarten Gradientenhistogrammen entstehen, kann nicht

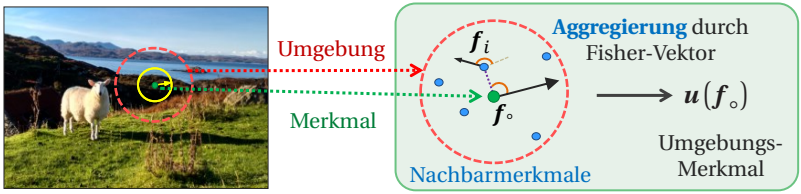

Abbildung 5.1.: Berechnung eines Umgebungsmerkmals basierend auf der Fisher Vektor Repräsentation der benachbarten lokalen Merkmale: Das lokale Merkmal $f_\circ$, im Eingangsbild gelb dargestellt, definiert durch seine Position im Bild und seine Skalierung eine Umgebung (rot), innerhalb derer die benachbarten lokalen Merkmale mittels des Fisher Vektors aggregiert werden.

davon ausgegangen werden, dass die 128 Dimensionen des Deskriptors unkorreliert sind. Nach Anwendung der PCA wird dies für die resultierenden 64 Dimensionen erreicht, sodass die Modellannahme in der Fisher Vektor Repräsentation hinsichtlich der diagonalen Kovarianzmatrizen der Komponenten des Gauß'schen Mischmodells besser erfüllt ist. Die Parameter der PCA werden aus zufälligen Deskriptoren des OXFORD5K Datensatzes berechnet.

Neben dem erscheinungsbasierten Teil der Nachbarmerkmale – also ihrer Deskriptoren – soll auch deren Anordnung berücksichtigt werden. Bei der Verwendung der Fisher Vektoren zur globalen Bildbeschreibung [Per10b] wird dies üblicherweise mit der sogenannten *Spatial Pyramid* Kodierung [Laz06] erreicht. Sie unterteilt ein Bild in eine feste Anzahl von – gegebenenfalls überlappenden und unterschiedlich großen – Teilbereichen, berechnet in jedem Teilbereich aus den darin liegenden Merkmalen einen separaten Fisher Vektor und konkateniert schließlich die einzelnen Fisher Vektoren. Die Gesamtdimensionalität steigt somit linear mit der Anzahl der Teilbereiche. Wenn außerdem nicht nur die reine zweidimensionale Lage der Merkmale im Bild, sondern auch weitere Informationen wie deren Skalierung und Orientierung berücksichtigt werden sollen, ist dieser Ansatz nicht zielführend, da mit jeder zusätzlichen Unterteilung weniger Merkmale für die einzelnen Fisher Vektor Repräsentationen zur Verfügung stehen.

Inspiriert durch [Sim13] wurde deshalb ein Ansatz gewählt, der die geometrische Anordnung der Nachbarmerkmale mit separaten Werten erfasst,

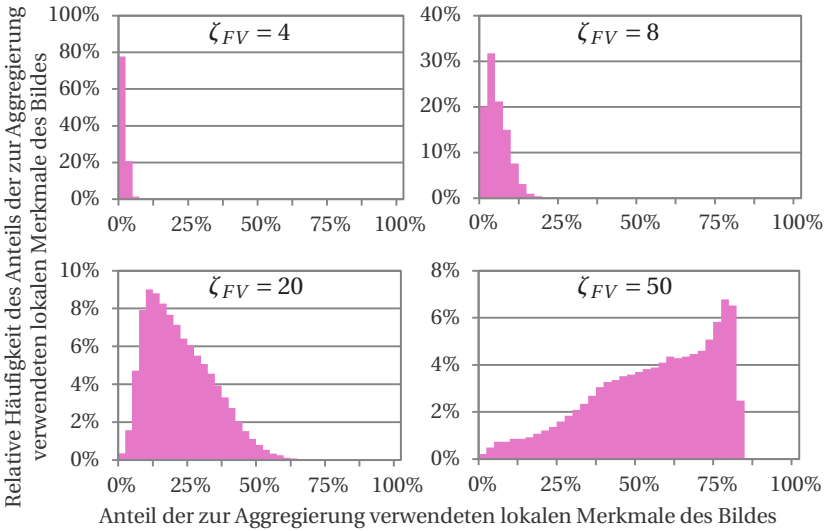

Abbildung 5.2.: Histogramm über den Anteil der lokalen Merkmale eines Bildes (in 2,5%-Intervallen), die für die Umgebungsmerkmale aus $\mathcal{K}_k^{Pa}$ aggregiert werden für vier verschiedene Umgebungsgrößen ζ_{FV}.

diese als zusätzliche Dimensionen an den Deskriptorvektor anhängt und anschließend die Fisher Vektor Repräsentation dieser erweiterten Vektoren berechnet. Im Gegensatz zu [Sim13], wo zur Gesichtswiedererkennung SIFT Deskriptoren um die relativen Bildkoordinaten ihrer lokalen Merkmale ergänzt wurden, sollen hier jedoch auch die Informationen bezüglich der Skalierung und Orientierung der lokalen Merkmale einfließen. Um dies zu erreichen, wird relativ zu einem lokalen Merkmal $\boldsymbol{f}_\circ = (\boldsymbol{x}_\circ, \sigma_\circ, \theta_\circ, \boldsymbol{d}_\circ)$ die Anordnung jedes Nachbarmerkmals $\boldsymbol{f}_i = (\boldsymbol{x}_i, \sigma_i, \theta_i, \boldsymbol{d}_i)$ durch die folgenden vier Werte beschrieben:

1. die normalisierte Distanz der Merkmale: $g_1 = \frac{\|\boldsymbol{x}_\circ - \boldsymbol{x}_i\|_2}{\sigma_\circ}$,
2. der Winkel, unter dem $\boldsymbol{f}_i$ aus Sicht von $\boldsymbol{f}_\circ$ erscheint – relativ zu $\theta_\circ$: $g_2 = \arctan2(\boldsymbol{x}_\circ - \boldsymbol{x}_i) - \theta_\circ$, wobei arctan2 die übliche Variante des Arkustangens bezeichnet, die mit zwei Argumenten arbeitet (X- und Y-Koordinate) und dadurch intern mit einer Fallunterscheidung alle vier Quadranten als Wertebereich abdecken kann,

3. das Skalierungsverhältnis der Merkmale: $g_3 = \frac{\sigma_\circ}{\sigma_i}$, und
4. der Orientierungsunterschied: $g_4 = \theta_\circ - \theta_i \pmod{2\pi}$.

Mit g_1 und g_2 wird also die Lage in Polarkoordinatendarstellung erfasst, während g_3 und g_4 die Orientierung und Skalierung der Merkmale berücksichtigen. Durch die entsprechenden Normierungen und Berechnungen relativ zu $\boldsymbol{f}_\circ$ ist sichergestellt, dass die Invarianzen der unterliegenden SIFT Merkmale gegenüber Translation, Skalierung sowie Rotation in der Bildebene erhalten bleiben. Bezüglich g_2 ist die arctan2-Funktion für den Nullvektor zwar üblicherweise als 0 definiert, um numerische Probleme zu vermeiden. Im vorliegenden Kontext bedarf dieser Fall aber ohnehin besonderer Aufmerksamkeit: Bei den verwendeten SIFT Merkmalen kommt $\boldsymbol{x}_\circ = \boldsymbol{x}_i$ immer dann vor, wenn sich bei der Bestimmung der Hauptgradientenrichtung kein ausgeprägtes Maximum im entsprechenden Gradientenrichtungshistogramm ergab. In diesem Fall werden bis zu drei weitere lokale Merkmale an derselben Position $\boldsymbol{x}_i$, aber mit unterschiedlichen Orientierungen $\theta_{i_1,\dots,i_3}$ (entsprechend den Nebenmaxima) erzeugt. Da die Deskriptoren relativ zur Merkmalsorientierung berechnet werden, ergeben sich auch unterschiedliche Deskriptoren. Für die angestrebte Umgebungsrepräsentation bieten diese zusätzlich erzeugten lokalen Merkmale allerdings keine Information und werden daher nicht berücksichtigt, d. h. vorab aus der Menge $\mathcal{U}_\circ$ entfernt, um die Fisher Vektor Repräsentation nicht zu beeinträchtigen.

Die Werte g_1, g_2, g_3 und g_4 werden anschließend an den oben beschriebenen, per PCA auf 64 Dimensionen reduzierten Deskriptor angefügt wobei die Werte der vier zusätzlichen Elemente zuvor normalisiert werden, um dem Wertebereich der übrigen PCA Dimensionen zu entsprechen.

Als nächstes wird das generative Modell in Form der Parameter für das Gauß'sche Mischmodell (Gleichung 2.22) benötigt. Aus dem Datensatz OXFORD5K werden dazu aus allen Bildern zufällig lokale Merkmale ausgewählt und die jeweiligen lokalen Merkmale der Umgebung ermittelt. Aus etwa zwei Millionen der so gesammelten 68-dimensionalen Merkmale werden dann per EM-Algorithmus [Dem77] die Modellparameter $\Theta = \left\{\alpha_j, \boldsymbol{\mu}_j, \Sigma_j\right\}_{j=1,\dots,K}$ für das GMM bestimmt, wobei K die Anzahl der Komponenten bezeichnet.

Um schließlich für ein Merkmal $\boldsymbol{f}_\circ$ das zugehörige Umgebungsmerkmal $\boldsymbol{u}(\boldsymbol{f}_\circ) \in \mathbb{R}^{\tilde{z}}$ zu berechnen, wird mittels Gleichung 2.28 die Fisher Vektor Repräsentation der 68-dimensionalen Vektoren in $\mathcal{U}_\circ$ erstellt:

$$\boldsymbol{u}\left(\boldsymbol{f}_\circ\right) = \boldsymbol{\Phi}\left(\mathcal{U}_\circ\right) \ . \tag{5.3}$$

Die Dimensionalität der Umgebungsrepräsentation entspricht $\tilde{z} = 2 \cdot K \cdot 68$. Wie in Abschnitt 2.3.1 beschrieben, werden die Vektoren der üblichen Nachverarbeitung durch Power Normalization [Per10a] und anschließender ℓ_2-Normierung [Per10b] unterzogen.

Für jedes Merkmalspaar der in Kapitel 4.2 beschriebenen Mengen von korrekten und inkorrekten BoW-Korrespondenzen zeigt Abbildung 5.3 die Resultate der Fisher Vektor Repräsentationen für $K = 32$ und $\zeta_{FV} = 20$. Als Histogramm zusammengefasst ist klar erkennbar, dass die $2 \cdot 32 \cdot 68 = 4\,352$-dimensionalen Fisher Vektoren der nichtkorrespondierenden Merkmale in der Regel eine größere euklidische Distanz aufweisen, während die Umgebungen von korrespondierenden Merkmalen zu Fisher Vektoren führen, die näher beieinander liegen. Dass die Distanzen für die korrekten Korrespondenzen im HOLIDAYS Datensatz im Mittel kleiner sind als bei OXFORD5K, PARIS6K und LANDMARKS, dass die Umgebungen also ähnlicher erscheinen, dürfte an der in Kapitel 4.1 beschriebenen Tatsache liegen, dass dort die Bildpaare der korrespondierenden Objekte und Szenen oft mit derselben Kamera in ähnlicher Pose aufgenommen wurden.

Evaluation

Die weiteren Untersuchungen erfolgen durch das in Kapitel 4 vorgestellten Evaluationsframeworks. Die Umgebungsmerkmale werden dazu mit einem Umgebungscodebook der Größe $\breve{k} = 3\,025$ quantisiert, das aus den Umgebungsmerkmalen der Merkmalsmenge $\mathcal{F}_M^{Ox}$ erstellt wurde. Anschließend werden für die Korrespondenzmengen $\mathcal{K}_k^*$ die Falsch-Negativ-Raten und für $\mathcal{K}_i^*$ die Falsch-Positiv-Raten ermittelt. Abbildung 5.4 zeigt die Evaluation für unterschiedliche Umgebungsgrößen ζ_{FV} und für unterschiedlich viele GMM Komponenten. Für $K = 16$, 32 und 64 ergeben sich für die Fisher Vektoren die Dimensionalitäten 2 176, 4 352 und 8 704. Da die Modelle auf dem OXFORD5K Datensatz trainiert wurden, ergibt sich für die ebenfalls auf OXFORD5K basierenden Korrespondenzmengen $\mathcal{K}_k^{Ox}$ und $\mathcal{K}_i^{OxPa}$ nur

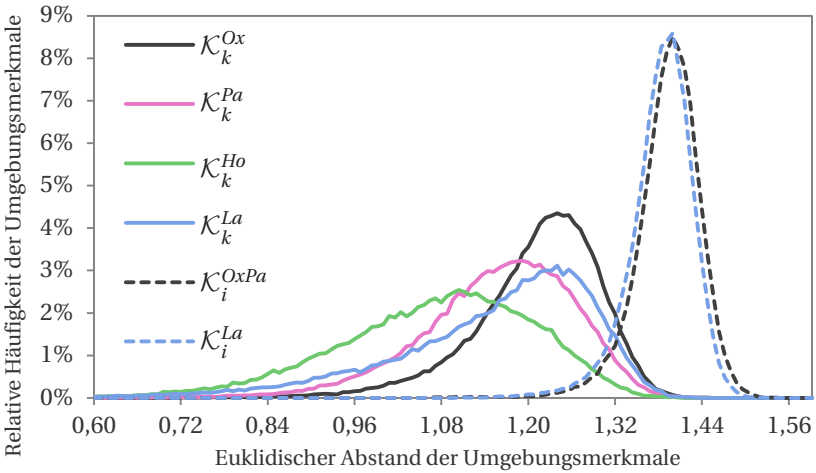

Abbildung 5.3.: Histogramm[2] über die Abstände der Umgebungsmerkmale von korrespondierenden (durchgezogene Linien, basierend auf den Korrespondenzmengen $\mathcal{K}_k^*$) und nichtkorrespondierenden Merkmalspaaren (gestrichelte Linien, basierend auf $\mathcal{K}_i^*$) für die Umgebungsgröße $\zeta_{FV} = 20$ und für $K = 32$ GMM Komponenten.

eine eingeschränkte Aussagekraft. Für die anderen Datensätze lässt sich feststellen:

- **Umgebungsgröße ζ_{FV}**: Für PARIS6K ergibt sich die geringste Falsch-Negativ-Rate bei einer relativen Umgebungsgröße von $\zeta_{FV} = 20$, wohingegen HOLIDAYS und LANDMARKS von noch größeren Umgebungen profitieren. Eine mögliche Erklärung ist, dass in PARIS6K (und in OXFORD5K) die gemeinsamen Objekte (Gebäude) manchmal nur in einem Teilbereich des Bildes sichtbar sind und die jeweiligen Objektumgebungen unterschiedlich sind, während in HOLIDAYS und LANDMARKS in der Regel ganze Szenen korrespondieren, sodass oft

2 In diesem und in einigen weiteren Diagrammen wurde aus Gründen der Übersichtlichkeit eine Liniendarstellung anstelle der für Histogramme üblichen Balkendarstellung gewählt. Die Werte dürfen daher keinesfalls interpoliert werden.

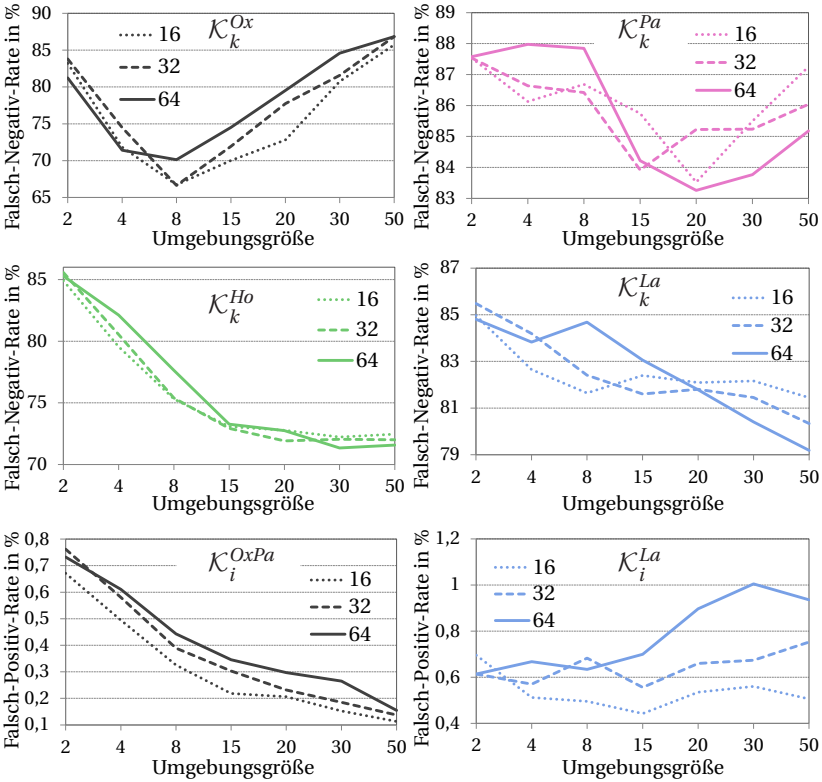

Abbildung 5.4.: Evaluation der Fisher Vektor-basierten Umgebungsmerkmale in Bezug auf die Größe der Umgebung ζ_{FV} und für 16, 32 und 64 Komponenten des Gauß'schen Mischmodells. Für jede der vier Mengen von korrekten Korrespondenzen $\mathcal{K}_k^*$ ist die Falsch-Negativ-Rate angegeben, d. h. der Anteil derjenigen Korrespondenzen, deren lokale Merkmale bezüglich ihrer quantisierten Umgebungsmerkmale nicht übereinstimmen, obwohl sie dasselbe Objekt beschreiben. Für die beiden Mengen von inkorrekten Korrespondenzen $\mathcal{K}_i^{OxPa}$ und $\mathcal{K}_i^{La}$ ist die Falsch-Positiv-Rate angegeben, d. h. der Anteil derjenigen Korrespondenzen, deren lokale Merkmale bezüglich ihrer quantisierten Umgebungsmerkmale übereinstimmen, obwohl sie unterschiedliche Objekte beschreiben.

das gesamte Bild als relevante Umgebung eines lokalen Merkmals angesehen werden kann.

- Bezüglich der **Anzahl der GMM Komponenten** K zeigt sich, dass für kleine Umgebungsgrößen wenige Komponenten vorteilhaft sind, während ab etwa $\zeta_{FV} = 20$ mehr Komponenten erforderlich scheinen, um die vielen Merkmale zu repräsentieren.

- Die **Falsch-Positiv-Rate** ist umso höher, je mehr GMM Komponenten für die Erstellung des Fisher Vektors genutzt werden. Prinzipiell würde sich bei der gewählten Größe des Umgebungscodebooks von $\check{k} = 3025$ für gänzlich zufällig verteilte Umgebungsmerkmale eine FPR von $\frac{1}{3025} \approx 0{,}033\%$ ergeben. Dass die beobachtete FPR hingegen um mehr als eine Größenordnung höher ausfällt, dürfte zum großen Teil daran liegen, dass viele der inkorrekten BoW-Korrespondenzen aus jeweils ähnlichen SIFT Deskriptoren entstanden sind, und sich diese Ähnlichkeit teilweise noch bis in die weitere Umgebung der beiden lokalen Merkmale erstreckt.

Als Nachverarbeitungsschritt werden die Fisher Vektoren üblicherweise einer Dimensionsreduktion durch PCA gefolgt von einem **Whitening** unterzogen [Jég12a]. Beim Whitening wird jedes Element des mittels PCA transformierten Vektors durch die Quadratwurzel des jeweiligen Eigenwertes der Kovarianzmatrix geteilt. Damit wird für die resultierenden Vektoren erreicht, dass die einzelnen Dimensionen nicht nur (aufgrund der PCA) unkorreliert sind, sondern auch die Varianz bezüglich jeder Dimension 1 entspricht. Die neue Kovarianzmatrix der durch PCA und Whitening nachverarbeiteten Vektoren entspricht somit der Einheitsmatrix. Die notwendigen Parameter für die PCA und das Whitening, also die Datenmittelwerte, die PCA Transformationsmatrix und die Eigenwerte der Kovarianzmatrix, werden dazu – wie für die Quantisierung auch – aus den Umgebungsmerkmalen der Merkmalsmenge $\mathcal{F}_M^{Ox}$ berechnet.

In Abbildung 5.5 sind die Auswirkungen der Dimensionsreduktion durch PCA und des Whitenings für den LANDMARKS Datensatz und $K = 32$ dargestellt. Eine bloße Dimensionsreduktion der ursprünglich 4352-dimensionalen Fisher Vektoren bringt kaum Vorteile für die nachfolgende Quantisierung. Nur wenn die Ergebnisvektoren nach der PCA Transformation einem Whi-

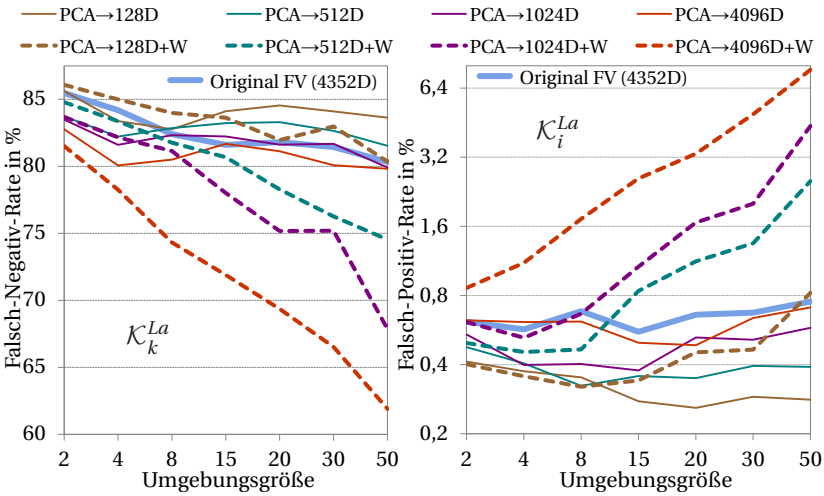

Abbildung 5.5.: Einfluss von PCA und Whitening („+W“) für den LANDMARKS Datensatz für verschiedene Umgebungsgrößen und für $K = 32$ GMM Komponenten. Die gestrichelten Linien bezeichnen Werte, die nach der PCA-basierten Dimensionsreduktion einem Whitening unterzogen wurden. Die Falsch-Positiv-Rate im rechten Diagramm ist in logarithmischer Skalierung dargestellt.

tening unterzogen werden, kann die FNR verbessert werden. Diese Verbesserung ist umso deutlicher, je mehr Dimensionen bei der Reduktion durch PCA erhalten bleiben und je größer die Umgebung ist. Gleichzeitig steigt allerdings die FPR sehr deutlich an: für $\zeta_{FV} = 50$ und 4 096 Dimensionen beispielsweise von 0,75% auf 7,67%. Um die maximal mögliche Reduzierung der FNR zu analysieren, wird im Folgenden die Dimensionalität durch die PCA nur marginal – aus Gründen der Implementierung und der Speicherausrichtung – auf die nächstgelegene Zweierpotenz reduziert: für $K = 16$ von 2 176 auf 2 048, für $K = 32$ von 4 352 auf 4 096 und für $K = 64$ von 8 704 auf 8 196 Dimensionen.

Abbildung 5.6 stellt die Ergebnisse von PCA und Whitening für alle Datensätze und verschiedene Werte von ζ_{FV} und K in Bezug auf die FNR dar. Es zeigt sich, dass durch das Whitening die FNR auch bei den Datensätzen PARIS6K und HOLIDAYS von einer größeren Umgebungsgröße profitiert. Die Auswirkungen auf die FPR sind in Tabelle 5.1 dargestellt. Für beide Mengen

von inkorrekten Korrespondenzen steigt die FPR durch PCA und Whitening bei zunehmender Umgebungsgröße immer stärker an – für $\zeta_{FV} = 50$ um mehr als den Faktor zehn.

Fazit

Bei der Umgebungsrepräsentation mittels Fisher Vektor, dessen GMM Parameter auf dem OXFORD5K Datensatz gelernt wurden, ergeben sich auf dem vergleichbaren Datensatz PARIS6K nur für weniger als jede fünfte korrekte Korrespondenz auch übereinstimmende Umgebungswörter. Für die Datensätze HOLIDAYS und LANDMARKS, bei denen oft komplette Szenen korrespondieren, lassen sich auch bei sehr ausgedehnten Umgebungen, also unter Nutzung von sehr vielen der lokalen Merkmale in der Umgebung nur geringfügig bessere Ergebnisse erzielen. PCA und Whitening können diesbezüglich bei den großen Umgebungen deutliche Vorteile bringen, allerdings sehr zu Lasten der Falsch-Positiv-Rate, sodass anstelle von grob jeder hundertsten inkorrekten Korrespondenz dann etwa jede zehnte nicht erkannt wird.

Die Fähigkeit zur Generalisierung scheint demnach eingeschränkt zu sein, oder es mangelt an weiteren repräsentativen Daten, um bessere GMM Parameter zu lernen. Dies motiviert die zweite vorgeschlagene Repräsentation im nächsten Abschnitt, die auf den Ergebnissen von faltenden neuronalen Netzen basiert. Diese vortrainierten Netze haben im Rahmen des Trainings mit den über eine Million Bilder des *ImageNet* Datensatzes eine große Zahl an unterschiedlichen semantischen Konzepten verarbeitet und können dadurch für die Unterscheidung zwischen korrekten und inkorrekten Korrespondenzen womöglich bessere Kompromisse realisieren.

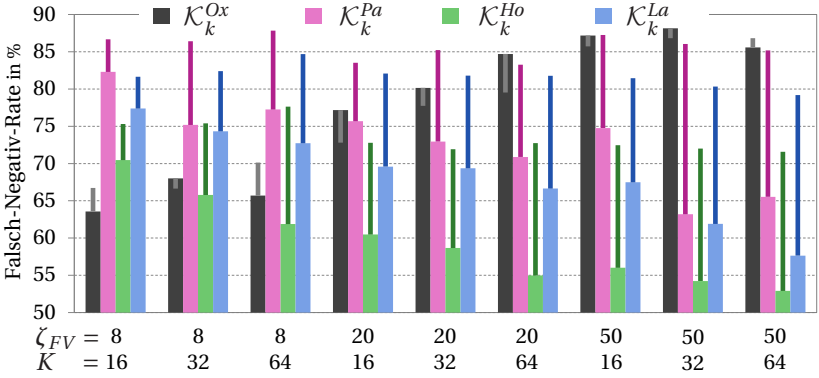

Abbildung 5.6.: Einfluss von PCA und Whitening auf die Fisher Vektor-basierten Umgebungsmerkmale für die Umgebungsgrößen 8, 20 und 50 sowie für 16, 32 und 64 Komponenten des Gauß'schen Mischmodells. Die dicken Balken stellen die FNR *nach* PCA und Whitening dar, während die aufgesetzten dünneren Linien die Originalwerte der Fisher Vektoren aus Abbildung 5.4 angeben. Die Länge der Linien visualisiert somit den Vorteil (bzw. Nachteil bei OXFORD5K) von PCA und Whitening.

Tabelle 5.1.: Einfluss von PCA und Whitening („+PCA+W") auf die Fisher Vektor-basierten Umgebungsmerkmale für die Umgebungsgrößen ζ_{FV} = 8, 20 und 50 sowie für K = 16, 32 und 64 Komponenten des Gauß'schen Mischmodells. Angegeben ist die Falsch-Positiv-Rate in % für die beiden Mengen von inkorrekten Korrespondenzen $\mathcal{K}_i^{OxPa}$ und $\mathcal{K}_i^{La}$.

ζ_{FV}	8			20			50		
K	16	32	64	16	32	64	16	32	64
$\mathcal{K}_i^{OxPa}$	0,33	0,39	0,44	0,21	0,23	0,30	0,11	0,14	0,15
+PCA+W	0,92	1,09	1,28	1,34	1,32	1,80	1,05	1,56	2,09
$\mathcal{K}_i^{La}$	0,50	0,68	0,63	0,54	0,66	0,90	0,51	0,75	0,94
+PCA+W	1,12	1,72	2,22	3,20	3,31	4,29	5,17	7,69	10,36

5.3. CNN-basierte Umgebungsrepräsentation

Um eine lokale Bildregion mit Merkmalen von faltenden neuronalen Netzen zu repräsentieren, wird in dieser Arbeit das von der Visual Geometry Group (Oxford University) (VGG) mit dem *ImageNet* Datensatz [Den09] vortrainierte *VGG16* Netz [Sim14] verwendet. Im Bereich der inhaltsbasierten Bildsuche ist dies das am weitesten verbreitete Netz, sowohl für die globalen Bildrepräsentationen [Bab15, Tol15, Kal15, Moh16], als auch für die Verfahren, die mit weiteren Daten nachtrainieren oder die Netzarchitekturen anpassen [Sal16, Ara16, Rad16, Gor16]. Das *VGG16* besteht, wie in Abbildung 5.7 dargestellt, aus 13 Faltungsschichten gefolgt von drei vollvernetzten Schichten. Die Faltungsschichten sind in fünf Blöcke eingeteilt, wobei die ersten beiden Blöcke aus jeweils zwei Schichten bestehen, und weitere drei Blöcke mit jeweils drei Schichten folgen. Die Feature Maps der Schichten innerhalb eines Blocks weisen dabei dieselbe Größe auf. Die Größe der Eingangsbilder sowie die Größe der Feature Maps des ersten Blocks ist 224×224 Pixel und nach jedem der fünf Blöcke erfolgt eine Halbierung der Breite und Höhe durch ein Maximum-Pooling in einem 2×2 Fenster mit Schrittweite zwei. Nach der letzten Faltungsschicht des letzten Blocks besitzt die Feature Map daher noch eine Breite und Höhe von $\frac{224}{2^4} = 14$ Pixel vor dem Pooling und entsprechend $\frac{224}{2^5} = 7$ Pixel nach dem Pooling . Alle Faltungen erfolgen mit einem Filter der Größe 3×3, Padding eins und Schrittweite eins. Alle Faltungsschichten innerhalb eines Blockes besitzen dieselbe Anzahl an Kanälen, die bei 64 beginnend nach jedem Block sukzessive bis auf 512 verdoppelt wird, bevor die erste vollvernetzte Schicht dann die 512 Feature Maps der Größe 7×7 zu einen 4 096-dimensionalen Vektor zusammenfasst.

Um die Bildregion einer Merkmalsumgebung zu repräsentieren, werden die Informationen aus der letzten Faltungsschicht *conv5_3* verwendet, da diese bereits für globale Bildrepräsentationen die besten Ergebnisse für die Bildsuche lieferte [Bab15, Tol15, Kal15, Moh16]. Bei der ursprünglichen, während des Trainings des Netzes verwendeten Bildgröße von 224×224 Pixeln ergeben sich als Ergebnis der letzten Faltungsschicht 512 Feature Maps der Größe 14×14. Die nachfolgenden vollvernetzten Schichten, deren Parameter zwingend Feature Maps einer festen Größe als Eingang voraussetzen, werden nicht verwendet. Daher können prinzipiell auch größere Bilder als Eingang der ersten Faltungsschicht verwendet werden und die Netzarchi-

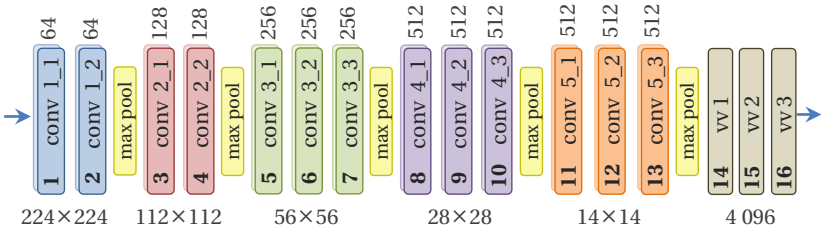

Abbildung 5.7.: Architektur des *VGG16* Netzes bezüglich der einzelnen Schichten. Am oberen Rand der Abbildung ist jeweils die Anzahl der Filterkanäle angegeben (64, 128, 256 und 512), und am unteren Rand die Größe der jeweiligen Feature Maps.

tektur kann als vollfaltendes Netz [Lon15] angesehen werden. Die Faltungen werden weiterhin mit den Filtern der Größe 3×3 durchgeführt, sodass sich lediglich andere Größen für die Feature Maps der Ein- und Ausgänge der Faltungsschichten ergeben. Da die Bibliotheken zur effizienten Berechnung der Netze von einer festen Bildgröße ausgehen, kann allerdings aus praktischen Gründen nicht für jede vorkommende Bildauflösung ein separates Netz geladen werden. Um dennoch die teilweise deutlich unterschiedlichen Bildauflösungen der verwendeten Datensätze zu berücksichtigen, werden in dieser Arbeit drei verschiedene Netzeingangsgrößen verwendet für Eingangsbilder mit Seitenlängen von 224, 448 und 896 Pixel. Außerdem werden jeweils separate Netzinstanzen verwendet für Bilder im Hochformat, im Querformat und für quadratische Bilder, sodass sich insgesamt neun Netzinstanzen ergeben, die also alle dieselben Parameter verwenden, aber für jeweils unterschiedlich große Eingangsbildgrößen bzw. Feature Maps. Für das Netz für die Bildgröße 896×672 Pixel ergibt sich nach der letzten Faltungsschicht beispielsweise für jeden der 512 Kanäle eine Feature Map der Größe 56×42 Pixel.

Um daraus eine Umgebungsrepräsentation $\boldsymbol{u} \in \mathbb{R}^{\check{z}}$ eines lokalen Merkmals zu erstellen, werden für jeden Kanal c diejenigen Werte in der Feature Map F_c durch Summierung oder Maximumbildung aggregiert, die räumlich in der größeren Umgebung $\bar{U}(\boldsymbol{f}_i)$ des lokalen Merkmals $\boldsymbol{f}_i$ entstehen, wobei im Folgenden ein Querstrich über den Größen bedeutet, dass sie sich auf

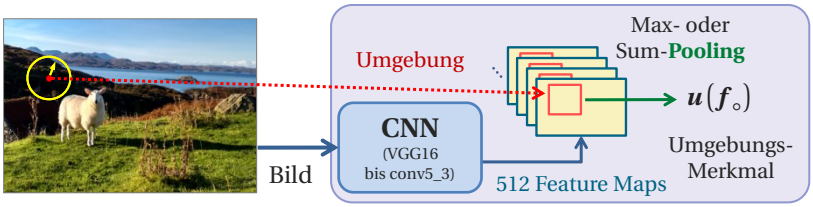

Abbildung 5.8.: Berechnung eines Umgebungsmerkmals basierend auf dem faltenden neuronalen Netz *VGG16*: Das lokale Merkmal (im Ursprungsbild in Gelb dargestellt) definiert durch seine Position im Bild und seine Skalierung eine quadratische Region (rot) in den Feature Maps. Pro Feature Map, also pro Faltungskanal werden die Werte in der Region anhand des Maximums oder der Summe aggregiert, sodass sich ein 512-dimensionaler Vektor für das Umgebungsmerkmal ergibt.

die Feature Maps beziehen und nicht auf das Eingangsbild:

$$u_c^{\text{sum}}(\boldsymbol{f}_i) = \sum_{(\bar{x}_j,\bar{y}_j)\in\bar{U}(\boldsymbol{f}_i)} F_c(\bar{x}_j,\bar{y}_j)\ , \tag{5.4}$$

$$u_c^{\max}(\boldsymbol{f}_i) = \max_{(\bar{x}_j,\bar{y}_j)\in\bar{U}(\boldsymbol{f}_i)} F_c(\bar{x}_j,\bar{y}_j)\ . \tag{5.5}$$

Abbildung 5.8 veranschaulicht für ein lokales Merkmal die Berechnung seines Umgebungsmerkmals. Die Region $\bar{U}(\boldsymbol{f}_i)$, in der die Werte in jeder Feature Map der Größe $W \times H$ aggregiert werden, wird von der Position (x_i, y_i) und der Skalierung σ_i des jeweiligen lokalen Merkmals $\boldsymbol{f}_i$ im Bild der Breite X und Höhe Y bestimmt und wird aus Gründen der effizienteren Berechnung als quadratisch modelliert:

$$\begin{aligned}\bar{U}(\boldsymbol{f}_i) = \Big\{(\bar{x},\bar{y}) \in \{1,\dots,W\}\times\{1,\dots,H\} \;\Big|\; \\ \left(\left|\bar{x} - \frac{x_i}{X}W\right| < \zeta_{CNN}\sigma_i\right) \wedge \\ \left(\left|\bar{y} - \frac{y_i}{Y}H\right| < \zeta_{CNN}\sigma_i\right)\Big\}\ .\end{aligned} \tag{5.6}$$

Im Vergleich zur idealen kreisrunden Region ergeben sich dabei aber nur sehr geringe Einschränkungen in Bezug auf die Rotationsinvarianz. Die abgebildeten Objekte und Szenen in den verwendeten Datensätzen sind außerdem überwiegend in 90-Grad-Schritten gedreht, sodass die quadratische

Region in diesen Fällen ein guter Kompromiss darstellt.

Im Anschluss an die Aggregierung werden die Umgebungsmerkmale einer ℓ_2-Normierung unterzogen. Neben der Eingangsbildgröße des CNNs und der Wahl der Aggregierungsmethode ist die relative Größe ζ_{CNN} der Umgebung ein entscheidender Parameter der vorgeschlagenen Umgebungsrepräsentation. Wenn sie zu klein gewählt wird, enthält sie zu wenig Informationen, um die korrekten von den inkorrekten BoW Korrespondenzen zu unterscheiden. Wird sie zu groß gewählt, degeneriert das Umgebungsmerkmal hingegen zu einer globalen Bildrepräsentation.

In Abbildung 5.9 ist dieser Sachverhalt für die in Kapitel 4.2 beschriebene Menge von korrekten BoW-Korrespondenzen $\mathcal{K}_k^{Pa}$ veranschaulicht. Für $\zeta_{CNN} = 8$ aggregieren 70% der Umgebungsmerkmale ihre Information aus weniger als 10% der Einträge der Feature Map, wohingegen für $\zeta_{CNN} = 50$ bereits mehr als die Hälfte der Umgebungsmerkmale sämtliche Einträge der Feature Map aggregieren und somit identisch sind, sofern sie aus demselben Bild stammen.

Für $\zeta_{CNN} = 20$ zeigt Abbildung 5.10 die euklidischen Abstände der Umgebungsmerkmale im 512-dimensionalen Raum für jedes Merkmalspaar der korrekten und inkorrekten BoW-Korrespondenzen. Als Histogramm zusammengefasst ist auch hier wieder erkennbar, dass die Umgebungsmerkmale der nichtkorrespondierenden BoW-Merkmale in der Regel eine größere euklidische Distanz aufweisen, während die Umgebungen von korrespondierenden BoW-Merkmalen zu Vektoren führen, die näher beieinander liegen.

Zur Untersuchung des Einflusses der Netzgröße und der Umgebungsgröße zeigen die Abbildungen 5.11 und 5.12 die Ergebnisse im Rahmen des Evaluationsframeworks. Die Falsch-Negativ-Raten und Falsch-Positiv-Raten wurden nach Quantisierung der Umgebungsmerkmale mit einem Codebook der Größe $\check{k} = 3025$ ermittelt. Die Ergebnisse zeigen die Aggregierung basierend auf Summierung (Gleichung 5.4), wobei sich folgendes feststellen lässt:

- **Netzgröße:** Die einheitlich in hoher Auflösung vorliegenden Bilder der Datensätze OXFORD5K, PARIS6K und HOLIDAYS (längste Seite jeweils 1 024 Pixel) profitieren von einer großen Netzgröße (896). Beim LANDMARKS Datensatz hingegen liegen viele Bilder in deutlich geringerer Auflösung vor, sodass sich für die mittlere Netzgröße (448) die besten Resultate ergeben. Dies erscheint plausibel, da dort bei Verwendung

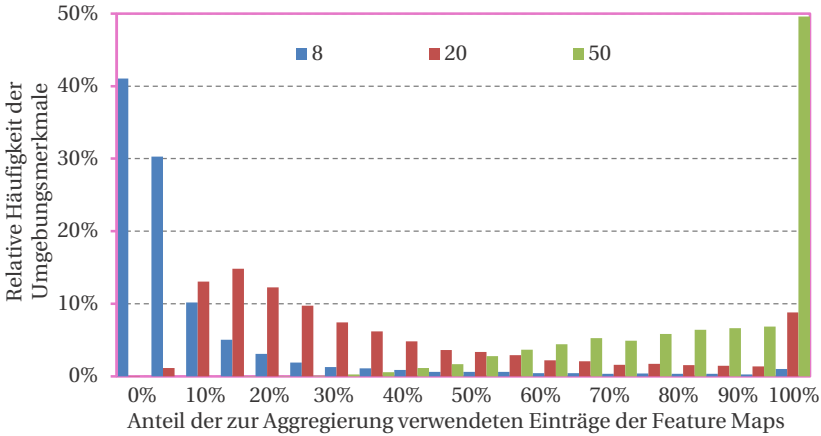

Abbildung 5.9.: Histogramm über den Anteil (in 5%-Intervallen) der Einträge der Feature Maps, die für die Umgebungsmerkmale aus $\mathcal{K}_k^{Pa}$ aggregiert werden, für die drei verschiedenen Umgebungsgrößen 8, 20 und 50.

des größeren Netzes die Bilder zunächst intern hochskaliert werden und die erste Faltungsschicht somit weniger Bildinformation in den jeweiligen rezeptiven Feldern zur Verfügung hat. Mit anderen Worten gelingt es der ersten Faltungsschicht dann nicht mehr, die durch die Filterparameter repräsentierten elementaren Bildstrukturen (Kanten, Ecken, etc.) zu erkennen, da sie im hochskalierten Eingangsbild aufgrund der Interpolation weniger ausgeprägt sind.

- **Umgebungsgröße:** Für OXFORD5K und PARIS6K ergibt sich die geringste FNR bei einer relativen Umgebungsgröße von $\zeta_{CNN} = 8$ bis $\zeta_{CNN} = 15$, wohingegen HOLIDAYS und LANDMARKS von noch größeren Merkmalsumgebungen ($\zeta_{CNN} = 30$) profitieren. Eine ähnliche Beobachtung ergab sich bereits bei den FV-basierten Umgebungsmerkmalen (Abbildung 5.4).

- **Falsch-Positiv-Rate:** Insbesondere bei LANDMARKS steigt die FPR bei großen Umgebungsgrößen (etwa ab $\zeta_{CNN} = 15$) deutlich an, wie die Abbildung 5.12 zeigt – vermutlich, weil in den meisten Bildern im oberen Bildbereich Himmel oder Wolken und im unteren Bildbereich Gras oder Asphalt sichtbar sind, sodass das Umgebungsmerkmal bei

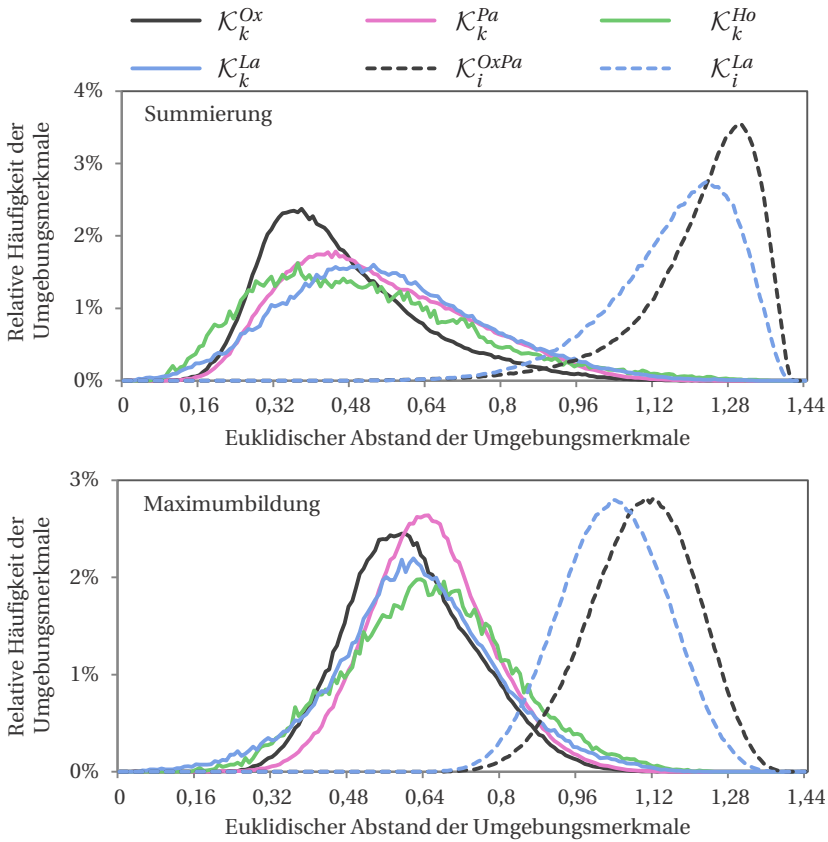

Abbildung 5.10.: Histogramm über die Abstände der Umgebungsmerkmale von korrespondierenden (durchgezogene Linien, basierend auf den Korrespondenzmengen $\mathcal{K}_k^*$) und nichtkorrespondierenden Merkmalen (gestrichelte Linien, basierend $\mathcal{K}_i^*$) für die beiden Aggregationsvarianten Summierung und Maximumbildung mit Umgebungsgröße $\zeta_{CNN} = 20$ und der Netzgröße 448.

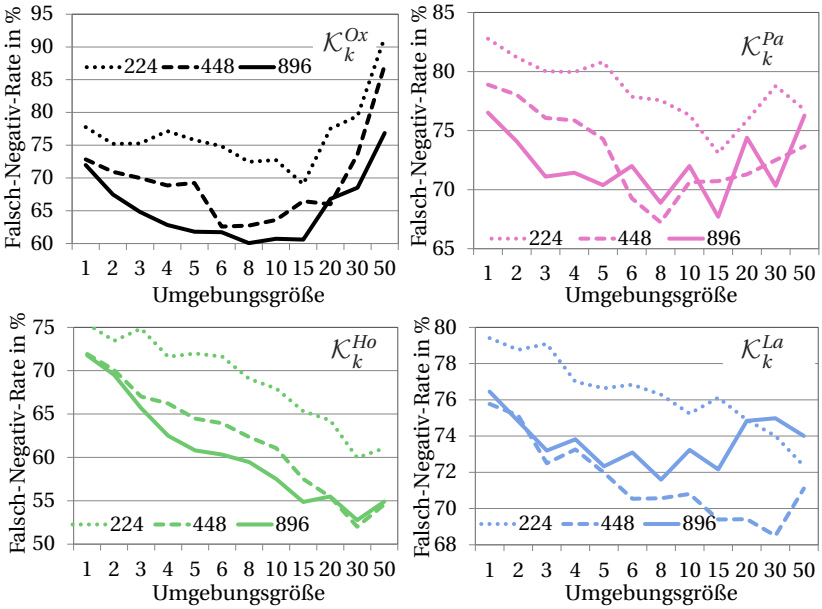

Abbildung 5.11.: Evaluation der CNN-basierten Umgebungsmerkmale in Bezug auf die Größe der Umgebung und die drei Netzgrößen 224, 448 und 896. Für jede der vier Mengen von korrekten Korrespondenzen $\mathcal{K}_k^{Ox}$, $\mathcal{K}_k^{Pa}$, $\mathcal{K}_k^{Ho}$ und $\mathcal{K}_k^{La}$ ist die Falsch-Negativ-Rate angegeben, d. h. der Anteil derjenigen Korrespondenzen, deren lokale Merkmale bezüglich ihrer quantisierten Umgebungsmerkmale nicht übereinstimmen, obwohl sie dasselbe Objekt beschreiben.

der Unterscheidung der eigentlichen Sehenswürdigkeiten von diesen typischen Umgebungen negativ beeinflusst wird.

Zur besseren Vergleichbarkeit wird für die weiteren Analysen in diesem Abschnitt eine einheitliche Umgebungsgröße von $\zeta_{CNN} = 15$ betrachtet. Für die Datensätze OXFORD5K, PARIS6K und HOLIDAYS wird die Netzgröße 896 verwendet und für LANDMARKS die Netzgröße 448.

Einen möglichen Nachverarbeitungsschritt der Umgebungsmerkmale stellt auch hier das Whitening dar, da es sich in der inhaltsbasierten Bildsuche mit globalen CNN-Merkmalen in der Regel als vorteilhaft erweist. Die notwendigen Daten für das Whitening, also die Eigenwerte der Kovarianz-

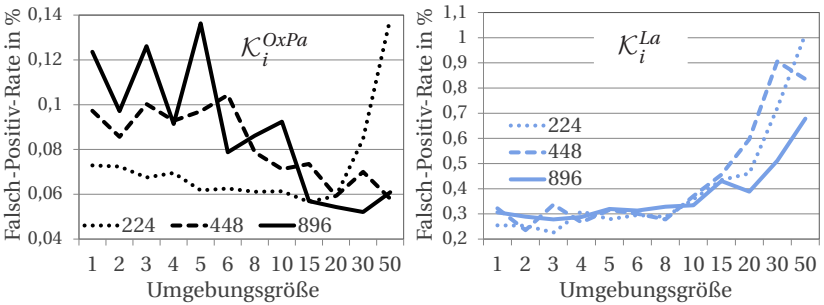

Abbildung 5.12.: Evaluation der CNN-basierten Umgebungsmerkmale in Bezug auf die Größe der Umgebung und die drei Netzgrößen 224, 448 und 896. Für die beiden Mengen von inkorrekten Korrespondenzen $\mathcal{K}_i^{OxPa}$ und $\mathcal{K}_i^{La}$ ist die Falsch-Positiv-Rate angegeben, d. h. der Anteil derjenigen Korrespondenzen, deren lokale Merkmale bezüglich ihrer quantisierten Umgebungsmerkmale übereinstimmen, obwohl sie unterschiedliche Objekte beschreiben.

matrix, wurden – wie für die Quantisierung auch – aus den Umgebungsmerkmalen der Merkmalsmenge $\mathcal{F}_M^{Ox}$ berechnet. Tabelle 5.2 stellt die Ergebnisse mit und ohne Whitening für die beiden Aggregationsvarianten Summierung und Maximumbildung dar. Das Whitening verbessert die Falsch-Negativ-Rate um ein paar Prozentpunkte während sich die Falsch-Positiv-Rate in etwa verdoppelt. Die Aggregierung durch Summierung liefert fast durchweg bessere Ergebnisse als die Variante mit Maximumbildung, die nur bezüglich der Falsch-Positiv-Rate beim LANDMARKS Datensatz geringe Vorteile bringt.

Tabelle 5.2.: Einfluss des Whitenings der CNN-basierten Umgebungsmerkmale auf die Falsch-Negativ-Rate (oben; in %) und die Falsch-Positiv-Rate (unten; in %) der korrekten bzw. inkorrekten Korrespondenzmengen.

Aggregation durch:	Summierung		Maximumbildung	
Whitening:	Nein	Ja	Nein	Ja
Oxford ($\mathcal{K}_k^{Ox}$)	60,60	53,32	65,92	63,31
Paris ($\mathcal{K}_k^{Pa}$)	67,71	67,19	78,05	75,94
Holidays ($\mathcal{K}_k^{Ho}$)	54,85	51,43	68,22	68,52
Landmarks ($\mathcal{K}_k^{La}$)	69,39	65,94	72,90	74,10
Oxford/Paris ($\mathcal{K}_i^{OxPa}$)	0,057	0,108	0,070	0,161
Landmarks ($\mathcal{K}_i^{La}$)	0,455	1,156	0,403	0,743

5.4. Vergleich der Umgebungsrepräsentationen

Durch das in Kapitel 4 beschriebene Evaluationsframework lassen sich die beiden vorgestellten Umgebungsrepräsentationen direkt miteinander vergleichen. In den Abbildungen 5.13 und 5.14 werden zunächst die originalen Umgebungsmerkmale – also ohne PCA oder Whitening – einander gegenübergestellt. Für jeden der vier Datensätze OXFORD5K, PARIS6K, HOLIDAYS und LANDMARKS ergibt sich ein Diagramm, das die Falsch-Negativ-Raten für korrekten Korrespondenzen $\mathcal{K}_k^*$ und die Falsch-Positiv-Raten für die inkorrekten Korrespondenzen $\mathcal{K}_i^*$ darstellt. Die FPR ist in den Diagrammen dieses Kapitels in logarithmischer Skalierung dargestellt. Verfahren, die links unten im Diagramm eingezeichnet sind, übertreffen prinzipiell die Verfahren, die rechts oben dargestellt sind, wobei, wie in Abschnitt 4.3 beschrieben, keine direkte Abwägung zwischen den beiden Fehlermaßen möglich ist. $\mathcal{K}_i^{OxPa}$ wird dabei im Folgenden auch für die Auswertungen des HOLIDAYS Datensatzes verwendet, um auch dort eine Gegenüberstellung von FNR und FPR zu erhalten. Mit CNN-max und CNN-sum werden im weiteren Verlauf der Arbeit jeweils die auf CNN basierenden Umgebungsmerkmale bezeichnet, die durch Maximumbildung bzw. Summierung der Einträge der Feature Maps aggregiert wurden.

Über die Datensätze hinweg und für die verschiedenen Umgebungs-

größen ζ_{CNN} beziehungsweise ζ_{FV} schneiden die CNN-basierten Umgebungsmerkmale klar besser ab als die Fisher Vektor-basierten. Beim HOLIDAYS Datensatz beispielsweise (Abbildung 5.14 oben) liegen die CNN-Sum-Umgebungsmerkmale bei der FNR etwa 20 Prozentpunkte vorn und bei der FPR schneiden sie etwa um den Faktor zwei besser ab. Während einer Suchanfrage würden somit deutlich mehr korrekte Korrespondenzen erhalten bleiben und gleichzeitig könnten doppelt so viele falsche Korrespondenzen von der weiteren Verarbeitung ausgeschlossen werden. Als Erklärung für die Überlegenheit der CNN-Varianten bieten sich folgende Aspekte an:

- Die FV-Repräsentation baut auf den lokalen SIFT Merkmalen auf und ist daher auf dessen Leistungsfähigkeit begrenzt - so können beispielsweise keine Farbinformationen in die Repräsentation einfließen und bei starken Beleuchtungsunterschieden oder invertiertem Kontrast können sich keine Ähnlichkeiten mehr ergeben.

- Obwohl beide Repräsentationen gelerntes Wissen beinhalten, unterscheiden sich die Mengen der dafür verwendeten Daten und der Modellparameter sehr deutlich: Die FV-Repräsentation erfasst die Eigenschaften des wenige Tausend Bilder umfassenden OXFORD5K Datensatzes mit den GMM-Modellparametern $(\alpha, \boldsymbol{\mu}, \Sigma)$ für maximal $K = 64$ Komponenten der 68-dimensionalen Eingangsvektoren, was in Summe $64 \cdot (1 + 68 + 68) = 8\,768$ Parametern entspricht. Das VGG16 Netz dagegen wurde mit 1,2 Millionen Bildern trainiert und umfasst das Gelernte in 144 Millionen Parametern.

Die Auswirkungen der Nachverarbeitung mittels PCA und Whitening sind in den Abbildungen 5.15 und 5.16 dargestellt. Im Interesse einer besseren Übersichtlichkeit sind die jeweils kleinsten Modelle (für CNN die Netzgröße 224 und für FV die Variante mit 16 GMM Komponenten) nicht dargestellt. Bei Anwendung der PCA wird für die CNN-Variante keine Dimensionsreduktion vorgenommen und für die FV-Variante nur die in Abschnitt 5.2 beschriebene marginale Reduktion, d. h. die gestrichelten Werte basieren auf den 4 096-dimensionalen FV und die durchgezogenen Werte auf 8192-dimensionalen Fisher Vektoren, die dann für die Evaluation quantisiert werden.

Generell ist festzustellen, dass für die CNN-Sum-Umgebungsmerkmale, die PCA und das Whitening für die FNR vorteilhaft ist, wenn auch die FPR dadurch ansteigt. Für CNN-Max gilt dies nicht pauschal wie insbesondere die Werte für HOLIDAYS und LANDMARKS zeigen (Abbildung 5.16). Die hochdimensionalere Fisher Vektor Repräsentation hingegen profitiert von PCA und Whitening noch deutlicher in Bezug auf die FNR und erreicht dadurch den Bereich der CNN-basierten Werte. Dieser Gewinn geht allerdings mit einer dramatischen Erhöhung der FPR einher, sodass im Gesamten betrachtet die FV-Variante den CNN-Varianten wieder klar unterlegen ist. Dass im OXFORD5K Datensatz (Abbildung 5.15 oben), auf dem sämtliche Modelle trainiert wurden, die PCA und das Whitening für die ausgedehnteren Umgebungsgrößen sogar Nachteile bezüglich beider Evaluationsmaße bringt, deutet zudem auf Overfitting-Effekte bei den Fisher Vektoren hin. Dies erscheint plausibel, denn durch die ausgedehnten Umgebungsgrößen können dieselben lokalen SIFT Merkmale durchaus sowohl für die Berechnung der PCA-Parameter als auch für Umgebungsmerkmale der beiden zur Evaluation verwendeten Mengen $\mathcal{K}_k^{Ox}$ und $\mathcal{K}_i^{OxPa}$ eine Rolle gespielt haben.

Fazit

- Die CNN-basierten Repräsentationen übertreffen die Fisher Vektor-basierten Varianten deutlich bezüglich beider Evaluationsmaße (FNR und FPR).
- Die besten Ergebnisse erzielte die CNN-basierte Repräsentation, die die Informationen der Feature Maps durch Summierung aggregiert. Sie kann außerdem am besten von PCA und Whitening profitieren, um die FNR zu verbessern – allerdings zu Lasten der FPR.

Ob die Informationen in den einzelnen Repräsentationen teilweise komplementär sind und sich somit ergänzen könnten, wird im nächsten Abschnitt untersucht, indem verschiedene Kombinationen verglichen werden.

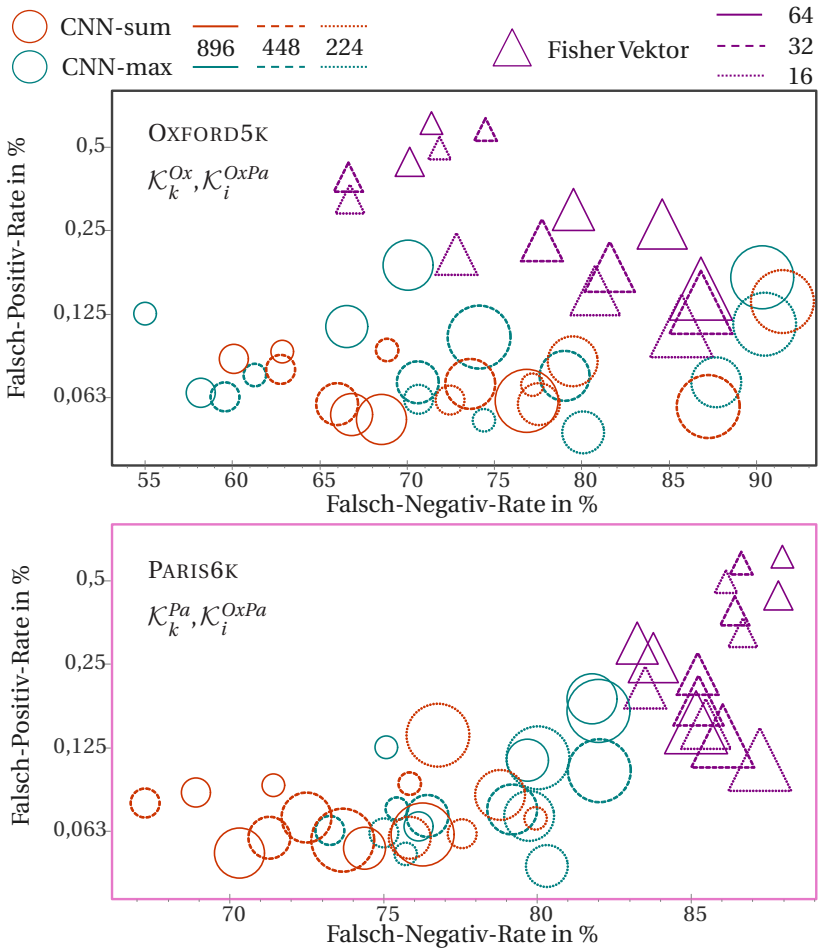

Abbildung 5.13.: Vergleich der auf FV und CNN basierenden Umgebungsmerkmale (ohne PCA/Whitening) für die Datensätze OXFORD5K und PARIS6K. Die Größe der Kreise bzw. Dreiecke signalisiert die verschiedenen Umgebungsgrößen ζ_{CNN} bzw. ζ_{FV} (4, 8, 20, 30 und 50). Die Linienarten kennzeichnen für CNN die Netzgrößen (224, 448 und 896) und für FV die Anzahl K der GMM-Komponenten (16, 32 und 64).

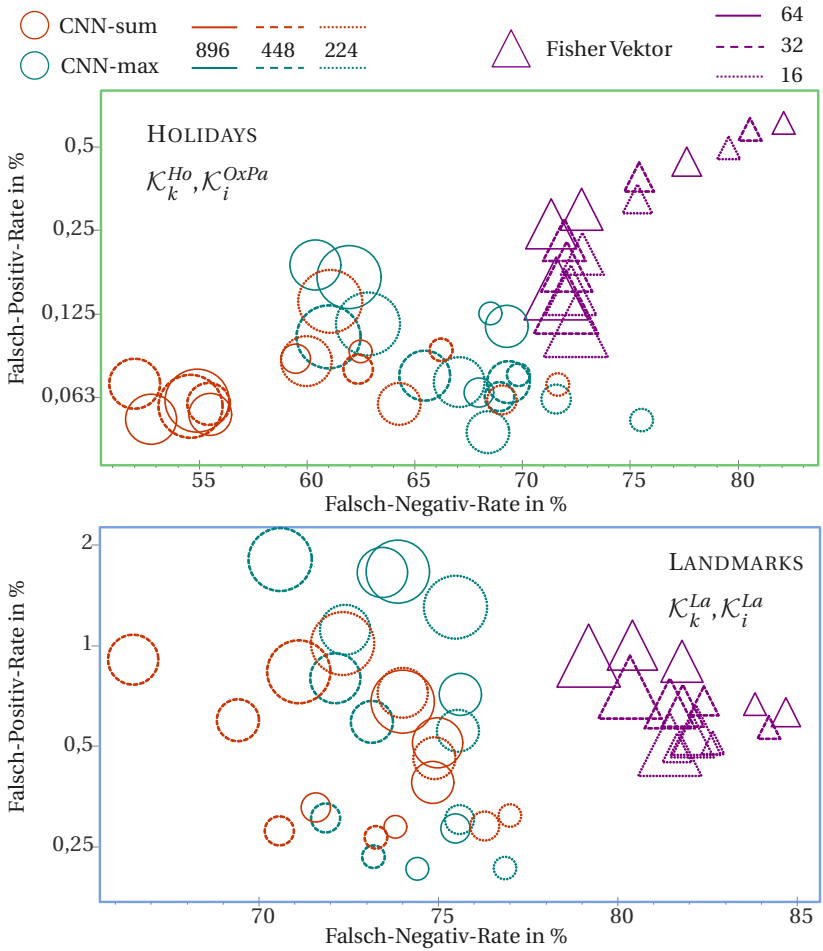

Abbildung 5.14.: Vergleich der auf FV und CNN basierenden Umgebungsmerkmale (ohne PCA/Whitening) für die Datensätze HOLIDAYS und LANDMARKS. Die Größe der Kreise bzw. Dreiecke signalisiert die verschiedenen Umgebungsgrößen ζ_{CNN} bzw. ζ_{FV} (4, 8, 20, 30 und 50). Die Linienarten kennzeichnen für CNN die Netzgrößen (224, 448 und 896) und für FV die Anzahl K der GMM-Komponenten (16, 32 und 64).

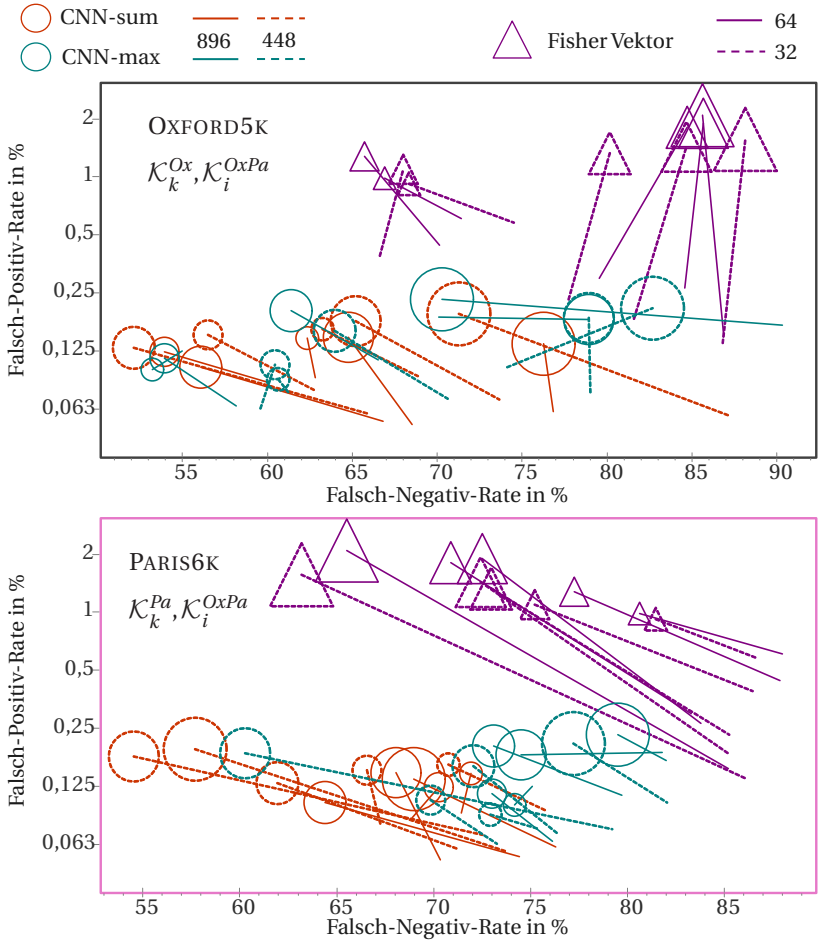

Abbildung 5.15.: Einfluss von PCA und Whitening auf die Umgebungsmerkmale für die Datensätze OXFORD5K und PARIS6K. Die Größe der Kreise bzw. Dreiecke signalisiert die verschiedenen Umgebungsgrößen ζ_{CNN} bzw. ζ_{FV} (4, 8, 20, 30 und 50). Die Linienarten kennzeichnen für CNN die Netzgrößen (448 und 896) und für FV die Anzahl K der GMM-Komponenten (32 und 64). Die Position der Kreise bzw. Dreiecke stellen die Werte *nach* PCA und Whitening dar, während die Enden der damit verbundenen Linien die originalen Merkmalswerte aus Abbildung 5.13 angeben, d. h. die Längen der Linien visualisieren die Veränderungen durch PCA und Whitening.

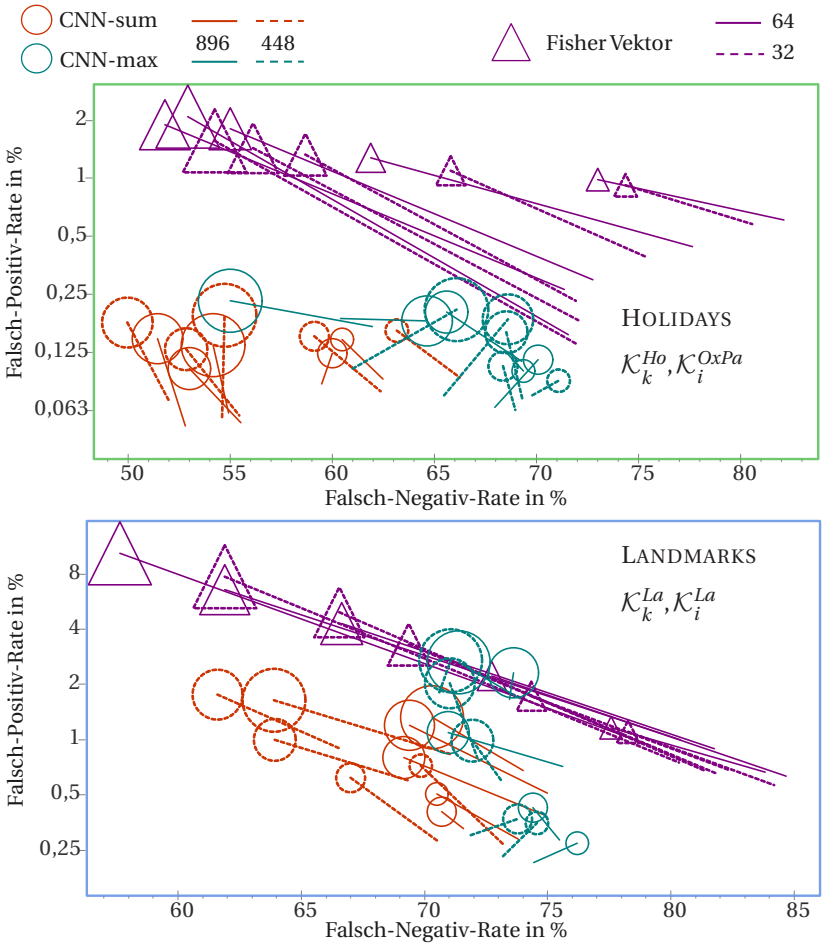

Abbildung 5.16.: Einfluss von PCA und Whitening auf die Umgebungsmerkmale für die Datensätze HOLIDAYS und LANDMARKS. Die Größe der Kreise bzw. Dreiecke signalisiert die verschiedenen Umgebungsgrößen ζ_{CNN} bzw. ζ_{FV} (4, 8, 20, 30 und 50). Die Linienarten kennzeichnen für CNN die Netzgrößen (448 und 896) und für FV die Anzahl K der GMM-Komponenten (32 und 64). Die Position der Kreise bzw. Dreiecke stellen die Werte *nach* PCA und Whitening dar, während die Enden der damit verbundenen Linien die originalen Merkmalswerte aus Abbildung 5.14 angeben.

5.5. Kombinationen

Bei der Analyse einer BoW-Korrespondenz zwischen zwei Bildern können die beiden lokalen Umgebungen durch die quantisierten Umgebungsmerkmale verglichen werden. Falls mehrere Varianten von Umgebungsmerkmalen – in dieser Arbeit also die beiden auf CNN basierenden und die Variante mit Fisher Vektoren – zum Einsatz kommen, kann, wie in Kapitel 3 motiviert, auch eine Kombination realisiert werden. Eine Ähnlichkeit kann dann auf zwei Arten definiert werden: entweder als ODER-Kombination oder als UND-Kombination. Bei der Letzteren müssen beide Varianten bezüglich ihrer quantisierten Werte übereinstimmen während bei der ODER-Kombination mindestens eine Variante übereinstimmen muss.

Die ODER-Kombination beider CNN-Varianten, also von CNN-Sum und CNN-Max, ist in Abbildung 5.18 als Übersicht über alle Datensätze veranschaulicht, wobei die ODER-Kombination in den folgenden Diagrammen mit „|“ abgekürzt wird und die UND-Kombination mit „&“. Durch die Kombination reduzieren sich die Falsch-Negativ-Raten für alle Datensätze um deutliche 10 bis 20 Prozentpunkte wobei sich auch hier wieder die FPR um mindestens Faktor zwei erhöht.

In Abbildung 5.17 ist die UND-Kombination von CNN-sum und CNN-max gezeigt. Erwartungsgemäß steigt dort die FNR deutlich an, da beide Werte übereinstimmen müssen, um ein Falsch-Negativ-Ergebnis zu verhindern. Dafür sinkt wiederum die FPR sehr deutlich, denn die inkorrekten BoW-Korrespondenzen stimmen in der Regel nur äußert selten bezüglich beider Varianten überein. Bei den beobachteten FPR-Werten von 0,003% würde während einer Suchanfrage nur etwa eine von 33 000 inkorrekten BoW-Korrespondenzen die Suchergebnisse beeinträchtigen.

Abbildung 5.18 geht hinsichtlich der Kombinationsmöglichkeiten außerdem noch einen Schritt weiter und verbindet alle drei vorgestellten Varianten in einer ODER-Kombination: beide CNN-basierten, sowie das FV-basierte Umgebungsmerkmal. Obwohl, wie in den Abbildungen 5.13 und 5.14 festgestellt, die FV-Variante einzeln betrachtet schlechter abschneidet als jede der beiden CNN-Varianten, vermag sie in dieser Dreierkombination die FNR jedoch um einige Prozentpunkte zu verbessern. In den Fisher Vektoren, die wiederum auf den SIFT Merkmalen aufbauen, ist demnach noch Informa-

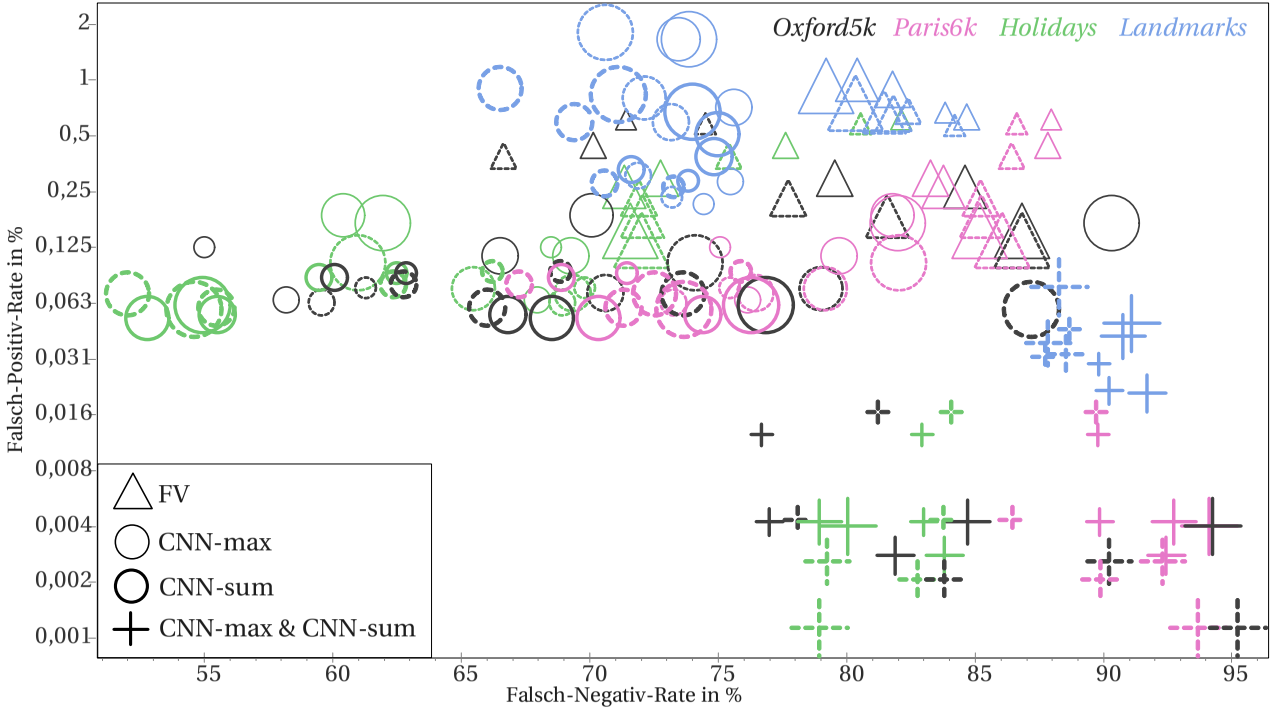

Abbildung 5.17.: UND-Kombination der beiden durch Summierung und Maximumbildung aggregierten CNN-basierten Umgebungsmerkmale (ohne PCA/Whitening). Die Größe der Kreise bzw. Dreiecke signalisiert die verschiedenen Umgebungsgrößen ζ_{CNN} bzw. ζ_{FV} (4, 8, 20, 30 und 50). Die Linienarten kennzeichnen für CNN die Netzgrößen (gestrichelt: 448, durchgezogen: 896) und für FV die Anzahl K der GMM-Komponenten (gestrichelt: 32, durchgezogen: 64).

tionsgehalt kodiert, der durch keine der CNN-Varianten bereits erfasst ist. Prinzipiell sind außerdem noch weitere Kombinationen denkbar:

- eine UND-Kombination aller drei Varianten: sie liefert jedoch für die Praxis untaugliche Falsch-Negativ-Raten jenseits der 95%. Außerdem können die Falsch-Positiv-Raten durch die begrenzte Größe der Korrespondenzmengen ($\hat{\mathcal{K}}_i^*$) nicht mehr repräsentativ bestimmt werden, denn für den Zähler des FPR Terms aus Gleichung 4.4 ergeben sich nur wenige Merkmalspaare.

- eine Kombination der FV-Variante mit nur einer CNN-Variante. Dies ergab jedoch keinerlei Vorteile gegenüber der Kombination beider CNN-Varianten.

Bei allen vorgestellten Kombinationen wurden jeweils die Werte derselben Umgebungsgröße und derselben Netzgröße kombiniert. Bei der Dreifach-Kombination wurde die Netzgröße 448 mit der FV-Variante mit 32 GMM Komponenten kombiniert und die Netzgröße 896 entsprechend mit 64 GMM Komponenten.

Der Einfluss von PCA und Whitening für die Kombinationen wird schließlich in Abbildung 5.19 dargestellt. Die ODER-Kombination der beiden CNN-Varianten kann dadurch fast durchgängig einen weiteren „Verschiebungsschritt“ zwischen FNR und FPR erreichen während für die UND-Kombination die PCA und das Whitening nicht in allen Fällen eine Verbesserung darstellt. Da in Abbildung 5.16 die Anwendung von PCA und Whitening bereits für CNN-Max keine nennenswerten Vorteile ergab, scheint sich eine weitere Kombination anzubieten: eine ODER-Kombination aus der CNN-sum-Variante *mit* PCA und Whitening und der CNN-max-Variante *ohne* PCA und Whitening. Interessanterweise ergaben sich dabei allerdings für beide Evaluationsmaße schlechtere Resultate als die in Abbildung 5.19 (oben) gezeigten Werte, in denen PCA und Whitening für beide Varianten erfolgt.

Fazit

Wenn der Fokus, wie in Abschnitt 4.3 erörtert, in erster Linie auf einer geringen Falsch-Negativ-Rate liegt, werden die besten Ergebnisse bei der Integration nur eines Umgebungsmerkmals für die durch Summierung aggregierte CNN-Variante erzielt. Eine deutliche Verbesserung ergibt sich durch die

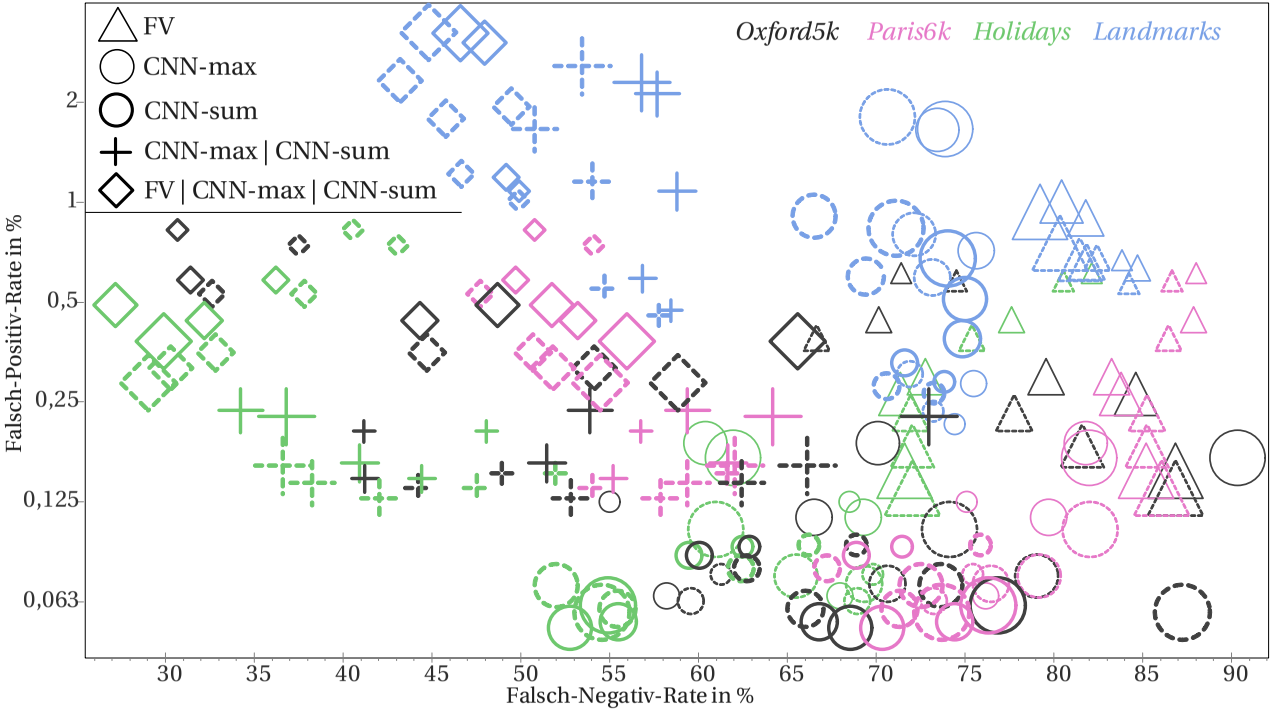

Abbildung 5.18.: ODER-Kombination der beiden durch Summierung und Maximumbildung aggregierten CNN-basierten sowie der FV-basierten Umgebungsmerkmale (ohne PCA/Whitening). Die Größe der Strukturen signalisiert die verschiedenen Umgebungsgrößen ζ_{CNN} bzw. ζ_{FV} (4, 8, 20, 30 und 50). Die Linienarten kennzeichnen für CNN die Netzgrößen (gestrichelt: 448, durchgezogen: 896) und für FV die Anzahl K der GMM-Komponenten (gestrichelt: 32, durchgezogen: 64).

Hinzunahme der Variante mit Maximumbildung im Rahmen einer ODER-Kombination. PCA und Whitening führen zu einer nochmaligen, wenn auch geringeren Verbesserung, die in etwa vergleichbar ist mit der Hinzunahme der FV-basierten Umgebungsmerkmale (CNN-sum ODER CNN-max ODER FV). Die Falsch-Positiv-Rate, auf der anderen Seite, wird durch jeden dieser zusätzlichen Verbesserungsschritte in etwa verdoppelt.

Um zu prüfen, wie gut diese Kompromisse zwischen FNR und FPR die letztlichen Suchergebnisse eines Gesamtsystems zur inhaltsbasierten Bildsuche vorgeben, werden im nächsten Kapitel die Umgebungsrepräsentationen für sämtliche lokalen Merkmale in allen Bilden der verwendeten Datensätze ausgewertet. Eine weitere interessante Frage wird sein, ob bei einer UND-Kombination, die praktisch sämtliche inkorrekten Korrespondenzen ausschließen kann, noch genügend korrekte Korrespondenzen erhalten bleiben, um relevante Bilder in großen Datenbanken zu finden.

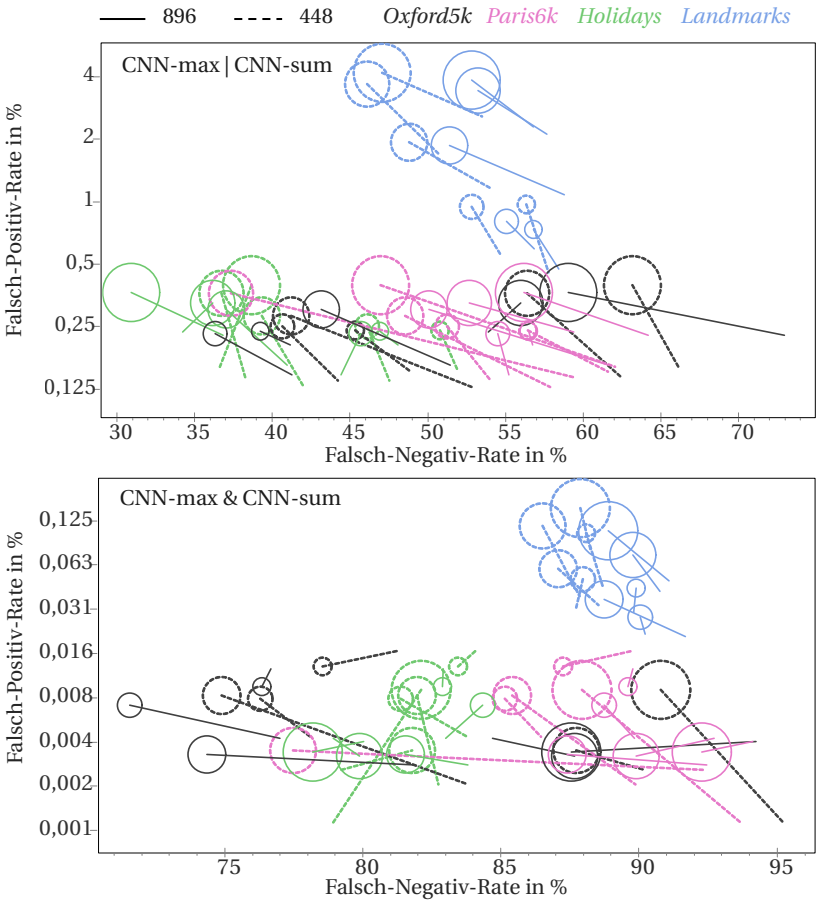

Abbildung 5.19.: ODER- bzw. UND-Kombination der beiden durch Summierung und Maximumbildung aggregierten CNN-basierten Umgebungsmerkmale, nachdem diese einer PCA und einem Whitening unterzogen wurden. Die Größe der Kreise bzw. Dreiecke signalisiert die verschiedenen Umgebungsgrößen ζ_{CNN} bzw. ζ_{FV} (4, 8, 20, 30 und 50). Die Linienarten kennzeichnen die CNN Netzgrößen (gestrichelt: 448, durchgezogen: 896). Die Position der Kreise stellt die Kombinationen der Werte *nach* PCA und Whitening dar, während die Enden der damit verbundenen Linien die Kombinationen der jeweiligen originalen Merkmalswerte angeben.

6

Evaluation im Rahmen der inhaltsbasierten Bildsuche

Im vorangegangenen Kapitel wurden verschiedene Varianten von Umgebungsmerkmalen vorgestellt und mit Hilfe des Evaluationsframeworks aus Kapitel 4 miteinander verglichen. Die dabei genutzten Falsch-Negativ- und Falsch-Positiv-Raten modellieren die interne Sicht eines Suchsystems, das die Umgebungsmerkmale verwendet, um inkorrekte BoW-Korrespondenzen während der Verarbeitung einer Suchanfrage auszuschließen. Die letztlichen Auswirkungen auf die Qualität der Suchergebnisse – also die Reihenfolge und die Ähnlichkeitswerte der Datenbankbilder – lassen sich daraus allerdings aus verschiedenen Gründen nicht direkt ermitteln:

- In die Berechnung der FNR und FPR fließen alle BoW Korrespondenzen gleichberechtigt ein. Abhängig vom Anfragebild und von den Datenbankbildern sind sie aber unterschiedlich schwierig bzw. ablenkend für die Ergebnisse einer Suchanfrage.
- Die *TF-IDF*-Gewichtung aus Gleichung 2.7 ist in den FNR und FPR Termen nicht abgebildet.
- Es wurde nur ein Bruchteil der kombinatorisch möglichen Korrespondenzen verwendet, um für diese die Umgebungsmerkmale zu berechnen.

Um den Gewinn für ein Gesamtsystem zur inhaltsbasierten Bildsuche zu ermitteln, ist es daher unumgänglich, die Verwendung der Umgebungsmerkmale direkt mit entsprechenden Suchanfragen zu evaluieren [Man17a]. Dafür wird die, in der inhaltsbasierten Bildsuche übliche Evaluationsmethodik verwendet, die auf annotierten Datensätzen basiert, und im folgenden Abschnitt beschrieben wird. Anschließend werden die gewählten Parameter für die Evaluation motiviert und die Ergebnisse vorgestellt.

6.1. Evaluationsmethodik und -maß

Die Evaluation eines Systems für die inhaltsbasierte Bildsuche erfolgt üblicherweise mit annotierten Daten, die aus drei Teilen bestehen:

1. **Anfragebilder**, typischerweise eingeteilt in Gruppen: Jede Gruppe besteht aus einem oder mehreren Anfragebildern, die ein Objekt oder eine Szene zeigen, die wiedergefunden werden soll.

2. **Referenzbilder**: Für jede Gruppe in den Anfragebildern existiert eine Menge von Referenzbildern, die dasselbe Objekt oder dieselbe Szene zeigen. Sie sollen während der einzelnen Suchanfragen basierend auf den Anfragebildern gefunden werden. Für jedes Anfragebild ist bekannt, welche Referenzbilder als ähnlich betrachtet werden und zwei Bilder aus verschiedenen Gruppen weisen keine relevanten Ähnlichkeiten auf.

3. **Ablenkungsbilder**: Sie dienen der Untersuchung der Skalierbarkeit der Modelle. Für die typischerweise eine Million oder mehr Ablenkungsbilder sollte sichergestellt sein, dass sie ebenfalls keine relevanten Ähnlichkeiten mit den Anfragebildern aufweisen. Gleichzeitig sollten sie aber genügend Ablenkungspotential besitzen, d. h. repräsentativ für die Herausforderungen einer potentiellen Anwendung sein. Bei der Suchanfrage mit einem Bild einer bestimmten Gruppe werden die Referenzbilder der übrigen Gruppen üblicherweise ebenfalls als Ablenkungsbilder angesehen.

Die Referenzbilder und die Ablenkungsbilder bilden die Datenbank und werden indexiert. Für jedes der n_A Anfragebilder wird anschließend eine

Suchanfrage durchgeführt, die als Ergebnis eine nach Ähnlichkeit sortierte Liste der indexierten Datenbankbilder liefert. Als Evaluationsmaß wird üblicherweise die *mean average precision* aus allen Ergebnislisten $\mathcal{L}_1, \ldots, \mathcal{L}_{n_A}$ berechnet:

$$map = \frac{1}{n_A} \sum_{i=1}^{n_A} ap(\mathcal{L}_i), \tag{6.1}$$

also der Mittelwert der *average precision (ap)* Werte der einzelnen Ergebnislisten. Die average precision für eine sortierte Ergebnisliste $\mathcal{L}_i$ einer Suchanfrage wiederum ist definiert als

$$ap(\mathcal{L}_i) = \frac{1}{\sum_{j=1}^{|\mathcal{L}_i|} t_j} \sum_{j=1}^{|\mathcal{L}_i|} t_j \frac{\sum_{j'=1}^{j} t_{j'}}{j}, \tag{6.2}$$

wobei t_j eine Indikatorvariable bezeichnet, die den Wert 1 annimmt, falls in der Ergebnisliste der Eintrag an Position j korrekt ist (also auf diesem Rang ein Referenzbild gefunden wurde) und 0 anderenfalls. Für jeden Rang mit einem korrekten Eintrag wird also die Genauigkeit (Anteil der korrekten Einträge) bis zu diesem Rang berechnet, und alle Werte werden gemittelt. Der Wertebereich ist somit $0 < map \leq 1$ und erreicht den Maximalwert 1, wenn für alle Anfragebilder jeweils alle zugehörigen Referenzbilder in den Ergebnislisten auf den ersten Rängen erscheinen und erst danach die Ablenkungsbilder folgen.

6.2. Daten- und Parameterauswahl

Als Datensätze werden wieder die in Kapitel 4.1 beschriebenen öffentlichen Datensätze verwendet. Die annotierten Datensätze OXFORD5K, PARIS6K, HOLIDAYS und LANDMARKS dienen jeweils als Anfrage- und Referenzbilder, wobei OXFORD5K und PARIS6K außerdem wenige tausend Ablenkungsbilder beinhalten. Als Anfragebild dient jeweils das komplette Bild. Als weitere

Ablenkungsbilder fungieren die eine Million[1] Bilder des MIRFLICKR1M Datensatzes. Tabelle 6.1 zeigt den Umfang der fünf verwendeten Datensätze.

Nachdem Evaluationsmethodik und Daten festgelegt sind, stellt sich als letztes noch die Frage nach den auszuwertenden Parametern. Generell wären hier die Ergebnisse für *alle* in Kapitel 5 untersuchten Parameterkombinationen interessant. Aus Laufzeitgründen kann dies aber nur für ausgewählte Parameter durchgeführt werden, denn für jedes zu indexierende Bild sind die folgenden Schritte erforderlich:

1. Berechnung der lokalen SIFT Merkmale
2. Quantisieren der Deskriptoren
3. Propagieren des Bildes durch das faltende neuronale Netz und Extraktion der Feature Maps
4. Für jedes lokale Merkmal:
   a) Berechnung eines Umgebungsmerkmals (bzw. mehrerer Varianten bei den Kombinationen)
   b) Gegebenenfalls Durchführung von PCA und Whitening
   c) Quantisierung des Umgebungsmerkmals
5. Indexierung der Merkmale in den Listen gemäß dem Inverted-File-Prinzip

Bei einer Gesamtdauer dieser Schritte von etwa einer Sekunde pro Bild mit der verwendeten Hardware (ein einzelner leistungsfähiger Server mit zwei CPUs, Details siehe Abschnitt 7.1) sind insgesamt elf Tage für die Indexierung erforderlich bei der verwendeten Datenbankgröße von einer Million Bildern. Die wenigen Suchanfragen, die für die Evaluation eines Datensatzes erforderlich sind (siehe letzte Spalte in Tabelle 6.1), fallen dagegen nicht

1 Angesichts der Größe dieses Datensatzes ist es nicht möglich, die Bilder manuell auf eventuelle Überlappungen mit den Anfragebildern der anderen Datensätzen zu überprüfen. Stattdessen wird ein halbautomatischer Prozess verwendet, bei dem die MIRFLICKR1M Bilder gesondert indexiert werden und anschließend die Suchergebnisse für die Anfragebilder aus den übrigen Datensätzen analysiert werden. In den jeweils ersten zwanzig Bildern der Ergebnislisten werden dann diejenigen erfasst, die augenscheinlich dieselbe Szene zeigen. Für jeden Datensatz ergibt sich so eine Liste von Ablenkungsbildern, die zwar indexiert werden, aber in der jeweiligen Evaluation außen vor bleiben. In Summe betrifft dies 2 182 der eine Million Bilder.

Tabelle 6.1.: Anzahl der Anfragebilder sowie der indexierten Bilder mit ihren lokalen Merkmalen, die für die Evaluation im Rahmen der inhaltsbasierten Bildsuche mit den verschiedenen Datensätzen verwendet werden.

Datensatz	Indexierte Bilder	Indexierte Merkmale	Anfragebilder
OXFORD5K	5 062	26 165 170	55
PARIS6K	6 392	30 336 151	55
HOLIDAYS	1 491	11 062 051	500
LANDMARKS	32 720	80 710 271	3 392
MIRFLICKR1M	999 859	1 277 447 284	-

ins Gewicht. Für die Auswertungen in diesem Kapitel werden daher die folgenden Parameter ausgewählt:

- Da in Kapitel 5.4 die **Umgebungsmerkmale** basierend auf den Fisher Vektoren praktisch in allen Belangen deutlich schlechter abschnitten als die auf neuronalen Netzen basierenden Umgebungsmerkmale, und sie zudem aufwändiger in der Berechnung sind, werden in diesem Kapitel nur die CNN Varianten analysiert.

- Bezüglich der **Netzgröße** lieferte das kleinste Netz (maximale Seitenlänge 224 Pixel) die schlechtesten Resultate. Das größte Netz (896) wiederum kann seine Vorteile, wie auf Seite 87 festgestellt, nur ausspielen, wenn alle Bilder in entsprechend großer Auflösung vorliegen. Dies ist für LANDMARKS nicht gegeben und vor allem für die Ablenkungsbilder aus dem MIRFLICKR1M Datensatz nicht, die in einer maximalen Seitenlänge von 500 Pixeln vorliegen. Damit die „Ablenkungsfähigkeit“ nicht eingeschränkt wird, wird als Kompromiss daher die mittlere Netzgröße von 448 für alle Auswertungen in diesem Kapitel verwendet.

- Hinsichtlich der **Umgebungsgröße** ζ_{CNN} werden die Werte 4, 8, 20 und 50 untersucht, da sich in Kapitel 5 diesbezüglich kein einheitliches Bild ergab.

- Beide Varianten der **Aggregierung** (Summierung und Maximumbildung), mit oder ohne Anwendung von PCA und Whitening sowie die in Kapitel 5.5 beschriebenen `UND` und `ODER` **Kombinationen**.

Der Aufbau des Systems entspricht den Darstellungen des grundlegenden Bag-of-(visual)-Words-Systems in Kapitel 2.2.2, wobei die Modelle aus den vorangegangenen Kapiteln verwendet werden, die alle auf den Bildern des OXFORD5K Datensatzes basieren:

- Das Codebook $\mathcal{C}$ für die Quantisierung der SIFT-Deskriptoren: Größe $k = 104\,976$, erzeugt durch hierarchisches k-Means-Clustering. Die Quantisierung selbst wird mit approximativer Nächste-Nachbar-Suche durchgeführt (randomisierte k-d-Bäume [Muj14] bei einer Genaugikeit von 90%, siehe Kapitel 4.2).

- Die jeweiligen Umgebungscodebooks für die Quantisierung der Umgebungsmerkmale: Größe $\check{k} = 3\,025$, erzeugt durch hierarchisches k-Means-Clustering der Umgebungsmerkmale, die für $\mathcal{F}_M^{Ox}$ berechnet wurden. Im Vergleich zum Codebook für die SIFT Merkmale wird hier die Quantisierung durch exakte Nächste-Nachbar-Suche auf der GPU realisiert, damit während der Indexierung sowohl CPU als auch GPU-Ressourcen vollständig ausgeschöpft werden können.

- Die Parameter für die PCA und das Whitening, ebenfalls ermittelt mit den Umgebungsmerkmalen, die für $\mathcal{F}_M^{Ox}$ berechnet wurden.

In den Listen $\mathcal{Q}_i^\star, i = 1,\ldots,k$ aus Gleichung 2.13, die das Inverted-File-Prinzip umsetzen, werden für jedes indexierte lokale Merkmal, abgesehen von seiner Bildnummer, nun auch die beiden quantisierten Werte der zugehörigen Umgebungsmerkmale (CNN-max und CNN-sum) gespeichert. Während einer Suchanfrage erhöhen dann aus jeder Liste nur diejenigen Merkmale den Akkumulator, die bezüglich dieser zusätzlichen Informationen übereinstimmen. In Kapitel 2.2.5 wurden die bislang dominierenden Verfahren zur Integration zusätzlicher Informationen in den Index in zwei Kategorien eingeteilt: Akkumulatorerweiterung und Filterung. Die Vorgehensweise in diesem Kapitel entspricht streng genommen noch der Filterung, da jeweils die komplette Liste eines visuellen Wortes traversiert werden muss. Im Vergleich zu den übrigen Ansätzen in der Literatur ist hier allerdings lediglich eine Vergleichsoperation nötig, um die Relevanz eines Eintrages der Liste zu prüfen. Außerdem könnten – falls nur ein Umgebungsmerkmal verwendet wird – die Einträge in den Listen prinzipiell auch sortiert

werden, sodass sich das Auffinden der relevanten Merkmale mit übereinstimmenden quantisierten Umgebungsmerkmalen durch Binärsuche noch beschleunigen ließe. Im nächsten Kapitel wird die Kategorie Filterung aber ohnehin gänzlich verlassen und die Merkmale werden in einem 2D-Index gespeichert. In diesem Kapitel liegt der Fokus der Auswertungen daher weiterhin auf der Genauigkeit der Suchergebnisse mit den vorgeschlagenen Repräsentationen.

6.3. Ergebnisse

Die Resultate in Form der *mean average precision* Werte für die vier Datensätze zeigt Abbildung 6.1. Auf der Abszisse werden in drei Blöcken die Ergebnisse für die ansteigende Datenbankgröße dargestellt, wobei der jeweils erste Block keine Ablenkungsbilder des MIRFLICKR1M Datensatzes beinhaltet, sondern nur die Bilder des jeweiligen Datensatzes. Es lässt sich folgendes feststellen:

- Bereits ohne zusätzliche Ablenkungsbilder kann die Integration der Umgebungsmerkmale die Resultate verbessern. Die Repräsentation ist damit nicht nur weniger anfällig für Ablenkung durch irrelevante Bilder in großen Datenbanken, sondern kann schon an sich mehr Information aufnehmen.

- Die Anwendung von PCA und Whitening bringt nur bei der größten Umgebungsgröße von 50 einen Vorteil, die aber ohnehin die schlechtesten Ergebnisse erzielt. Auch die Tatsache, dass OXFORD5K dort deutlicher von PCA und Whitening profitiert als die übrigen Datensätze, spricht dafür, dass für die Ermittlung der PCA Parameter nicht genügend repräsentative Daten zur Verfügung stehen und sich daher Overfitting-Effekte ergeben.

- Die ODER-Kombination kann die Ergebnisse durchweg stark verbessern, was beim LANDMARKS Datensatz besonders deutlich wird. Den Verlust der Suchgenauigkeit durch das Hinzufügen von einer Million Ablenkungsbildern kann diese Kombination vollständig ausgleichen und das BoW-Modell ohne Ablenkungsbilder sogar noch etwas übertreffen.

- Selbst die UND-Kombination liefert teilweise noch vergleichbare Performance zum BoW-Modell obwohl um mehrere Größenordnungen weniger Merkmale für die Ermittelung der Suchergebnisse verwendet werden.

Bezüglich der Umgebungsgröße zeigt Abbildung 6.2 die Werte der vier untersuchten Umgebungsgrößen im Vergleich zur gestrichelten Basis des BoW-Modells für die maximale Datenbankgröße mit einer Million Ablenkungsbildern. Für $\zeta_{CNN} = 8$ ergibt sich der beste Kompromiss. Abweichend von den Resultaten aus Kapitel 5 schneidet die größte Umgebung ($\zeta_{CNN} = 50$) deutlich schlechter ab. Eine mögliche Erklärung dafür ist, dass bei der Zusammenstellung der korrekten BoW-Korrespondenzen $\mathcal{K}_k^*$ nur „hochwertige" Korrespondenzen verwendet wurden, deren Plausibilität durch weitere Verifizierungen mit den Originalmerkmalen überprüft wurde. Dadurch könnten vor allem Merkmale in der Mitte der Objekte gesammelt worden sein, die auch bezüglich einer großen Umgebung noch übereinstimmen. Die in diesem Kapitel für die Ergebnisse des Gesamtsystems aber ebenso wichtigen Merkmale am Rand der Objekte oder solche mit weniger lokaler Übereinstimmung profitieren hingegen nicht von der maximalen Umgebungsgröße.

In Abbildung 6.3 sind die positiven Auswirkungen der Umgebungsrepräsentation an einem Beispiel aus dem HOLIDAYS Datensatz dargestellt. Das rechts unten abgebildete nichtkorrespondierende Bild weist von allen Ablenkungsbildern des HOLIDAYS Datensatzes die meisten BoW-Korrespondenzen auf, sodass sich dafür – selbst nach der *TF-IDF*-Gewichtung – eine größere Ähnlichkeit ergibt als für das laut Annotierung korrespondierende Bild (rechts oben). Durch die Hinzunahme der Umgebungsmerkmale werden die inkorrekten Korrespondenzen nahezu vollständig eliminiert und von den korrekten Korrespondenzen bleiben viele erhalten, sodass sich das korrespondierende Bild als ähnlichstes ergibt.

Tabelle 6.2 zeigt einen Vergleich der Ergebnisse mit den Werten anderer Ansätze in der Fachliteratur, die ebenfalls Umgebungsinformationen in den Index integrieren. Der Fokus liegt dabei auf Arbeiten, die Ergebnisse auf den in dieser Dissertation verwendeten Datensätzen ausweisen, und nicht nur auf Fast-Duplikat-Datensätzen evaluieren. Um den eigentlichen Gewinn der Integration von Umgebungsinformation für große Datenbanken einzuschätzen, sind die Ergebnisse mit einer Million Ablenkungsbildern dargestellt. Auf den Datensätzen PARIS6K und HOLIDAYS kann die ODER-Kombination

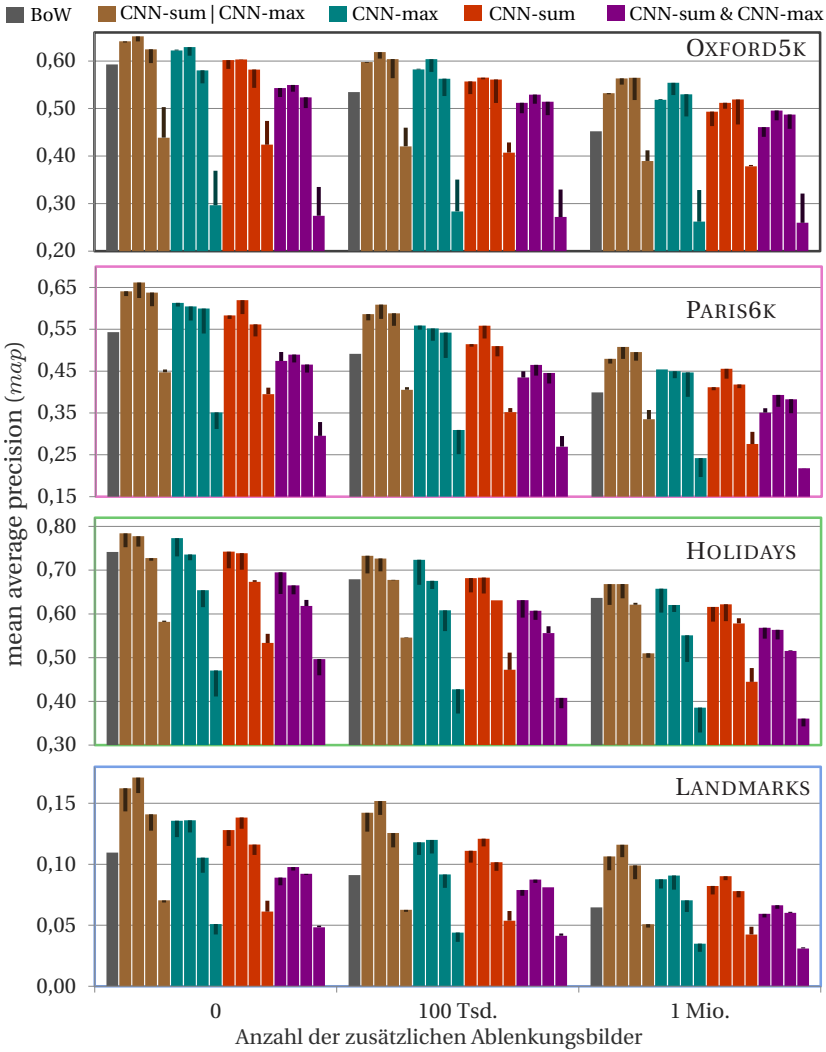

Abbildung 6.1.: Ergebnisse der Integration der Umgebungsmerkmale und ihrer Kombinationen. Durch die dünnen Linien auf jedem Balken wird der (überwiegend negative) Einfluss von PCA und Whitening auf die *map*-Werte gekennzeichnet. Die jeweils vier nebeneinander angeordneten gleichfarbigen Balken jeder Konfiguration zeigen die Werte für die vier von links nach rechts angeordneten Umgebungsgrößen $\zeta_{CNN} = 4, 8, 20$ und 50.

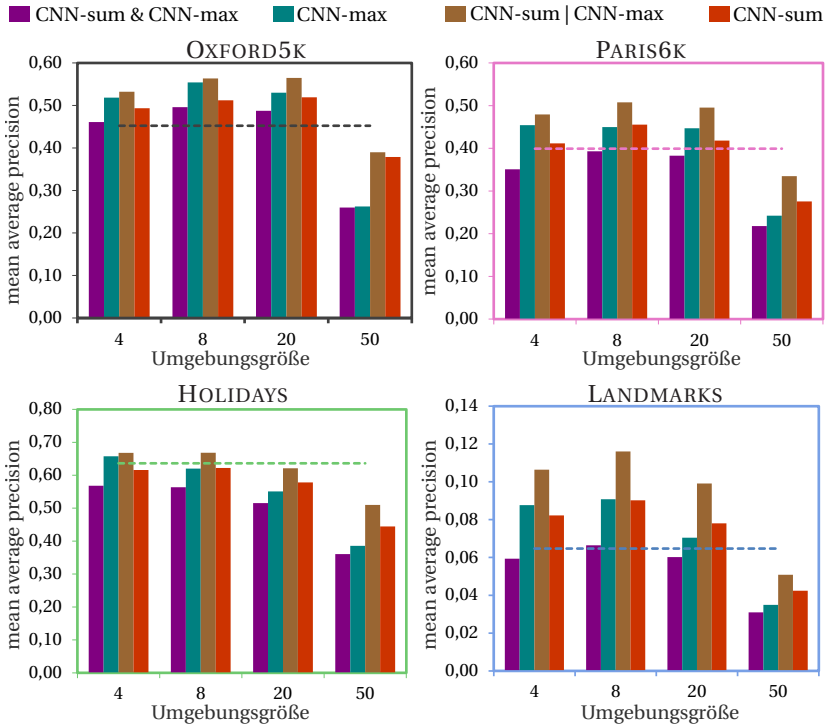

Abbildung 6.2.: Einfluss der Umgebungsgröße auf die Qualität der Suchergebnisse (*map*) bei einer Datenbankgröße mit einer Million Ablenkungsbildern. Die waagrechten gestrichelten Linien kennzeichnen die Basisergebnisse ohne Umgebungsmerkmale, also mit einem rein BoW-basiertes System.

die aufgeführten Verfahren zwar übertreffen. Dennoch unterliegen die dargestellten Werte einer eingeschränkten Vergleichbarkeit, denn die einzelnen Systeme zur inhaltsbasierten Bildsuche sind aus vielen Einzelkomponenten zusammengesetzt, die sich meist in wesentlichen Aspekten unterscheiden:

- Extraktion der lokalen Merkmale: Zur Detektion markanter Punkte (Kapitel 2.1) werden unterschiedliche Verfahren verwendet (DoG oder verschiedene affin-invariante Detektoren). Oft wird auch auf eine Rotationsinvarianz verzichtet, da die Anfrage- und Referenzbilder in den Datensätzen in der Regel in derselben Ausrichtung aufgenommen

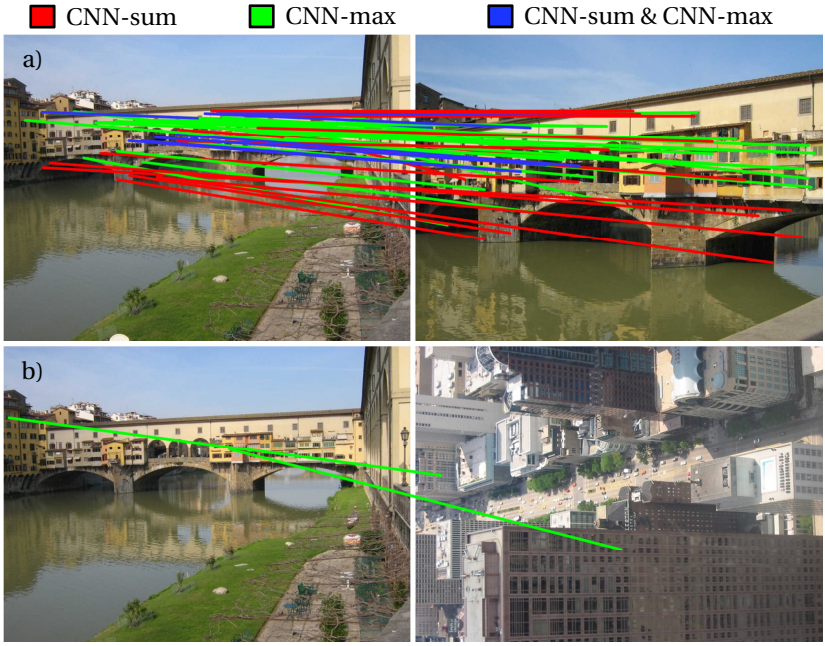

Abbildung 6.3.: Qualitatives Ergebnis der vorgeschlagenen CNN-basierten Umgebungsrepräsentation am Beispiel eines korrespondierenden und eines nichtkorrespondierenden Bildpaars für die Umgebungsgröße $\zeta_{CNN} = 8$. Die Linien zeigen diejenigen BoW-Korrespondenzen, die bezüglich der Umgebungswörter übereinstimmen. In Rot ist die Variante mit Aggregierung durch Summierung dargestellt, in Grün die durch Maximumbildung. Die `UND`-Kombination beider Varianten ist blau eingefärbt, die `ODER`-Kombination ergibt sich demnach durch alle dargestellten Korrespondenzen eines Bildpaares. Die Anzahl der lokalen Merkmale beträgt im jeweils linken Anfragebild 4 708, im dazu korrespondierenden Bild (rechts oben) 3 750, und im nichtkorrespondierenden, aber komplexeren Bild (rechts unten) 8 964. Im korrespondierenden Bildpaar ergeben sich für 272 der lokalen Merkmale des linken Bildes mindestens eine BoW-Korrespondenz im rechten Bild. Im nichtkorrespondierenden Bildpaar ist dies bei 427 der lokalen Merkmale der Fall. Mit der vorgestellten Umgebungsrepräsentation können also nahezu alle inkorrekten BoW-Korrespondenzen im unteren Bildpaar eliminiert werden. Die drei Bilder stammen aus dem HOLIDAYS Datensatz [Jég08].

sind.

- Normalisierung der lokalen Merkmale: die älteren Ansätze vor 2012 nutzen noch nicht die RootSIFT Normalisierung, die beim OXFORD5K-Datensatz und 100 000 Ablenkungsbildern bereits eine Verbesserung der *map* um 0,06 ergibt [Ara12].

- Wenn die Modelle – allen voran das Codebook – auf demselben Datensatz trainiert wurden, der auch zur Evaluation verwendet wird, ergeben sich dafür bessere Resultate.

- Unterschiede in der Deskriptorquantisierung (Größe des Codebooks, Verfahren zur (approximativen) Quantisierung, Soft Quantization etc.).

- Unterschiedliche Anfragebilder: für OXFORD5K und PARIS6K sind ausgeschnittene und nicht ausgeschnittene Varianten spezifiziert; für HOLIDAYS werden oft die um 90 Grad gedrehten Bilder manuell korrigiert und dann auf rotationsinvariante Repräsentationen verzichtet.

- Unterschiedliche Datensätze als Ablenkungsbilder: Flickr1M, MIRFLICKR1M [MJH10], Panoramio1M [Cao10a] oder selbst heruntergeladene Bilder.

- Die Überlappung der Anfragebilder mit den Ablenkungsbildern wird selten thematisiert. Gerade bei Verwendung großer Datensätzen zur Ablenkung ist die Wahrscheinlichkeit aber hoch, dass sehr ähnliche Bilder enthalten sind, die die Ergebnisse verfälschen können.

Außerdem sind die Ergebnisse immer in Verbindung mit der Laufzeit und mit dem Speicherbedarf für den Index sowie für den Akkumulator zu betrachten. Für den Speicherbedarfs sind in Tabelle 6.2 jeweils beide Angaben aufgeführt. Bei der Anzahl der Felder für den Akkumulator bedeutet der Wert 2 048 des Verfahrens von Shen *et al.* [She12] etwa, dass für eine Million Bilder in der Datenbank ein Akkumulator der Größe $10^6 \cdot 2\,048 \cdot 4$ Byte $\approx 7{,}6$ GiB[1] erforderlich ist, der für jede Suchanfrage initialisiert, befüllt und

1 Um Mehrdeutigkeiten zu vermeiden, werden in dieser Arbeit die Binärprefixe verwendet, d.h. 1 GiB = 1 Gibibyte = 1 024 Mebibyte = 2^{30} Byte

ausgewertet werden muss, anstelle von knapp 4 MiB wie für die Verfahren der anderen Strategien einschließlich dieser Arbeit.

Hinsichtlich des Speicherbedarfs, um pro indexiertem Merkmal die jeweiligen Umgebungscharakteristiken im Index zu speichern, sind für diese Dissertation 56 Bit angegeben, die sich zusammensetzen aus den üblichen 32 Bit für die Bildnummer sowie jeweils 12 Bit für die CNN-sum und CNN-max-Werte. Genau hier knüpft der wesentliche Unterschied von dieser Arbeit zu den Verfahren an, die mit Filterung oder Akkumulatorerweiterung arbeiten: die Merkmale können direkt entlang mehrerer Dimensionen indexiert werden, sodass nur noch die Bildnummern gespeichert werden müssen und der Zugriff nur auf einen Bruchteil des Index erforderlich ist. Im nächsten Kapitel wird beides anhand eines konkreten Systems untersucht, bei dem der Index nicht mehr im Arbeitsspeicher liegt sondern allein auf einem SSD-Laufwerk.

Tabelle 6.2.: Vergleich unterschiedlicher Arbeiten, die den Kontext von lokalen Merkmalen im Index berücksichtigen. Alle Werte geben die mean average precision an und wurden berechnet auf dem jeweiligen Datensatz (OXFORD5K, PARIS6K oder HOLIDAYS) unter Verwendung von einer Million Ablenkungsbildern (je nach Arbeit aus unterschiedlichen Quellen, siehe Text). „**+K**“ bezeichnet jeweils die Ergebnisse, die den Kontext der lokalen Merkmale nutzen, während die Werte **BoW** die lokalen Merkmale gemäß dem Bag-of-Words Modell nur anhand ihres quantisierten Deskriptors indexieren. **Index Bit/M.** gibt den Speicherbedarf in Bit pro indexiertem Merkmal an, einschließlich der 32 Bit für die Bildnummer. **Akk. Felder** kennzeichnet die Anzahl der Felder im Akkumulator, die jedes Datenbankbild beansprucht.

		Index	Akk.	OXFORD5K		PARIS6K		HOLIDAYS	
Ansatz	**Strategie**	**Bit/M.**	**Felder**	**BoW**	**+ K**	**BoW**	**+ K**	**BoW**	**+ K**
Diese Arbeit (`ODER` -Kombination)	2D-Index	56	1	0,452	0,496	0,399	0,508	0,636	0,668
Coupled Multi-index [Zhe14a]	2D-Index	54	1	-	-	-	-	0,230	0,360
Spatial Bag-of-Features [Cao10a]	Filterung	64	1	0,408	0,550	0,278	0,391	-	-
Self-contained contextual binary code [Liu14b]	Filterung	160	1	-	-	0,208	0,271	0,318	0,484
Multi-order visual phrase [Zha13b]	Filterung	96	1	0,493	0,621	-	-	-	-
Geometry-preserving visual phrases [Zha11b]	Akk'Erw'	34	100	0,413	0,532	-	-	-	-
Visual Phraselet [Zhe13a]	Akk'Erw'	39	100	0,447	0,557	0,290	0,391	-	-
Spatially constrained similarity measure [She12]	Akk'Erw'	40	2 048	0,535	0,685	-	-	-	-
Weak geometric consistency [Jég08]	Akk'Erw'	41	128	-	-	-	-	0,320	0,440

7

Speicherauslegung des Index

Wie sich im vorangegangenen Kapitel gezeigt hat, bietet die Hinzunahme der Umgebungsinformation eines lokalen Merkmals die Chance, die Suchergebnisse in großen Datenbanken zu verbessern, da viele nichtrelevante Merkmale aufgrund ihrer abweichenden Umgebung nicht in die Ähnlichkeitsbewertung einfließen. Dies trifft für die in Abschnitt 2.2.5.2 genannten bisherigen Ansätze in der Fachliteratur, die mit der Filterung der Merkmale im Index arbeiten, ebenfalls zu. Der in dieser Dissertation gewählte Ansatz der Quantisierung der Merkmalsumgebungen leistet aber noch mehr: die vielen nichtrelevanten Merkmale haben hier nicht nur keinen Einfluss auf die Ähnlichkeiten, sondern müssen gar nicht erst aus dem Index gelesen werden. Die Merkmale im Index werden direkt anhand mehrerer Dimensionen adressiert – anhand des visuellen Wortes und anhand einer oder zweier Umgebungswörter. Dadurch müssen nur die zutreffenden Zellen im Index traversiert werden, um die Stimmen in den Akkumulator einzutragen. Insgesamt müssen also weniger Zellen gelesen werden, welche zudem weniger Merkmale beinhalten. Tabelle 7.1 zeigt oben die durchschnittliche Anzahl der quantisierten Merkmale im Index, die für ein Anfragebild des PARIS6K Datensatzes bei einer Million Ablenkungsbildern verarbeitet werden. Durch die Hinzunahme einer Umgebungsdimension sind etwa Faktor 200 bis 300 weniger Zugriffe erforderlich. Bei der Verwendung beider Umgebungsdimensionen im Rahmen einer UND-Kombination sind gar 2 000 mal weniger Zugriffe nötig. Dies wirft die Frage auf, ob der Index dann über-

haupt noch zwingend im Arbeitsspeicher (Random-Access Memory (RAM)) vorliegen muss. Denn falls die aktuellen nichtflüchtigen Speichermedien auf Halbleiterbasis ihre Stärken hinsichtlich schneller Zugriffszeiten ausspielen können, wird es möglich, den Index auf einem SSD-Laufwerk zu belassen. Dieses Kapitel widmet sich dieser Fragestellung, indem zunächst die dafür relevanten Eigenschaften der SSD-Laufwerke kurz beschrieben werden. Anschließend werden die Vorteile anhand eines konkreten Systemaufbaus evaluiert [Man18]. Die indexierten Merkmale werden dabei während einer Suchanfrage ausschließlich von einem SSD-Laufwerk gelesen, sodass der Arbeitsspeicher nur für den Akkumulator verwendet wird.

7.1. SSD-Laufwerke für die Bildsuche

Solid-State-Disks (SSDs) sind nichtflüchtige, elektronische Speichermedien, die Informationen in Flash-Halbleiterspeicherzellen ablegen. Aktuelle Modelle arbeiten mit der sogenannten V-NAND-Technologie, die mehrere – derzeit z. B. 48 [Kan17] – Speicherchips vertikal stapelt und untereinander verbindet, um bei konstanter Grundfläche eine möglichst hohe Datendichte zu erreichen. Eine erschöpfende Darstellung der Eigenschaften und Besonderheiten aktueller SSDs würde den Umfang dieser Arbeit weit übertreffen, daher werden hier nur die für die Bildsuche relevantesten Aspekte behandelt, welche die Leseoperationen auf zufällig verteilte Speicherbereiche betreffen[1]:

- **Aufbau**: Ein Bit wird in einer Speicherzelle mit Floating-Gate Transistoren gespeichert. Mehrere Speicherzellen werden zu Seiten gruppiert, mehrere Seiten wiederum zu Blöcken und mehrere Blöcke bilden eine Ebene. Die kleinstmögliche Einheit für das Lesen und Schreiben der Daten ist eine Seite, d. h. selbst beim Lesen nur eines Bytes wird stets eine ganze Seite gelesen.

1 Aufgrund der internen Parallelisierung – etwa durch die Verteilung der Daten auf mehrere Speicherchips – haben jedoch auch die Schreiboperationen einen Einfluss auf die Lesezugriffe, sodass größere, inhaltlich korrelierende Daten immer gemeinsam geschrieben werden sollten. Da im vorliegenden Anwendungsfall aber beim Schreiben des Index nicht bekannt ist, welche Merkmale später für ein beliebiges Anfragebild inhaltlich korrelieren werden, wird dieser Aspekt hier nicht weiter vertieft.

- **Ausrichtung**: Die Daten sollten bei Schreibzugriffen an der Seitengröße ausgerichtet werden, damit spätere Leseoperationen nicht unnötig über Seitengrenzen hinweg erfolgen.

- **Adressierung**: Um die gleiche Schnittstelle wie herkömmliche magnetische Festplattenlaufwerke anzubieten, und weil die Speicherzellen sich durch die Schreibzyklen abnutzen, besitzen SSD-Laufwerke einen aufwändigen Controller, der die logischen Adressen in physische Adressen im Flash Speicherraum übersetzt und die Parallelisierung und die Abnutzung der einzelnen Blöcke berücksichtigt.

- **Parallelisierung**: Die maximale Lesegeschwindigkeit ist nur zu erreichen wenn mehrere Leseanfragen parallel gestellt werden, damit der Controller die verschiedenen Arten der internen Parallelität nutzen kann. In den Datenblättern wird die Anzahl der parallelen Zugriffe meist als *Query Depth* bezeichnet.

Für weitergehende Informationen über die technischen Details von SSDs und die Konsequenzen für die Softwareentwicklung sei an dieser Stelle auf eine lesenswerte Artikelreihe von Emmanuel Goossaert [Goo14] verwiesen.

Die für den vorliegenden Anwendungszweck relevante Kenngröße in den Datenblättern von SSDs ist die Anzahl zufälliger Leseoperationen pro Sekunde – Input/Output Operations Per Second (IOPS) –, die in der Regel für eine Größe von 4 KiB pro Lesevorgang angegeben werden. Damit die maximale Leistung einer SSD abgerufen werden kann, müssen aber neben der Speicherausrichtung, Zugriffsgröße etc. auch die Umgebungsbedingungen stimmen, d. h. die Hardwareanbindung, Treiber, Protokolle etc. aufeinander abgestimmt sein. Alle Experimente in dieser Dissertation wurden mit der folgenden Hardware durchgeführt: ein Server mit zwei Intel Xeon E5-2630v4 2,2 Ghz CPUs unter Nutzung aller 20 CPU Kerne. Als separate SSD, die nur den Index speichert, kam eine Samsung SSD 960 Pro M.2 mit 1 TiB, die per NVMe Protokoll und PCIe 3.0x4 angebunden ist, zum Einsatz. Abbildung 7.1 zeigt für dieses System die gemessene Anzahl der zufälligen Leseoperationen pro Sekunde in Abhängigkeit der Größe der zu lesenden Daten und für unterschiedlich viele parallele Leseoperationen. Dabei zeigt sich einerseits, dass der maximale Lesedurchsatz erst bei ca. 20 parallelen Anfragen erreicht wird, und andererseits, dass es keinen Unterschied macht, ob jeweils lediglich

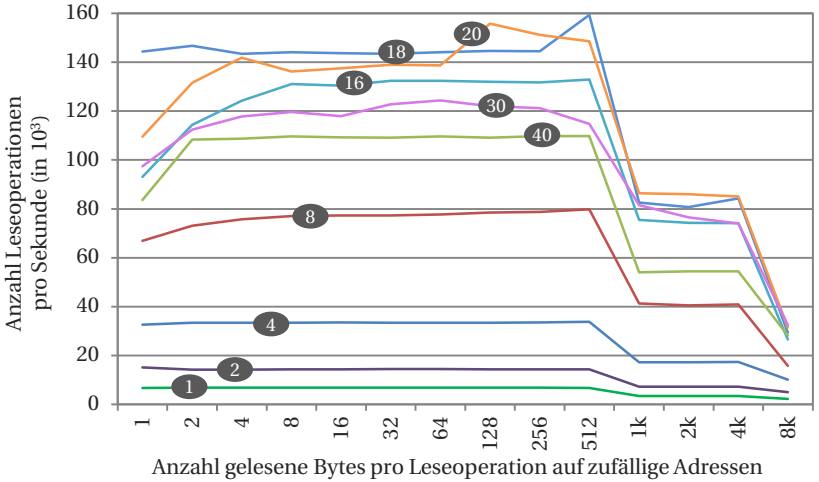

Abbildung 7.1.: Anzahl der Leseoperationen pro Sekunde auf zufällige Adressen in Abhängigkeit der Größe der zu lesenden Daten und für unterschiedlich viele parallele Zugriffe (1 – 40). Angegeben sind die Mittelwerte aus zehn Durchläufen, wobei jeweils 10 000 zufällige Leseoperationen in einer Datei der Größe 200 GiB durchgeführt wurden und vor jeder Messung der Cache des Betriebssystems bereinigt wurde, um dessen Caching Effekte auszuschließen.

ein Byte oder 512 Byte pro Leseoperation angefordert werden. Erst ab dem 513. Byte müssen offensichtlich weitere Speicherseiten adressiert werden, was den Durchsatz deutlich verringert. Im Datenblatt der verwendeten SSD werden „bis zu 440 000 IOPS“ bei 32 parallelen zufällig verteilten Zugriffen und für eine Größe von 4 KiB pro Lesevorgang angegeben. Für die Praxis sind solche Herstellerangaben und selbst viele Benchmarks jedoch leider nur begrenzt aussagekräftig [Ti17], da sie beispielsweise nur einen sehr kleinen Teil des logischen Adressraums verwenden, sodass durch sogenannte Read-ahead-Mechanismen die Daten häufig bereits vor der eigentlichen Leseanfrage „auf Verdacht“ im Cache gehalten werden.

Tabelle 7.1.: Anzahl der indexierten Merkmale, die während der Suchanfragen verarbeitet werden und somit den Akkumulator erhöhen. Die Datenbank besteht aus den Datensätzen PARIS6K und MIRFLICKR1M. Die Werte zeigen jeweils den Mittelwert über alle 55 Anfragebilder von PARIS6K. Da die Anzahl der verwendeten Merkmale (sowie die Zeitdauern der Suchanfragen in Tabelle 7.2) linear mit der – in PARIS6K sehr unterschiedlichen – Anzahl der Merkmale der Anfragebilder steigt, stellt der Mittelwert ein aussagekräftiges Maß dar und die Angabe von Standardabweichungen bezüglich der 55 Anfragebilder ist nicht informativ.

Index / Konfiguration	**Anzahl verarbeiteter Merkmale**			
Index im Arbeitsspeicher				
BoW				25 886 315
+CNN-sum				102 852
+CNN-max				76 343
+CNN-max | CNN-sum				166 023
+CNN-max & CNN-sum				13 172
Index auf einer SSD	**bei einer maximale Zellengröße von**			
	1	**4**	**16**	**32**
BoW				
+CNN-sum	4 481	14 052	31 507	41 798
+CNN-max	4 284	13 039	28 895	38 659
+CNN-max | CNN-sum	8 765	27 091	60 402	80 457
+CNN-max & CNN-sum	868	2 307	4 646	6 139

7.2. Realisierung eines 2D-Index mit einem SSD-Laufwerk

Für die Realisierung eines konkreten Systems zur Bildsuche, bei dem der Index allein auf einem SSD Laufwerk vorliegt, werden die Modelle (Codebooks) und Parameter (Netzgröße 448 und Umgebungsgröße $\zeta_{CNN} = 8$) aus Kapitel 6 verwendet, sodass die Vergleichbarkeit der Ergebnisse gewährleistet ist. Indexiert und zur Evaluation verwendet wird der PARIS6K Datensatz, wobei der Datensatz MIRFLICKR1M wieder zur Ablenkung dient. Die Wahl fällt auf PARIS6K, da dieser Datensatz ähnliche Eigenschaften wie OXFORD5K besitzt, auf dem wiederum sämtliche Modelle trainiert wurden. Das Codebook für die lokalen Merkmale umfasst wieder $18^4 = 104\,976$ visuelle Wörter und das

Umgebungscodebook der CNN Merkmale besteht aus 3 025 Umgebungswörtern. Daraus ergeben sich im Falle des 2D-Index $18^4 \cdot 3\,025 = 317\,552\,400$ Zellen, auf die sich alle $1{,}3 \cdot 10^9$ Merkmale der Datenbank verteilen, was im Mittel etwa vier Merkmalen pro Zelle entspricht. Um während einer Suchanfrage die Merkmale innerhalb einer Indexzelle schnell lokalisieren zu können, ist eine feste maximale Größe der Zellen erforderlich, denn anderenfalls müssten die Startadressen der jeweils unterschiedlich großen Zellen verwaltet werden. Das würde wiederum entweder die Häufigkeit der Zugriffe auf die SSD verdoppeln (falls die Startadressen der Zellen auf der SSD gespeichert würden), oder aber die Vorteile des 2D-Index zunichte machen (falls die Startadressen bei jedem Programmstart in den Arbeitsspeicher gelesen werden müssten).

Bei der Wahl der maximalen Zellengröße ist zu berücksichtigen, dass sich die Merkmale keineswegs gleichmäßig auf die Zellen verteilen. Für die Verteilung der Merkmale hinsichtlich der visuellen Wörter ergibt sich beispielsweise ein Gini-Koeffizient von 0,2 und bezüglich der Umgebungsmerkmale von 0,6. Bei einer zu kleinen Zellengröße finden somit viele Merkmale keinen Platz im Index und müssen verworfen werden, wohingegen eine zu große Zellengröße mehr Speicherplatz benötigt und gleichzeitig den mittleren Belegungsgrad der Zellen reduziert. In diesem Kapitel werden daher verschiedene Größen hinsichtlich der jeweils resultierenden Suchgenauigkeit untersucht.

Für die beiden unterschiedlichen Varianten der Umgebungsmerkmale (Maximum- und Sum-Pooling) wird jeweils ein separater 2D-Index erstellt. In beiden 2D-Indizes sind also dieselben Merkmale indexiert, allerdings in einer unterschiedlichen Anordnung bezüglich der zweiten Dimension. Zur Realisierung der `UND` und `ODER`-Kombinationen wiederum wäre eigentlich ein 3D-Index erstrebenswert. Allerdings ergäben sich hierbei $18^4 \cdot 3\,025 \cdot 3\,025 \approx 960 \cdot 10^9$ Zellen, was die Größe der SSD bereits für die minimale Zellengröße von einem Merkmal je Zelle klar übersteigt. Um die Kombinationen dennoch auf Basis eines SSD-Laufwerkes realisieren zu können, wird ein Kompromiss verwendet, der mit den beiden separaten 2D-Indizes auskommt. Pro lokalem Merkmal im Anfragebild werden die korrespondierenden Merkmale dazu aus beiden 2D-Indizes gelesen mit den folgenden Besonderheiten:

- Bei der `ODER`-Kombination werden aus Effizienzgründen die beiden

gelesenen Merkmalsmengen nicht auf gemeinsame Elemente untersucht, sprich auf Merkmale, die bezüglich beider Umgebungsdimensionen übereinstimmen. Sie stimmen im Akkumulator somit doppelt ab. Dies kann als eine Art Gewichtung interpretiert werden, wenn man argumentiert, dass solche Merkmale, die bezüglich beider Umgebungswörter übereinstimmen, noch wahrscheinlicher korrekte Korrespondenzen darstellen als solche, die nur bezüglich einer Umgebungsdimension übereinstimmen.

- Für die `UND`-Kombination wird ebenfalls auf einen Vergleich der jeweils gelesenen Merkmalsmengen verzichtet, indem mit dem Filteransatz (vgl. Abschnitt 2.2.5.2) gearbeitet wird. Dazu wird im 2D-Index, der die Merkmale anhand des Maximum-Poolings indexiert, für jedes Merkmal nicht nur seine Bildnummer gespeichert, sondern zusätzlich auch sein Umgebungswort bezogen auf das Sum-Pooling. In den Akkumulator werden dann aus dem zum Maximum-Pooling gehörenden 2D-Index nur diejenigen gelesenen Merkmale eingetragen, die den passenden Sum-Pooling Wert aufweisen. Im Vergleich zu den in Abschnitt 2.2.5.2 genannten Verfahren benötigt dies jedoch praktisch keinen zusätzlichen Aufwand, da einerseits die Merkmalsmengen sehr klein verglichen mit den inversen Listen im BoW-Modell sind, und sich die Filteroperation pro Merkmal auf den Vergleich eines Wertes beschränkt.

Jedes Merkmal wird im 2D-Index mit sechs Byte repräsentiert: vier Byte für die Bildnummer und zwei Byte für das Umgebungswort der dritten Indexdimension, um die `UND`-Kombination, wie oben beschrieben, zu realisieren. Wenn die maximale Zellengröße auf 32 Merkmale festgelegt wird, resultiert dies in einem 2D-Index der Größe $18^4 \cdot 3\,025 \cdot 32 \cdot 6$ Byte ≈ 57 GiB. Der Speicherbedarf pro Zelle liegt mit $32 \cdot 6 = 192$ Byte damit unter der in Abschnitt 7.1 ermittelten Grenze von 512 Byte, ab der sich die Zugriffsrate hardwarebedingt reduziert.

7.3. Evaluation

Den Einfluss der maximalen Anzahl an Merkmalen pro Indexzelle auf die Qualität der Suchergebnisse zeigt der obere Teil von Abbildung 7.2. Im un-

teren Teil ist außerdem der Anteil aller Merkmale dargestellt, die für die jeweilige begrenzte Zellengröße im Index Platz finden[1]. Gleichzeitig bleiben in jedem der beiden 2D-Indizes von den ca. 317 Mio. Indexzellen etwa 44% leer, da kein Merkmal in den Datensätzen eine entsprechende Kombination aus dem visuellem Wort und dem Umgebungswort aufweist. Abbildung 7.2 ist folgendes zu entnehmen:

- Die ODER Konfiguration übertrifft das Basis-BoW-Modell bereits mit maximal zwei Merkmalen pro Zelle, obwohl dann nur etwa 20% aller ursprünglichen Merkmale im Index Platz finden.
- Interessanterweise liefert der SSD-basierte 2D-Index ab einer gewissen maximalen Anzahl von Merkmalen pro Zelle sogar bessere Ergebnisse als die Vergleichsvariante, die mit *allen* Merkmalen im Arbeitsspeicher arbeitet. Das lässt vermuten, dass durch die Begrenzung der Merkmale pro Zelle diejenigen Merkmale außen vor bleiben, die oft in derselben Umgebung vorkommen. Im MIRFLICKR1M Datensatz könnte dies für diverse Wasserzeichen oder Textelemente gelten, die in einigen Bildern vorkommen. Zusätzlich stellt die Begrenzung eine Art Behandlung des Burstiness Effekts aus Kapitel 2.2.4 dar, der Ergebnisse beeinträchtigt. Tabelle 7.1 führt im unteren Teil die mittlere Anzahl der quantisierten Merkmale im SSD-Index auf, die für ein Anfragebild verarbeitet werden und zeigt, dass bei einer maximalen Zellengröße von 16 Merkmalen offensichtlich bereits etwa ein Drittel der Merkmale genügen, um die Resultate aus Kapitel 6 zu übertreffen.

Tabelle 7.2 zeigt schließlich die Auswirkungen des 2D-Index auf die Verarbeitungsgeschwindigkeit einer Suchanfrage. Bei den Werten mit den SSD-Indizes wurde vor jeder der vier Auswertungen der Cache des Betriebssystems zurückgesetzt, da sonst die zu lesenden Merkmale oft noch im Arbeitsspeicher liegen und die resultierenden Zeiten sich dadurch etwa halbieren. Es zeigt sich, dass die Konfigurationen, die die Merkmale aus dem

1 Hinsichtlich dieser Begrenzung stellt sich für die „überbelegten“ Zellen die Frage, welche Merkmale in der Zelle und somit im SSD Index gespeichert werden und welche nicht. In den Experimenten in diesem Kapitel wurde die Auswahl per Zufall getroffen, da die Datenbankgröße im Voraus bekannt ist. In Systemen mit einer über die Zeit ansteigenden Datenbankgröße ist dies jedoch gesondert zu betrachten.

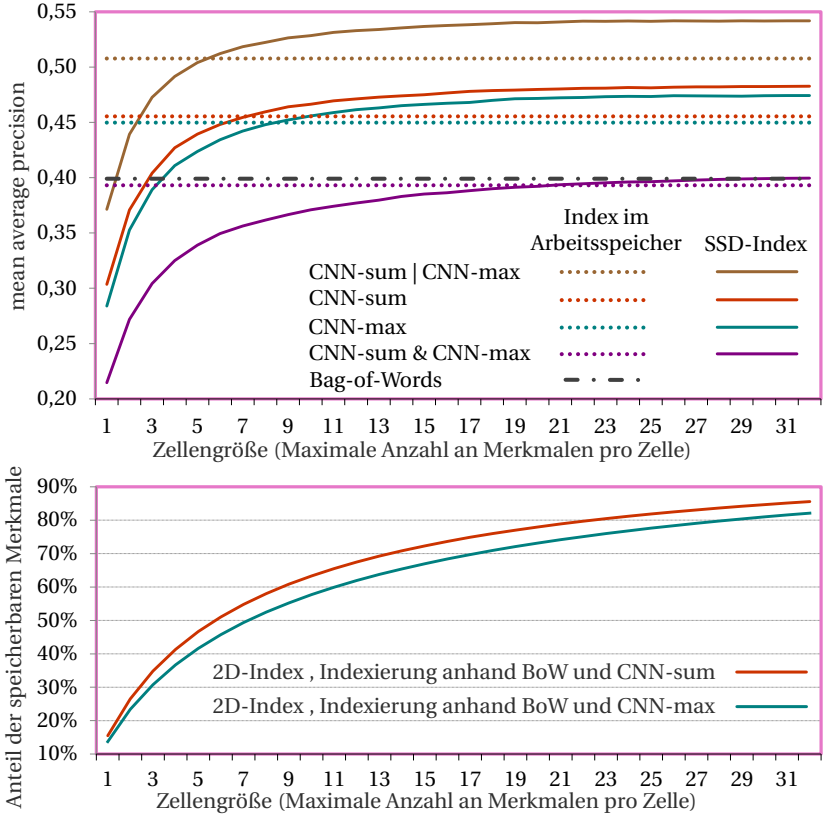

Abbildung 7.2.: Auswirkungen der limitierten Zellengröße des SSD-Index für den Datensatz PARIS6K bei einer Million Ablenkungsbildern von MIR-FLICKR1M für die Umgebungsgröße $\zeta_{CNN} = 8$.
Oben: Auswirkungen auf die Suchergebnisse. Die waagrechten gestrichelten Linien kennzeichnen zum Vergleich die Ergebnisse aus Kapitel 6, also die Varianten, bei denen alle indexierten Merkmale im Arbeitsspeicher vorliegen und für die Suche verwendet werden können.
Unten: Anteil der $1{,}3 \cdot 10^9$ zu indexierenden Merkmale, die aufgrund der beschränkten Zellengröße jeweils im SSD-Index Platz finden.

Tabelle 7.2.: Zeitdauer für eine Suchanfrage in einer Datenbank bestehend aus den PARIS6K und MIRFLICKR1M Datensätzen für verschiedene Index-Konfigurationen. Gemessen wurde die Zeitdauer für das Ermitteln der Merkmale im Index und das Eintragen in den Akkumulator. Die von der jeweiligen Konfiguration unabhängige Zeitdauern für die Merkmalsextraktion und die Akkumulatorauswertung sind dagegen nicht berücksichtigt. Die Werte zeigen jeweils den Mittelwert über alle 55 Anfragebilder von PARIS6K.

Konfiguration	Index Speicherort	Dauer in ms
BoW	RAM	311,2
Globale Merkmale, erschöpfende Suche in 10^6 Vektoren $\in \mathbb{R}^{256}$	RAM	28,1
BoW + CNN-max	SSD	35,5
BoW + CNN-sum	SSD	36,2
BoW + CNN-max & CNN-sum	SSD	35,4
BoW + CNN-max | CNN-sum	SSD	63,1

SSD-Index lesen beinahe um eine Größenordnung schneller sind als das BoW-Modell, obwohl dort die Merkmale im vielfach schnelleren Arbeitsspeicher vorliegen. Das ist nur möglich, weil durch die Hinzunahme der Umgebungsmerkmale die Diskriminanz erhöht wird und dadurch sehr viel weniger Merkmale während einer Suchanfrage verarbeitet werden müssen. Zum Vergleich ist in Tabelle 7.2 außerdem die Verarbeitungsdauer für ein System angegeben, das mit globalen Merkmalen arbeitet und eine erschöpfende Suche in allen 1 Million Bildvektoren durchführt. Der Vergleichswert geht von einer 256-dimensionalen Repräsentation eines Bildes aus, wurde auf derselben Hardware berechnet, und wie bei den übrigen Konfigurationen werden alle 20 CPU Kerne genutzt.

In Tabelle 7.3 wird die Verarbeitungsgeschwindigkeit einer Suchanfrage schließlich mit den Verfahren aus der Fachliteratur verglichen. Da die Zeitdauer einer Suchanfrage bei Indexierung auf Basis der einzelnen lokalen Merkmale (Gleichung 2.14) linear von der Anzahl der Merkmale im Anfragebild abhängt, muss diese beim Vergleich prinzipiell berücksichtigt werden. Für die durchschnittliche Anzahl der Merkmale eines Anfragebildes ergibt sich in dieser Arbeit 5 811 für OXFORD5K, 5 081 für PARIS6K und 5 287 für HOLIDAYS. Klar erkennbar ist, dass die Integration des Kontextes in den

Index typischerweise zu einer mitunter starken Erhöhung der Anfragezeit führt. Nur für die in dieser Dissertation gewählte Strategie des 2D-Index gilt das Gegenteil, da viele der inkorrekten Korrespondenzen gar nicht erst verarbeitet werden müssen.

An dieser Stelle sei darauf hingewiesen, dass sich mit der UND-Kombination prinzipiell ein System realisieren ließe, das noch einmal um eine Größenordnung schneller wäre. Das würde aber ein SSD-Laufwerk erfordern, die Platz für eine Indexierung entlang aller drei Dimensionen böte (BoW × CNN-max × CNN-sum). Für die Parameter in dieser Arbeit wären das bei einer Zellengröße von 16 Merkmalen etwa 86 TiB. Obwohl bereits SSD-basierte Speichersysteme in dieser Größenordnung erhältlich sind, sind auch auf kleineren Datenträgern noch weitere Kompromisse zwischen Speicherbedarf und Zugriffszeiten möglich, etwa mit anderen Codebookgrößen oder durch Hashing-Verfahren oder Datenbanksysteme.

Fazit

Die Vermutung, dass die diskriminante Umgebungsrepräsentation der lokalen Merkmale sich ideal mit den Hardwareeigenschaften von aktuellen SSD-Laufwerken ergänzen könnte, hat sich in eindrucksvoller Weise bestätigt. Die Einschränkung auf eine feste Zellengröße für die Implementierung des 2D-Index auf einem SSD-Laufwerk führte nicht zu schlechteren, sondern sogar zu geringfügig besseren Ergebnissen. Gleichzeitig ist die Zeitdauer für eine Suchanfrage geringer als bei bisherigen Verfahren, die den Merkmalskontext auf andere Weise im Index integrieren und dadurch auf den Arbeitsspeicher als Speicherort des Index angewiesen sind.

Tabelle 7.3.: Durchschnittliche Zeitdauern in Millisekunden für eine Suchanfrage in einer Datenbank mit über einer Million Bilder für Systeme, die den Kontext von lokalen Merkmalen im Index berücksichtigen. Die letzte Spalte gibt den Datensatz an, der zusätzlich zu den (nicht überall identischen) eine Million Ablenkungsbildern indexiert wurde und aus dem sich auch die Anfragebilder ergeben. Die Fragezeichen in der Spalte „**System (CPU)**“ bedeuten, dass in den Veröffentlichungen nur die Taktrate, aber nicht die Anzahl der Kerne der verwendeten Hardware angegeben wurde. „**Dauer in ms + Kontext**“ bezeichnet jeweils die Ergebnisse, die den Kontext der lokalen Merkmale nutzen, während die Werte in der Spalte links davon die lokalen Merkmale gemäß dem Bag-of-Words Modell nur anhand ihres quantisierten Deskriptors verwenden.

Ansatz	Strategie	System (CPU)	Dauer in ms BoW	Dauer in ms + Kontext	Datensatz (+ 1M)
Diese Arbeit (CNN-sum)	2D-Index	20 × 2,2 Ghz	311	36	PARIS6K
Diese Arbeit (CNN-sum ODER CNN-max)	2D-Index	20 × 2,2 Ghz	311	63	PARIS6K
Coupled Multi-index [Zhe14a]	2D-Index	? × 3,5 Ghz	2 720	1 410	HOLIDAYS
Spatial Bag-of-Features [Cao10a]	Filterung	4 × 2,4 Ghz	42	56	OXFORD5K
Self-Contained Contextual Binary Code [Liu14b]	Filterung	? × 2,4 Ghz	-	711	PARIS6K
Multi-order Visual Phrase [Zha13b]	Filterung	4 × 3,4 Ghz	233	251	OXFORD5K
Geometry-preserving visual phrases [Zha11b]	Akk'Erw'	4 × 2,2 Ghz	137	248	OXFORD5K
Visual Phraselet [Zhe13a]	Akk'Erw'	16 × 2,4 Ghz	677	2 780	OXFORD5K
Weak geometric consistency [Jég08]	Akk'Erw'	4 × 2,6 Ghz	620	2 100	HOLIDAYS

8

Zusammenfassung und Ausblick

8.1. Zusammenfassung

Für die inhaltsbasierte Bildsuche auf Basis des Bag-of-Words-Modells wurden in dieser Dissertation verschiedene Repräsentationen vorgestellt, die die lokalen Merkmale im Index um Informationen aus den jeweiligen Bildumgebungen erweitern. Im Gegensatz zu den bisher üblichen Ansätzen, wird der Merkmalskontext als eigenständiges Merkmal erfasst und quantisiert. Dies erweitert den BoW-basierten Index um eine zusätzliche Dimension, sodass alle indexierten Merkmale anhand beider Dimensionen adressiert werden können.

Zunächst wurde dazu ein Evaluationsframework entworfen, das die Ziele dieser Repräsentationen abbildet – nämlich, die korrekten von den inkorrekten BoW-Korrespondenzen zu unterscheiden, die im Rahmen der erforderlichen Quantisierung entstehen. Für die Repräsentation wurden die beiden erfolgversprechendsten Verfahren aus der globalen Bildbeschreibung verwendet und entsprechend angepasst, um die jeweilige lokale Bildumgebung der Merkmale unter Berücksichtigung der Invarianzeigenschaften zu erfassen. Die erste Repräsentation aggregiert dafür die benachbarten lokalen Merkmale mittels des Fisher Vektors, während die zweite die Ergebnisse eines faltenden neuronalen Netzes (CNN) verwendet. Im Rahmen des Evaluationsframeworks wurden beide Repräsentationen sowie Kombina-

tionen davon miteinander verglichen, wobei die CNN-Varianten am besten abschnitten. Die anschließende Evaluation eines kompletten Systems zur inhaltsbasierten Bildsuche auf vier öffentlichen Datensätzen bestätigte die Leistungsfähigkeit der CNN-basierten Repräsentation: Für die Suche in einer Million Bildern verbessert die vorgeschlagene Erweiterung die Suchergebnisse des BoW-Modells deutlich und übertrifft die vergleichbaren Verfahren der Fachliteratur auf zwei Datensätzen.

Motiviert durch den wesentlich diskriminanteren Zugriff auf die Merkmale des 2D-Index sowie durch die Hardwareeigenschaften aktueller SSD-Laufwerke, wurde eine Realisierung der inhaltsbasierten Suche vorgeschlagen, die für den Index nicht mehr den Arbeitsspeicher, sondern eine SSD-Laufwerke vorsieht. Anhand eines konkreten Systemaufbaus wurden die resultierenden Vorteile demonstriert: die Suche mit einem Anfragebild in einer Million Datenbankbildern erfolgt in weniger als 100 Millisekunden und ist somit vergleichbar schnell wie Verfahren, die mit kompakten globalen Bildmerkmalen und Nächster-Nachbar-Suche arbeiten. Gleichzeitig entfällt eine der bisherigen Haupteinschränkungen von BoW-basierten Systemen hinsichtlich der Skalierungsfähigkeit, denn der Index kann nun auf die deutlich größeren und preisgünstigeren SSD-Laufwerke verlagert werden. Davon profitieren bereits heutige Systeme, die durch die vorgeschlagene Repräsentation bei vergleichbarer Hardware deutlich mehr Bilder für die inhaltsbasierte Bildsuche mit lokalen Merkmalen verarbeiten können.

8.2. Ausblick

Für die vorgestellte Repräsentation der Umgebung sind zunächst verschiedene inkrementelle Verbesserungen der einzelnen Komponenten vielversprechend. Für die CNN-Varianten bieten sich die neueren Netzarchitekturen wie beispielsweise *GoogleNet* [Sze15] oder *ResNet* [He16] an, und hinsichtlich der Indexierungsgeschwindigkeit die optimierten Netze *SqueezeNet* [Ian16] oder *MobileNet* [How17]. Von der Architektur der neuronalen Netze abgesehen sind bei angepassten Trainingsdaten ebenfalls bessere Ergebnisse zu erwarten. Wenn die spätere Anwendung etwa auf die Erkennung von Sehenswürdigkeiten zielt, kommen Netze in Frage, die auf dem *Places* Datensatz [Zho17] vortrainiert wurden, da dieser nur relevante Kategorien

beinhaltet und beispielsweise keine Tiere, wie dies im *ImageNet* Datensatz [Den09] unter anderem der Fall ist.

Bezüglich weiterer Daten ist aber vor allem beim Lernen des Umgebungscodebooks noch großes Potential zu erwarten. Die in dieser Arbeit gewählte Größe des Umgebungscodebooks von lediglich $\check{k} = 3025$ liegt in der Tatsache begründet, dass im verwendeten OXFORD5K Datensatz zwar tausende Bilder, aber nur elf unterschiedliche annotierte Objekte mit mehreren Ansichten enthalten sind. Ein öffentlicher Datensatz mit wesentlich mehr unterschiedlichen, aber mehrfach vorkommenden Objekten würde daher größere Umgebungscodebooks ermöglichen und zu einer repräsentativeren Unterteilung des Raums der Umgebungsmerkmale in Umgebungswörter führen.

Auch für die Evaluation wäre ein geeigneterer Datensatz wünschenswert, in dem überwiegend kleine, von sehr viel Hintergrund umgebene Objekte abgebildet sind, denn damit könnte der Vorteil der Umgebungsrepräsentation noch intensiver untersucht werden.

Aus Sicht des Gesamtsystems erscheint darüber hinaus die Integration der in den Abschnitten 2.2.4 und 2.2.3 beschriebenen ergänzenden Module Soft Quantization, Re-Ranking und Query Expansion vielversprechend. Diese sind zwar prinzipiell methodisch als orthogonal zu den Zielen dieser Dissertation anzusehen, aber durch den schnelleren und diskriminanteren Zugriff auf die indexierten Merkmale ergeben sich hier zusätzliche Fragestellungen. So sind etwa beim Soft Quantization Verfahren interessante Abwägungen zwischen den lokalen Merkmalen und den Umgebungsmerkmalen zu erwarten. Im Falle des SSD-Index könnte Soft Quantization nicht nur im Anfragebild, sondern auch für die Datenbankbilder realisiert werden, ohne die Gesamtgröße des SSD-Index zu erhöhen – schließlich sind viele der Zellen ohnehin nicht mit der maximalen Anzahl an Merkmalen belegt. Die Ansätze zu Re-Ranking und Query Expansion könnten ebenfalls von der vorgestellten Repräsentation profitieren, da durch die zwei- oder dreidimensionale Adressierung der indexierten Merkmale mehr Bilder berücksichtigt werden können und gleichzeitig weniger Ausreißer durch inkorrekte Korrespondenzen zu erwarten sind.

A

Anhang

A.1. Erforderliche Größe des visuellen Codebooks

Obwohl Bilder in Computern durch eine äußerst hochdimensionale Repräsentation verarbeitet werden, treten nur sehr wenige aller kombinatorisch möglichen Bilder in der Realität auch auf. Schon die Aufnahmeoptik und der typische Abstand der Kamera zur aufgenommenen Szene sorgen dafür, dass benachbarte Pixel mit hoher Wahrscheinlichkeit ähnliche Werte annehmen. Gleichzeitig ähneln sich die in der Natur entstandenen Strukturen wie z.B. Bäume oder Blätter. Für von Menschen gemachte Objekte gilt dies noch viel mehr, sodass sich Gradienten in lokalen Bildbereichen sehr viel öfter in vertikaler und horizontaler Richtung ausprägen als in den übrigen Richtungen. Auch bezüglich der lokalen Merkmale in Bildern wirken sich diese Wiederholungen aus, sodass der Anspruch, für eine jeweilige Anwendung nahezu alle später einmal relevanten Bilder mit einem Codebook einer begrenzten Größe erfassen zu können, durchaus gerechtfertigt erscheint. Dies soll anhand des folgenden Experiments veranschaulicht werden. Untersucht wird dabei die Frage, ob sich, ausgehend von einer vorhandenen Menge an gesehenen Bildern, mit dem sich daraus ergebenden Codebook weitere, unbekannte Bilder aufgrund der beschriebenen Wiederholungen hinreichend genau repräsentieren lassen.

Zunächst erfordert dies die Festlegung, ab wann ein Bild mit seinen lokalen Merkmalen durch ein Codebook – bis auf ein festes, aber sinnvolles Epsilon – als hinreichend genau repräsentiert gilt[1]. In diesem Experiment soll dies gelten, wenn für mindestens 75% der lokalen Merkmale eines Bildes ein visuelles Wort im Codebook existiert, das bezüglich der euklidischen Distanz zum Merkmal einen Schwellwert $\check{\varepsilon}$ unterschreitet. Dieser Schwellwert wird empirisch ermittelt, indem im OXFORD5K Datensatz für 100 laut Annotierung korrespondierende Bildpaare die jeweiligen Merkmalskorrespondenzen $\mathcal{M}_\varepsilon$ berechnet werden. Dies wird mittels des üblichen Abstandsverhältnisses von zweit-nächstem zum nächsten Nachbar im Deskriptorraum (siehe Gleichung 2.2) umgesetzt. Für die sich so ergebenden ca. 23 000 Merkmalskorrepondenzen (also im Mittel 230 pro Bildpaar) werden die Abstände im Deskriptorraum betrachtet und $\check{\varepsilon}$ wird so gewählt, dass 90% der Merkmalskorrespondenzen bezüglich ihres Abstandes unter diesen Schwellwert fallen, was $\check{\varepsilon} = 0{,}156247$ ergibt. Mit einer Teilmenge der Bilder des MIRFLICKR1M Datensatzes werden anschließend mittels hierarchischem k-Means Clusterings Codebooks unterschiedlicher Größen $(10^4, \ldots, 27^4)$ erstellt. Dabei werden jeweils so viele Bilder verwendet, dass im Mittel pro visuellem Wort 50 Merkmale für das Clustering zur Verfügung stehen, d. h. die Anzahl der „gesehenen“ Bilder soll linear mit der Codebookgröße korrelieren.

Für eine zweite, zur ersten disjunkten, Teilmenge des MIRFLICKR1M Datensatzes, bestehend aus 100 000 Bildern, wird dann evaluiert, wie groß der Anteil dieser unbekannten Bilder ist, die jeweils mit den unterschiedlichen Codebooks im oben formulierten Sinne hinreichend genau repräsentiert werden können.

Im Diagramm in Abbildung A.1 wird deutlich, dass für die gewählten Parameter bei einer Codebookgröße von ca. 250 000, bereits nahezu alle unbekannten Bilder hinreichend genau repräsentiert werden können. Da die Bilder des MIRFLICKR1M Datensatzes durchschnittlich 1 200 Merkmale aufweisen, umfasst die Menge an gesehenen Bildern, die für die Erstellung dieses Codebooks verwendet wurden, etwa 10 000 Bilder. Mit 35 000 visuel-

1 Die Repräsentation bezieht sich hier allein auf die lokalen SIFT Merkmale und nicht auf den gesamten Bildinhalt. Dass sich ein Bild aber anhand seiner lokalen Merkmale zu einem gewissen – für den Betrachter durchaus wiedererkennbaren – Teil rekonstruieren lässt, wurde in [Wei11] gezeigt.

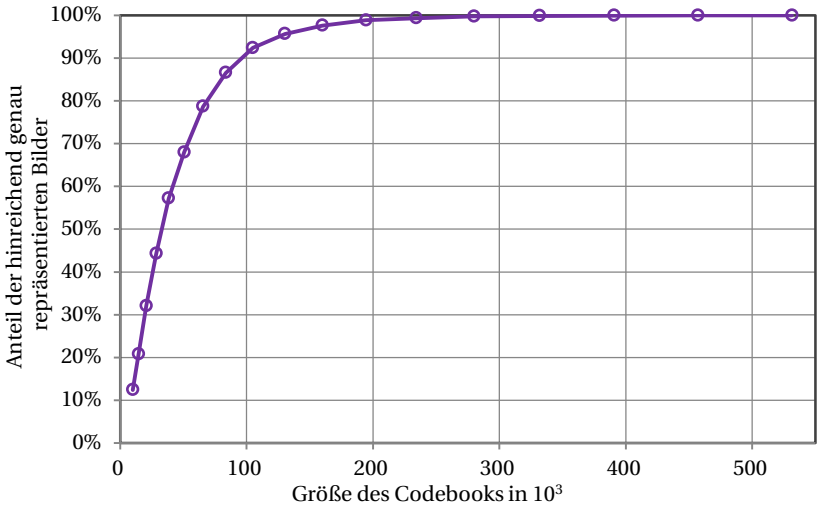

Abbildung A.1.: Erforderliche Größe des Codebooks, um unbekannte Bilder hinreichend genau (siehe Text) zu repräsentieren. Die Ordinate gibt den Anteil der 100 000 unbekannten Bilder an, der mit einem Codebook einer gewissen Größe repräsentiert werden kann.

len Worten bzw. 1 500 Bildern lassen sich bereits die Hälfte der unbekannten Bilder hinreichend genau repräsentieren. Bei der maximalen dargestellten Codebookgröße von $27^4 = 531\,441$ visuellen Wörtern dagegen gelingt die Repräsentation nur bei 29 der 100 000 Bilder nicht, was einem Anteil der hinreichend genau repräsentierten Bilder von 99,971% entspricht.

Literaturverzeichnis

[AH06] ABDEL-HAKIM, Alaa E and FARAG, Aly A: CSIFT: A SIFT descriptor with color invariant characteristics, in: *Conference on Computer Vision and Pattern Recognition*, vol. 2, IEEE (2006), pp. 1978–1983

[Ala12] ALAHI, Alexandre; ORTIZ, Raphael and VANDERGHEYNST, Pierre: Freak: Fast retina keypoint, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2012), pp. 510–517

[Aly11] ALY, Mohamed; MUNICH, Mario and PERONA, Pietro: Distributed kd-trees for retrieval from very large image collections, in: *British Machine Vision Conference*, vol. 17, Citeseer (2011)

[Ara12] ARANDJELOVIC, Relja and ZISSERMAN, Andrew: Three things everyone should know to improve object retrieval, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2012), pp. 2911–2918

[Ara13] ARANDJELOVIC, Relja and ZISSERMAN, Andrew: All about VLAD, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2013), pp. 1578–1585

[Ara16] ARANDJELOVIC, Relja; GRONAT, Petr; TORII, Akihiko; PAJDLA, Tomas and SIVIC, Josef: NetVLAD: CNN architecture for weakly supervised place recognition, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2016), pp. 5297–5307

[Avr15] AVRITHIS, Yannis; KALANTIDIS, Yannis; ANAGNOSTOPOULOS, Evangelos and EMIRIS, Ioannis Z: Web-scale image clustering revisited, in: *International Conference on Computer Vision*, IEEE (2015), pp. 1502–1510

[Azi15] AZIZPOUR, Hossein; SHARIF RAZAVIAN, Ali; SULLIVAN, Josephine; MAKI, Atsuto and CARLSSON, Stefan: From generic to specific deep representations for visual recognition, in: *Conference on Computer Vision and Pattern Recognition Workshops*, IEEE (2015), pp. 36–45

[Bab14] BABENKO, Artem; SLESAREV, Anton; CHIGORIN, Alexandr and LEMPITSKY, Victor: Neural codes for image retrieval, in: *European Conference on Computer Vision*, Springer (2014), pp. 584–599

[Bab15] BABENKO, Artem and LEMPITSKY, Victor: Aggregating local deep features for image retrieval, in: *International Conference on Computer Vision*, IEEE (2015), pp. 1269–1277

[Bay06] BAY, Herbert; TUYTELAARS, Tinne and VAN GOOL, Luc: SURF: Speeded Up Robust Features, in: *European Conference on Computer Vision*, Springer (2006), pp. 404–417

[Bit17] BITCOM: Zukunft der Consumer Technology 2017 (2017), URL https://www.bitkom.org/Bitkom/Publikationen/Zukunft-der-Consumer-Technology-2017.html

[Bos08] BOSCH, Anna; ZISSERMAN, Andrew and MUÑOZ, Xavier: Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2008), vol. 30(4):pp. 712–727

[Cal10] CALONDER, Michael; LEPETIT, Vincent; STRECHA, Christoph and FUA, Pascal: BRIEF: Binary robust independent elementary features, in: *European Conference on Computer Vision*, Springer (2010), pp. 778–792

[Cao10a] CAO, Yang; WANG, Changhu; LI, Zhiwei; ZHANG, Liqing and ZHANG, Lei: Spatial-bag-of-features, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2010), pp. 3352–3359

[Cao10b] CAO, Yang; WANG, Hai; WANG, Changhu; LI, Zhiwei; ZHANG, Liqing and ZHANG, Lei: Mindfinder: interactive sketch-based image search on millions of images, in: *18th ACM international conference on Multimedia*, ACM (2010), pp. 1605–1608

[Cao11] CAO, Yang; WANG, Changhu; ZHANG, Liqing and ZHANG, Lei: Edgel index for large-scale sketch-based image search, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2011), pp. 761–768

[Che11] CHEN, David; TSAI, Sam; CHANDRASEKHAR, Vijay; TAKACS, Gabriel; CHEN, Huizhong; VEDANTHAM, Ramakrishna; GRZESZCZUK, Radek and GIROD, Bernd: Residual enhanced visual vectors for on-device image matching, in: *Asilomar Conference on Signals, Systems and Computers*, IEEE (2011), pp. 850–854

[Chu04] CHUM, Ondrej; MATAS, Jırı and OBDRZALEK, Stepan: Enhancing RANSAC by generalized model optimization, in: *Asian Conference on Computer Vision*, vol. 2 (2004), pp. 812–817

[Chu05] CHUM, Ondrej and MATAS, Jiri: Matching with PROSAC-progressive sample consensus, in: *Conference on Computer Vision and Pattern Recognition*, vol. 1, IEEE (2005), pp. 220–226

[Chu07b] CHUM, Ondrej; PHILBIN, James; SIVIC, Josef; ISARD, Michael and ZISSERMAN, Andrew: Total recall: Automatic query expansion with a generative feature model for object retrieval, in: *International Conference on Computer Vision*, IEEE (2007), pp. 1–8

[Chu09] CHUM, Ondrej; PERD'OCH, Michal and MATAS, Jiri: Geometric min-hashing: Finding a (thick) needle in a haystack, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2009), pp. 17–24

[Chu10] CHUM, Ondřej and MATAS, Jiří: Unsupervised discovery of co-occurrence in sparse high dimensional data, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2010), pp. 3416–3423

[Chu11] CHUM, Ondřej; MIKULIK, Andrej; PERDOCH, Michal and MATAS, Jiří: Total recall II: Query expansion revisited, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2011), pp. 889–896

[Cum08] CUMMINS, Mark and NEWMAN, Paul: FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research* (2008), vol. 27(6):pp. 647–665

[Dat04] DATAR, Mayur; IMMORLICA, Nicole; INDYK, Piotr and MIRROKNI, Vahab S: Locality-sensitive hashing scheme based on p-stable distributions, in: *20th annual symposium on Computational geometry*, ACM (2004), pp. 253–262

[Dat08] DATTA, Ritendra; JOSHI, Dhiraj; LI, Jia and WANG, James Z: Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)* (2008), vol. 40(2):p. 5

[Del13] DELHUMEAU, Jonathan; GOSSELIN, Philippe-Henri; JÉGOU, Hervé and PÉREZ, Patrick: Revisiting the VLAD image representation, in: *21st ACM international conference on Multimedia*, ACM (2013), pp. 653–656

[Dem77] DEMPSTER, Arthur P; LAIRD, Nan M and RUBIN, Donald B: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977):pp. 1–38

[Den09] DENG, Jia; DONG, Wei; SOCHER, Richard; LI, Li-Jia; LI, Kai and FEI-FEI, Li: Imagenet: A large-scale hierarchical image database, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2009), pp. 248–255

[Dol12] DOLEZALEK, Stephan and FREED, Joshua: An American Kodak Moment (2012), URL http://content.thirdway.org/publications/514/Third_Way_Report_-_An_American_Kodak_Moment.pdf

[Dou10] DOUBEK, Petr; MATAS, Jiri; PERDOCH, Michal and CHUM, Ondrej: Image matching and retrieval by repetitive patterns, in: *International Conference on Pattern Recognition*, IEEE (2010), pp. 3195–3198

[Far13] FARABET, Clement; COUPRIE, Camille; NAJMAN, Laurent and LECUN, Yann: Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2013), vol. 35(8):pp. 1915–1929

[Fis81] FISCHLER, Martin A and BOLLES, Robert C: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* (1981), vol. 24(6):pp. 381–395

[Fon09] FONSECA, Manuel J; FERREIRA, Alfredo and JORGE, Joaquim A: Sketch-based retrieval of complex drawings using hierarchical topology and geometry. *Computer-Aided Design* (2009), vol. 41(12):pp. 1067–1081

[Fre03] FREUND, Yoav; IYER, Raj; SCHAPIRE, Robert E and SINGER, Yoram: An efficient boosting algorithm for combining preferences. *Journal of machine learning research* (2003), vol. 4(Nov):pp. 933–969

[Gon14] GONG, Yunchao; WANG, Liwei; GUO, Ruiqi and LAZEBNIK, Svetlana: Multi-scale orderless pooling of deep convolutional activation features, in: *European Conference on Computer Vision*, Springer (2014), pp. 392–407

[Gon15] GONG, Yunchao; PAWLOWSKI, Marcin; YANG, Fei; BRANDY, Louis; BOURDEV, Lubomir and FERGUS, Rob: Web scale photo hash clustering on a single machine, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2015), pp. 19–27

[Goo14] GOOSSAERT, Emmanuel: Coding for SSDs (2014), URL http://codecapsule.com/2014/02/12/

[Gor16] GORDO, Albert; ALMAZÁN, Jon; REVAUD, Jerome and LARLUS, Diane: Deep image retrieval: Learning global representations for image search, in: *European Conference on Computer Vision*, Springer (2016), pp. 241–257

[He16] HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing and SUN, Jian: Deep residual learning for image recognition, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2016), pp. 770–778

[How17] HOWARD, Andrew G; ZHU, Menglong; CHEN, Bo; KALENICHENKO, Dmitry; WANG, Weijun; WEYAND, Tobias; ANDREETTO, Marco and ADAM, Hartwig: Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017)

[Ian16] IANDOLA, Forrest N; HAN, Song; MOSKEWICZ, Matthew W; ASHRAF, Khalid; DALLY, William J and KEUTZER, Kurt: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016)

[Jaa99] JAAKKOLA, Tommi and HAUSSLER, David: Exploiting generative models in discriminative classifiers, in: *Advances in neural information processing systems* (1999), pp. 487–493

[Jég08] JÉGOU, Hervé; DOUZE, Matthijs and SCHMID, Cordelia: Hamming embedding and weak geometric consistency for large scale image search, in: *European Conference on Computer Vision*, Springer (2008), pp. 304–317

[Jég09a] JÉGOU, Hervé; DOUZE, Matthijs and SCHMID, Cordelia: On the burstiness of visual elements, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2009), pp. 1169–1176

[Jég09b] JÉGOU, Hervé; DOUZE, Matthijs and SCHMID, Cordelia: Packing bag-of-features, in: *International Conference on Computer Vision*, IEEE (2009), pp. 2357–2364

[Jég10a] JÉGOU, Hervé; DOUZE, Matthijs and SCHMID, Cordelia: Improving bag-of-features for large scale image search. *International Journal of Computer Vision* (2010), vol. 87(3):pp. 316–336

[Jég10b] JÉGOU, Hervé; DOUZE, Matthijs; SCHMID, Cordelia and PÉREZ, Patrick: Aggregating local descriptors into a compact image representation, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2010), pp. 3304–3311

[Jég11] JÉGOU, Hervé; DOUZE, Matthijs and SCHMID, Cordelia: Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2011), vol. 33(1):pp. 117–128

[Jég12a] JÉGOU, Hervé and CHUM, Ondřej: Negative evidences and cooccurences in image retrieval: The benefit of PCA and whitening, in: *European Conference on Computer Vision*, Springer (2012), pp. 774–787

[Jég12b] JÉGOU, Hervé; PERRONNIN, Florent; DOUZE, Matthijs; SÁNCHEZ, Jorge; PEREZ, Patrick and SCHMID, Cordelia: Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2012), vol. 34(9):pp. 1704–1716

[Joh17] JOHNSON, Jeff; DOUZE, Matthijs and JÉGOU, Hervé: Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734* (2017)

[Kal15] KALANTIDIS, Yannis; MELLINA, Clayton and OSINDERO, Simon: Cross-dimensional weighting for aggregated deep convolutional features. *arXiv preprint arXiv:1512.04065* (2015)

[Kan17] KANG, Dongku; JEONG, Woopyo; KIM, Chulbum; KIM, Doo-Hyun; CHO, Yong Sung; KANG, Kyung-Tae; RYU, Jinho; KANG, Kyung-Min; LEE, Sungyeon; KIM, Wandong ET AL.: 256 Gb 3 b/cell V-NAND Flash memory with 48 stacked WL layers. *IEEE Journal of Solid-State Circuits* (2017), vol. 52(1):pp. 210–217

[Kha12] KHAN, Fahad Shahbaz; ANWER, Rao Muhammad; VAN DE WEIJER, Joost; BAGDANOV, Andrew D; VANRELL, Maria and LOPEZ, Antonio M: Color attributes for object detection, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2012), pp. 3306–3313

[Kri12] KRIZHEVSKY, Alex; SUTSKEVER, Ilya and HINTON, Geoffrey E: Imagenet classification with deep convolutional neural networks, in: *Advances in neural information processing systems* (2012), pp. 1097–1105

[Lam09] LAMPERT, Christoph H: Detecting objects in large image collections and videos by efficient subimage retrieval, in: *International Conference on Computer Vision*, IEEE (2009), pp. 987–994

[Lan12] LAN, Tian; YANG, Weilong; WANG, Yang and MORI, Greg: Image retrieval with structured object queries using latent ranking svm, in: *European Conference on Computer Vision*, Springer (2012), pp. 129–142

[Laz06] LAZEBNIK, Svetlana; SCHMID, Cordelia and PONCE, Jean: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: *Conference on Computer Vision and Pattern Recognition*, vol. 2, IEEE (2006), pp. 2169–2178

[LeC98] LECUN, Yann; BOTTOU, Léon; BENGIO, Yoshua and HAFFNER, Patrick: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* (1998), vol. 86(11):pp. 2278–2324

[LeC10] LeCun, Yann; Kavukcuoglu, Koray and Farabet, Clément: Convolutional networks and applications in vision, in: *International Symposium on Circuits and Systems*, IEEE (2010), pp. 253–256

[LeC16] LeCun, Yann: Deep Learning and the Future of AI, CERN Colloquium (2016), URL https://indico.cern.ch/event/510372/

[Leu11] Leutenegger, Stefan; Chli, Margarita and Siegwart, Roland Y: BRISK: Binary robust invariant scalable keypoints, in: *International Conference on Computer Vision*, IEEE (2011), pp. 2548–2555

[Lev16] Levi, Gil and Hassner, Tal: LATCH: learned arrangements of three patch codes, in: *Winter Conference on Applications of Computer Vision*, IEEE (2016), pp. 1–9

[Lew06] Lew, Michael S; Sebe, Nicu; Djeraba, Chabane and Jain, Ramesh: Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* (2006), vol. 2(1):pp. 1–19

[Li06] Li, Xirong; Chen, Le; Zhang, Lei; Lin, Fuzong and Ma, Wei-Ying: Image annotation by large-scale content-based image retrieval, in: *14th ACM international conference on Multimedia*, ACM (2006), pp. 607–610

[Lia08] Liang, Shuang and Sun, Zhengxing: Sketch retrieval and relevance feedback with biased SVM classification. *Pattern Recognition Letters* (2008), vol. 29(12):pp. 1733–1741

[Lin10] Lin, Zhe and Brandt, Jonathan: A local bag-of-features model for large-scale object retrieval, in: *European Conference on Computer Vision*, Springer (2010), pp. 294–308

[Liu07] Liu, Ying; Zhang, Dengsheng; Lu, Guojun and Ma, Wei-Ying: A survey of content-based image retrieval with high-level semantics. *Pattern recognition* (2007), vol. 40(1):pp. 262–282

[Liu12] Liu, Zhen; Li, Houqiang; Zhou, Wengang and Tian, Qi: Embedding spatial context information into inverted filefor large-scale image retrieval, in: *20th ACM international conference on Multimedia*, ACM (2012), pp. 199–208

[Liu14b] LIU, Zhen; LI, Houqiang; ZHOU, Wengang; ZHAO, Ruizhen and TIAN, Qi: Contextual hashing for large-scale image search. *IEEE Transactions on Image Processing* (2014), vol. 23(4):pp. 1606–1614

[Llo82] LLOYD, Stuart: Least squares quantization in PCM. *IEEE transactions on information theory* (1982), vol. 28(2):pp. 129–137

[Lon15] LONG, J.; SHELHAMER, E. and DARRELL, T.: Fully Convolutional Models for Semantic Segmentation, in: *Conference on Computer Vision and Pattern Recognition* (2015)

[Low99] LOWE, David G: Object recognition from local scale-invariant features, in: *International Conference on Computer Vision*, vol. 2, IEEE (1999), pp. 1150–1157

[Low04] LOWE, David G: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* (2004), vol. 60(2):pp. 91–110

[Mad17] MADEO, Simone and BOBER, Miroslaw: Fast, compact, and discriminative: Evaluation of binary descriptors for mobile applications. *IEEE Transactions on Multimedia* (2017), vol. 19(2):pp. 221–235

[Mak10] MAKADIA, Ameesh: Feature tracking for wide-baseline image retrieval, in: *European Conference on Computer Vision*, Springer (2010), pp. 310–323

[Mat04] MATAS, Jiri; CHUM, Ondrej; URBAN, Martin and PAJDLA, Tomás: Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing* (2004), vol. 22(10):pp. 761–767

[Mik10] MIKULÍK, Andrej; PERDOCH, Michal; CHUM, Ondřej and MATAS, Jiří: Learning a fine vocabulary, in: *European Conference on Computer Vision*, Springer (2010), pp. 1–14

[MJH10] MARK J. HUISKES, B. Thomee and LEW, Michael S.: New Trends and Ideas in Visual Concept Detection: The MIR Flickr Retrieval Evaluation Initiative, in: *ACM International Conference on Multimedia Information Retrieval*, ACM, New York, NY, USA (2010), pp. 527–536

[Moh16] MOHEDANO, Eva; MCGUINNESS, Kevin; O'CONNOR, Noel E; SALVADOR, Amaia; MARQUÉS, Ferran and GIRÓ-I NIETO, Xavier: Bags of local convolutional features for scalable instance search, in: *ACM International Conference on Multimedia Retrieval*, ACM (2016), pp. 327–331

[Muj14] MUJA, Marius and LOWE, David G: Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2014), vol. 36(11):pp. 2227–2240

[Nis06] NISTER, David and STEWENIUS, Henrik: Scalable recognition with a vocabulary tree, in: *Conference on Computer Vision and Pattern Recognition*, vol. 2, IEEE (2006), pp. 2161–2168

[Oli01] OLIVA, Aude and TORRALBA, Antonio: Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision* (2001), vol. 42(3):pp. 145–175

[Per07] PERRONNIN, Florent and DANCE, Christopher: Fisher kernels on visual vocabularies for image categorization, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2007), pp. 1–8

[Per09] PERD'OCH, Michal; CHUM, Ondrej and MATAS, Jiri: Efficient representation of local geometry for large scale object retrieval, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2009), pp. 9–16

[Per10a] PERRONNIN, Florent; LIU, Yan; SÁNCHEZ, Jorge and POIRIER, Hervé: Large-scale image retrieval with compressed fisher vectors, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2010), pp. 3384–3391

[Per10b] PERRONNIN, Florent; SÁNCHEZ, Jorge and MENSINK, Thomas: Improving the fisher kernel for large-scale image classification, in: *European Conference on Computer Vision*, Springer (2010), pp. 143–156

[Phi07] PHILBIN, James; CHUM, Ondrej; ISARD, Michael; SIVIC, Josef and ZISSERMAN, Andrew: Object retrieval with large vocabularies and fast spatial matching, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2007), pp. 1–8

[Phi08a] PHILBIN, James; CHUM, Ondrej; ISARD, Michael; SIVIC, Josef and ZISSERMAN, Andrew: Lost in quantization: Improving particular object retrieval in large scale image databases, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2008), pp. 1–8

[Phi08b] PHILBIN, James and ZISSERMAN, Andrew: Object mining using a matching graph on very large image collections, in: *6^{th} Indian Conference on Computer Vision, Graphics & Image Processing*, IEEE (2008), pp. 738–745

[Phi10] PHILBIN, James; ISARD, Michael; SIVIC, Josef and ZISSERMAN, Andrew: Descriptor learning for efficient retrieval, in: *European Conference on Computer Vision*, Springer (2010), pp. 677–691

[Qin11] QIN, Danfeng; GAMMETER, Stephan; BOSSARD, Lukas; QUACK, Till and VAN GOOL, Luc: Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2011), pp. 777–784

[Rad16] RADENOVIĆ, Filip; TOLIAS, Giorgos and CHUM, Ondřej: CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples, in: *European Conference on Computer Vision*, Springer (2016), pp. 3–20

[Ren15] REN, Shaoqing; HE, Kaiming; GIRSHICK, Ross and SUN, Jian: Faster R-CNN: Towards real-time object detection with region proposal networks, in: *Advances in neural information processing systems* (2015), pp. 91–99

[Ros58] ROSENBLATT, Frank: The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review* (1958), vol. 65(6):p. 386

[Rub11] RUBLEE, Ethan; RABAUD, Vincent; KONOLIGE, Kurt and BRADSKI, Gary: ORB: An efficient alternative to SIFT or SURF, in: *International Conference on Computer Vision*, IEEE (2011), pp. 2564–2571

[Rui99] RUI, Yong; HUANG, Thomas S and CHANG, Shih-Fu: Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation* (1999), vol. 10(1):pp. 39–62

[Rum85] RUMELHART, David E; HINTON, Geoffrey E and WILLIAMS, Ronald J: Learning internal representations by error propagation, Tech. Rep., California Univ San Diego La Jolla Inst for Cognitive Science (1985)

[Sah12] SAHA, Sajib and DÉMOULIN, Vincent: ALOHA: An efficient binary descriptor based on Haar features, in: *International Conference on Image Processing*, IEEE (2012), pp. 2345–2348

[Sal88] SALTON, Gerard and BUCKLEY, Christopher: Term-weighting approaches in automatic text retrieval. *Information processing & management* (1988), vol. 24(5):pp. 513–523

[Sal16] SALVADOR, Amaia; GIRÓ-I NIETO, Xavier; MARQUÉS, Ferran and SATOH, Shin'ichi: Faster R-CNN features for instance search, in: *Conference on Computer Vision and Pattern Recognition Workshops*, IEEE (2016), pp. 9–16

[Sán13] SÁNCHEZ, Jorge; PERRONNIN, Florent; MENSINK, Thomas and VERBEEK, Jakob: Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision* (2013), vol. 105(3):pp. 222–245

[Sch07] SCHINDLER, Grant; BROWN, Matthew and SZELISKI, Richard: City-scale location recognition, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2007), pp. 1–7

[She12] SHEN, Xiaohui; LIN, Zhe; BRANDT, Jonathan; AVIDAN, Shai and WU, Ying: Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2012), pp. 3013–3020

[Shi15] SHI, Miaojing; AVRITHIS, Yannis and JÉGOU, Hervé: Early burst detection for memory-efficient image retrieval, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2015), pp. 605–613

[Sim12] SIMONYAN, Karen; VEDALDI, Andrea and ZISSERMAN, Andrew: Descriptor learning using convex optimisation, in: *European Conference on Computer Vision*, Springer (2012), pp. 243–256

[Sim13] SIMONYAN, Karen; PARKHI, Omkar M; VEDALDI, Andrea and ZISSERMAN, Andrew: Fisher vector faces in the wild, in: *British Machine Vision Conference*, vol. 1 (2013), p. 7

[Sim14] SIMONYAN, Karen and ZISSERMAN, Andrew: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)

[Siv03] SIVIC, J. and ZISSERMAN, A.: Video Google: a text retrieval approach to object matching in videos, in: *International Conference on Computer Vision*, IEEE (2003), pp. 1470–1477 vol.2

[Sme00] SMEULDERS, Arnold WM; WORRING, Marcel; SANTINI, Simone; GUPTA, Amarnath and JAIN, Ramesh: Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2000), vol. 22(12):pp. 1349–1380

[Sou10] SOUSA, Pedro and FONSECA, Manuel J: Sketch-based retrieval of drawings using spatial proximity. *Journal of Visual Languages & Computing* (2010), vol. 21(2):pp. 69–80

[SR14] SHARIF RAZAVIAN, Ali; AZIZPOUR, Hossein; SULLIVAN, Josephine and CARLSSON, Stefan: CNN features off-the-shelf: an astounding baseline for recognition, in: *Conference on Computer Vision and Pattern Recognition Workshops*, IEEE (2014), pp. 806–813

[Ste12] STEWÉNIUS, Henrik; GUNDERSON, Steinar H and PILET, Julien: Size matters: exhaustive geometric verification for image retrieval, in: *European Conference on Computer Vision*, Springer (2012), pp. 674–687

[Str12] STRECHA, Christoph; BRONSTEIN, Alex; BRONSTEIN, Michael and FUA, Pascal: LDAHash: Improved matching with smaller descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2012), vol. 34(1):pp. 66–78

[Sze15] SZEGEDY, Christian; LIU, Wei; JIA, Yangqing; SERMANET, Pierre; REED, Scott; ANGUELOV, Dragomir; ERHAN, Dumitru; VANHOUCKE, Vincent and RABINOVICH, Andrew: Going deeper with convolutions, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2015), pp. 1–9

[Tan15b] TANG, Xiaoxin; HUANG, Zhiyi; EYERS, David; MILLS, Steven and GUO, Minyi: Efficient selection algorithm for fast k-nn search on gpus, in: *International Parallel and Distributed Processing Symposium*, IEEE (2015), pp. 397–406

[Ti17] TECH-INSIGHTS, Seagate: Lies, Damn Lies And SSD Benchmark Test Result (2017), URL https://www.seagate.com/de/de/tech-insights/lies-damn-lies-and-ssd-benchmark-master-ti

[Tol10] TOLA, Engin; LEPETIT, Vincent and FUA, Pascal: Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2010), vol. 32(5):pp. 815–830

[Tol13] TOLIAS, Giorgos; AVRITHIS, Yannis and JÉGOU, Hervé: To aggregate or not to aggregate: Selective match kernels for image search, in: *International Conference on Computer Vision*, IEEE (2013), pp. 1401–1408

[Tol14a] TOLIAS, Giorgos and JÉGOU, Hervé: Visual query expansion with or without geometry: refining local descriptors by feature aggregation. *Pattern recognition* (2014), vol. 47(10):pp. 3466–3476

[Tol15] TOLIAS, Giorgos; SICRE, Ronan and JÉGOU, Hervé: Particular object retrieval with integral max-pooling of CNN activations. *arXiv preprint arXiv:1511.05879* (2015)

[Tor08a] TORRALBA, Antonio; FERGUS, Rob and FREEMAN, William T: 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2008), vol. 30(11):pp. 1958–1970

[Tor11] TORII, Akihiko; SIVIC, Josef and PAJDLA, Tomas: Visual localization by linear combination of image descriptors, in: *International Conference on Computer Vision*, IEEE (2011), pp. 102–109

[Tor13] TORII, Akihiko; SIVIC, Josef; PAJDLA, Tomas and OKUTOMI, Masatoshi: Visual place recognition with repetitive structures, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2013), pp. 883–890

[Tur09] TURCOT, Panu and LOWE, David G: Better matching with fewer features: The selection of useful features in large database recognition problems, in: *International Conference on Computer Vision*, IEEE (2009), pp. 2109–2116

[VDS10] VAN DE SANDE, Koen; GEVERS, Theo and SNOEK, Cees: Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2010), vol. 32(9):pp. 1582–1596

[VdW06] VAN DE WEIJER, Joost; GEVERS, Theo and BAGDANOV, Andrew D: Boosting color saliency in image feature detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2006), vol. 28(1):pp. 150–156

[Wak14] WAKATANI, Akiyoshi and MURAKAMI, Akio: GPGPU implementation of nearest neighbor search with product quantization, in: *International Symposium on Parallel and Distributed Processing with Applications*, IEEE (2014), pp. 248–253

[Wan10b] WANG, Xin-Jing; ZHANG, Lei; LIU, Ming; LI, Yi and MA, Wei-Ying: Arista-image search to annotation on billions of web photos, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2010), pp. 2987–2994

[Wan11a] WANG, Jingdong and HUA, Xian-Sheng: Interactive image search by color map. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2011), vol. 3(1):p. 12

[Wan11b] WANG, Xiaoyu; YANG, Ming; COUR, Timothee; ZHU, Shenghuo; YU, Kai and HAN, Tony X: Contextual weighting for vocabulary tree based image retrieval, in: *International Conference on Computer Vision*, IEEE (2011), pp. 209–216

[Wan17] WANG, Jingdong; ZHANG, Ting; SEBE, Nicu; SHEN, Heng Tao ET AL.: A survey on learning to hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017)

[Wei09] WEISS, Yair; TORRALBA, Antonio and FERGUS, Rob: Spectral hashing, in: *Advances in neural information processing systems* (2009), pp. 1753–1760

[Wei11] WEINZAEPFEL, Philippe; JÉGOU, Hervé and PÉREZ, Patrick: Reconstructing an image from its local descriptors, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2011), pp. 337–344

[Wie16] WIESCHOLLEK, Patrick; WANG, Oliver; SORKINE-HORNUNG, Alexander and LENSCH, Hendrik: Efficient large-scale approximate nearest neighbor search on the gpu, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2016), pp. 2027–2035

[Wit99] WITTEN, Ian H; MOFFAT, Alistair and BELL, Timothy C: Managing gigabytes: compressing and indexing documents and images (1999)

[Wu09a] WU, Zhong; KE, Qifa; ISARD, Michael and SUN, Jian: Bundling features for large scale partial-duplicate web image search, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2009), pp. 25–32

[Xia15] XIAO, Changcheng; WANG, Changhu; ZHANG, Liqing and ZHANG, Lei: Sketch-based image retrieval via shape words, in: *5th ACM on International Conference on Multimedia Retrieval*, ACM (2015), pp. 571–574

[Xie11b] XIE, Hongtao; GAO, Ke; ZHANG, Yongdong; TANG, Sheng; LI, Jintao and LIU, Yizhi: Efficient feature detection and effective post-verification for large scale near-duplicate image search. *IEEE Transactions on Multimedia* (2011), vol. 13(6):pp. 1319–1332

[Xu10a] XU, Hao; WANG, Jingdong; HUA, Xian-Sheng and LI, Shipeng: Image search by concept map, in: *33rd international ACM SIGIR conference on Research and development in information retrieval*, ACM (2010), pp. 275–282

[Xu10b] XU, Hao; WANG, Jingdong; HUA, Xian-Sheng and LI, Shipeng: Interactive image search by 2D semantic map, in: *19th international conference on World wide web*, ACM (2010), pp. 1321–1324

[Zei14] ZEILER, Matthew D and FERGUS, Rob: Visualizing and understanding convolutional networks, in: *European Conference on Computer Vision*, Springer (2014), pp. 818–833

[Zha08] ZHANG, Jiangong; LONG, Xiaohui and SUEL, Torsten: Performance of compressed inverted list caching in search engines, in: *17th international conference on World Wide Web*, ACM (2008), pp. 387–396

[Zha09c] ZHANG, Yimeng and CHEN, Tsuhan: Efficient kernels for identifying unbounded-order spatial features, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2009), pp. 1762–1769

[Zha11b] ZHANG, Yimeng; JIA, Zhaoyin and CHEN, Tsuhan: Image retrieval with geometry-preserving visual phrases, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2011), pp. 809–816

[Zha13b] ZHANG, Shiliang; TIAN, Qi; HUANG, Qingming; GAO, Wen and RUI, Yong: Multi-order visual phrase for scalable image search, in: *5th International Conference on Internet Multimedia Computing and Service*, ACM (2013), pp. 145–149

[Zha13d] ZHANG, Shiliang; YANG, Ming; WANG, Xiaoyu; LIN, Yuanqing and TIAN, Qi: Semantic-aware co-indexing for image retrieval, in: *International Conference on Computer Vision*, IEEE (2013), pp. 1673–1680

[Zhe13a] ZHENG, Liang and WANG, Shengjin: Visual phraselet: Refining spatial constraints for large scale image search. *IEEE Signal Processing Letters* (2013), vol. 20(4):pp. 391–394

[Zhe13b] ZHENG, Liang; WANG, Shengjin; LIU, Ziqiong and TIAN, Qi: Lp-norm idf for large scale image search, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2013), pp. 1626–1633

[Zhe14a] ZHENG, Liang; WANG, Shengjin; LIU, Ziqiong and TIAN, Qi: Packing and padding: Coupled multi-index for accurate image retrieval, in: *Conference on Computer Vision and Pattern Recognition*, IEEE (2014), pp. 1939–1946

[Zho11a] ZHOU, Wengang; LI, Houqiang; LU, Yijuan and TIAN, Qi: Large scale image search with geometric coding, in: *19th ACM international conference on Multimedia*, ACM (2011), pp. 1349–1352

[Zho12] ZHOU, Wengang; LU, Yijuan; LI, Houqiang and TIAN, Qi: Scalar quantization for large scale image search, in: *20th ACM international conference on Multimedia*, ACM (2012), pp. 169–178

[Zho17] ZHOU, Bolei; LAPEDRIZA, Agata; KHOSLA, Aditya; OLIVA, Aude and TORRALBA, Antonio: Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017)

[Zob06] ZOBEL, Justin and MOFFAT, Alistair: Inverted files for text search engines. *ACM computing surveys (CSUR)* (2006), vol. 38(2):p. 6

Veröffentlichungen

[Brü14] BRÜSTLE, Stefan; MANGER, Daniel; MÜCK, Klaus and HEINZE, Norbert: Identification of spatially corresponding imagery using content-based image retrieval in the context of UAS video exploitation, in: *Proc. SPIE 9076, Airborne Intelligence, Surveillance, Reconnaissance Systems and Applications XI*, International Society for Optics and Photonics (2014), p. 907603

[Cra13] CRABBE, Stephen; BLACK, Peter Ambs Sue; WILKINSON, Caroline; BIKKER, Jan; HERZ, Norbert; MANGER, Daniel; PAPE, René and SEIBERT, Helmut: Results of the FASTID project, in: *8th Security Research Conference on Future Security* (2013)

[Gri12] GRINBERG, Michael; SCHNEIDER, Nick; PAGEL, Frank; MANGER, Daniel and WILLERSINN, Dieter: Towards Video Processing in Vehicles under Adverse Weather Conditions, in: *Proc. SPIE 8550, Optical Systems Design*, vol. 8550, International Society for Optics and Photonics (2012), pp. 855020–1

[Her11] HERRMANN, Christian; MANGER, Daniel and METZLER, Jürgen: Feature-based localization refinement of players in soccer using plausibility maps, in: *International Conference on Image Processing, Computer Vision, and Pattern Recognition*, vol. 2 (2011)

[Man12] MANGER, Daniel: Large-scale tattoo image retrieval, in: *Ninth Conference on Computer and Robot Vision*, IEEE (2012), pp. 454–459

[Man13] MANGER, Daniel; WIDAK, Heiko and PAGEL, Frank: Mobile object retrieval in server-based image databases, in: *Proc. SPIE 8755,*

Mobile Multimedia/Image Processing, Security, and Applications 2013, International Society for Optics and Photonics (2013), p. 875515

[Man14] MANGER, Daniel and METZLER, Jürgen: Object detection in MOUT: evaluation of a hybrid approach for confirmation and rejection of object detection hypotheses, in: *IS&T/SPIE Electronic Imaging*, International Society for Optics and Photonics (2014), p. 90240P

[Man15a] MANGER, Daniel; MÜLLER, Markus and KUBIETZIEL, Markus: Filtering local features for logo detection and localization in sports videos, in: *International Conference on Signal and Image Processing Applications*, IEEE (2015), pp. 505–508

[Man15b] MANGER, Daniel and PAGEL, Frank: Camera-based forecasting of insolation for solar systems, in: *SPIE/IS&T Electronic Imaging*, International Society for Optics and Photonics (2015), p. 94050M

[Man16a] MANGER, Daniel; HERRMANN, Christian and WILLERSINN, Dieter: Towards Extending Bag-of-Words-Models Using Context Features for an 2D Inverted Index, in: *International Conference on Digital Image Computing: Techniques and Applications*, IEEE (2016), pp. 1–5

[Man16b] MANGER, Daniel; PAGEL, Frank; ARNOLDT, Alexander and WARWEG, Oliver: Camera-based forecasting of cloud coverage for optimization of energy grids, in: *SPIE Remote Sensing*, International Society for Optics and Photonics (2016), p. 100010T

[Man17a] MANGER, Daniel and WILLERSINN, Dieter: Enlarging the Discriminability of Bag-of-Words Representations with Deep Convolutional Features (Best Paper Award | 1st runner up), in: *7th International Conference on Image Processing Theory, Tools and Applications* (2017)

[Man17b] MANGER, Daniel and WILLERSINN, Dieter: Extending the Bag-of-Words Representation with Neighboring Local Features and Deep Convolutional Features, in: *Irish Machine Vision and Image Processing Conference* (2017)

[Man18] MANGER, Daniel; WILLERSINN, Dieter and BEYERER, Jürgen: Towards Large-scale Image Retrieval with a Disk-only Index, in: *13th*

International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, vol. 5 (2018), pp. 367–372

[Mül11] MÜLLER, Thomas; MANGER, Daniel and METZLER, Jürgen: Recognition of Soccer Players after Occlusions using Temporal Color Signatures, in: *International Conference on Image Processing, Computer Vision, and Pattern Recognition* (2011)

[Mül13] MÜLLER, Thomas and MANGER, Daniel: Person detection in LWIR imagery using image retrieval, in: *Proc. SPIE 8744, Automatic Target Recognition XXIII*, International Society for Optics and Photonics (2013)

[Sch12] SCHNEIDER, Nick; GRINBERG, Michael; PAGEL, Frank; MANGER, Daniel and WILLERSINN, Dieter: Fahrzeugtracking unter widrigen Witterungsbedingungen, in: *Forum Bildverarbeitung 2012*, KIT Scientific Publishing (2012), p. 291

[Tüz18] TÜZKÖ, Andras; HERRMANN, Christian; MANGER, Daniel and BEYERER, Jürgen: Open Set Logo Detection and Retrieval, in: *13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 5 (2018), pp. 284–292

[Wil16] WILLERSINN, Dieter; MANGER, Daniel; ERDNÜSS, Bastian; BAUMGÄRTNER, Dietmar and GIRAUD, Frédéric: Safety Assessment of Windshield Washing Technologies. *International Journal of Automotive Engineering* (2016), vol. 7(3):pp. 91–98

Quellenverzeichnis

- Abbildung 1.2 b): Von Frank Schulenburg, https://commons.wikimedia.org/wiki/File:Tacorón_Creek-El_Hierro.jpg,
  Lizenz: CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0/deed.en
- Abbildung 1.2 c): Von Simoncio, https://commons.wikimedia.org/wiki/File:Flysch_-_Zumaia_(3).JPG,
  Lizenz: CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0/deed.en
- Abbildung 1.2 d): Von Tobias Helfrich, https://commons.wikimedia.org/wiki/File:Irland_parknasilla_ring_of_kerry.jpg,
  Lizenz: CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0/deed.en
- Abbildung 1.2 e): Von Frank Schulenburg, https://commons.wikimedia.org/wiki/File:Northern_California_Coast_as_seen_from_Muir_Beach_Overlook.jpg,
  Lizenz: CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0/deed.en

Abkürzungen

AKM Approximatives k-Means 18

BoW Bag-of-(visual)-Words 15

CBIR Content-Based Image Retrieval iii

CN ColorName 38

CNN Convolutional Neural Network 46

CPU Central Processing Unit 13

DoG Difference-of-Gaussian 10

EM Expectation-Maximization 42

FNR Falsch-Negativ-Rate 69

FV Fisher Vektor 72

FPR Falsch-Positiv-Rate 69

GMM Gauß'sches Mischmodell 42

GPU Graphics Processing Unit 13

GVP Geometry-Preserving Visual Phrase 33

HE Hamming Embedding 26

HKM Hierarchisches *k*-Means 18

IDF Inverse Document Frequency 19

IOPS Input/Output Operations Per Second 121

LDA Linear Discriminant Analysis 15

LSH Locality Sensitive Hashing 14

MSER Maximally Stable Extremal Regions 35

MVP Multi-Order Visual Phrase 36

PCA Principle Component Analysis 27

RAM Random-Access Memory 120

RPN Region-Proposal-Network 49

SIFT Scale-Invariant Feature Transform 5

SQ Soft Quantization 25

SSD Solid-State-Disk ii

SVM Support Vektor Maschine 30

TF Term Frequency 19

VLAD Vector of Locally Aggregated Descriptors 44

VGG Visual Geometry Group (Oxford University) 84

WGC Weak Geometric Consistency 32

Symbolverzeichnis

a_σ, a_θ	Anzahl der Akkumulatorintvervalle (→WGC)
b_{cnn}	Offset-Wert eines Perzeptrons (→CNN)
b	Komponente einer Binärsignatur (→HE)
c	Kanal einer Feature Map (→CNN)
$g_1, \ldots, g_4$	Größen, die die relative Anordnung von Merkmalen beschreiben
h	Hamming-Distanz (→HE)
k	Größe des Codebooks der visuellen Wörter
$\check{k}$	Größe des Codebooks der Umgebungswörter
$\check{k}$	Indexvariable für eine der K Komponente des GMM
k_{gvp}	Länge der Geometry-preserving Visual Phrase (→GVP)
k_{mvp}	Ordnung der Multi-order Visual Phrase (→MVP)
m	Medianwert (→HE)
m_{gvp}	Wert im Akkumulator des MVP
n	Anzahl der lokalen Merkmale eines Bildes
n_A	Anzahl der Anfragebilder
n_{cnn}	Anzahl der Eingänge eines Perzeptrons (→CNN)
n_q	Anzahl der lokalen Merkmale in einem Anfragebild
n_d	Anzahl der lokalen Merkmale in einem Datenbankbild
z	Dimensionalität der lokalen Merkmale (z. B. 128 für SIFT)
$\grave{z}$	Dimensionalität der binären HE-Signatur
$\check{z}$	Dimensionalität der Umgebungsmerkmale

$\tilde{z}$	Dimensionalität der Fisher Vektoren
q	Quantisierungsfunktion
s	Ähnlichkeit zweier Bilder bei Indexierung der visuellen Wörter
$s^\star$	Ähnlichkeit zweier Bilder bei Indexierung der lokalen Merkmale
t	Indikatorvariable für die Korrektheit eines Ergebnisbildes
w	Komponente eines BoW-Vektors
x	X-Koordinate eines lokalen Merkmals im Bild
y	Y-Koordinate eines lokalen Merkmals im Bild
y_{cnn}	Ausgabe eines Perzeptrons (→CNN)

$\boldsymbol{b}$	Binärsignatur des Hamming Embeddings (→HE)
$\boldsymbol{c}$	Visuelles Wort (d. h. ein Element eines Codebooks)
$\boldsymbol{d}$	Deskriptor eines lokalen Merkmals
$\grave{\boldsymbol{d}}$	Projizierter Deskriptor (→HE)
$\boldsymbol{f}$	Lokales Merkmal
$\boldsymbol{m}$	Medianvektor für ein visuelles Wort (→HE)
$\boldsymbol{s}$	ColorName Deskriptor
$\boldsymbol{u}$	Umgebungsmerkmal
$\boldsymbol{w}$	BoW-Vektor
$\boldsymbol{w}_{cnn}$	Gewichtsvektor eines Perzeptrons (→CNN)
$\boldsymbol{x}_{cnn}$	Eingang eines Perzeptrons (→CNN)

A	Akkumulator (bei Indexierung der visuellen Wörter)
$A^\star$	Akkumulator (bei Indexierung der lokalen Merkmale)
B	Anzahl der Parameter des generativen Modells (→GMM)
F	Feature Map (→CNN)
G	Gradient der Log-Likelihood-Funktion (→FV)
H	Höhe einer Feature Map (→CNN)

I	Bild
K	Anzahl der Mixturen im Gauß'schen Mischmodell (→GMM)
K_h	Hellinger Kernel
L	Cholesky-Zerlegung von Ψ^{-1} (→FV)
N	Anzahl der Bilder in einer Datenbank
P	Projektionsmatrix (→HE)
U	Umgebung in einer Feature Map (→CNN)
W	Breite einer Feature Map (→CNN)
X	Bildbreite
Y	Bildhöhe

$\mathcal{C}$	Visuelle Wörter des Codebooks
$\mathcal{D}$	Deskriptoren
$\mathcal{F}$	Lokale Merkmale
$\mathcal{K}_k$	Korrekte Korrespondenzen
$\mathcal{K}_i$	Inkorrekte Korrespondenzen
$\mathcal{L}$	Ergebnisliste einer Suchanfrage
$\mathcal{M}_\varepsilon$	Korrespondenzen (anhand des Abstandsverhältnisses)
$\mathcal{M}_g$	Korrespondenzen (anhand plausibler Nachbarkorrespondenzen)
$\mathcal{M}_\mathcal{C}$	Korrespondenzen (basierend auf einem Codebook $\mathcal{C}$)
$\mathcal{N}$	Normalverteilung
$\mathcal{Q}$	Inverse Liste (bei Indexierung der visuellen Wörter)
$\mathcal{Q}^\star$	Inverse Liste (bei Indexierung der lokalen Merkmale)
$\mathcal{U}$	Merkmale innerhalb einer Umgebung

α	Gewichtung einer Komponente des GMM
α_{mvp}	Gewichtung bei der Multi-order Visual Phrase (→MVP)
α_{cw}	Gewichtung beim Contextual Weighting [Wan11b]

γ	Zuordnung zu einer Komponente des GMM
δ	Kronecker-Delta
ε	Schwellwert bei der Berechnung von Korrespondenzen
$\check{\varepsilon}$	Schwellwert für Korrespondenzen im Experiment in Anhang A.1
ζ_{CNN}	Faktor für die Größe der Umgebung eines lokalen Merkmals (→CNN)
ζ_{FV}	Faktor für die Größe der Umgebung eines lokalen Merkmals (→FV)
η	Anzahl der Bilder, die beim Re-Ranking berücksichtigt werden
θ	Orientierung eines lokalen Merkmals
κ	Größe eines Faltungskernes (→CNN)
λ	Anzahl der Faltungskanäle = Anzahl der Feature Maps (→CNN)
ξ	Anzahl der nächsten Nachbarn bei Soft Quantization (→SQ)
ρ	Grenze für den Skalierungsunterschied der lokalen Merkmale
σ	Skalierung eines lokalen Merkmals
τ	Schwellwert für den Vergleich von Binärsignaturen des HE
ϕ	Komponente des Fisher Vektors
φ	Nichtlineare Funktion eines Perzeptrons (→CNN)
$\boldsymbol{\beta}$	Parameter des probabilistischen Modells (→FV)
$\boldsymbol{\mu}$	Mittelwert einer Komponente des GMM
$\boldsymbol{\sigma}$	Varianz einer Komponente des GMM (Diagonale von Σ)
$\boldsymbol{\omega}$	Gewichteter BoW-Vektor
Υ	Fisher Kernel (→FV)
Σ	Diagonale Kovarianzmatrix einer Komponente des GMM
Ψ	Fisher Informationsmatrix (→FV)
$\boldsymbol{\Phi}$	Fisher Vektor (FV)