DISS. ETH NO. 23673

Improving image retrieval by introducing locality sensitive encoding to visual similarity measures

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZÜRICH
(Dr. sc. ETH Zürich)

presented by

Danfeng Qin
Msc. ITEC. ETH
born on October 24, 1984
citizen of China

accepted on the recommendation of

Prof. Dr. Luc Van Gool, examiner
Prof. Dr. Tinne Tuytelaars co-examiner
Dr. Matthieu Guillaumin, co-examiner

2016

Abstract

The objective of this thesis is to develop a large scale image retrieval system in which search is purely based on visual analysis. Specifically, given a query image and a large database of reference images, we are interested in retrieving images in the database that depict the same object as the query one. Even though astonishing progress has been made in recent years in terms of scalability and precision, accuracy on common retrieval benchmarks still shows room for significant improvements. Therefore, the main focus of this thesis is to improve the accuracy of retrieval systems while keeping search speed near real-time and memory consumption manageable.

At the heart of many image retrieval systems lies the pairwise image similarity measure. Over the years through much experimentation we have found out that many available similarity measures are only reliable in a localized region of the descriptor space. They should not be but are commonly used to measure visual similarities globally. To address this problem, we present four contributions to make these measures more accurate by adapting them using locality information.

As a first contribution, we present a probabilistic framework to model feature-to-feature similarity for high-dimensional local features. We show by experiment that the de facto standard Euclidean distance is discriminating locally but not globally, and therefore propose to adapt the original Euclidean distance by a local neighborhood statistic of a query feature. We also propose a function to score the individual feature-to-feature contributions to an image-to-image similarity. Experimental results show that our method consistently gives a significant boost to retrieval accuracy.

As a second contribution, we analyze the commonly used hand crafted methods for measuring pairwise similarity between bag of visual words (BoVW) histograms, and propose a simple linear additive function which can well approximates these functions. We illustrate that an approximation of the mean

average precision (mAP) of the retrieval system can be directly maximized by optimizing the parameters of this linear model. We also show how our model integrates into an efficient inverse file structure and thus how to use it in large-scale retrieval scenarios. Our experimental results confirm the effectiveness of our method.

The third contribution is a method for improving image retrieval precision and recall by introducing k-reciprocal nearest neighbors as blind relevance feedback to rerank images. Due to the curse of dimensionality and the highly inhomogeneous nature of image space, most existing measures for modeling pairwise image similarity can only work locally. Comparing pairwise image similarity in one region of image space to pairwise similarity in another is generally problematic. However, to obtain the size of the neighborhood in which a given similarity measure works reliably is very difficult. In experiments, we observe that k-reciprocal nearest neighbors is a surprisingly reliable measurement. Therefore, we propose to expand our knowledge of the query according to its k-reciprocal neighbors, and rerank other images in the database according to their similarity to this relevant set of images. We evaluate our approach on common object retrieval benchmarks and demonstrate a significant improvement over a standard bag-of-words retrieval.

As a fourth contribution, we introduce a simple yet powerful family of kernels, quantized kernels (QK), which model non-linearities and heterogeneities in the data efficiently and effectively. In essence, we build on the fact that vector quantizers project data into a finite set of N elements, the index space, and on the simple observation that kernels on finite sets are fully specified by the Gram matrix of these elements (the kernel matrix), which we propose to learn directly. Thus, QKs are piecewise constant locally but arbitrary globally, making them very flexible. Since the learnt kernel matrices are positive semi-definite, we directly obtain the corresponding explicit feature mappings and exploit their potential low rank. As a result, we obtain state-of-the-art matching performance on a standard benchmark dataset using only a few bits to represent each feature dimension.

Zusammenfassung

Das Ziel dieser Arbeit ist es ein im groen Umfang angelegtes Bildsuchsystem zu entwickeln, welches rein auf visueller Analyse basiert. Besonders, gegeben ein Bild und eine groe Datenbank an Referenzbildern, sind wir daran interessiert Bilder aus dieser Datenbank wiederzufinden, welche die gleichen Objekte wie das Suchbild zeigen. Nebst dem unglaublichen Erfolg der in den letzten Jahren im Bezug auf Skalierbarkeit und Przision erreicht werden konnte, kann die Genauigkeit auf standard Such-Benchmarks noch weiter verbessert werden. Daher ist der Hauptfokus dieser Arbeit jener, die Genauigkeit der Bildsuchsysteme zu verbessern und zugleich die Geschwindigkeit echtzeit-fhig und den Speicherbedarf handhabbar zu halten.

Im Herzen von vielen Bildsuchsystemen liegt der paarweise Vergleich von Bildhnlichkeiten. ber die Jahre und durch viele Experimente konnten wir herausfinden, dass die Bildhnlichkeit nur in lokalen Bereichen des Beschreibungsraum stabil und verlsslich ist. Diese Mae sollten nicht - werden aber dennoch - fr den globalen visuellen Vergleich von Bildhnlichkeiten verwendet. Um dieses Problem zu adressieren, schlagen wir vier wissenschaftliche Beitrge vor, um diese hnlichkeitsmasse genauer zu machen mittels adaptiver lokaler Information.

Als erster Beitrag prsentieren wir ein wahrscheinlichkeitstheoretisches Framework um die Bildmerkmal zu Bildmerkmal-hnlichkeit fr hoch-dimensionierte lokale Bildmerkmale zu modellieren. Wir zeigen durch Experimente, dass der de facto Standard der Euklidischen Distanz nur lokal diskriminativ ist jedoch nicht global. Daher schlagen wir schlagen vor, die originale Euklidische Distanz zu adaptieren, in dem eine lokale Nachbarschaftstatistik fr jedes Bildmerkmal verwendet wird. Weiters schlagen wir eine Funktion vor, welche die individuellen Bildmerkmal zu Bildmerkmal hnlichkeiten in eine bessere Bild zu Bild hnlichkeit auswertet. Experimentelle Ergebnisse zeigen dass unsere Methode konsistent signifikante Steigerungen der Suchgenauigkeit liefert.

Als zweiter Beitrag analysieren wir die blich handgefertigten Methoden, um paarweise hnlichkeit zwischen Bag of Visual Words (BoVW) Histogrammen zu messen, und schlagen einfache linear additive Funktionen vor, welche eine gute Annherung bieten. Wir illustrieren wie die Annherung des mean average precision (mAP) Werts des Suchsystems durch Optimierung der Parameter des linearen Modells maximiert werden kann. Weiters zeigen wir wie unser Modell in das effiziente inverted file structure Framework integriert werden kann. Unsere experimentellen Ergebnisse besttigen die Effektivitt unserer Methode.

Der dritte Beitrag ist eine Methode um die Precision und Recall Werte des Bildsuchsystems zu verbessern, indem ein k-reziprokaler Nachbarvergleich als blindes Relevanz-Feedback fr eine Neusortierung eingefhrt wird. Wegen dem Fluch der Dimensionen und der hohen Inhomogenitt des Bildraums, knnen viele der paarweisen Bildvergleich nur lokal arbeiten. Damit ist der Vergleich der paarweisen hnlichkeit von einem Bereich des Bildraums mit einem anderen Bereich generell problematisch. Jedoch die genaue Gre der Nachbarschaft fr welches ein hnlichkeitsma verlsslich funktioniert zu bestimmen ist schwer. In Experimenten, konnten wir beobachten dass k-reziprokaler Nachbarvergleiche erstaunlich verlssliche Ergebnisse liefert. Daher schlagen wir vor die Suchergebnisse durch deren k-reziprokalen Nachbarn zu erweitern, um so andere Bilder nach dieser Relevanzliste neu zu sortieren. Wir evaluieren unseren Ansatz auf standard Objektsuchdatenbanken und zeigen signifikante Verbesserungen im Vergleich zu standard Bag of Words Systemen.

Als vierten Beitrag, stellen wir eine einfache jedoch mchtige Familie von Kerneln (quantisierte Kernel) vor, welche Nichtlinearitten und Heterogenitten in Daten effizient und effektiv modellieren knnen. Im Wesentlichen berufen wir uns auf dem Fakt, dass Vektorquantifizierung die Daten auf eine endliche Menge von Elementen, dem Indexraum, projiziert. Weiters, verwenden wir die Beobachtung, dass Kernel auf endlichen Mengen vollstndig durch die Gram Matrix auf diesen Elementen (kernel matrix) bestimmt sind und schlagen vor, diese direkt zu lernen. Demnach sind QKs lokal stckweise konstant aber global beliebig, und somit sehr flexibel. Da die gelernten Kernelmatrizen positiv semidefinit sind, erhalten wir direkt die korrespondierende explizite Bildmerkmal-Abbildung und knnen deren mglichen niedrigen Rang ausnutzen. Somit knnen wir state-of-the-art Ergebnisse auf blichen Datenbanken erreichen, wobei wir nur wenige Bits pro Dimension bentigen.

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Luc Van Gool. He gave me so much freedom to pursue my own research interests, and gave me endless support through difficulties both in research and in my personal life. Ever a source of knowledge and inspiration, he helped me find the way many a time I was lost.

I would like to thank Stephan Gammeter for getting me started in research. While working with him at the beginning of my PhD, he taught the nuts and bolts of research, and how to write and present a paper. He set the example for me to follow to be a good PhD student and engineer.

I would also like to thank Lukas Bossard, whose expertise I could always rely on when an engineering problem became too tough. Matthieu Guillaumin, for his excellent knowledge of machine learning theory, and being a model research scientist. Christian Wengert, for teaching me a lot of retrieval system hacks. Till Quack, for being an amazing mentor and manager. All of the aforementioned, for being such a pleasure to work with.

Angela Yao, Valeria De Luca and Dengxin Dai were good company and moreover great friends, who brought sunshine to my day whenever I was stuck under a cloud. They helped me through many dull and difficult moments. I also thank my many other wonderful colleagues at ETH, who I will dearly miss, along with the excellent lab cake parties there.

A very special acknowledgement goes to James Lyon, who gave me unlimited support and love during the final, critical months of my dissertation.

Lastly, I could not have done this without endless love and support from my parents. They always valued education very highly, and offered all they had to allow me to follow my dreams.

Contents

List of Figures ix

List of Tables xiii

1 Introduction 1
1.1 Objective . . . 1
1.1.1 Applications . . . 2
1.2 Challenges . . . 4
1.3 Contributions . . . 7

2 Overview of Retrieval System 11
2.1 Image Representation Based Local Features . . . 12
2.1.1 Local Features . . . 12
2.1.2 Bag of features representation . . . 16
2.1.3 Aggregating local features into a global representation . . . 21
2.2 Global Descriptors based Image Representation . . . 22

3 Dataset and Evaluation 25
3.1 Dataset . . . 25
3.1.1 Oxford Buildings . . . 25
3.1.2 Paris Buildings . . . 26
3.1.3 UKbench . . . 26
3.1.4 Holidays . . . 26
3.2 Evaluation Protocol . . . 26

4 Query Adaptive Similarity 29
4.1 Introduction . . . 29
4.2 Related Work . . . 30
4.3 Our Approach . . . 32

4.3.1 A probabilistic view of similarity estimation 32
4.3.2 Estimation accuracy 33
4.3.3 Ground truth data generation 34
4.3.4 Query adaptive distance 35
4.3.5 Similarity function 37
4.3.6 Overall method 39
4.4 Experiments 41
4.4.1 Implementation details 41
4.4.2 Parameter selection 41
4.4.3 Effectiveness of our method 42
4.5 Results 43
4.5.1 Comparison with state-of-the-art 44
4.5.2 Computational Complexity 47
4.6 Conclusion 48

5 Learning to rank Bag-of-Words Histogams 49
5.1 Introduction 49
5.2 Related work 50
5.3 Learning to rank histograms by similarity 53
5.3.1 A linear approximation of histogram similarity . . . 53
5.3.2 Learning to rank query-document pairs 54
5.3.3 Robust parameter estimation 56
5.3.4 Integrating in an inverted index 57
5.4 Experiments 58
5.4.1 Dataset, feastures, evaluation protocol, and implementation details 58
5.4.2 Qualitative analysis of our approach 59
5.4.3 Comparison to the state of the art 61
5.5 Conclusion 64

6 Reciprocal Nearest Neighbor 65
6.1 Introduction 65
6.2 Related work 66
6.3 Motivation 67
6.4 Our Approach 71
6.4.1 Close set construction 71
6.4.2 Far set re-ranking 74
6.5 Experiments 75

6.5.1 Evaluated datasets . . . 75
6.5.2 Close set accuracy . . . 76
6.5.3 Far set accuracy . . . 76
6.5.4 Full method . . . 77
6.6 Conclusion . . . 82

7 Quantized Kernel Learning For Feature Matching 85
7.1 Introduction . . . 85
7.2 Related work . . . 87
7.3 Quantized kernels . . . 88
7.3.1 Definition and properties . . . 88
7.3.2 Learning quantized kernels . . . 89
7.3.3 Interval quantization optimization for a single dimension . . . 90
7.3.4 Learning higher dimensional quantized kernels . . . 92
7.4 Results . . . 94
7.5 Conclusion . . . 102

8 Conclusions 103
8.1 Contributions . . . 103
8.2 Future Work . . . 104

List of Figures

1.1 Object of interest could be blurred when taken on a foggy or snowy day. . . . 5
1.2 Object of interest could look very different when taken under different viewpoint. . . . 6
1.3 Object of interest could be occluded or cropped. . . . 6
1.4 Object of interest could be of very low resolution. . . . 7
1.5 Object of interest could look very different when taken at a different time of a day. . . . 7

4.1 Corresponding image patches for two randomly selected points of the simulated data . . . 35
4.2 Distribution of the Euclidean distance for two points from the simulated data. The solid lines show the distribution for corresponding features $T(x_i)$, whereas the dotted line depict non-corresponding ones $F(x_i)$. . . . 36
4.3 The comparison of our adaptive distance to the Euclidean distance on dataset $\mathcal{D}$. The solid lines are the distance distribution of the matched pairs, and the dotted lines are the distance distribution of non-matched pairs. The green dashed lines denotes where the probability of the non-matching distance exceed 0.1%, i.e, the non-matching feature is very likely to dominate our observation. A comparison of the right tails of both distributions is shown in (c). . . . 38
4.4 Feature similarity evaluated on dataset $\mathcal{D}$. Red lines are the visual similarity for different c evaluated on the simulated data. The blue line is our final similarity function with $\alpha = 9$. . . . 39

4.5 Comparison of our adaptive distance with Euclidean distance on Oxford5k dataset 43
4.6 Retrieval Performance by using top k nearest neighbor as similar features [Jégou *et al.*, 2011b] 44

5.1 Visual word log-frequency in Oxford105k. Colors illustrate the proposed clustering with 5 groups. 55
5.2 Weight pattern learnt on the UKbench^s data. This pattern was learnt for $N_v = 5$, $N_w = 8$ and enforcing symmetry. . 55
5.3 Comparison of the performance of learning-to-rank formulations. 60
5.4 Weights learnt for UKbench (pattern shown in Fig. 5.2) for each of the $N_v = 5$ word group. 60
5.5 Influence of the number of training samples on the top-4 score. 60
5.6 Learnt weight pattern for Oxford105k^s ($N_v = 5, N_w = 5$) . 60
5.7 Learnt weight pattern for Holidays^s ($N_v = 2, N_w = 6$, sym.) 60
5.8 Learnt weight pattern for UKbench^s ($N_v = 5, N_w = 8$, sym.) 60

6.1 Degradation of similarity in document space for a typical query from the Oxford5k data set. The y-axis shows $\mathrm{sim(q, d)}$, the x-axis the rank of retrieved images. The red circles show true positives at their similarity and rank. . . 68
6.2 Difference between the unidirectional nearest neighbor set $top(2, q)$ and the 2-reciprocal nearest neighbor set $R(2, q)$. 69
6.3 Precision and recall of $R(k, q)$ in comparison to the top(k, q). 71
6.4 Overview over the expansion rules. For the new set $N_{q,t+1}$ only nodes are considered which either occur in the first half of the top-k list of the query image ($q \rightarrow n_3$), or if the query image occurs within the top list of the image ($n_1 \rightarrow q$). 73
6.5 Example for the *close set* in the expansion step. 74
6.6 Mean average precision for Paris. 77
6.7 Mean average precision for Oxford5k. 78
6.8 Mean average precision for Oxford105k. 78
6.9 Top-4 score for Kentucky. 79
6.10 Mean average precision for INRIA Holidays. 79

6.11 Average precision (*AP*) of the baseline versus AP of our method 80

7.1 Impact of N, the number of quantization intervals 94
7.2 Impact of G, the number of dimension groups 94
7.3 Our learned feature maps and additive quantized kernel of a single dimension. (a) shows the quantized kernel in index space, while (b) is in the original feature space for the first quantizer. (c,d) show the two corresponding feature maps, and (e,f) the related rank-1 kernels. 97
7.4 ROC curves when evaluating Notre Dame (top) and Liberty (bottom) from Yosemite 100

List of Tables

4.1 Influence of the size of the random feature set for each inverted list on Oxford5k 42
4.2 Influence of the cut-off value β on Oxford5k 42
4.3 Performance of our method on public datasets. 44
4.4 Comparisons with state-of-the-art methods without applying image level post-processing. * indicates the score of merging Oxford5k and Paris and 100K distractor images. ** denotes the result obtained by manually rotating all images in the Holidays dataset to be upright. 46
4.5 Comparisons with the state of art methods with post-processing in image level. ** denotes the result obtained by manually rotating all images in the Holidays dataset to be upright. 46
4.6 Comparison to Jégou *et al.* [2011a] using more bytes per feature. 47

5.1 Comparison of ad-hoc similarities on the original versions of the datasets (mAP or top-4 score depending on the dataset). In bold we show the best results: they are comparable with state-of-the-art results available in the literature for comparable approaches. 63
5.2 Comparison to alternative similarities. We report the average performance over the 10 splits of the data (mAP or top-4 score depending on the dataset) and in parenthesis the number of runs where the method is the best. In bold is the best result for each dataset. 63

6.1 mAP for different datasets compared to results of state of the art results. 81

6.2 Additional memory overhead per dataset and average time overhead per query. 82

7.1 Impact of quantization optimization for different quantization strategies 94
7.2 Performance of kernels on different datasets with different descriptors. AQK(N) denotes the additive quantized kernel with N quantization intervals. Following [Brown and Lowe, 2007], we report the False positive rate (%) at 95% recall. The best results for each descriptor are in bold. 96
7.3 Performance comparison of different compact feature encoding. The number in the table is reported as False positive rate (%) at 95% recall. The best results for each group are in bold. 101

1

Introduction

1.1 Objective

We are living in an era of exponential data growth: with every passing minute 2.5 million pieces of content are shared on Facebook, nearly 300,000 tweets are sent on Twitter, nearly 220,000 photographs are posted on Instagram, 72 hours of video is uploaded to YouTube, nearly 50,000 apps are downloaded from Apple's app store, over 200 million emails are delivered and Amazon generates over $\$80,000$ in online sales[1]. With the booming of multimedia data on the web from around the world, there is an urgent need to effectively and efficiently index and retrieve this data, particularly image data.

Advanced systems for information retrieval have revolutionized many aspects of our daily lives. They provide us an efficient and convenient way to acquire knowledge. They help us to make good decisions, whether about which documents to read, which products to buy, or which places to visit. However, the applicability of traditional text based retrieval systems is inherently limited since they only accept text queries and search documents purely based on textual analysis. In many situations, it is actually hard for us to express our needs in the form of keywords. For instance, it is almost impossible to figure out what we are looking at by trying to summarize the appearance of an unknown object with a few keywords and query over internet. As vision is the most important sense that connects us to the world, the ability to query by visual example becomes critical, especially during moments when we want to interact across both the physical world and the digital one.

[1] According to an infographic published by DOMO in 2014. https://www.domo.com/blog/2014/04/data-never-sleeps-2-0/

The objective of this thesis is to develop a large scale image retrieval system in which search is purely based on visual analysis. Specifically, given a query image and a large database of reference images, we are interested in retrieving images in the database that depict the same object as the query one. Notably, we aim to search at the level of a specific object, such as finding images containing the "Eiffel Tower" as the query image does, but not images containing other buildings. The typical object types we work with consist of landmark buildings, scenery or other rigid 3D objects.

Retrieving images in a large scale dataset is a very challenging problem. The principal difficulties are 1) unstructured dataset, 2) significant variation in object appearance under different viewing conditions, 3) the large scale of the dataset and 4) the requirement of real time performance. Following is a list of a high level overview about how we deal with these challenges. A big challenge in image retrieval is that the reference database is usually unstructured: we do not know how many objects it contains, and which of them contain the same object. Accordingly, we can hardly pre-learn a specific model for each object in the dataset as done in generic object recognition. Instead, in this thesis, we aim at deriving a general and object independent similarity function which models the likelihood that a pair of images contains the same object. Another main challenge of image retrieval is raised by imaging conditions. Under different lighting conditions or viewpoints, object appearances vary enormously. Thereby it is in general very difficult to establish correspondences between images of the same object under unknown geometric and illumination transformation. Towards this end, this thesis aims at developing a visual similarity function that is invariant to unknown imaging conditions. Similar to text retrieval system, the performance of image retrieval system largely depends on the speed of query processing and the scale of the database. However, how to scale up the available image retrieval system to another level, or how to speed up is out of the scope of this thesis. We mainly focus in improving retrieval accuracy by developing more accurate visual models while keeping the search speed in near real time and operate on datasets at the scale of millions to billions of images.

1.1.1 Applications

Large scale object retrieval system has a plethora of applications, some of which are listed here.

Product Recognition Powered by advanced image retrieval system, the technology of real time product recognition on mobile devices could revolutionize the way we shop. It can provide an unprecedented and superior in-store experience: grocery shoppers wearing Google Glass will see coupons for a new cereal flash before their eyes as they search for Cheerios, while those browsing the dairy section will have information about the health benefits of soy milk pop up automatically on their iPads. Moreover, it can make shopping available anywhere: with no more than a snapshot of the item we are interested in, such as a fancy bike we saw on the street, a bottle of splendid red wine we just tasted in a friends' place, or an interesting book we just browsed in the office's library, product recognition technology is able to identify the product and reveal to us a wealth of information about it. Information including the brand, material, color, price, user ratings and places to buy it can provide us a good understanding of the product, and allow us to buy it instantly on our phones.

Geolocalization A picture is worth a thousands words. Rich visual cues in picture such as landmarks, region specific items, languages, and plant life sometimes provides telltale evidence of its location. By matching a photo against a large collection of geotagged images, image retrieval system is able to discover the location of a photo without any help of GPS device. Accordingly, in the scenarios where GPS device can't function properly or simply not available, image retrieval technology could be an alternative solution to geolocalization.

Intelligent Photo Album Assistant Does this sound like you: to capture an important moment, you take a handful of pictures of the same thing from slightly different angles or in slightly different lighting just to be sure you get it right. Most of those repeat photos land in a dusty corner of your hard drive, never to be seen again. It may even become a source of pain later on when browsing photo albums full of nearly identical images. With the help of an image retrieval system, repeated or visually overlapping photos can be automatically grouped into photosets. Combining such a system with modern image processing techniques, similar images can be either combined into vivid animations or stitched into gorgeous panoramas.

Automatic Annotation An image retrieval system can also facilitate personal photo album organization if used to power content search. Scrolling

through a photo collection containing thousands of images to find a particular one that you want to show your friend can be extremely tedious, while an image retrieval system makes keyword queries such as "the photo of me taken at the movie premiere of 'Alice Through the Looking Glass'" possible.

Interestingness Discovery There may be no better way to capture a single moment and strengthen our visual memory than taking a photo. With the increasing popularity of photo sharing, social networking and food sharing apps such as Instagram, Facebook and TripAdvisor, along with the widespread availability of high quality cameras in mobile phones, the number of photos we take recording our daily life is booming. Frequently occurring objects or scenery across the photo collections of a large group of people can be taken as a strong indication of public interest: what is the most popular landmark in Paris, and what are the top dishes in the restaurant behind your hotel? Taking into account the huge scale of photo collections to be considered, and the need to discover specific objects that occur across multiple images, image retrieval serves as one of the most effective and efficient techniques to mine the frequently occurring objects in a large image dataset.

3D reconstruction The high scalability and fast speed of image retrieval systems make it possible to automatically group the colossal amount of tourist photos available in photo sharing platforms such as FLICKR at the level of individual buildings. In combination with advanced 3D reconstruction techniques city scale 3D modeling, such as "Building Rome in a day" Agarwal *et al.* [2011], becomes feasible.

1.2 Challenges

There are many challenges in large scale object retrieval, which we will examine in detail in this section.

Appearance Variation The appearance of the same object in different images could vary tremendously. Taking photographs of Eiffel Tower as an example,

Figure 1.1: Object of interest could be blurred when taken on a foggy or snowy day.

- the images could be blurred because they were taken on a foggy or snowy day as in Figure 1.1,
- the images could differ greatly because they were taken from very different viewpoints as shown in Figure 1.2,
- the images could show only a part of it because it is occluded by other objects, or they were cropped manually as shown in Figure 1.3,
- the images could vary widely in quality due to resolution settings or JPEG artifacts if some were taken using professional cameras while others shot on mobile phones as shown in Figure 1.4.
- the lighting in the images could be vastly different if some were captured in the morning and others at night as shown in Figure 1.5.

Dataset Diversity A dataset may contain a wide variety of visually different kinds of object. Some objects have little texture, while others have fine details which are lost in poor quality images. Some objects usually occupy a large area of an image, while others may be inherently small and appear in cluttered backgrounds. Some objects are best described by their shapes while others are characterized by their texture or color. The intrinsic diversity in object appearance poses a great challenge in the selection and combination of visual cues, making the design or learning of an effective visual representation extremely hard. Furthermore the object occurrence frequency, i.e. the amount of relevant

Figure 1.2: Object of interest could look very different when taken under different viewpoint.

Figure 1.3: Object of interest could be occluded or cropped.

items, differs tremendously across the dataset. There may be many thousands of images of a famous landmark such as the Eiffel Tower but only a handful of a local merchant. The highly unbalanced distribution of data poses a great challenge to algorithms with fixed parameters.

Unstructured Database What makes image retrieval radically different from generic object recognition is the unstructured dataset. Typically an image retrieval system can be used on an arbitrary collection of images without requiring any further knowledge about the relationships between images, such as which ones contain the same object. This setting makes image retrieval highly flexible and widely applicable, but also makes exploring a wide range of powerful supervised learning algorithms extremely difficult.

Figure 1.4: Object of interest could be of very low resolution.

Figure 1.5: Object of interest could look very different when taken at a different time of a day.

Scalability and Speed To facilitate numerous potential applications, it is critical that an image retrieval system can scale up to millions or even billions of images, and process queries in near realtime. Processing such a massive amount of data in a short time requires a highly compact visual representation of the image as well as efficient indexing and searching.

1.3 Contributions

Visual content based retrieval is a long studied problem. Research into it can even be traced back as far as the early eighties. Even though astonishing progress has been made in recent years in terms of scalability and precision,

accuracy on common retrieval benchmarks still shows room for significant improvements. Accordingly, the objective of this thesis is to improve the accuracy of retrieval systems while keeping search speed near real-time and memory consumption manageable.

At the heart of many image retrieval systems lies the pairwise image similarity measure. Over the years through many experiments we have found out that many available similarity measures are only reliable in a localized region of the descriptor space. They should not be but are commonly used to measure visual similarities globally. In the following, we summarize our contributions to image similarity measurement. We introduce locality to available measures, both in different stages of image retrieval systems and different descriptor levels of image representations. Further, we show that introducing locality makes these measures more accurate, and demonstrate that improving locality could be very beneficial for retrieval accuracy.

As a first contribution, we systematically analyze how to model feature-to-feature similarity for high-dimensional local features under a probabilistic framework. We find the de facto standard Euclidean distance is discriminating locally but not globally. A location sensitive distance measurement is therefore necessary to fully exploit the discriminating power of Euclidean distance. Hence we derive a query adaptive distance which is appropriate for global similarity evaluation. We show that the expected distance to the non-corresponding features can be used effectively to adapt the original Euclidean distance. By introducing a small set of random features as negative examples, we demonstrate that this query adaptive measure can be computed very efficiently. Furthermore, we propose a function to score the individual contributions into an image to image similarity within the probabilistic framework. Experimental results show that our method consistently gives a significant boost to retrieval accuracy. Moreover, our result compares favorably to the state-of-the-art.

As a second contribution, we propose a simple and general linear function to model the visual similarity between bag of visual words (BoVW) features, whose parameters can be efficiently trained to maximize retrieval accuracy. To do so, we first analyze the commonly used hand crafted methods for measuring pairwise similarity between BoVW histograms. We then propose a simple additive approximation of the available methods that leads to our linear model for similarity. We also analyze how this model can implicitly integrate various statistical properties of visual data. We then show how the learning problem can be seen as learning to rank from pairs of images. For this, we optimize

a loss function inspired by rank-SVM so as to maximize an approximation of the mean average precision (mAP) of the system. We also show how our model integrates into an efficient inverse file structure and thus how to use it in large-scale retrieval scenarios. In our experiments, we show that our method consistently and significantly outperforms existing similarity measures on four standard image retrieval benchmarks.

The third contribution is a method for improving image retrieval precision and recall by introducing k-reciprocal nearest neighbors as blind relevance feedback to rerank images. Due to the curse of dimensionality and the highly inhomogeneous nature of image space, most existing measures in modeling pairwise image similarity can only work locally. Comparing pairwise image similarity in one region of image space to pairwise similarity in another is generally problematic. However, to obtain the size of the neighborhood in which a given similarity measure works reliably is very difficult. In experiments, we observed that k-reciprocal nearest neighbors is a surprisingly reliable measurement. Across a wide range of k, the k-reciprocal nearest neighbor of query remains almost the same, and most of them are in fact true positive matches to the query. Based on this simple observation, we propose to expand our knowledge of the query according to its k-reciprocal neighbors, and rerank other images in the database according to their similarity to this relevant set of images. We evaluate our approach on common object retrieval benchmarks and demonstrate a significant improvement over a standard bag-of-words retrieval. Since our method obtained relevance feedback without relying on lower-level information such as the geometric arrangement of features or the geometry of the descriptor space, it can be used in a wide variety of settings. We also achieve very competitive results at reasonable memory overhead and very little additional computational complexity at query time.

As a fourth contribution, we introduce a simple yet powerful family of kernels, quantized kernels (QK), which model non-linearities and heterogeneities in the data efficiently and effectively. In essence, we build on the fact that vector quantizers project data into a finite set of N elements, the index space, and on the simple observation that kernels on finite sets are fully specified by the NN Gram matrix of these elements (the kernel matrix), which we propose to learn directly. Thus, QKs are piecewise constant locally but arbitrary globally, making them very flexible. Since the learnt kernel matrices are positive semi-definite, we directly obtain the corresponding explicit feature mappings and exploit their potential low rank. As a result, we obtain state-of-the-art matching

performance on a standard benchmark dataset using only a few bits to represent each feature dimension. QKs also have explicit non-linear, low-dimensional feature mappings that grant access to Euclidean geometry for uncompressed features. Although we do not show direct experimental results of applying QK to Image Retrieval, incorporating QK in the geometric verification step in the post processing stage of a retrieval system should be straightforward.

2

Overview of Retrieval System

The objective of an image retrieval system is, when given a query image, to generate a ranked list of database images that are sorted according their similarity to the given query. Essentially, this is a ranking problem. The task is to learn a unary score function for each query, which takes a database image as input and outputs its similarity to the query. However, pre-learning a score function for every query is not possible due to the overwhelming number of potential queries. The standard approach to account for this difficulty is to use a universal, query independent, bivariate score function which models visual similarity between two arbitrary images. In fact, the design of this pairwise similarity function lies at the heart of every image retrieval system. Its accuracy and efficiency impacts the performance of the retrieval system enormously.

To quantify the similarity between images, the question of how to describe an image arises naturally. As the appearance of the same object in different images varies vastly due to the large changes in viewpoints, illuminations, background, and occlusion, the robustness of the image representation to these changes is critical to the success of the retrieval system. Throughout literature, the most popular choice of image representation is a bag of local features. As only the similarity between matched local features is counted but not the dissimilar pairs, it is by design highly robust to background clustering and occlusion. Moreover, as local features are extracted in a multi-scale and multi-location manner, the bag-of-features representation is robust to geometric variations as well. An alternative choice of image representation is global features, which aims at modeling the image in its entirety and is able to capture larger context in the image. Although global feature are better at modeling highly level semantic information, they are in typically suffered from the problem of occlusion or

viewpoint differences. In Section 2.1 and 2.2, we will give a detailed review both the local feature based representation and global visual descriptors.

A dual aspect of image representation is the measure of visual similarity, which is responsible for deciding whether a pair of image features matches or not. Although many simple and efficient dis-/similarity measures are available, such as Euclidean Distance and Cosine distance, they can be hardly adapted to the data of interest, and therefore lead to limited performance of the retrieval system. In Chapter 4, 5, 6, 7, we will discuss why we have to go beyond the existing simple measures, how to learn better similarity measures, and demonstrate the importance of similarity measure in retrieval system by running our algorithms on benchmark datasets.

2.1 Image Representation Based Local Features

As a matter of fact, the best existing retrieval systems rely on local features. In this section, we will discuss retrieval systems in which image representation is built on local descriptors. We begin with a introduction of local features in the context of image retrieval: what is it, what its advantages are, and how to describe it. Then we will discuss how to encode local features efficiently in retrieval systems. Finally, we will discuss global image representations which goes beyond local features.

2.1.1 Local Features

Local features are one of the major topics in computer vision. In this section, we will review local features only in the context of image retrieval. To be precise, below we refer a local feature as an image region centered at a point.

Why local features? As mentioned previously, the most popular image representations in image retrieval are built on local features. Its great success can be attributed to several factors. Firstly, local features are by nature robust to background clustering, occlusion, and geometric transformations. As pointed out by Tuytelaars and Mikolajczyk [2008], despite the great difference in appearance and large regions of dissimilarity, two images of the same object still share a large number of similar local regions. Moreover, local features are

important and essential for human vision system to recognize objects: removing local patches corresponding to corners and junctions from images hampers human recognition, while removing most of the straight edges does not. Additionally, modeling objects using local features leads to a compact representation, as common local parts are shared across different objects: eyes of cats and dogs look similar, as do wheels of buses and trucks.

Interest point detectors. The most common way to extract regions of local features is an interest point detector. It was initially developed to find point correspondences in wide baseline stereo matching. To cope with the great differences in images of the same scene under different viewing conditions, interest point detectors are designed to extract regions that are covariant to geometric transformations. However, detecting covariant regions is very difficult for the applicable geometric transformation is not known. One way to handle this difficulty is to assume that a certain image structure exists in the reference image, such as an isotropic blob with unit scale. Hence, if a deformed blob is detected, then the covariant region is found, and thereby the geometric transformation as well. According to the image structure they work on and the the geometric transformations they deal with, interest point detectors can be categorized in groups. The first group of detectors responds to blob structure. For instance, SIFT proposed by Lowe [2004] detects scale and rotation invariant regions by extracting local extrema of the Difference of Gaussians (DoG) pyramid. To speed up, SURF proposed by Bay *et al.* [2006] utilizes integral image algorithms. To generalize to affine transformations, the Hessian-Affine detector developed by Mikolajczyk and Schmid [2004] detects affine deformed isotropic structure by estimating the anisotropic shape using a second moment matrix. Alternatively, the second group of detectors responds to other image structures like corners, edges, and stable regions. Representative examples in this group include MSER Matas *et al.* [2002] and Harris-Laplacian Mikolajczyk and Schmid [2004]. To be specific, MSER constructs a sequence of extremal regions by incrementally thresholding the image, and selects the locally stable regions as interest regions. Harris-Laplacian detects corners using Harris cornerness measure and selects scales at the local extremal of Laplacian scale space.

In the context of wide baseline matching, a thorough evaluation is given by Mikolajczyk *et al.* [2005], in which MSER is reported to give the best performance. However, experiments Gammeter [2013] in image retrieval show that Hessian-

Affine and Harris-Laplacian detector works the best among interest point detectors. As a result, most literature in retrieval adopts the Hessian-Affine detector. Moreover, multiple works Perdoch *et al.* [2009]; Mikulík *et al.* [2010] reported the great impact of small twists in the implementation of the interest point detectors: the performance of retrieval system improves greatly if applying the Hessian Affine detector to an upsampled image with a factor of 2 and enlarging the detected scale by 3 times.

Dense sampling Another common way to extract regions for local features is to extract regions on a regular spatial grid at various scales.

Dense sampling or interest point detector? Thanks to the simplicity of the sampling procedure, dense sampling has various advantages. It generates a large number of local features across dense locations and scales at a negligible cost. It provides a uniform and dense coverage of the image. It is highly adaptive, as its parameters can be easily tuned according to the needs of applications. For instance, we can easily control the number of generated regions, the density of coverage, and the degree of scale variance. However, it suffers from several drawbacks. Regardless of the local content of the image, it generates the same number of regions throughout the image. It gives no guarantee about the repeatability nor discriminability of the extracted regions. As a result, to measure similarity between images, we cannot rely on computing it by counting the number of matched local feature pairs, but have to resort to some overall statistics. On the other hand, interest point detectors extract regions that are covariant to geometric transformations. Their detected regions are typically sparse and repeatable. Thus local features of these regions can be matched individually, and image-wise similarity can be computed by accumulating feature-wise similarities. Accordingly, the problem of occlusion and background clustering in image matching can be easily addressed, by considering only matched feature pairs and ignoring the unmatched ones. Furthermore, the unknown geometric transformation between images can be estimated by inspecting the spatial consistency between matched feature pairs. Despite the great advantages of interest point detectors, they also suffer from several limitations. First and foremost, they are typically hand engineered to respond to a single type of low level structure only, and hence their detected regions are limited to that specific type of image structure and lack semantic interpretation. Second, the design of interest point detectors typically involves many complex

hand-crafted steps, making it extremely difficult to optimize their parameters for a particular application like image retrieval. To give an example, even the number of detected regions can scarcely be adjusted!

Patch Descriptors As raw pixel values of local regions vary dramatically to changes in imaging conditions, early research efforts were devoted to engineering more invariants and discriminating local patch descriptors for robust local feature matching. A landmark paper in the area was the work of Lowe [2004], which firstly partitions a image patch into 4×4 regular spatial grids, then describes each of this grids using a histogram of angle-quantized gradients, and finally concatenates the descriptors of those grids. Ever since its advent, SIFT descriptor has quickly became de facto standard local region descriptor, and led to ground breaking progress in a vast range of vision applications, such as object recognition, 3D reconstruction, image segmentation and object tracking, just to name a few. Inspired by the great success of SIFT descriptor, a handful of patch descriptors were developed to further improve on its speed and accuracy. Famous examples includes SURF Bay *et al.* [2006], GLOH Mikolajczyk and Schmid [2005], DAISY Tola *et al.* [2010], BRIFE Calonder *et al.* [2010] and LBP Guo *et al.* [2010]. As pointed out by Boix *et al.* [2013], despite the very different appearance of these descriptors, most of them share the same modular framework, consisting of four main stages: smoothing, filtering, pooling, and normalization. First, a smoothing step is introduced to compensate for the differing amount of lens blur due to different imaging scales. Second, a filtering step is exploited to capture image patterns. The most popular choice is a kind of differentiated filter. Examples includes the gradient, used in SIFT Lowe [2004], first and second order derivatives in SURF Bay *et al.* [2006], and randomly selected point difference in BRIEF Calonder *et al.* [2010]. Besides their simplicity, their robustness to the linear illuminations changes is likely to explain their popularity.

Third, the pooling step accounts for the geometric offset arising from local region detection errors. By aggregating the filter responses on pixels inside a spatial neighborhood together, spatial pooling results in a descriptor invariant to small location offsets. Fourth and finally, a normalization step is included to remove the descriptor dependency on image contrast.

Later on, modern learning techniques were widely employed in descriptor design to further improve parameter selection. For instance, DAISY Tola *et al.*

[2010] utilizes Powell minimization to learn the parameters of daisy-like spatial pooling regions, while Simonyan *et al.* [2014] proposed a convex objective to further optimize the parameter selection. A more dedicated review on learning based descriptors is given in Chapter 7.

2.1.2 Bag of features representation

To profit from the nice properties of local features, such as robustness to background clustering and small viewpoint changes, many image representations are built upon local features. Bag of local features (BoF) is undoubtedly one of the most commonly used representations among them. Applying the BoF representation to image retrieval, the fundamental image matching problem can be reformulated as matching one set of features to the other. To be robust to background clustering and local detector errors, the matching of BoF is always done on the basis of individual feature matching, *i.e.* finding the best match to each feature in one set by scanning through all features in the other set. However, matching features individually leads to many problems. Firstly, the straightforward one-to-one matching algorithm is of quadratic complexity and therefore not suitable for real time large scale image retrieval. Moreover, different kinds of local features depict different type of image structure, and may carry varying amount of information. Accordingly, their discriminability should also be taken into account when computing the image similarity. Nevertheless, multiple local features may belong to the same macro image structure and tend to occur together frequently. Therefore, counting their contribution separately could result in overestimating the actual similarity. In fact, the three problems given above, *i.e.*, time complexity, feature discrimination and feature correlations, are the main challenges in applying BoF model in image retrieval. In the following, we will discuss in detail the methods that were developed to addressed these problems.

Bag-of-Words Model

Without doubt, bag of visual words model Sivic and Zisserman [2003] is one of the most prevalent and influential image representations in computer vision literature. Originating from text retrieval, the spirit of the bag of words model(BoW) is to only count the number of occurrences of a word, disregarding the word order. The ignorance of geometric locations of word occurrences

not only makes the BoW representation simple to implement, but also robust to document cropping and permutation. Moreover, the fixed length property of the BoW representation is highly desirable as many machine learning algorithms require equal sized feature vectors. Thanks to these many advantages, BoW model is commonly used in natural language processing and information retrieval today. However, this model does not work in image retrieval, as images are typically two dimensional continuous signals without well defined semantic breaks. To address this difficulty, a seminal work from Sivic and Zisserman [2003] considered an image as a collection of image segments (local features), and proposed to encode each of these segments to a particular word in a pre-learnt vocabulary. In their work, vocabulary is learnt through k-means clustering, and each cluster center is regarded as a visual word. An image segment is assigned to the visual word that is closest to it. Despite the simplicity of BoVW model, it quickly swept through many domains of computer vision after its advent, and demonstrated its superiority to many counterparts. An important follow-up work adopting BoVW in image retrieval is Philbin *et al.* [2007], in which the use of a huge vocabulary(with 1 million to 16 million visual words) is shown to boost the performance significantly compared to the original small size dictionary (10k visual words) used in Sivic and Zisserman [2003]. To tackle the computational bottleneck of ordinary k-means clustering in building a such large vocabulary, Nistér and Stewénius [2006] proposed to build the dictionary in a hierarchical manner. Furthermore, Philbin *et al.* [2007] utilized the randomized k-d forest structure to facilitate speedy nearest neighbor search, and built a large and flat vocabulary, yielding much better results than hierarchical ones Nistér and Stewénius [2006]. It can therefore be concluded that the usage of a huge vocabulary can not only enhance the retrieval accuracy markedly, but also improve the retrieval speed substantially. This is because the large dictionary leads to extremely sparse BoVW vectors, producing short lists of documents associated with each visual word while indexing with an inverted file system. Since each image query only contains a few thousand words, only a few thousand inverted lists need to be touched, resulting in a short processing time. To compute the visual similarity between images represented by BoVW vectors, a tf-idf scheme borrowed from text retrieval is commonly used. More precisely, the final visual similarities between images are computed using the dot product of the word occurrence in the both documents (term frequency) weighted by the inverse of the occurrence frequency of each visual word in the whole set of documents (inverted document frequency).

It is not hard to see that BoVW is in fact a highly simplified version of BoF model. Its simplicity makes it remarkably space and time efficient – with an ordinary desktop computer, an image retrieval system based on BoVW model enables searching through a database of tens of millions images in near real time.

Improving feature discriminability

Alleviating quantization artifacts Due to the hard partitioning of the local descriptor space, the BoVW model is inevitably vulnerable to quantization artifacts, *i.e.* features located near to quantization boundaries, in spite of their proximity, may be assigned to different visual words. This problem becomes more serious when the vocabulary is learnt from an independent image dataset as reported in Philbin *et al.* [2008]. To diminish quantization artifacts, Philbin *et al.* [2008] proposed to assign each local feature softly to multiple visual words, and use a Gaussian kernel to weight its proximity to nearby visual words. This method was confirmed to be very effective by many later research works but comes at a cost of multiple-times increase in CPU and memory usage. Another milestone work in improving the discriminability of local features against hard assignment is the Hamming embedding scheme introduced by Jégou *et al.* [2008]. In their work, each local feature is encoded not only with its nearest visual word, but with a 64 bit binary vector as well. The binary vector is obtained by firstly applying a random orthogonal projection to the local feature descriptors and then thresholding them according to the median of transformed features associated with the corresponding visual word. Essentially, the binary signature provides a more accurate spatial encoding of local features in the descriptor space, enabling more precise feature matching and leading to a much higher retrieval quality. Similarly, the product quantization scheme Jégou *et al.* [2011b] also provides a compact and accurate encoding of local features, and can be used as a replacement for Hamming embedding to improve retrieval accuracy. In parallel, Jégou and Chum [2012] demonstrated that merging BoVW vectors of independently trained dictionaries can also alleviate the quantization effect. This is done by applying PCA on top of the concatenation of these BoVW vectors to remove the correlation and preserve the additional information from different quantizations. Nevertheless, Philbin *et al.* [2010] proposed to address the quantization problem at the source. To do so, they constructed both a linear and nonlinear projection from the raw feature

space to a new Euclidean space, in which the matching features are more likely to be assigned to the same visual word.

Learning semantically plausible similarities In spite of the convenience of using Euclidean distance in computing feature similarities, such as compatibility with k-means clustering, highly efficient approximate nearest neighbor search algorithms like LSH Datar *et al.* [2004] and product quantization Jégou *et al.* [2011b], Euclidean distance is generally believed not to be optimal for matching histogram based descriptors like SIFT. This is best illustrated by the work Arandjelovic and Zisserman [2012]. Their work achieved a dramatic boost in retrieval performance by simply replacing Euclidean distance by a Hellinger kernel. Thinking along the same lines, taking into account of the high inhomogeneity of descriptor space, we Qin *et al.* [2013] argued that using the same metric across the whole feature space was not appropriate, and proposed a probability model to adapt Euclidean distance locally, which was shown to improve the retrieval quality significantly. Observing the fact that most retrieval systems consist of a vector quantization step due to time and space constraints, several works attempted to integrate a more semantically plausible similarity measure into soft assignment rules. For instance, Mikulík *et al.* [2010] proposed to partition the descriptor space into very fine Voronoi cells, and learnt the semantic correlations between these cells based on the matching feature tracks collected from a large auxiliary image dataset using a wide-baseline method. Similarly, Gao *et al.* [2012] proposed to augment the visual vocabulary with the geometric relationship learnt between visual words under various simulation viewpoints, and demonstrated the robustness of their method against serious affine distortions.

Aggregating feature similarity into image similarity

One assumption made by the BoVW model is that features of different visual words are independent. However, as pointed out by Chum and Matas [2010], this assumption does not always hold as spatially nearby features maybe belong to the same image structure. Treating these co-occurring features independently may overcount their contributions and eventually harm retrieval quality. To tackle this problem, Chum and Matas [2010] proposed to use Min-Hash to discover sets of features co-occurring more often than random, and down weighted their contribution in computing pairwise image similarity. On the

other hand, Quack *et al.* [2006] argued that certain frequently occurring spatial configurations of local features may act as stable and discriminant features that well represent an object, and proposed to utilize frequent item set mining techniques to discover such distinctive patterns and boost recognition performance. Furthermore, to reduce the effect of redundantly representing the same part of an object by multiple frequent patterns, Fernando and Tuytelaars [2013] propose to select the smallest set of patterns that can best describe the query object according to the minimal description principle.

Two problems occur when using the tf-idf scheme to compute image similarity. The first problem is term frequency weighting due to burstiness. As pointed out by Jégou *et al.* [2009], burstiness is a phenomenon that a visual word may appear more frequently than a statistically independent model predicts, *i.e.* if a visual word appears in an image once, it is more likely to appear again. Accordingly, counting similarity by directly multiplying term frequencies regardless of burstiness produces inferior results. In Jégou *et al.* [2009], multiple weighting functions are presented to deal with the burstiness effect. Although it sounds intuitive and reasonable to down-weight visual words of high frequency, the way that idf assigns weights is too coarse and heuristic. More carefully engineered methods Zheng *et al.* [2013] thus proposed to incorporate more features to construct the weighting term including term frequency, document frequency, statistics from images and dictionary information. In spite of the effectiveness of these methods, their handcrafted nature makes it hard to optimize. Accordingly, in Chapter 5 we propose to address both problems in a unified model.

Since BoVW is purely appearance based, many works have attempted to extend it by exploiting more informative cues such as spatial configurations and context. To be specific, the first line of work in this group utilizes the spatial extent of local features to boost performance. For instance, Jégou *et al.* [2008] used characteristic scales and dominant orientations of local features to estimate geometric transformations, and penalized inconsistent matches. Perdoch *et al.* [2009] quantized geometric representations of local features, and used them to verify the spatial consistency of matched features subject to the gravity vector assumption. The second group makes use of the coordinates of local features. In Philbin *et al.* [2007], point correspondences are established by assigning features to visual words, and RANSAC was used to estimate the associated affine transformation between images. Image are only considered to be similar if the number of spatially consistent matches is above a certain

threshold. To improve efficiency and robustness, Zhang *et al.* [2011] and Shen *et al.* [2012] firstly discretized the parameter space of the translation/similarity transform, and then cast the votes of spatially compatible matches to the Hough voting space softly. In addition to being able to compute a more accurate visual similarity between images, methods in this group can also localize or register the object and thus make other applications possible like augmented reality. The final group employs context information. For example, Knopp *et al.* [2010] shows that detecting regions populated with confusing features and suppressing features coming from them significantly improves place recognition performance.

2.1.3 Aggregating local features into a global representation

Matching local features individually comes with many benefits such as being highly robust to viewpoint changes, image cropping, occlusion or background clutter, and makes it possible to establish the spatial correspondences of the query image to the database one, which is critical for applications like 3D reconstruction or Virtual Reality Augmentation. However, matching thousands of local features of the query image to billions of local features of database images is extremely difficult either in terms of memory footprint, computational complexity or speed. Although BoVW methods provide a highly efficient way to compare local features on a nearly individual basis by partitioning the descriptor space into very fine Voronoi cells, and matching features based on their co-ocurence in the same cell, their efficiency comes with a high cost and significant compromises. Firstly, to match local features approximately pairwise requires a huge vocabulary. Training a vocabulary with a size of millions of codewords is very difficult, since keeping the codewords of a huge vocabulary in computer memory is already a big problem, not to mention the difficulty to find the nearest codeword in a huge set for a given local feature within a tight time and computational complexity budget. Secondly and more importantly, to match local features individually in an efficient way, BoVW methods must store the encoding of each feature in memory. Although a highly compact encoding such as Hamming embedding or product quantization makes it possible to encode more information of a local feature to its assigned codeword, the very tight memory constraint greatly limit the space BoVW methods can explore to improve the feature encoding.

To avoid running into the trouble of training a massive scale vocabulary and not being limited by very tight memory budgets due to the requirement of encoding each local feature, another group of methods matches images based on the aggregated statistical distributions of local features rather than accumulating pairwise local feature similarities. A seminal work in this group is the Fisher kernel Perronnin and Dance [2007] Perronnin *et al.* [2010], which assumes all local features of an image are i.i.d. samples generated from a Gaussian Mixture Model, whose parameters are estimated on a set of training samples. It then uses the gradient of the sample's likelihood with respect to the parameters of this distribution, scaled by the inverse square root of the Fisher information matrix, to describe an image. This representation basically encodes the direction about how the learnt generative model should be modified to fit the observed image. As shown in Perronnin and Dance [2007], Fisher kernels actually model higher order statistics of local features to learn visual word centers, such as expected mean distances and the variances, while BoVW only encodes zero order information like counts. A simplification of fisher kernel, VLAD, is introduced by Jégou *et al.* [2010b]. They replace the modes of a Gaussian Mixture Model by a codebook of visual words learnt with k-means, and accumulate the differences between local features to their closest visual words to characterize the distribution of the local features of an image with respect to the learnt codebook. It is demonstrated by Jégou *et al.* [2010b] that VLAD is a highly compact representation which makes searching 10 millions of images in real time possible. Although retrieval systems build on VLAD could have fast speed and great scalability, their retrieval performance is still far behind the BoVW counterparts. To improve the retrieval accuracy of VLAD, many works has been proposed, either by introducing better normalization scheme Delhumeau *et al.* [2013]; Arandjelovic and Zisserman [2013], encoding weakly spatial information Tolias *et al.* [2014] or data whitening Jégou and Chum [2012].

2.2 Global Descriptors based Image Representation

A parallel line of work to local descriptor based image representation is global descriptors. An early representative work in this field is the GIST descriptor Oliva and Torralba [2001]. A set of perceptual dimensions, such as naturalness, openness, roughness, expansion, ruggedness are used to represent the dominant spatial structure of a scene. Although all these perceptual dimensions

are intuitive and informative elements in representing an image, how to model them and combine them are extremely hard. Hand engineering nature of GIST makes it impossible to adapt to the image retrieval dataset and lead to inferior performance Douze *et al.* [2009]. An alternative solution is to use a list of visual attributes to represent an image. For instance, Douze *et al.* [2011] uses a image descriptor which is constructed by a list of classification confidence scores.

Another way to characterize a high level visual content of an image is to use the activations invoked by an image within the top layers of a large convolutional neural network. Babenko *et al.* [2014] investigate the performance of neural nets on image retrieval while trained with an unrelated classification dataset like Image-Net, and particular object instance classification dataset. It is shown that neural codes perform competitively to other global descriptors. A further work from Babenko and Lempitsky [2015] shows that deep features and traditional shallow hand engineered features have quite different distribution of pairwise similarities, and thus they propose to aggregate both together. Their new compact global descriptor improves the state-of-the-art on common image retrieval benchmarks considerably. Recently, Arandjelović *et al.* [2015] introduce a new local descriptor aggregation layer which is pluggable into available CNN architecture and amenable to training via backpropagation. Since the parameters of the network could be learnt in an end-to-end manner, their proposed architecture shows significant improvement over non-learnt image representations and off-the-shelf CNN descriptors on two challenging place recognition benchmarks.

3

Dataset and Evaluation

3.1 Dataset

3.1.1 Oxford Buildings

Oxford5k consists of 5062 high-resolution images collected from Flickr by searching for particular Oxford landmarks. Ground truth for 11 different landmarks is generated by manual annotation. For each landmark, 5 images are selected as queries to cover different views of a building. For each query image, a bounding box is provided as the region of interest (ROI) to delimit the building of interest. Moreover, database images are divided into three groups for each query. Images are labeled as *Good* if the corresponding landmark is clearly visible, *OK* if more than 25% is visible, *Bad* if the object is not present, and *Junk* if less than 25% of the landmark is visible or the image is greatly occluded or distorted. In evaluation, *Good* and *OK* images are used as relevant images, while *Bad* images are used as non-relevant ones. *Junk* images are ignored by taken out from the retrieved list. On average, 10% of the database images are relevant for a query. This dataset is quite challenging for the following reasons: 1) relevant images of a given query could demonstrate substantial visual difference due to geometric or illumination variations. 2) the dataset is highly inhomogeneous for the high variation in the number of relevant images for difference queries.

Oxford105k contains Oxford5k and $100,000$ distractor images. The distractor images are downloaded from Flickr with the same resolution as the original Oxford5k images ranging between 768×1024 and 1024×1024.

3.1.2 Paris Buildings

Similar to Oxford5k, Paris dataset consists of 6412 high-resolution images collected from Flickr by searching for particular Paris landmarks. One common usage of Paris dataset is to test the generalizability of the learnt vocabulary by learning a vocabulary on Paris but evaluating on Oxford5k.

3.1.3 UKbench

UKbench consists of 10,200 images of 2,550 indoor objects. Each object is represented by 4 images taken from four different viewpoints. All the images have resolution of 640×480. Many image are taken with the object in the center and with a clean background. Each image is used as a query with the other 3 images of the same object as relevant ones. This dataset is especially collected to evaluate object recognition systems. However, no bounding box is provided through out the dataset.

3.1.4 Holidays

The Holidays dataset collected by Jégou *et al.* [2008] mainly consists of personal holiday photos. Images in this dataset have a very high resolution of 1600x1200. It contains 500 image groups and $1,491$ images in total. Each group represents a distinct scene or object. The first image of each group is the query image and the correct retrieval results are the other images of the group. For a large portion of the queries in this dataset there are only 1 or 2 relevant images. Region of interest is not provided since many images of this dataset depict a scene rather than an object. As a result, this dataset is more suitable for evaluating image retrieval rather than particular object retrieval.

3.2 Evaluation Protocol

The most popular measure to evaluate a retrieval system is mean average precision. Retrieval quality for a single query is measured in terms of precision-recall curve. Precision is defined as the proportion of true positives in the retrieved images. Recall is the ratio of retrieved true positives to the total number of positives for the query. In simple terms, precision measures the purity

of the retrieved list, while recall measures what fraction of the total number of known positives is discovered. The average precision is simply the area under the precision-recall curve. Mean average precision (mAP) is computed by taking the mean of the average precision of each query.

4

Query Adaptive Similarity

4.1 Introduction

We consider the problem of content-based image retrieval for applications such as object recognition or similar image retrieval. This problem has applications in web image retrieval, location recognition, mobile visual search, and tagging of photos.

Most of the recent state-of-the-art large scale image retrieval systems rely on local features, in particular the SIFT descriptor [Lowe, 2004] and its variants. Moreover, these descriptors are typically used jointly with a bag-of-words (BOW) approach, reducing considerably the computational burden and memory requirements in large scale scenarios.

The similarity between two images is usually expressed by aggregating the similarities between corresponding local features. However, to the best of our knowledge, few attempts have been made to systematically analyze how to model the employed similarity measures.

In this chapter we present a probabilistic view of the feature to feature similarity. We then derive a measure that is adaptive to the query feature. We show - both on simulated and real data - that the Euclidean distance density distribution is highly query dependent and that our model adapts the original distance accordingly. While it is difficult to know the distribution of true correspondences, it is actually quite easy to estimate the distribution of the distance of non-corresponding features. The expected distance to the non-corresponding features can be used to adapt the original distance and can be efficiently estimated by introducing a small set of random features as negative examples. Furthermore, we derive a global similarity function that scores the feature to

feature similarities. Based on simulated data, this function approximates the analytical result.

Moreover, in contrast to some existing methods, our method does not require any parameter tuning to achieve its best performance on different datasets. Despite its simplicity, experimental results on standard benchmarks show that our method improves the retrieval accuracy consistently and significantly and compares favorably to the state-of-the-art.

Furthermore, all recently presented post-processing steps can still be applied on top of our method and yield an additional performance gain.

The rest of this paper is organized as follows. Section 4.2 gives an overview of related research. Section 4.3 describes our method in more detail. The experiments for evaluating our approach are described in Section 4.4. Results in a large scale image retrieval system are presented in Section 4.5 and compared with the state-of-the-art.

4.2 Related Work

Most of the recent works addressing the image similarity problem in image retrieval can be roughly grouped into three categories.

Feature-feature similarity The first group mainly works on establishing local feature correspondence. The most famous work in this group is the bag-of-words (BOW) approach [Sivic and Zisserman, 2003]. Two features are considered to be similar if they are assigned to the same visual word. Despite the efficiency of the BOW model, the hard visual word assignment significantly reduces the discriminative power of the local features. In order to reduce quantization artifacts, Philbin *et al.* [2008] proposed to assign each feature to multiple visual words. In contrast, Jégou *et al.* [2008] rely on using smaller codebooks but in conjunction with short binary codes for each local feature, refining the feature matching within the same Voronoi cell. Additionally, product quantization [Jégou *et al.*, 2011b] was used to estimate the pairwise Euclidean distance between features, and the top k nearest neighbors of a query feature is considered as matches. Recently, several researchers have addressed the problem of the Euclidean distance not being the optimal similarity measure in most situations. For instance in Mikulík *et al.* [2010], a probabilistic relationship between visual words is learned from a large collection of corresponding feature tracks.

Alternatively, in Philbin *et al.* [2010], they learn a projection from the original feature space to a new space, such that Euclidean metric in this new space can appropriately model feature similarity.

Intra-image similarity The second group focuses on effectively weighting the similarity of a feature pair considering its relationship to other matched pairs.

Several authors exploit the property that the local features inside the same image are not independent. As a consequence, a direct accumulation of local feature similarities can lead to inferior performance. This problem was addressed in Chum and Matas [2010] by down-weighting the contribution of non-incidentally co-occurring features. In Jégou *et al.* [2009] this problem was approached by re-weighting features according to their burstiness measurement.

As the BOW approach discards spatial information, a scoring step can be introduced which exploits the property that the true matched feature pairs should follow a consistent spatial transformation. Philbin *et al.* [2007] proposed to use RANSAC to estimate the homography between images, and only count the contribution of feature pairs consistent with this model. Zhang *et al.* [2011] and Shen *et al.* [2012] propose to quantize the image transformation parameter space in a Hough voting manner, and let each matching feature pair vote for its correspondent parameter cells. A feature pair is considered valid if it supports the cell of maximum votes.

Inter-image similarity Finally, the third group addresses the problem of how to improve the retrieval performance by exploiting additional information contained in other images in the database, that depict the same object as the query image. Chum *et al.* [2007] rely on query expansion. That is, after retrieving a set of spatially verified database images, this new set is used to query the system again to increase recall. In Qin *et al.* [2011], a set of relevant images is constructed using k-reciprocal nearest neighbors, and the similarity score is evaluated on how similar a database image is to this set.

Our work belongs to the first group. By formulating the feature-feature matching problem in a probabilistic framework, we propose an adaptive similarity to each query feature, and a similarity function to approximate the quantitative result. Although the idea of adapting similarity by dissimilarity has already been exploited in Jégou *et al.* [2011a] and Omercevic *et al.* [2007], we propose

to measure dissimilarity by mean distance of the query to a set of random features, while theirs use k nearest neighbors (kNN). According to the fact that, in a realistic dataset, different objects may have different numbers of relevant images, it is actually quite hard for the kNN based method to find an generalized k for all queries. Moreover, as kNN is an order statistic, it could be sensitive to outliers and can't be used reliably as an estimator in realistic scenarios. In contrast, in our work, the set of random features could be considered as a clean set of negative examples, and the mean operator is actually quite robust as shown later.

Considering the large amount of data in a typical large scale image retrieval system, it is impractical to compute the pairwise distances between high-dimensional original feature vectors. However, several approaches exist to relieve that burden using efficient approximations such as Jégou *et al.* [2011b], Jégou *et al.* [2011c], Babenko and Lempitsky [2012] and Hwang *et al.* [2012]. For simplicity, we adopt the method proposed in Jégou *et al.* [2011b] to estimate the distance between features.

4.3 Our Approach

In this section, we present a theoretical framework for modeling the visual similarity between a pair of features, given a pairwise measurement. We then derive an analytical model for computing the accuracy of the similarity estimation in order to compare different similarity measures. Following the theoretical analysis, we continue the discussion on simulated data. Since the distribution of the Euclidean distance varies enormously from one query feature to another, we propose to normalize the distance locally to obtain similar degree of measurement across queries. Furthermore, using the adaptive measure, we quantitatively analyze the similarity function on the simulated data and propose a function to approximate the quantitative result. Finally, we discuss how to integrate our findings into a retrieval system.

4.3.1 A probabilistic view of similarity estimation

We are interested in modeling the visual similarity between features based on a pairwise measurement.

Let us denote as x_i the local feature vectors from a query image and as $\mathcal{Y} = \{y_1, ..., y_j, ..., y_n\}$ a set of local features from a collection of database images. Furthermore, let $m(x_i, y_j)$ denote a pairwise measurement between x_i and y_j. Finally $T(x_i)$ represents the set of features which are visually similar to x_i, and $F(x_i)$ as the set of features which are dissimilar to x_i. Instead of considering whether y_j is similar to x_i and how similar they look, we want to evaluate how likely y_j belongs to $T(x_i)$ given a measure m. This can be modeled as follows

$$f(x_i, y_j) = p(y_j \in T(x_i) \mid m(x_i, y_j)) \tag{4.1}$$

For simplicity, we denote $m_j = m(x_i, y_j)$, $T_i = T(x_i)$, and $F_i = F(x_i)$. As y_j either belongs to T_i or F_i, we have

$$p(y_j \in T_i \mid m_j) + p(y_j \in F_i \mid m_j) = 1 \tag{4.2}$$

Furthermore, according to the Bayes Theorem

$$p(y_j \in T_i \mid m_j) = \frac{p(m_j \mid y_j \in T_i) \times p(y_j \in T_i)}{p(m_j)} \tag{4.3}$$

and

$$p(y_j \in F_i \mid m_j) = \frac{p(m_j \mid y_j \in F_i) \times p(y_j \in F_i)}{p(m_j)} \tag{4.4}$$

Finally, by combining Equations 4.2, 4.3 and 4.4 we get

$$p(y_j \in T_i \mid m_j) = \left(1 + \frac{p(m_j \mid y_j \in F_i)}{p(m_j \mid y_j \in T_i)} \times \frac{p(y_j \in F_i)}{p(y_j \in T_i)}\right)^{-1} \tag{4.5}$$

For large datasets the quantity $p(y_j \in T_i)$ can be modeled by the occurrence frequency of x_i. Therefore, $p(y_j \in T_i)$ and $p(y_j \in F_i)$ only depend on the query feature x_i.

In contrast, $p(m_j \mid y_j \in T_i)$ and $p(m_j \mid y_j \in F_i)$ are the probability density functions of the distribution of m_j, for $\{y_j \mid y \in T_i\}$ and $\{y_j \mid y \in F_i\}$. We will show in Section 4.3.3, how to generate simulated data for estimating these distributions. In Section 4.3.5 we will further exploit these distributions in our framework.

4.3.2 Estimation accuracy

Since the pairwise measurement between features is the only observation for our model, it is essential to estimate its reliability. Intuitively, an optimal measurement should be able to perfectly separate the true correspondences from

the false ones. In other words, the better the measurement distinguishes the true correspondences from the false ones, the more accurately the feature similarity based on it can be estimated. Therefore, the measurement accuracy can be modeled as the expected pureness. Let $\mathcal{T}$ be a collection of all matched pairs of features, i.e,

$$\mathcal{T} = \{(x, y) \mid y \in T(x))\} \tag{4.6}$$

The probability that a pair of features is a true match given the measurement value z can be expressed as

$$p(\mathcal{T} \mid z) = p((x, y) \in \mathcal{T} \mid m(x, y) = z) \tag{4.7}$$

Furthermore, the probability of observing a measurement value z given a corresponding feature pair is

$$p(z \mid \mathcal{T}) = p(m(x, y) = z \mid (x, y) \in \mathcal{T}) \tag{4.8}$$

Then, the accuracy for the similarity estimation is

$$Acc(m) = \int_{-\infty}^{\infty} p(\mathcal{T} \mid z) \times p(z \mid \mathcal{T}) \mathrm{d}z \tag{4.9}$$

with m some pairwise measurement and $Acc(m)$ the accuracy of the model based on m. Since

$$p(\mathcal{T} \mid z) \leq 1 \text{ and } \int_{-\infty}^{\infty} p(z \mid \mathcal{T}) \mathrm{d}z = 1 \tag{4.10}$$

the accuracy of a measure m is

$$Acc(m) \leq 1 \tag{4.11}$$

and

$$Acc(m) = 1 \Leftrightarrow p(\mathcal{T} \mid z) = 1, \forall p(z \mid \mathcal{T}) > 0 \tag{4.12}$$

This measure allows to compare the accuracy of different distance measurements as will be shown in the next section.

4.3.3 Ground truth data generation

In order to model the property of $T(x_i)$, we simulate corresponding features using the following method: First, regions $r_{i,0}$ are detected on a random set

of images by the Hessian Affine detector [Mikolajczyk and Schmid, 2004]. Then, we apply numerous random affine warpings (using the affine model proposed by ASIFT [Yu and Morel, 2011]) to $r_{i,0}$, and generate a set of related regions. Finally, SIFT features are computed on all regions resulting in $\{x_{i,1}, x_{i,2}, ..., x_{i,n}\}$ as a subset of $T(x_{i,0})$.

The parameters for the simulated affine transformation are selected randomly and some random jitter is added to model the detection errors occurring in a practical setting. The non-corresponding features $F(x_i)$ are simply generated by selecting $500K$ random patches extracted from a different and unrelated dataset. In this way, we also generate a dataset $\mathcal{D}$ containing $100K$ matched pairs of features from different images, and $1M$ non-matched pairs. Figure 4.1 depicts two corresponding image patches randomly selected from the simulated data.

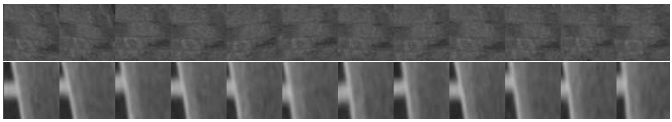


Figure 4.1: Corresponding image patches for two randomly selected points of the simulated data

4.3.4 Query adaptive distance

It has been observed that the Euclidean distance is not an appropriate measurement for similarity [Philbin *et al.*, 2010; Mikulík *et al.*, 2010; Jégou *et al.*, 2011a]. We argue that the Euclidean distance is a robust estimator when normalized locally.

As an example, Figure 4.2 depicts the distributions of the Euclidean distance of the corresponding and non corresponding features for the two different interest points shown in Figure 4.1. For each sample point x_i, we collected a set of 500 corresponding features $T(x_i)$ using the procedure from Section 4.3.3 and a set of $500K$ random non-corresponding features $F(x_i)$. It can be seen, that the Euclidean distance separates the matching from the non-matching features quite well in the local neighborhood of a given query feature x_i.

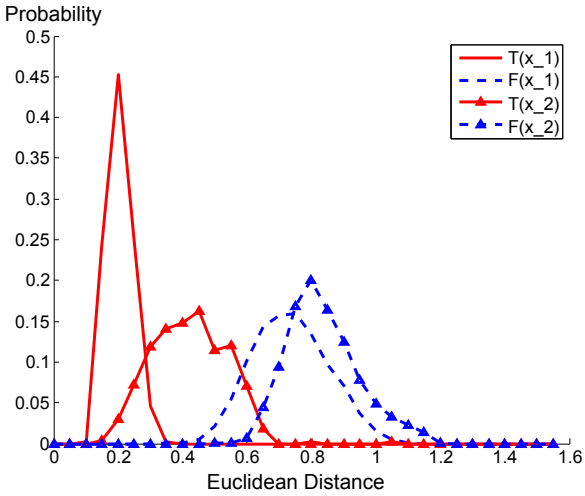


Figure 4.2: Distribution of the Euclidean distance for two points from the simulated data. The solid lines show the distribution for corresponding features $T(x_i)$, whereas the dotted line depict non-corresponding ones $F(x_i)$.

However, by averaging the distributions of $T(x_i)$ and $F(x_i)$ respectively for all queries x_i, the Euclidean distance loses its discriminative power. This explains, why the Euclidean distance has inferior performance in estimating visual similarity from a global point of view. A local adaptation is therefore necessary to recover the discriminability of the Euclidean Distance.

Another property can also be observed in Figure 4.2: if a feature has a large distance to its correspondences, it also has a large distance to the non-matching features. By exploiting this property, a normalization of the distance can be derived for each query feature

$$d_n(x_i, y_j) = d(x_i, y_j)/N_{d(x_i)} \tag{4.13}$$

where $d_n(\,\cdot\,,\,\cdot\,)$ represents the normalized distance, $d(\,\cdot\,,\,\cdot\,)$ represents the original Euclidean distance and $N_{d(x_i)}$ represents the expected distance of x_i to its non-matching features. It is intractable to estimate the distance distribution between all feature and their correspondences, but it is simple to estimate the expected distance to non-corresponding features. Since the non-corresponding

features are independent from the query, a set of randomly sampled, thus unrelated features can be used to represent the set of non-correspondent features to each query. Moreover, if we assume the distance distribution of the non-corresponding set to follow a normal distribution $\mathcal{N}(\mu, \sigma)$, then the estimation error of its mean based on a subset follows another normal distribution $\mathcal{N}(0, \sigma/N)$, with N the size of the subset. Therefore, $N_{d(x_i)}$ can be estimated sufficiently well and very efficiently from even a small set of random, i.e. non-corresponding features.

The probability that an unknown feature matches to the query one when observing their distance z can be modeled as,

$$\begin{aligned} p(\mathcal{T} \mid z) &= \frac{N_T \times p(z \mid \mathcal{T})}{N_T \times p(z \mid \mathcal{T}) + N_F \times p(z \mid \mathcal{F})} \\ &= \{1 + \frac{N_F}{N_T} \times \frac{p(z \mid \mathcal{F})}{p(z \mid \mathcal{T})}\}^{-1} \end{aligned} \tag{4.14}$$

with N_T and N_F the number of corresponding and non-corresponding pairs respectively. In practical settings, N_F is usually many orders of magnitude larger than N_T. Therefore, once $p(z \mid \mathcal{F})$ starts getting bigger than 0, $p(\mathcal{T} \mid z)$ rapidly decreases, and the corresponding features would be quickly get confused with the non-corresponding ones.

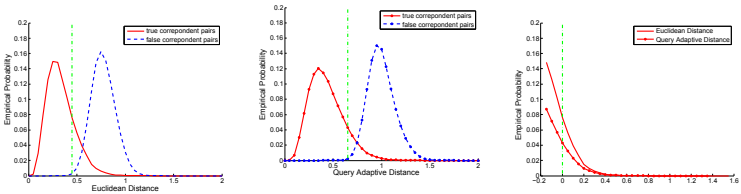
Figure 4.3 illustrates how the adaptive distance recovers more correct matches compared to the Euclidean distance.

Moreover, by assuming that $N_F/N_T \approx 1000$ the measurement accuracy following Equation 4.9 can be computed. For the Euclidean distance, the estimation accuracy is 0.7291, and for the adaptive distance, the accuracy is 0.7748. Our proposed distance thus significantly outperforms the Euclidean distance.

4.3.5 Similarity function

In this section, we show how to derive a globally appropriate feature similarity in a quantitative manner. After having established the distance distribution of the query adaptive distance in the previous section, the only unknown in Equation 4.5 remains $\frac{p(y_j \in F_i)}{p(y_j \in T_i)}$.

As discussed in Section 4.3.1, this quantity is inversely proportional to the occurrence frequency of x_i, and it is generally a very large term. Assuming



(a) Euclidean Distance (b) Query Adaptive Distance (c) Comparison

Figure 4.3: The comparison of our adaptive distance to the Euclidean distance on dataset $\mathcal{D}$. The solid lines are the distance distribution of the matched pairs, and the dotted lines are the distance distribution of non-matched pairs. The green dashed lines denotes where the probability of the non-matching distance exceed 0.1%, i.e, the non-matching feature is very likely to dominate our observation. A comparison of the right tails of both distributions is shown in (c).

$c = \frac{p(y_j \in F_i)}{p(y_j \in T_i)}$ being between 10 and 100000, the full similarity function can be estimated and is depicted in Figure 4.4.

The resulting curves follow an inverse sigmoid form such that the similarity is 1 for $d_n \to 0$ and 0 if $d_n \to 1$. They all have roughly the same shape and differ approximately only by an offset. It is to be noted, that they show a very sharp transition making it very difficult to correctly estimate the transition point and thus to achieve a good separation between true and false matches.

In order to reduce the estimation error due to such sharp transitions, a smoother curve would be desirable. Since the distance distributions are all long-tailed, we have fitted different kinds of exponential functions to those curves. However, we observe similar results. For the reason of simplicity, we choose to approximate the similarity function as

$$f(x_i, y_j) = \exp(-\alpha \times d_n(x_i, y_j)^4) \tag{4.15}$$

As can be seen in Figure 4.4, this curve is flatter and covers approximately the full range of possible values for c.

In Equation 4.15, α can be used to tune the shape of the final function and roughly steers the slope of our function, we achieved best results with $\alpha = 9$ and keep this value throughout all experiments.

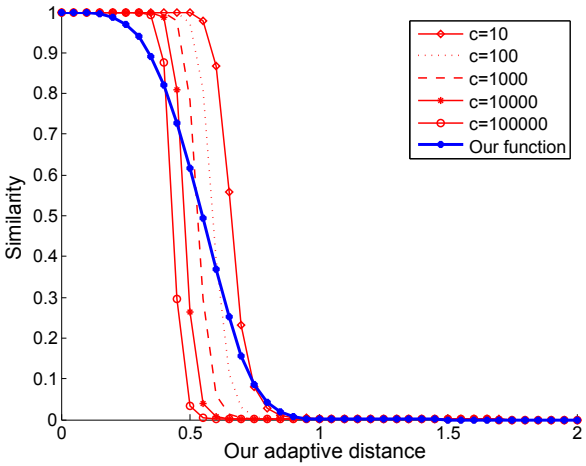


Figure 4.4: Feature similarity evaluated on dataset $\mathcal{D}$. Red lines are the visual similarity for different c evaluated on the simulated data. The blue line is our final similarity function with $\alpha = 9$.

In the next section, the robustness of this function in real image retrieval system will be evaluated.

4.3.6 Overall method

In this section we will integrate the query adaptive distance measurement and the similarity function presented before into an image retrieval system.

Let the visual similarity between the query image $q = \{x_1, ..., x_m\}$ and a database image $d = \{y_1, ..., y_n\}$ be

$$sim(q, d) = \sum_{i=1}^{m} \sum_{j=1}^{n} f(x_i, y_j) \tag{4.16}$$

with $f(x_i, y_j)$ the pairwise feature similarity as in Equation 4.15. As mentioned before, $d_n(x_i, y_j)$ and $N_{d(x_i)}$ are estimated using the random set of features.

For retrieval, we use a standard bag-of-words inverted file. However, in order to have an estimation of the pairwise distance $d(x_i, y_j)$ between query and database features, we add a product quantization scheme as in Jégou *et al.* [2011b] and select the same parameters as the original author. The feature space is firstly partitioned into $N_c = 20'000$ Voronoi cells according to a coarse quantization codebook $\mathcal{K}_c$. All features located in the same Voronoi cell are grouped into the same inverted list. Each feature is further quantized with respect to its coarse quantization centroid. That is, the residual between the feature and its closest centroid is equally split into $m = 8$ parts and each part is separately quantized according to a product quantization codebook $\mathcal{K}_p$ with $N_p = 256$ centroids. Then, each feature is encoded using its related image identifier and a set of quantization codes, and is stored in its corresponding inverted list.

We select random features from Flickr and add 100 of them to each inverted list. For performance reasons, we make sure that the random features are added to the inverted list before adding the database vectors.

At query time, all inverted lists whose related coarse quantization centers are in the k nearest neighborhood of the query vector are scanned.

With our indexing scheme, the distances to non-matching features are always computed first, with their mean value being directly $N_{d(x_i)}$. Then, the query adaptive distance $d_n(x_i, y_j)$ to each database vector can directly be computed as in Equation 4.13. In order to reduce unnecessary computation even more, a threshold β is used to quickly drop features whose Euclidean distance is larger than $\beta \times N_{d(x_i)}$. This parameter has little influence on the retrieval performance, but reduces the computational load significantly. Its influence is evaluated in Section 4.4.

As pointed out by Jégou *et al.* [2009], local features of an image tend to occur in bursts. In order to avoid multiple counting of statistically correlated features, we incorporate both "intra burstiness" and "inter burstiness" normalization [Jégou *et al.*, 2009] to re-weight the contributions of every pair of features. The similarity function thus changes to

$$sim(q, d) = \sum_{i=1}^{m} \sum_{j=1}^{n} w(x_i, y_j) f(x_i, y_j) \tag{4.17}$$

with $w(x_i, y_j)$ the burstiness weighting.

4.4 Experiments

In this part, we first introduce the evaluation protocol. Then we give some implementation details of our algorithm. Furthermore, we discuss the influence of each parameter and experimentally select the best ones. Finally, we evaluate each part of our method separately.

4.4.1 Implementation details

Preprocessing For all experiments, all images are resized such that their maximum resolution is 1024×768. In each image, interest points are detected using the Hessian Affine detector and a SIFT descriptor is computed around each point. As in Arandjelovic and Zisserman [2012] a square root scaling is applied to each SIFT vector, yielding a significantly better retrieval performance when using the Euclidean metric.

Codebook training The vocabularies were trained on an independent dataset of images randomly downloaded from Flickr in order to prevent overfitting to the datasets.

Random feature dataset preparation Random images from Flickr (however different from the codebook training dataset) are used to generate the random feature dataset.

4.4.2 Parameter selection

In this section, we evaluate the retrieval performance of our approach on the Oxford5K dataset for different settings of parameters. There are two parameters in our method: the number of random features in each inverted list, and the cut-off threshold β for filtering out features whose contribution is negligible.

The influence of the number of the random features Table 4.1 shows the retrieval performance by varying the number of random features for each inverted list. The performance remains almost constant for a very large range of number of random features. This supports the assumption, that the mean distance of a query feature to the dissimilar features can be robustly estimated

Length	50	100	500	1000	10000
mAP	0.739	0.739	0.739	0.739	0.738

Table 4.1: Influence of the size of the random feature set for each inverted list on Oxford5k

β	0.80	0.85	0.9	0.95
similarity score	0.025	0.009	0.003	0.001
#selected features	13	43	124	292
mAP	0.733	0.739	0.740	0.739

Table 4.2: Influence of the cut-off value β on Oxford5k

even with a small number of random features. We select 100 random features per inverted list throughout the rest of this paper.

The influence of the cut-off threshold β Table 4.2 shows that features with a distance larger than $\beta \times N_{d(x_i)}$ with $\beta \in [0.8, 0.95]$ have almost no contribution to the retrieval performance. In order to reduce the number of updates of the scoring table, we select $\beta = 0.85$ for all experiments.

4.4.3 Effectiveness of our method

Local adaptive distance In order to compare the adaptive distance function to the Euclidean distance, we use a threshold for separating matching and non-matching features. Figure 4.5 shows the retrieval performance for a varying threshold both for the Euclidean distance as well as for the adaptive distance. Overall, the best mAP using the adaptive distance is 3% better than the Euclidean distance. Furthermore, the adaptive distance is less sensitive when selecting a non-optimal threshold. It is to be noted that in the final setup, our method does not require any thresholding.

Contributions of other steps In order to justify the contribution of other steps that are contained in our method, we evaluate the performance of our method by taking them out of the pipeline. For the experiment on Oxford5k, we find out that without the feature scaling, mAP will drop from 0.739 to 0.707, while without burstiness weighting, mAP will drop to 0.692. With multi-assignment

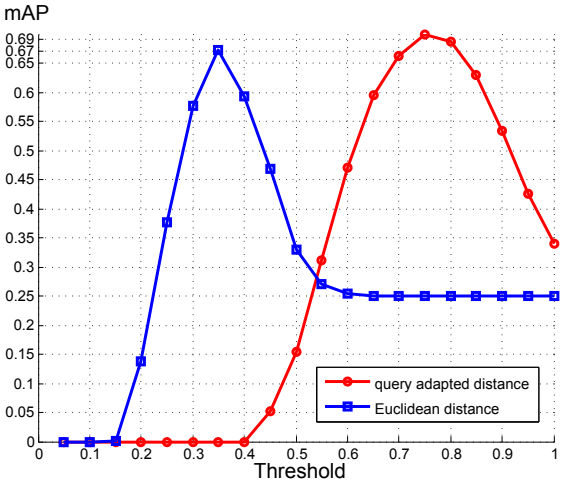


Figure 4.5: Comparison of our adaptive distance with Euclidean distance on Oxford5k dataset

only on the query side, mAP can increase from 0.739 to 0.773 for $MA = 5$, and 0.780 for $MA = 10$. MA denotes the number of inverted lists that are traversed per query feature.

4.5 Results

Throughout all experiments, the set of parameters was fixed to the values obtained in the previous section and vocabularies were trained always on independent datasets. Table 4.3 shows the retrieval performance on all typical benchmarks both with single assignment (SA) and multi-assignment ($MA = 10$). As expected, multi-assignment (scanning of several inverted lists) reduces the quantization artifacts and improves the performance consistently, however, in exchange for more computational load.

Furthermore, we applied an image level post-processing step on top of our method. We choose to use reciprocal nearest neighbors (RNN) [Qin *et al.*, 2011], for the reason that it can be easily integrated on top of a retrieval system independently from the image similarity function. We adopt the publicly

available code [RNN] provided by the original authors and the default settings. RNN significantly improves the results on Oxford5K and Paris datasets, but slightly lowers the result on Holidays. Considering that RNN tries to exploit additional information contained in other relevant database images, which are scarce in Holidays (in average only 2 to 3 relevant database images per query), it is difficult for query expansion methods to perform much better.

Dataset	SA	MA	MA + RNN
Oxford5k	0.739	0.780	0.850
Oxford105k	0.678	0.728	0.816
Paris	0.703	0.736	0.855
Holidays	0.814	0.821	0.801

Table 4.3: Performance of our method on public datasets.

4.5.1 Comparison with state-of-the-art

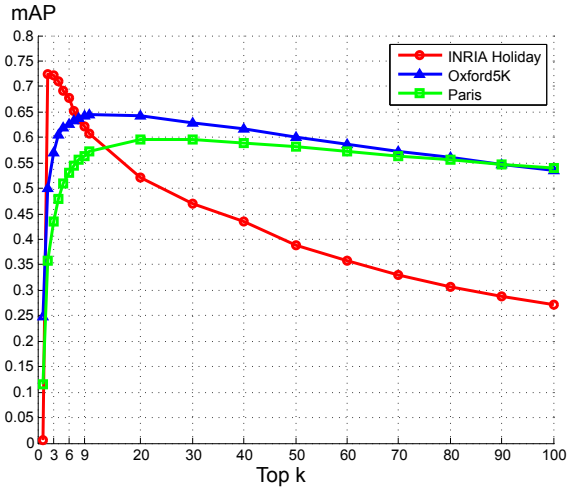


Figure 4.6: Retrieval Performance by using top k nearest neighbor as similar features [Jégou *et al.*, 2011b]

We first compare the performance of our method to Jégou *et al.* [2011b] which relies on using the top k nearest neighbors of the Euclidean distance for selecting the similar features of a query. This work is closest to ours, both in memory overhead and computational complexity. It can be seen in Figure 4.6, that no single k maximizes the performance for all datasets, showing that this parameter is very sensitive to the data. Moreover, our method outperforms the peak results from Jégou *et al.* [2011b] consistently by roughly 10 points of mAP.

Table 4.4 shows the comparison to several other methods without applying any image-level post-processing step. As pointed out by Jégou *et al.* [2010a], training a vocabulary on independent data rather than the evaluated dataset itself can better represent the search performance in a very large dataset. We only compare to state-of-the-art methods using codebooks trained on independent datasets. We achieve the best performance for Oxford5k, Oxford105k, and Holidays and fall only slightly behind Mikulík *et al.* [2010] on Paris.

Dataset	Ours	Mikulík *et al.* [2010]	Jain *et al.* [2011]	Perdoch *et al.* [2009]
Oxford5k	**0.780**	0.742	0.704	0.725
Oxford105k	**0.728**	0.675*	-	0.652
Paris	0.736	**0.749**	-	-
Holidays	**0.821**	0.749**	0.817	0.769/0.818**

Table 4.4: Comparisons with state-of-the-art methods without applying image level post-processing. * indicates the score of merging Oxford5k and Paris and 100K distractor images. ** denotes the result obtained by manually rotating all images in the Holidays dataset to be upright.

Dataset	Ours+RNN	Mikulík *et al.* [2010]	Perdoch *et al.* [2009]	Arandjelovic and Zisserman [2012]
Oxford5k	**0.850**	0.849	0.822	0.809
Oxford105k	**0.816**	0.795	0.772	0.722
Paris	**0.855**	0.824	-	0.765
Holidays	**0.801**	0.758**	0.78	-

Table 4.5: Comparisons with the state of art methods with post-processing in image level. ** denotes the result obtained by manually rotating all images in the Holidays dataset to be upright.

Furthermore, Table 4.5 gives a comparison for the results when additional image-level post-processing steps are applied. We argue, that any post-processing step can directly benefit from our method and illustrate with RNN as example that the best performance can be achieved.

In all of the previous experiments, each feature costs 12 bytes of memory. Specifically, 4 bytes is used for the image identifier and 8 bytes for the quantization codes. As Jégou *et al.* [2011a] mainly show results using more bytes for feature encoding, we also compare our method to theirs with more bytes per feature. As shown in Table 4.6, using more bytes further improves the retrieval results. Even with less bytes than Jégou *et al.* [2011a], better performance is achieved on all datasets.

Dataset	Ours	Ours	Jégou *et al.* [2011a]
Bytes	12	36	44
Oxford5k	0.780	**0.831**	0.764
Paris	0.736	**0.756**	0.728
Holidays	0.821	**0.844**	**0.844**

Table 4.6: Comparison to Jégou *et al.* [2011a] using more bytes per feature.

In all experiments, we compare favorably to the state-of-the-art by exploiting a simple similarity function without any parameter tuning for each dataset. The good results justify our previous analysis and the effectiveness of our method.

4.5.2 Computational Complexity

In a small scale experiment, e.g for Oxford5k, we observe that our method is 30% faster than the original product quantization algorithm [Jégou *et al.*, 2011b] while traversing the inverted lists, for the reason that our method requires no heap structure. However, for a large scale experiment, we observe similar timing of our method to theirs as each inverted list contains a very long list of database features, and thus the computation of the Euclidean distance will dominate the computational time.

4.6 Conclusion

In this paper, we present a probabilistic framework for the feature to feature similarity for high-dimensional local features such as SIFT. We then propose a query adaptive feature to feature distance measurement and derive a global image to image similarity function. Despite the simplicity of this approach, it achieves consistently good results on all evaluated datasets, supporting the validity of our model. Furthermore, it does not require parameter tuning to achieve optimal performance.

5

Learning to rank Bag-of-Words Histogams

5.1 Introduction

Retrieving images of a particular query object in a large database of images is an important problem for computer vision with applications in object discovery [Gammeter *et al.*, 2009], 3D reconstruction [Agarwal *et al.*, 2011], location recognition [Schindler *et al.*, 2007] and mobile visual search [Girod *et al.*, 2011]. Most recent state-of-the-art large-scale image retrieval systems rely on local feastures, in particular the SIFT descriptor [Lowe, 2004] and its variants. Typically, those local descriptors are aggregated into a histogram-based representation of the image referred to as the Bag-of-Words model (BoW) [Sivic and Zisserman, 2003]. BoW models considerably reduce the computational burden and the memory footprint of the systems, because local descriptors are quantised into *visual words*.

For BoW histograms, it is common to use simple similarity functions such as the inner product or cosine similarity [Philbin *et al.*, 2007, 2008; Chum *et al.*, 2011; Qin *et al.*, 2011]. However, such functions are not optimal for modelling the visual similarity between BoW feastures and thus lead to sub-optimal performance for retrieval [Jégou *et al.*, 2009; Jégou and Chum, 2012; Zhu *et al.*, 2013]. The potential problems are the following: a) The evidence coming from co-missing visual feastures is under-estimated [Jégou and Chum, 2012]; b) The similarity between a query image and a database image should not be symmetric [Zhu *et al.*, 2013]; c) Statistical properties of visual words are not taken into account [Chum *et al.*, 2008; Jégou *et al.*, 2009; Zheng *et al.*, 2013].

Even though different methods have been proposed to address each of these problems individually, none provides a satisfying solution to properly account for all of them. Moreover, most authors propose ad-hoc solutions by means of functions controlled by very few parameters. These parameters are then hand-tuned or exhaustively searched on validation/test data to adapt them to each dataset. In this work, we address the problem in a different way, by learning the values of the similarity function directly. Because the number of parameters becomes too large to be set by hand, we learn them using training data.

In the following, we make the following contributions. Firstly, we propose a simple additive approximation of the methods discussed above that leads to our linear model for similarity. We analyze how this model can integrate various statistical properties of the data implicitly. Secondly, in order to learn the parameters of our model on training data, we show how the learning problem can be seen as learning to rank from pairs of images. For this, we optimize a loss function inspired by rank-SVM [Joachims, 2002] so as to maximize an approximation of the mean average precision (mAP) of the system. We also show how our model integrates into an efficient inverse file structure and thus how to use it in large-scale retrieval scenarios. In our experiments, we show that our method consistently and significantly outperforms existing similarity measures on four standard image retrieval benchmarks.

This paper is organized as follows. In Sec. 5.2, we summarize related work. We describe our model and contributions in Sec. 5.3. In Sec. 5.4 we present our experimental validation and we draw conclusions in Sec. 5.5.

5.2 Related work

The Bag-of-Words (BoW) representation has become the *de facto* standard for large-scale image and object retrieval [Philbin *et al.*, 2007]. In this model, the space $\mathbb{R}^D$ of local features is clustered using k-means into K bins, and an image x, viewed as an unordered set of local descriptors, is represented by $x = [x_i]_{1\dots K} \in \mathbb{N}^K$ by counting how many local descriptors of x fall in bin i.

Most recent works in retrieval have focused on improving this model in various ways. The first direction is to improve the different components, such as the local feature representation [Philbin *et al.*, 2010; Simonyan *et al.*, 2014; Wengert *et al.*, 2011; Zheng *et al.*, 2014] or the visual codebook [Mikulík *et al.*, 2010; Gao *et al.*, 2012]. Another is to add more information in the BoW model, such

as spatial layout [Zhang *et al.*, 2011; Shen *et al.*, 2012], attributes [Su and Jurie, 2011] or higher-order statistics [Wu *et al.*, 2009]. Several works also propose to combine the BoW model with post-processing techniques to further filter retrieval results and obtain state-of-the-art performance. For instance: spatial verification using RANSAC [Philbin *et al.*, 2007], voting based on the Hough transform [Jégou *et al.*, 2008; Zhang *et al.*, 2011; Shen *et al.*, 2012], query expansion [Chum *et al.*, 2007, 2011] or reciprocal nearest neighbours [Qin *et al.*, 2011; Zhang *et al.*, 2012].

Our work focuses on an important component of the retrieval system, namely the similarity function used to rank results. Below, we describe in detail the main similarity functions proposed in the literature and used in state-of-the-art methods.

Weighted cosine similarity. The weighted cosine similarity $s_{\cos}$ is the most common measure for BoW in the literature [Philbin *et al.*, 2007, 2008; Chum *et al.*, 2011; Qin *et al.*, 2011]. It is simply computed as a weighted, normalised inner product between the query q and database images d:

$$s_{\cos}(q,d) = \frac{1}{\|q\|\|d\|} \sum_{i=1}^{K} w_i q_i d_i, \tag{5.1}$$

where the weights w_i account for the relative importance of visual words. A common approach is to use the squared Inverse Document Frequency (idf) of the database $\mathcal{D}$:

$$\sqrt{w_i} = \mathrm{idf}(i) = \log|\mathcal{D}| - \log|\{d \in \mathcal{D} : d_i > 0\}|. \tag{5.2}$$

Negative evidence. One problem of the cosine similarity is that only the co-occurrence of visual words is counted as an evidence of similarity while the visual similarity coming from co-missing ones is ignored [Jégou and Chum, 2012]. Firstly, Eq. (5.1) accumulates the similarity only over co-occurring visual words. Secondly, the normalization term is also unaltered by absent features. To take this *negative evidence* into account, Jegou *et al.* proposed to transform the original BoW vector by substracting the average vector $\bar{d}$ of $\mathcal{D}$: $x' = x - \beta\bar{d}$, where β is a tuning parameter. With the transformed features q' and d' as input, Eq. (5.1) now gives a positive contribution for co-missing words. β is tuned by brute force search.

Asymmetric dissimilarity. Another problem is that, in many real world retrieval applications, the (dis-)similarity measurement between a query image

and a database image should not be symmetric [Zhu *et al.*, 2013]. In several retrieval scenarios, the query image is restricted to only contain the object of interest, with minimal background. On the other side, database images are unrestricted. Therefore, the presence or absence of features should be weighted differently whether it is in q or d: $-s_{\text{asym}}(q,d) = \sum_{i=1}^{K}(d_i - q_i)^p \mathbb{I}(d_i > q_i) + \lambda \sum_{i=1}^{K}(q_i - d_i)^p \mathbb{I}(q_i > d_i)$, where p is a constant in $\{1,2\}$, $\lambda > 0$ and $\mathbb{I}(\cdot)$ is the indicator function. In other words, a difference between d_i and q_i is penalized by 1 if $d_i > q_i$ and by λ otherwise. Zhu *et al.* [Zhu *et al.*, 2013] proposed, for $p = 1$, to vary λ as a function of the query and a tuning parameter α, using $\lambda = \alpha \frac{\sum_{j=1}^{N}\sum_{i=1}^{K} d_i^j}{\sum_{j=1}^{N}\sum_{i=1}^{K} \min(q_i, d_i^j)} - 1$.

Burstiness weighting. The idf alone is not sufficient to model all the statistical properties of visual words. A missing aspect is that visual words do not appear independently but in bursts [Jégou *et al.*, 2009]. The above metrics typically over-estimate the similarity for visual words with many occurrences. Instead, the penalty for visual word count difference should be attenuated as the raw value grows. This effect is obtained with a sub-linear transformation of the features [Jégou *et al.*, 2009]. Using simple algebra, we can show that the inter-image and intra-image burstiness models are equivalent to using the following similarity:

$$s_{\text{burst}}(q,d) = \sum_i \left(\text{idf}(i)^2 q_i \sqrt[4]{d_i} \Big/ \sqrt{\sum_{j=1}^{N} \sqrt{d_i^j}} \right). \tag{5.3}$$

In summary, all of these similarity measures can be written in a very general form as:

$$s(q,d) = \tau(q)\tau(d) \sum_{i=1}^{K} s_i(q_i, d_i), \tag{5.4}$$

where the specific choice of τ and s_i help address a specific problem. This approach is not entirely satisfying, as it is challenging to design τ and s_i to account for all of these observations simultaneously and to adapt them to new phenomena to be discovered in the future. Instead, we propose to learn their values directly from training data, as we show below.

5.3 Learning to rank histograms by similarity

Following previous work [Jégou *et al.*, 2009; Jégou and Chum, 2012; Zhu *et al.*, 2013], we start from Eq. (5.4) to define the similarity between two BoW histograms. As explained before, authors often motivate their choices of τ and s_i by aiming at the correction of potential shortcomings of previous choices. Instead, we propose to resort to learning and discover the patterns of a good similarity function for image search, automatically from training data. We describe below our model in Sec. 5.3.1, how we learn its parameters in Sec. 5.3.2, then devise in Sec. 5.3.3 simple techniques to improve the robustness of the system by reducing the number of parameters to learn. Finally, we describe in Sec. 5.3.4 how to integrate our model into an inverted index to allow the use of large-scale databases.

5.3.1 A linear approximation of histogram similarity

Looking at Eq. (5.4), we aim at learning the values $s_i(q_i, d_i)$ directly. This is notably impractical, as each q_i and d_i can be arbitrarily large. However, state-of-the-art methods use very large visual codebooks ($K \approx 10^6$) leading to sparse BoW representations, with few occurrences of any visual word in any given image.[1] As a result, using a *truncated histogram* $\hat{q}_i = \min(q_i, n)$ with $n \in \mathbb{N}^+$ will provide an excellent approximation of the original histogram while limiting the number of possible values of $s_i(\hat{q}_i, \hat{d}_i)$ to $(n+1)^2$.

Additionally, because we learn the values of $s_i(\hat{q}_i, \hat{d}_i)$ directly, these terms can be learned to incorporate a *contribution to the normalisation functions*. This leads to a modified similarity $\hat{s}_i$ and our approximated model becomes additive and writes as:

$$s(q,d) = \tau(q)\tau(d) \sum_{i=1}^{K} s_i(q_i, d_i) \approx \sum_{i=1}^{K} \hat{s}_i(\hat{q}_i, \hat{d}_i), \tag{5.5}$$

where $\hat{s}_i(j,l)$ for $j, l \in [0, n]$ are the $K \cdot (n+1)^2$ parameters to learn. Notably, this additive approximation allows to rewrite Eq. (5.5) as a linear combination of indicator functions:

$$\hat{s}_i(\hat{q}_i, \hat{d}_i) = w_{i\hat{q}_i\hat{d}_i} = \sum_{j=0}^{n} \sum_{l=0}^{n} w_{ijl} \mathbb{I}(\hat{q}_i = j) \mathbb{I}(\hat{d}_i = l), \tag{5.6}$$

[1] As an example, in the UKbench dataset, more than 81% of the visual words occur at most twice in any image and more than 97% of them occur at most 5 times.

where $w_{ijl} = \hat{s}_i(j, l)$. In other words, if we define $\mathbf{\Psi}(q, d)$ as the binary vector indexed by (i, j, l) such that $\mathbf{\Psi}_{ijl}(q, d) = \mathbb{I}(\hat{q}_i = j)\mathbb{I}(\hat{d}_i = l)$ and define $\mathbf{w} = [w_{ijl}]_{i,j,l}$, then:

$$s(q, d) \approx \mathbf{w}^\top \mathbf{\Psi}(q, d). \tag{5.7}$$

Importantly, Eq. (5.7) highlights that $\mathbf{\Psi}$ acts as a feature encoding for the query-document pair (q, d) in a linear prediction model. Despite its simplicity, this model is very general and flexible, and is able to incorporate many of the properties discussed in Sec. 7.2, and potentially others, without having to explicitly model them.

To illustrate this, let us first consider the simple case of $n = 1$. In such case, the truncated histogram $\hat{q}$ simply encodes the absence or presence of visual words (an encoding often referred to as *max-pooling* or *binary bag-of-words*), and there are only 4 weights to learn per visual word: co-absence $\hat{s}_i(0, 0)$, co-occurrence $\hat{s}_i(1, 1)$ and either case of mutual exclusion $\hat{s}_i(0, 1)$ and $\hat{s}_i(1, 0)$. If we learn that $\hat{s}_i(0, 0) > \hat{s}_i(0, 1)$, then not only have we implicitly learned that co-absence of the visual word i contribute more to the similarity than mutual exclusion (as argued by [Jégou and Chum, 2012]) but also exactly by which amount. If we learn that $\hat{s}_i(0, 1) \neq \hat{s}_i(1, 0)$, then this implies that the ideal similarity is indeed asymmetric [Zhu *et al.*, 2013]. Finally, learning all the weights together allows to identify which visual words are more important than others, as indicated by the relative weight of $\hat{s}_i(1, 1)$ and $\hat{s}_j(1, 1)$. Hence, it automatically models re-weighting schemes such as idf. Finally, when $n > 1$, phenomena such as burstiness [Jégou *et al.*, 2009] are also learnt.

5.3.2 Learning to rank query-document pairs

In this section, we delve into the details of learning the parameters of our similarity function. Let $\mathcal{D} = \{d^1, \ldots, d^{N_\mathcal{D}}\}$ be the database of $N_\mathcal{D}$ images with their BoW representation d^i. Similarly, let $\mathcal{Q} = \{q^1, \ldots, q^{N_\mathcal{Q}}\}$ be a set of $N_\mathcal{Q}$ query images. For each query $q \in \mathcal{Q}$, $\mathcal{D}$ is partitioned in a relevant subset $U_\mathcal{D}(q)$ and an irrelevant one $V_\mathcal{D}(q)$. Ideally, the similarity function s would satisfy the following constraints:

$$\forall q \in Q,\ \forall (u, v) \in U_D(q) \times V_D(q), \quad s(q, u) > 1 + s(q, v), \tag{5.8}$$

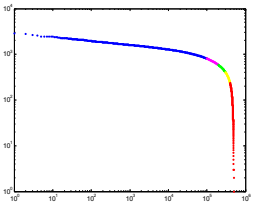


Figure 5.1: Visual word log-frequency in Oxford105k. Colors illustrate the proposed clustering with 5 groups.

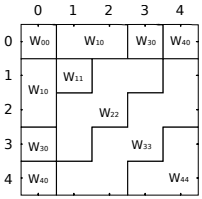


Figure 5.2: Weight pattern learnt on the UKbench^s data. This pattern was learnt for $N_v = 5$, $N_w = 8$ and enforcing symmetry.

which translates the idea that, for each query, any relevant document should yield a larger similarity than any irrelevant one by a margin of 1. Considering that our model in (5.7) is linear, we can simply resort to the following rank-SVM [Joachims, 2002] formulation to learn the weights that minimize the number of violated constraints (*i.e.*, the mAP of the system):

$$\underset{\mathbf{w}}{\operatorname{argmin}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{q\in\mathcal{Q}} \sum_{u\in U_{\mathcal{D}}(q)} \sum_{v\in V_{\mathcal{D}}(q)} \max\left(0, 1 - \mathbf{w}^\top \left(\mathbf{\Psi}(q,u) - \mathbf{\Psi}(q,v)\right)\right), \tag{5.9}$$

where λ controls the trade-off between the regularization of $\mathbf{w}$ and the data loss term.

We propose to use a variant of this model to prevent relevant documents that are very far in the retrieval list to incur very large penalties, and normalise query-specific losses as in [Cao *et al.*, 2006]. This consists in first rewriting Eq. (5.8) with:

$$\forall q \in \mathcal{Q},\ \forall u \in U_{\mathcal{D}}(q), \quad s(q,u) > 1 + \max_{v\in V_{\mathcal{D}}(q)} s(q,v), \tag{5.10}$$

which leads to the corresponding optimisation problem:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{q\in\mathcal{Q}} \frac{1}{|U_{\mathcal{D}}(q)|} \sum_{u\in U_{\mathcal{D}}(q)} \max\left(0, 1 - \mathbf{w}^\top \mathbf{\Psi}(q,u) + \max_{v\in V_{\mathcal{D}}(q)} \mathbf{w}^\top \mathbf{\Psi}(q,v)\right). \tag{5.11}$$

Both problems are convex, hence can be optimized in the primal using sub-gradient descent. As we will show in our experiments, our proposed formula-

tion in Eq. (5.11) leads to a faster optimization and learns more robust parameters compared to [Joachims, 2002].

5.3.3 Robust parameter estimation

As it is described above, our model suffers from an important issue. On one side, retrieval systems improve with larger visual vocabularies. On the other side, the number of parameters to estimate grows with K and, the larger the K, the fewer training data we can obtain to estimate the $(n+1)^2$ weights for each visual word. Learning $K(n+1)^2$ weights is simply unreasonable. Below, we propose two solutions to reduce the number of parameters and thus avoid overfitting. First, we group visual words together and learn a single set of weights for each group. Second, we reduce the number of weights in each set.

Grouping visual words by frequency

Using large vocabularies causes the number of parameters to grow too large. Hence, we propose to identify groups of visual words and share weights among visual words within each group. Similar to previous works that relate visual word frequency with visual word importance (*e.g.*, using idf), we also decide to associate similar frequencies with similar weights. In practice, we resort to a simple unsupervised clustering where we sort the K visual words according to their frequency in the database $\mathcal{D}$ and create N_v groups $\{g_1, \ldots, g_{N_v}\}$ such that each group g_k has the same number K/N_v of visual words.

This grouping can be easily incorporated in our optimization problem of Eq. (5.11) by summing the components of $\mathbf{\Psi}$ corresponding to the visual words in each group g_k, leading to $\mathbf{\Psi}'(q,d) = [\mathbf{\Psi}'_{kjl}(q,d)]_{k,j,l}$ with $\mathbf{\Psi}'_{kjl}(q,d) = \sum_{i \in g_k} \mathbf{\Psi}_{ijl}(q,d)$. Notably, this grouping is performed only during training to learn the value of w_{kjl} for all k, j, and l. At test time, visual words are kept separate to retain they discriminative power and we use for w_{ijl} the value w_{kjl} of the group g_k containing i. Fig. 5.1 illustrates the distribution of visual words frequencies in Oxford105k and its clustering.

Grouping weights with patterns

We also propose to share weights within the same visual word group. First, because small variations in x_i may not have enough impact to justify having

different weights. But also because as n grows, many specific combinations of visual word counts will become too rare to learn their corresponding weight robustly.

Jointly over all the visual word groups, we propose to cluster the $(n+1)^2$ weights into N_w bins $\{b_1, \ldots, b_{N_w}\}$. We resort to the following iterative procedure. We start with $(n+1)^2$ bins, *i.e.* each weight index (j, l) being associated with its own bin b_m. At each subsequent iteration, we merge the 2 adjacent bins[2] whose merging yields the best *mAP* on training data. We iterate until there are only N_w bins left. We refer to the resulting binnings as *weight patterns*, and we illustrate them in Fig. 5.2. If needed, we can enforce symmetry of the weights.

Importantly, although the visual word groups share the same weight patterns, they still have their own specific values for the weights themselves. That is, the number of parameters of the model is $N_v \times N_w$. The weight patterns are incorporated in the training algorithm in the same way as the visual word grouping.

As a short remark, weight patterns can be viewed as a generalization of truncated histograms, as the latter already correspond to weight sharing for histogram values beyond n (*c.f.* Sec. 5.3.1). In a way, we are here searching for an optimal projection of $\mathbb{N}^2$ to $[1, N_w]$ such that the corresponding similarity values maximize the retrieval performance of the system.

5.3.4 Integrating in an inverted index

An important aspect of object retrieval system is its ability to scale gracefully with the number of images in the database and the size of the visual codebook. Using Eq. (5.7) to this end is not practical. For a given query, it involves building the joint feature encoding $\mathbf{\Psi}$ with each document in the database and then applying a potentially large vector-matrix multiplication. It turns out that, using simple algebra, we can rewrite $s(q, d) = \sum_i w_{iq_i d_i}$ the following way:

[2] Two bins b_s and b_t are adjacent if they contain indices $(j, l) \in b_s$ and $(j', l') \in b_t$ such that $|j - j'| + |l - l'| \leq 1$.

$$
\begin{aligned}
s(q,d) =& \sum_i w_{i0d_i} + \sum_{\substack{i \text{ s.t.} \\ q_i>0}} (w_{iq_i0} - w_{i00}) + \sum_{\substack{i \text{ s.t.} \\ q_i>0 \\ d_i>0}} (w_{iq_id_i} - w_{iq_i0} - w_{i0d_i} + w_{i00}) \\
=& s(0,d) + \overline{s}(q,0) + \sum_{\substack{i \text{ s.t.} \\ q_i>0 \\ d_i>0}} \overline{w}_{iq_id_i},
\end{aligned}
\tag{5.12}
$$

where $s(0,d) = \sum_i w_{i0d_i}$ represents the similarity with the empty query and can be pre-computed offline for the database images, $\overline{s}(q,0) = \sum_{\substack{i \text{ s.t.} \\ q_i>0}} (w_{iq_i0} - w_{i00})$ depends only on the query (thus can be ignored for ranking the database images) and finally $\overline{w}_{ijl} = w_{ijl} - w_{ij0} - w_{i0l} + w_{i00}$ are the only terms that need to be explicitly computed for a query. Since they sum only over visual words that are present in both the query and the document, Eq. (5.12) has exactly the form needed to benefit from using an inverted file structure [Sivic and Zisserman, 2003].

At query time, the pre-computed vector with elements $S_d = s(0,d)$ is loaded, then for each visual word i occurring in q, the inverted file structure is used to add $\overline{w}_{iq_id_i}$ to S_d for documents that also contain i. Finally, the documents are sorted by $S_d = s(q,d)$.[3]

5.4 Experiments

In this section, we present our experiments on four standard benchmark data sets for retrieval. We first present them below in Sec. 5.4.1. We study the different parameters of our approach in Sec. 5.4.2 and then compare it to the state of the art in Sec. 5.4.3.

5.4.1 Dataset, feastures, evaluation protocol, and implementation details

We evaluated our method on the standard Oxford5k [Philbin *et al.*, 2007; oxf], Oxford105k [Philbin *et al.*, 2007], University of Kentucky (UKbench) [Nistér

[3] Actually, $[s(0,d)]_{d\in\mathcal{D}}$ can also be sorted offline and an adaptive sort used to just update the ranking.

and Stewénius, 2006; nis] and the INRIA Holidays [Jégou *et al.*, 2008; inr] benchmark datasets.

On those datasets, we computed RootSIFT [Arandjelovic and Zisserman, 2012] descriptors on improved Hessian Affine interest points [Perdoch *et al.*, 2009]. Using approximate k-means [Philbin *et al.*, 2007], we trained a visual vocabulary of $500{,}000$ visual words for each of Oxford5k, UKbench and Holidays.

As our method requires data to learn its parameters, we generate training and test splits of the datasets by randomly dividing them into two halves. During the process, we ensure that relevant queries and documents remain in the same split. Hence, we guarantee that the system will not see at test time any image of any object used to learn its parameters. To prevent results from depending on the actual random choice of split, we generate 10 splits for each dataset and report the average performance. We refer to this protocol as Oxford5k^s, Oxford105k^s, Holidays^s and UKbench^s, resp. To measure performance, we use mean average precision (*mAP*), except for UKbench where we use the top-4 score [Nistér and Stewénius, 2006].

For our approach, three parameters have to be set. The histogram truncation n is set by ensuring that less than 10% of the visual words occur more than n times in any image. This results in $n = 2$ for Oxford and $n = 4$ for Holidays and UKbench. For N_v and N_w, we resort to cross-validation. For all choices of N_v between 1 and 10, we consider all possible numbers of weight patterns. Those are limited since the truncation leaves only few weights. We then select the combination of N_v and N_w that maximizes cross-validation accuracy.

5.4.2 Qualitative analysis of our approach

In this section, we study the following four components of our approach: the choice of optimization problem, the grouping of visual words and patterns and the size of the training set. For this analysis, we use the UKbench dataset.

Problem formulation. We start by comparing the influence of problem formulation on the performance of the system. Fig. 5.3 shows the top-4 score on the test set as a function of the training iterations for the standard rank-SVM cost [Joachims, 2002], Eq. (5.9), and our modified cost, Eq. (5.11). As one can see, the performance of our proposed formulation converges faster (in about 200 iterations) and yields better accuracy.

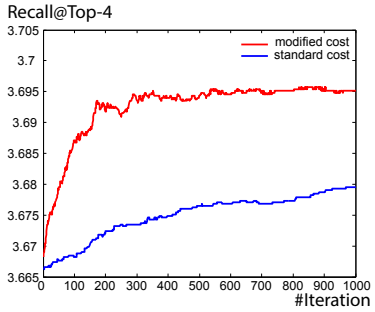


Figure 5.3: Comparison of the performance of learning-to-rank formulations.

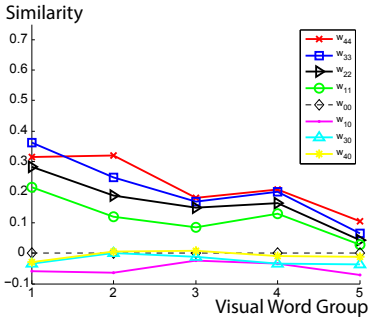


Figure 5.4: Weights learnt for UKbench (pattern shown in Fig. 5.2) for each of the $N_v = 5$ word group.

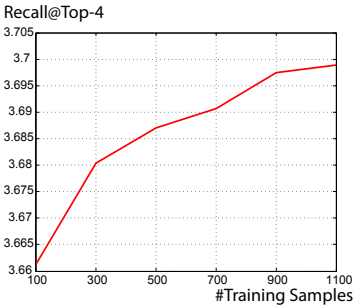


Figure 5.5: Influence of the number of training samples on the top-4 score.

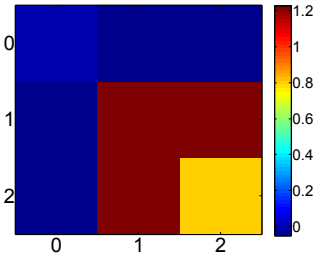


Figure 5.6: Learnt weight pattern for Oxford105k^s ($N_v = 5, N_w = 5$)

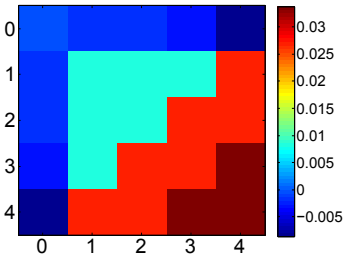


Figure 5.7: Learnt weight pattern for Holidays^s ($N_v = 2, N_w = 6$, sym.)

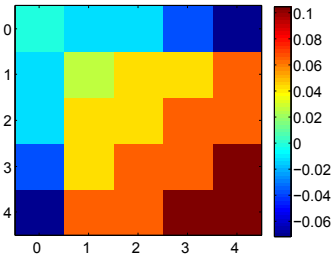


Figure 5.8: Learnt weight pattern for UKbench^s ($N_v = 5, N_w = 8$, sym.)

Learnt weights. Fig. 5.4 shows the weights learnt for each of the $N_v = 5$ visual word group, using the pattern shown in Fig. 5.2, and normalized such that $w_{00} = 0$. We make three main observations. First, in each group, the weights on the diagonal (*i.e.*, w_{11} to w_{44}) are larger than w_{00} while most of the weights corresponding to mutual exclusion are smaller. This shows that co-missing evidence is indeed learnt. Second, those diagonal weights also show a diminishing return as the word frequency increases. This is consistent with burstiness modelling. Third, when comparing the different visual word groups, the corresponding weights tend to decrease as we move to more frequent visual words. This is a property of idf weighting.

Number of training samples. Fig. 5.5 shows the top-4 score with weights learnt from varying numbers of training samples. As we see, the performance of our model increases with more training data, and does not reach a plateau. This highlights the flexibility of our model and the interest of creating larger training datasets in the future to further improve performance.

Weight patterns. Finally, Figures 5.6, 5.7 and 5.8 show the weight patterns that we learnt for Oxford, Holidays and UKBench datasets, respectively. Notably, the weights for Oxford are not enforced to be symmetric, as we observe $w_{01} = -0.0412$ and $w_{10} = -0.0493$.

5.4.3 Comparison to the state of the art

In this section we compare our approach with ad-hoc similarities used in state-of-the-art methods and then report performance on the datasets with train-test splits.

Similarities in the state of the art. For this experiment, we have implemented several similarities used in state-of-the-art approaches (*c.f.* Sec. 5.2): a) Cosine similarity [Chum *et al.*, 2011]; b) Burstiness weighting [Jégou *et al.*, 2009]; c) Negative evidence [Jégou and Chum, 2012]; and d) Adaptive asymmetric similarity [Zhu *et al.*, 2013]. Tab. 5.1 shows their performances on the original datasets using the same system (as described in Sec. 5.4.1) for fair comparison. We observe in Tab. 5.1 that the best similarity to use depends on the dataset. The asymmetric assumption fits Oxford well, whereas Cosine and Burstiness are better similarities for Holidays and UKbench. Importantly, these numbers are comparable with similar state-of-the-art systems in the literature (*e.g.*,

single-assignment BoW on Hessian Affine points, without post-processing), so we can confidently use them on our train-test splits as strong competitors.

Evaluation on train-test splits. We can now compare the same state-of-the-art similarities to our approach on the train-test splits of the benchmark datasets. Tab. 5.2 summarizes the performances over 10 such splits as described in Sec. 5.4.1. As each split is smaller than the original database – thus making each search task slightly easier –, the reported performances are slightly increased. Still, the relative ranking of the ad-hoc similarities [Chum *et al.*, 2011; Jégou and Chum, 2012; Jégou *et al.*, 2009; Zhu *et al.*, 2013] is consistent with Tab. 5.1 on the original datasets.

From Tab. 5.2, we make the following observations. Clearly, our method consistently and significantly improves over all other similarities on all datasets. Over the 40 total runs (10 splits, 4 datasets), our method does not achieve the best performance for only 1 run. This is true even when the number of training samples is small: with only about 30 queries per split for Oxford5k and Oxford105k, we still obtain $+3\%$ and $+6\%$ improvement, resp., over the best competitor [Zhu *et al.*, 2013]. We also show improvements on Holiday^s ($+1\%$) and UKbench^s ($+0.16$, *i.e.* $+4\%$, relatively). This shows that our model is very general and very flexible, and can adapt to the specificities of each dataset, and that our contributions in Sec. 5.3.3 indeed allow to learn robust weights and prevent overfitting.

These results are also very promising considering that our approach has the potential to exploit more training data (*c.f.* Fig. 5.5) and that the procedures for identifying visual word groups and weight patterns have room for improvement. Despite these limitations, we report state-of-the-art results for 4 different datasets without the burden of hand-crafting similarity functions to adapt to the specificities of the data. Moreover, our system can seamlessly benefit from improvements in interest point detection, local descriptors, visual vocabulary learning and post-processing techniques such as geometrical verification. Such techniques typically bring improvements in the order of $+10\%$ [Simonyan *et al.*, 2014].

	Oxford5k	Oxford105k	Holidays	UKbench
Cosine Similarity [Chum *et al.*, 2011]	0.769	0.679	**0.848**	3.42
Burstiness Weighting [Jégou *et al.*, 2009]	0.777	0.703	0.847	**3.46**
Negative Evidence [Jégou and Chum, 2012]	0.771	0.629	0.836	3.35
Adaptive Asymmetric Similarity [Zhu *et al.*, 2013]	**0.793**	**0.719**	0.783	3.30
Comparable literature	0.795 [Simonyan *et al.*, 2014]	0.723 [Perdoch *et al.*, 2009]	0.793 [Jégou *et al.*, 2009]	3.50 [Qin *et al.*, 2011]

Table 5.1: Comparison of ad-hoc similarities on the original versions of the datasets (mAP or top-4 score depending on the dataset). In bold we show the best results: they are comparable with state-of-the-art results available in the literature for comparable approaches.

	Oxford5k^s	Oxford105k^s	Holidays^s	UKbench^s
Cosine Similarity [Chum *et al.*, 2011]	0.819 (0)	0.725 (0)	0.862 (0)	3.51 (0)
Burstiness Weighting [Jégou *et al.*, 2009]	0.826 (0)	0.748 (0)	0.858 (0)	3.54 (0)
Negative Evidence [Jégou and Chum, 2012]	0.830 (0)	0.684 (0)	0.848 (0)	3.44 (0)
Adaptive Asymmetric Similarity [Zhu *et al.*, 2013]	0.839 (1)	0.758 (0)	0.795 (0)	3.38 (0)
Learnt Histogram Similarity (this paper)	**0.870 (9)**	**0.816 (10)**	**0.871 (10)**	**3.70 (10)**

Table 5.2: Comparison to alternative similarities. We report the average performance over the 10 splits of the data (mAP or top-4 score depending on the dataset) and in parenthesis the number of runs where the method is the best. In bold is the best result for each dataset.

5.5 Conclusion

In this chapter, we have presented a novel framework to directly learn the visual similarity that maximizes the accuracy of an object retrieval system. Our model is very flexible and allows us to seamlessly integrate statistical properties of BoW histograms without modelling them explicitly. In our experiments, we have shown the superiority of our similarities compared to those used in state-of-the-art approaches.

6

Reciprocal Nearest Neighbor

6.1 Introduction

In this chapter we try to exploit another characteristic specific to visual data in order to improve accuracy of object retrieval results: often, the reference database contains many images showing the same object covering it from varying viewpoints *etc.*. We make use of this by constructing a graph on the image database connecting each image with likely related images. At query time this graph is used to construct a set of database images that are closely related to the query image, then based on this *close set* the rest of the database is re-ranked. As we will show this has two benefits: first, treating the two sets with different similarity measures allows for compensation for the "curse of dimensionality", *i.e.* the degradation of distance functions in high dimensional spaces. Second, it allows for dealing with the uneven distribution of images in the data space. Dealing with both challenges has very beneficial effect on retrieval accuracy.

The main contribution of this chaper is a method that improves image retrieval purely on the bag-of-words level. It does so without relying on lower-level information like for instance the geometric arrangement of features or the geometry of the descriptor space. As such our method can be used in a wide variety of settings. We also achieve very competitive results at reasonable overhead in memory usage and very little additional computational complexity during query time.

The remaining part of the chapter is structured as follows: We first discuss related work in the immediately following section. Section 6.3 lays out the basis for our method, by discussing some key characteristics of visual words based object retrieval. We introduce our method for more accurate object re-

trieval in Section 6.4. Experiments and analysis of the effects of optimization on retrieval tasks follow in Section 6.5. Section 6.6 concludes this chapter.

6.2 Related work

Our work relates to recent contributions in the field of object retrieval with visual vocabularies in several aspects. The relevant works build on the common bag-of-features retrieval approach and have proposed improvements, which can be roughly grouped into three categories.

A first group of works deals with improvements on the feature level. In descriptor space the Euclidian distance is often used to assess the similarity of features. However, it has been shown this is not the optimal similarity measure in most situations. In the context of large scale image retrieval this problem has recently been addressed by several works. For instance in [Mikulík *et al.*, 2010] a probabilistic relationship between visual words is proposed as an alternative distance measure. It is based on an "oversegmentation" of the descriptor space with an extremely large vocabulary, and probabilistic relations between the visual words. This way, for each feature mapped to a visual word, a statistic of alternative visual words is learned. The relations are learned offline from a large set of feature tracks. Slightly similar is the work [Philbin *et al.*, 2010], where data is used to learn a projection from SIFT feature space to a new Euclidean space, such that clustering is more likely to put matching descriptors into the same visual words.

A second group of works deals with the quantization artifacts introduced while assigning features to visual words. The most common effect of quantization artifacts is, that for two images showing the same object, corresponding features are not assigned to the same visual word. One way of dealing with this problem is by assigning each feature descriptor to multiple visual words as proposed in [Philbin *et al.*, 2008], however the more words are assigned to a feature, the more posting lists in the inverted index have to be traversed, thus increasing the query time. Jégou *et al.* [2008] addressed this problem by first constructing a relatively coarse vocabulary plus a binary signature for each feature. When a feature of the query images is assigned to a visual word of the coarse vocabulary, the binary signature is used to filter out database features by setting a threshold on the Hamming distance.

A third group of works deals with shortcomings on the document retrieval or database level. Chum *et al.* [2007] adopt query expansion (that originated in text retrieval) to the visual domain. Strict geometric verification is applied to the initial top list in order to extract a set of images that are very likely to be relevant to the query. Then a generative model is used to fuse the information provided by the additional images into a new query, which significantly increases recall.

A common cause of problems is due to the independence assumption between visual words, commonly used because of efficiency reasons. In reality this independence assumption is violated and some visual words co-occur more often than others. This can severely degrade retrieval accuracy. If for instance the query contains a set of frequently co-occurring visual words, then it is likely to match to unrelated images that contain the same set of co-occurring visual words. These sets are commonly referred to as *bursts* [Jégou *et al.*, 2009] or *co-ocsets* [Chum and Matas, 2010]. Jégou *et al.* [2009] evaluated several voting schemes that account for intra- and inter-image *bursts*. Chum and Matas [2010] addressed this problem by finding and removing sets of frequently co-occurring visual words. In both cases improvement in retrieval accuracy was demonstrated. Also operating on the document vector level, Jégou *et al.* [2007, 2010c] improved the accuracy of visual word retrieval, by accounting for changes in the local distributions of the visual word vectors. To this extent, they introduced an iterative update scheme that modifies the distance function between vectors in a way that nearest neighborhood relationships become more symmetric.

Most similar to this paper are probably Chum *et al.* [2007] and Jégou *et al.* [2010c]. As we will explain in the following sections in more detail, the key differences to our work are that we do not rely on lower level information like for instance the geometric arrangement of features and we do not symmetrize nearest neighbor relationships (in contrast to Jégou *et al.* [2010c]).

6.3 Motivation

In this section we motivate our approach for improving accuracy of object retrieval by two key observations.

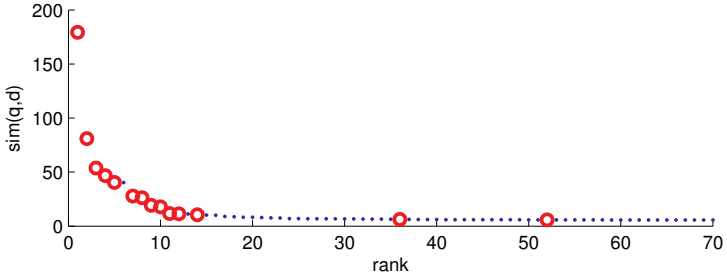


Figure 6.1: Degradation of similarity in document space for a typical query from the Oxford5k data set. The y-axis shows $\mathrm{sim}(\mathrm{q},\mathrm{d})$, the x-axis the rank of retrieved images. The red circles show true positives at their similarity and rank.

For all experiments in this chapter we use the same similarity function as Jégou *et al.* [2008], which corresponds to the bag-of-words model with an additional inverse document frequency weighting term:

$$\mathrm{sim}\,(q,d) = \frac{\sum_{i=1}^{K} q_i\, d_i\, idf(i)^2}{\|q\|\,\|d\|} \tag{6.1}$$

$$\mathrm{idf}\,(i) = \log\left(\frac{\sum_{i=1}^{K}\sum_{d\in D} d_i}{\sum_{d\in D} d_i}\right) \tag{6.2}$$

where q and d are visual word vectors of length K and $D = \{d^1 \ldots d^N\}$ is the set of database vectors.

Observation 1: Similarity functions degrade quickly in high dimensional spaces. A fundamental issue for visual word based image retrieval is the high dimensionality of the visual word vector space. While this high dimensionality facilitates fast search, it also has the effect, that most distance or similarity measures quickly degenerate at points far away from the query vector.

An illustration of this phenomenon is shown in Figure 6.1 where we plot the similarity measure $\mathrm{sim}(\mathrm{q},\mathrm{d})$ (*c.f.* Equation 6.1) for the top 70 ranked images in the Oxford5k data set [oxf] for a given query. Correctly retrieved matches from the evaluation data set are denoted by a red circle. Most images with high similarity are of course true positives, however the similarity curve quickly

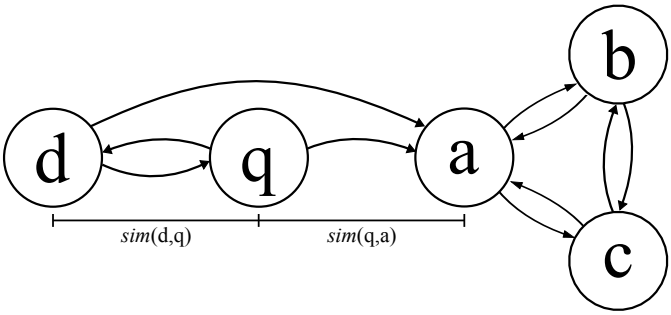


Figure 6.2: Difference between the unidirectional nearest neighbor set $top(2, q)$ and the 2-reciprocal nearest neighbor set $R(2, q)$.

flattens out giving relevant and non relevant images almost the same score at lower ranks. So the similarity measure is very useful for images close to the query, but it loses its utility far away from the query. One way of dealing with this problem is by modifying the similarity measure $\mathrm{sim}(\mathrm{q}, \mathrm{d})$ in a way that more relevant images are pushed closer to the query and irrelevant images are pushed away from the query. For instance, Hamming embedding [Jégou *et al.*, 2008] or soft visual word assignment [Philbin *et al.*, 2008] do this by reducing quantization artifacts.

Descriptor space learning techniques [Philbin *et al.*, 2010; Mikulík *et al.*, 2010] push relevant images closer to the query by correcting for the fact that the Euclidean norm is not a perfect distance measure in SIFT or SURF descriptor spaces. In addition, filtering irrelevant images from the ranked lists is typically achieved by geometric verification [Philbin *et al.*, 2007] or similar methods (*e.g.* Jégou *et al.* [2008]).

In this chapter we try to address these effects of the curse of dimensionality in a slightly different way. Accepting that the similarity measure can degrade quickly, we will split the database vectors at query time into two groups. One group which is close to the query, and for which the regular similarity measure $\mathrm{sim}(\mathrm{q}, \mathrm{d})$ can still correctly separate relevant from non-relevant vectors, and a second group, for which the similarity measure can not distinguish between relevant from non-relevant vectors anymore. For the second group we use a different similarity measure.

Observation 2: Non-reciprocity of near neighbor relationships. Let us define the k-nearest neighbors (*i.e.* the top-k list) of a vector q as the k most similar vectors in the database D:

$$\begin{aligned} \text{top}(k,q) &\subset D && (6.3)\\ |\text{top}(k,q)| &= k && (6.4)\\ \text{sim(q, a)} > \text{sim(q, b)} &\quad \forall \quad a \in \text{top}(k,q) \\ & \qquad\quad b \in D \setminus \text{top}(k,q) && (6.5) \end{aligned}$$

While the similarity measure $\text{sim(q, d)} = \text{sim(d, q)}$ itself is symmetric, nearest neighbor relationships are not. This means that $a \in \text{top}(k,b)$ does not imply $b \in \text{top}(k,a)$ in general.

We define the set of k-reciprocal nearest neighbors $\text{R}(k,a)$ of a as

$$\text{R}(k,a) = \{b \in \text{top}(k,a) \wedge a \in \text{top}(k,b)\} \tag{6.6}$$

which is of course trivially symmetric. The k-reciprocal nearest neighborhood relationship $b \in \text{R}(k,a)$ is also a much stronger indicator of similarity than the unidirectional nearest neighborhood relationship $b \in \text{top}(k,a)$, since it takes into account the local densities of vectors around a and b.

We illustrate in Figure 6.2 the difference between the unidirectional nearest neighbor set $top(2,q)$ and the 2-reciprocal nearest neighbor set $R(2,q)$. $top(2,q)$ contains the node a and d, $R(2,q)$ only contains node d, even though a and d are at the same distance from the query q. In such a situation it makes sense to assume that d is more relevant to the query q than a, since a has a high similarity to other nodes that share no connection to q.

We are of course not the first to make this observation. Contextual dissimilarity measures [Jégou *et al.*, 2007, 2010c] for instance are based on exactly this idea. However unlike Jégou *et al.* [2007, 2010c], we do not directly symmetrize nearest neighborhood relationships in this work. Instead we use k-reciprocal nearest neighbors as a tool to find images which are very likely to be related and to disambiguate database vectors that are far away from the query vector.

These two observations are the basis for our object retrieval method, which will be discussed in the following section.

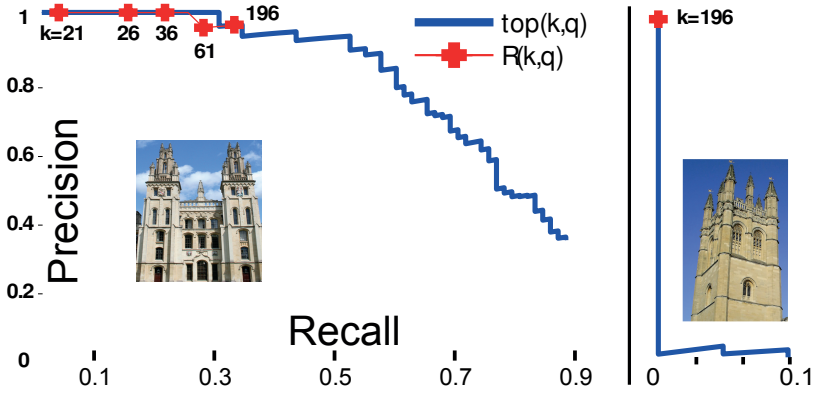


Figure 6.3: Precision and recall of $R(k, q)$ in comparison to the top(k, q).

6.4 Our Approach

At query time we want to separate the database into two disjoint sets, the *close set* which contains images highly relevant to the query and the *far set* which simply refers to the rest of the database. The final ranking list is the concatenation of the *close set* for which parts internally are ranked according to the original similarity measure $sim(q, d)$ (*c.f.* Equation 6.1) and the *far set* which is ranked according to a different similarity measure. We first discuss how the *close set* is constructed and then describe the similarity measure that is used for the *far set*.

6.4.1 Close set construction

In order to identify images highly related to the initial query image q, we start by adding the k-reciprocal nearest neighbors $R(k, q)$ of the query to the *close set*.

In Figure 6.3 we show for a query in the Oxford5k data set how precision and recall of $R(k, q)$ change for various values of k. With higher values of k, recall is increased and saturates while precision rarely decreases. Since in practice some images have very few k-reciprocal nearest neighbors, even for very large k, we grow the initial *close set* $N_{q,t=0}$ by iteratively adding neighboring nodes

to increase recall. Nodes are only added if a set of conditions are met which are designed in a way, that only images that are very likely to be related to the query image are added.

We first define the forward rank f-rank(a, q) of a as the position that a has in the top list of q and the backward rank b-rank(a, q) is defined as the position that q occupies in the top-k list of a:

$$\text{f-rank}(a,q) = k \iff a \in \text{top}(k,q) \setminus \text{top}(k\text{-}1,q) \tag{6.7}$$

$$\text{b-rank}(a,q) = \text{f-rank}(q,a) \tag{6.8}$$

$$a \in \text{R}(k,q) \iff \text{f-rank}(a,q) < k \wedge \text{b-rank}(a,q) < k \tag{6.9}$$

Since we are only interested in finding nodes close to the query, we only consider nodes $d \in D$ if their forward and backward rank relative to the query q do not exceed a certain threshold k_{max}:

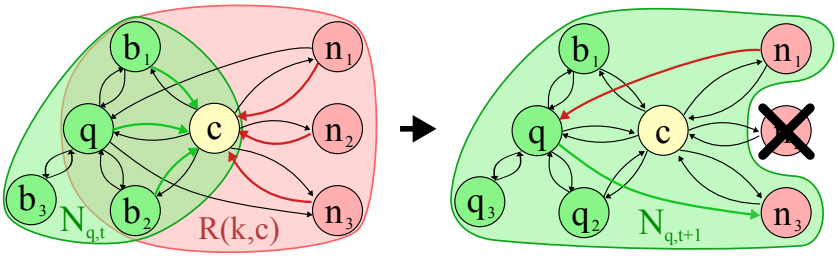
$$\text{f-rank}(q,d) < k_{max} \ \vee \ \text{b-rank}(q,d) < \frac{1}{2}\, k_{max} \tag{6.10}$$

Ignoring all nodes which do not satisfy these constraints we grow $N_{q,t=0}$ by the following procedure as described in Algorithm 1.

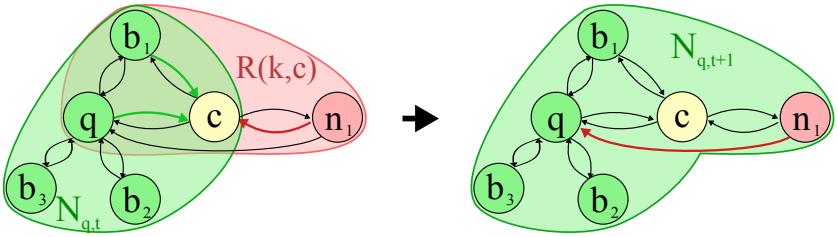
for $t \leftarrow 0$ **to** 2 **do**
  $N_{q,t+1} \leftarrow N_{q,t}$;
  foreach $n \in N_{q,t}$ **do**
    if $|N_{q,t} \cap R(k,n)| > \frac{1}{2}\,|N_{q,t}|$ **then**
      $N_{q,t+1} \leftarrow R(k,n) \cup N_{q,t+1}$;
    end
    if $|N_{q,t} \cap R(k,n)| > |R(k,n) \setminus N_{q,t}|$ **then**
      $N_{q,t+1} \leftarrow R(k,n) \cup N_{q,t+1}$;
    end
  end
end

Algorithm 1: Expansion step

The first condition allows only nodes which are connected to at least half of the *close set* to bring in their neighbors. This high connectivity ensures that added nodes are very likely to be relevant to the query. The second condition relaxes this restriction slightly by allowing weakly connected nodes to bring in



(a) Node C's neighborhood is considered, if it contains more than the half of the initial set, *i.e.* $|N_{q,t} \cap \mathrm{R}(k,c)| > \frac{1}{2}\,|N_{q,t}|]$



(b) Node C's neighborhood is considered, if it adds less unknown nodes than known, *i.e.* $|N_{q,t} \cap \mathrm{R}(k,c)| > |\mathrm{R}(k,c) \setminus N_{q,t}|]$

Figure 6.4: Overview over the expansion rules. For the new set $N_{q,t+1}$ only nodes are considered which either occur in the first half of the top-k list of the query image ($q \rightarrow n_3$), or if the query image occurs within the top list of the image ($n_1 \rightarrow q$).

their neighbors if the amount of new neighbors is smaller than the amount of connections already made to the *close set*. Nodes added to $N_{q,t+1}$ are sorted according to $sim(q,d)$ and inserted in this order into the final *close set*. This procedure can be seen as a form of query expansion, however unlike Chum *et al.* [2007] we do not rely on any geometric or other lower-level information. Figure 6.4 gives an overview of the two conditions for better visualization and in Figure 6.5 we give a real world example of the growing procedure.

In order to efficiently construct the *close set* for a new query, we pre-compute a directed graph $(d_1 \ldots d_n \in D, (u,v)_i \in V)$ on top of the image database. In this graph every node represents an image and a connection $(u,v) \in V$

Figure 6.5: Example for the *close set* in the expansion step.

from node u to another node v is made if node v appears in the truncated top($k_{\max}$, $\cdot$) list of u:

$$(u, v) \in V \iff v \in \text{top}(k_{\max}, u) \tag{6.11}$$

where we used $k_{\max} = 1000$ for all experiments in this paper. Using Equations 6.8 and 6.9, k-reciprocal neighborhoods R(k, q) can be efficiently determined.

In order to construct this graph, we query our retrieval system with every image in the database. While this step is quadratic in the number of images, we do not see this yet as a fundamental restriction since computation is trivially distributable. Also the query operation is quite fast, for a set of one million images we could compute the aforementioned graph in less than 5 hours using only 8 machines. It is reasonable to assume, that calculating the graph for 10 million images would still be feasible. For larger data sets, approximations may be used like for instance min hash [Chum *et al.*, 2008] which has only linear complexity in the number of images. However since we can still deal quite comfortably with up to one million images we have not evaluated this for the purpose of our method. At query time the similarity of the query image to all database images is calculated and the graph is updated to include a node for the query image.

6.4.2 Far set re-ranking

Once the *close set* is constructed, it is used to re-rank the rest of the database. Since images outside of the *close set* are likely to have a low similarity to the query, the original similarity measure $sim(q, d)$ is not useful anymore. From the vantage point of the query, images outside of the *close set* all look equally dissimilar. However if we turn the tables, and look from the position of an element in the far set this might not be true.

Images in the *far set* which are closely surrounded by other images in the *far set* will populate their top(k, $\cdot$) list with their close neighbors but not the ones from the *close set*. However images which are dissimilar to the entire database

but still rather close to the initial query can populate their top$(k, \cdot)$ list with images from the *close set*.

Intuitively it also makes sense that images which do not have any close neighbors except for images in the *close set* are more likely to be relevant to the query, than images that have close neighbors which are not related to the *close set*. In order to make use of this contextual similarity we calculate for each document in the *far set* ($f \in D \setminus N_{q,2}$) the average rank that images in the *close set* would have if image f were used as a query:

$$\text{sim}(f, q) = \text{cutoff} - \frac{1}{|N_{q,2}|} \sum_{c \in N_{q,2}} \min\left(\text{b-rank}(f, c), \text{cutoff}\right) \tag{6.12}$$

where we use a cutoff to account for the fact that only a truncated version of the ranking lists is present at retrieval time. We used cutoff $= 3000$ for all experiments in this paper.

6.5 Experiments

In this section we evaluate the performance of our method on five different datasets. First we give an overview of the datasets. Then we asses the performance of the *close set* and the performance of the *far set* re-ranking separately. At the end a comparison of our full method to the baseline approach is given.

6.5.1 Evaluated datasets

We evaluated our method on the Oxford5k [Philbin *et al.*, 2007; oxf], Oxford105k [Philbin *et al.*, 2007], Paris [Philbin *et al.*, 2008; par], University of Kentucky [Nistér and Stewénius, 2006; nis] and the INRIA Holidays [Jégou *et al.*, 2008; inr] dataset.

We used Hessian Affine SIFT descriptors and approximate k-means [Philbin *et al.*, 2007] to cluster a visual vocabulary with 500 000 centroids for Oxford5k, Paris and Kentucky each. For Oxford105k we used the same vocabulary as for Oxford5k. For the Holidays dataset we received the pre-calculated visual words for a $200k$ visual vocabulary from the authors of Jégou *et al.* [2008]. Thus our baseline and the one from Jégou *et al.* [2008] are exactly the same.

As performance measure, we used mean average precision (*mAP*) on the Oxford5k, Oxford105k, Paris and Holidays dataset while for the University of Kentucky dataset we use the top-4 score as defined by Nistér and Stewénius [2006].

6.5.2 Close set accuracy

In the first part we demonstrate, that the construction of the *close set* which forms the first part of the final ranking list leads to higher accuracy than simply taking the top-k elements of the original ranking list. The size of the *close set* for a given query is dependent on the number of similar images in the database. Queries with many similar images in the database have a larger *close set* than queries with only few similar images in the database. Furthermore, the size of the *close set* depends on the threshold k. By varying k we produce *close sets* of different sizes. The *far set* for which in this experiment regular ranking (*c.f.* Equation 6.1) is used, is appended to the *close set* to form the final ranking list.

As can be seen by the blue lines in Figures 6.6, 6.7, 6.8, 6.9 and 6.10 this gives a major improvement on all datasets for a wide range of k. The mAP and top-4 score give high importance to the first part of a ranking list, which is exactly where the *close sets* increases accuracy.

6.5.3 Far set accuracy

We investigate the effect of replacing the *close set* by simply a truncated top-k list for different values of k and re-rank the *far set* using this list according to Equation 6.12.

As can be seen by the green lines in Figures 6.6, 6.7, 6.8, 6.9 and 6.10 this gives an improvement on all datasets for small k. However as k increases the performance asymptotically degrades back to the base line, since larger and larger portions of the beginning of the final ranking list are ranked using the same similarity measure as the baseline. This is especially visible for the Kentucky dataset. Since the top-4 score only considers the first 4 positions of the ranking list, for $k > 4$ the performance is equal to the baseline.

6.5.4 Full method

The full method combines the aforementioned rank list construction methods, such that the first entries of the final ranking list consist of the *close set* to which the *far set* is appended.

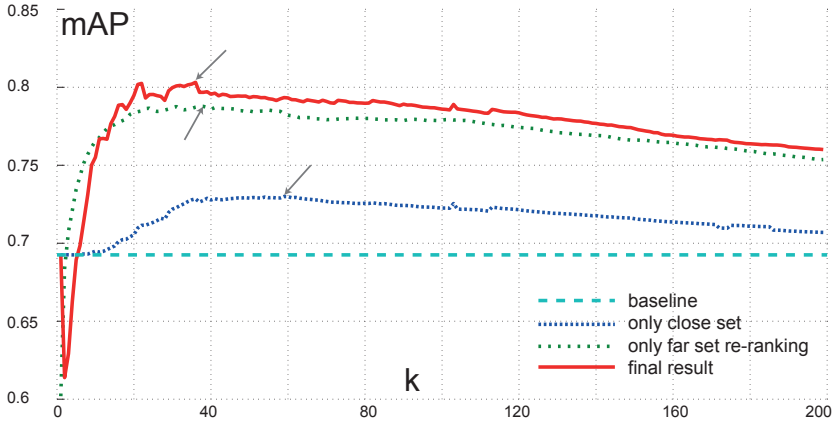


Figure 6.6: Mean average precision for Paris.

The red lines in Figures 6.6, 6.7, 6.8, 6.9 and 6.10 show the final result of the whole method for different thresholds k. The combination of *close set* construction and *far set* re-ranking leads to superior results over the baseline in all cases.

Figure 6.11 shows the average precision for the baseline versus the average precision of our improved method for individual query images for a fixed $k = \text{argmax}_{\hat{k}} \, \text{mAP}(\hat{k})$. Off-diagonal markers in the upper left triangle show a performance improvement, markers in the lower right triangle a degradation. For the University of Kentucky dataset a slightly different visualisation approach was taken. The top-4 score of the baseline method is plotted against the top-4 score of our new method. Each of the bubble's area corresponds to the number of images at this coordinate.

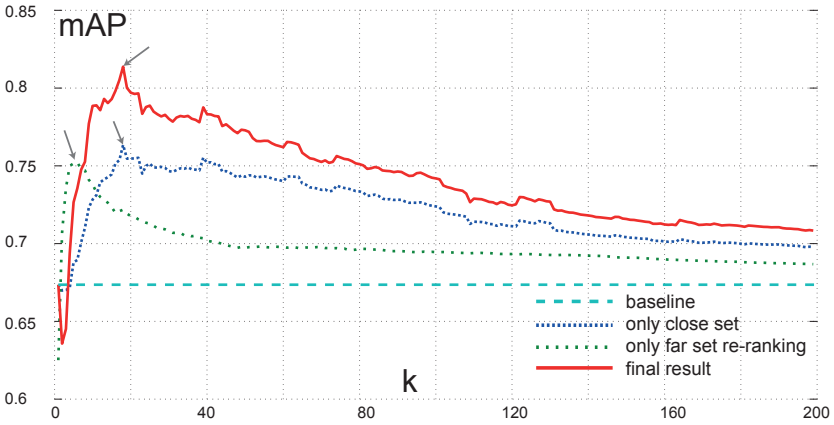


Figure 6.7: Mean average precision for Oxford5k.

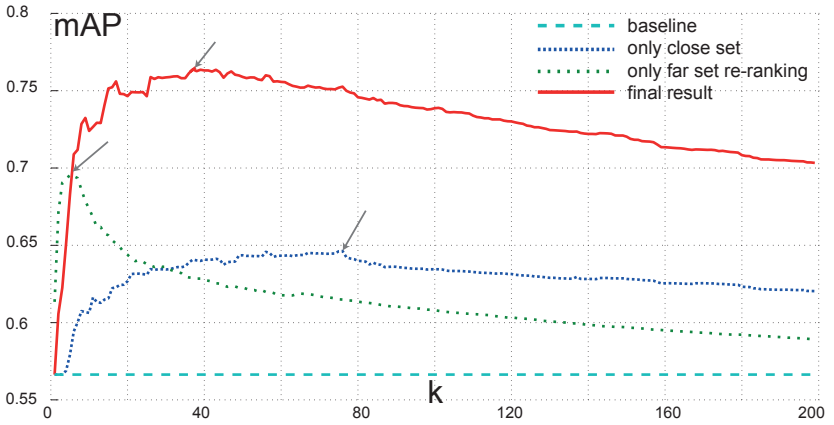


Figure 6.8: Mean average precision for Oxford105k.

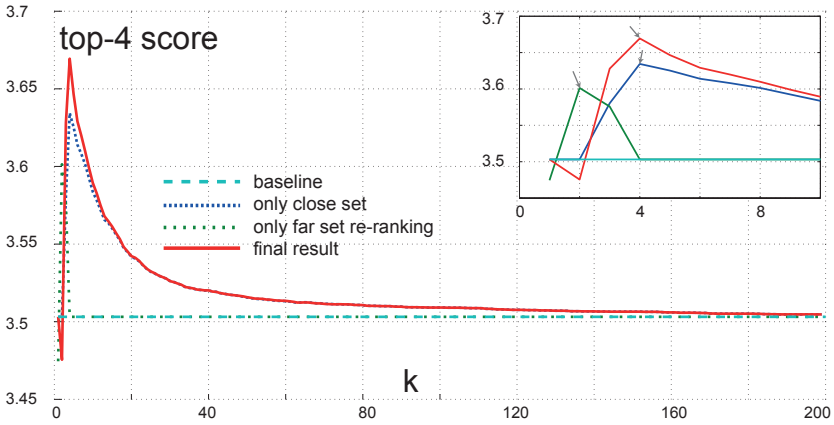


Figure 6.9: Top-4 score for Kentucky.

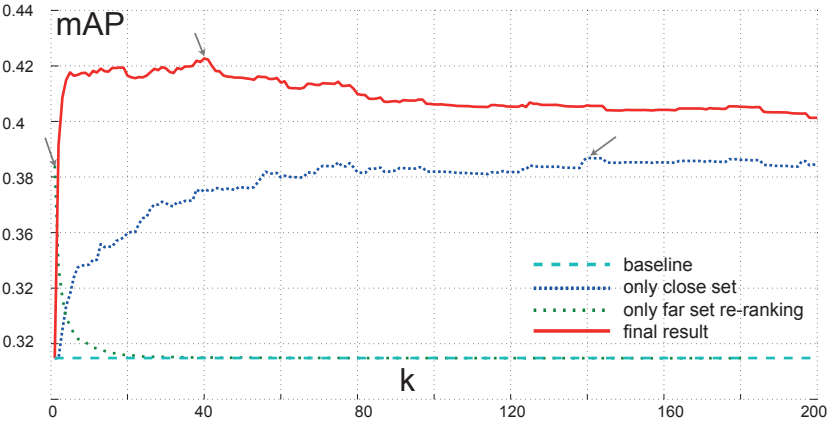
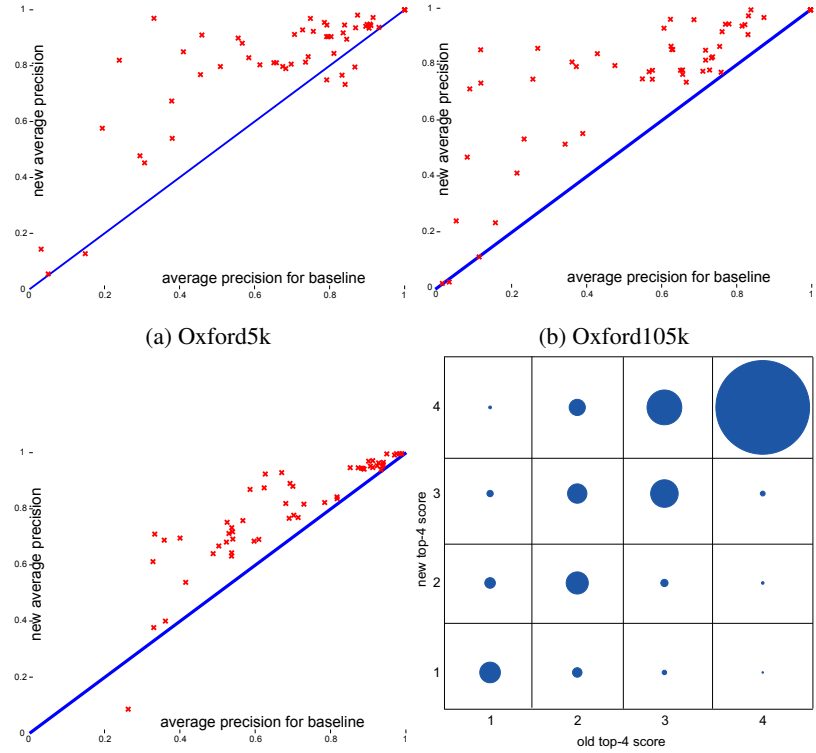


Figure 6.10: Mean average precision for INRIA Holidays.



(a) Oxford5k (b) Oxford105k

(c) Paris (d) Kentucky

Figure 6.11: Average precision (*AP*) of the baseline versus AP of our method

Dataset	Baseline	Our method	Jégou *et al.* [2010c]	Jégou *et al.* [2009]	Mikulík *et al.* [2010]
Oxford5k [Philbin *et al.*, 2007; oxf]	0.674	0.814		0.685	**0.849**
Oxford105k [Philbin *et al.*, 2007]	0.567	0.767			0.795
Paris [Philbin *et al.*, 2008; par]	0.693	0.803			**0.824**
INRIA+1 Mio [Jégou *et al.*, 2008; inr]	0.315	0.423		**0.77**	
Kentucky [Nistér and Stewénius, 2006; nis]	3.5	3.67	**3.68**	3.64	

Dataset	Baseline	Our method	Chum *et al.* [2007]	Chum and Matas [2010]	Philbin *et al.* [2010]
Oxford5k [Philbin *et al.*, 2007; oxf]	0.674	0.814			0.707
Oxford105k [Philbin *et al.*, 2007]	0.567	0.767	0.782	**0.864**	0.615
Paris [Philbin *et al.*, 2008; par]	0.693	0.803			0.689
INRIA+1 Mio [Jégou *et al.*, 2008; inr]	0.315	0.423			
Kentucky [Nistér and Stewénius, 2006; nis]	3.5	3.67			

Table 6.1: mAP for different datasets compared to results of state of the art results.

Dataset	Oxford5k	Oxford105k	Paris	INRIA	Kentucky
Memory [GiB]	0.16	2.35	0.13	22.35	0.23
Avg. time [ms]	5	6	8	30	4

Table 6.2: Additional memory overhead per dataset and average time overhead per query.

The combination of both methods yields in all datasets to superior results over the baseline. As the performance decays slowly, k is not as dataset specific as it might seem. Setting it to somewhere between 20 and 40 gives good results for real world datasets used in image retrieval applications. As can be seen in Table 6.1 for Oxford5k, Oxford105k and Paris we compete with the state of the art, however we do so without exploiting lower level information. For the Kentucky dataset we miss the state of the art only by 0.01 of top-4 precision. For the Holidays dataset it is well know that Hamming Embedding and Weak Geometric Consistency Constraint can greatly improve results, further more the $200k$ visual vocabulary is quite small for such a large dataset. We chose to evaluate our method on this challenging dataset to demonstrate that even under very unfavorable conditions we achieve a significant improvement.

Table 6.2 shows an overview over the total memory overhead per dataset and the average query time overhead for each query. As for each image in the database the forward- and the backward ranking lists need to be stored, the memory overhead grows linearly with the database size. This overhead is in the same order of magnitude as for instance Hamming Embedding [Jégou *et al.*, 2008]. The query time overhead is mainly dependent on the length of the backward ranking list and the chosen threshold k as this restricts the size of the *close* and *far set*.

6.6 Conclusion

We have demonstrated that a significant improvement in bag-of-words retrieval can be achieved, without considering the geometric arrangement of features in an image nor by modifying the feature quantization step. Our method uses k-reciprocal nearest neighbors to identify an initial set of highly relevant images in the database which are then used to re-rank the remaining part of the database. On many data sets our approach competes with the state of the art.

The memory overhead of our method is linear in the number of documents while the average query time overhead is neglectable.

As a secondary contribution, we make a binary executable and a C++ implementation of our method available at our homepage[1]. Additionally we publish the pre-calculated visual words for the Oxford5k, Oxford105k, Paris and Kentucky dataset together with an evaluation package to reproduce our results at the same place.

[1] http://www.vision.ee.ethz.ch/datasets/

7

Quantized Kernel Learning For Feature Matching

7.1 Introduction

Matching local visual features is a core problem in computer vision with a vast range of applications such as image registration [Yi *et al.*, 2008], image alignment and stitching [Brown and Lowe, 2007] and structure-from-motion [Agarwal *et al.*, 2011]. To cope with the geometric transformations and photometric distorsions that images exhibit, many robust feature descriptors have been proposed. In particular, histograms of oriented gradients such as SIFT [Lowe, 2004] have proved successful in many of the above tasks. Despite these results, they are inherently limited by their design choices. Hence, we have witnessed an increasing amount of work focusing on automatically learning visual descriptors from data via discriminative embeddings [Hua *et al.*, 2007; Boix *et al.*, 2013] or hyper-parameter optimization [Brown *et al.*, 2011; Simonyan *et al.*, 2014; Trzcinski *et al.*, 2012, 2013].

A dual aspect of visual description is the measure of visual (dis-)similarity, which is responsible for deciding whether a pair of features matches or not. In image registration, retrieval and 3D reconstruction, for instance, nearest neighbor search builds on such measures to establish point correspondences. Thus, the choice of similarity or kernel impacts the performance of a system as much as the choice of visual features [Arandjelovic and Zisserman, 2012; Maji *et al.*, 2013; Perronnin *et al.*, 2010]. Designing a good similarity measure for matching is difficult and commonly used kernels such as the linear, intersection, χ^2 and RBF kernels are not ideal as their inherent properties (*e.g.*, stationarity, homogeneity) may not fit the data well.

Existing techniques for automatically learning similarity measures suffer from different limitations. Metric learning approaches [Weinberger *et al.*, 2006] learn to project the data to a lower-dimensional and more discriminative space where the Euclidean geometry can be used. However, these methods are inherently linear. Multiple Kernel Learning (MKL) [Bach *et al.*, 2004] is able to combine multiple base kernels in an optimal way, but its complexity limits the amount of data that can be used and forces the user to pre-select or design a small number of kernels that are likely to perform well. Additionally, the resulting kernel may not be easily represented in a reasonably small Euclidean space. This is problematic, as many efficient algorithms (*e.g.* approximate nearest neighbor techniques) heavily rely on Euclidean geometry and have non-intuitive behavior in higher dimensions.

In this chapter, we introduce a simple yet powerful family of kernels, *Quantized Kernels* (QK), which (a) model non-linearities and heterogeneities in the data, (b) lead to compact representations that can be easily decompressed into a reasonably-sized Euclidean space and (c) are efficient to learn so that large-scale data can be exploited. In essence, we build on the fact that vector quantizers project data into a finite set of N elements, the *index space*, and on the simple observation that kernels on finite sets are fully specified by the $N \times N$ Gram matrix of these elements (the *kernel matrix*), which we propose to learn directly. Thus, QKs are piecewise constant but otherwise arbitrary, making them very flexible. Since the learnt kernel matrices are positive semi-definite, we directly obtain the corresponding explicit feature mappings and exploit their potential low-rankness.

In the remainder of the chapter, we first further discuss related work (Sec. 7.2), then present QKs in detail (Sec. 7.3). As important contributions, we show how to efficiently learn the quantizer and the kernel matrix so as to maximize the matching performance (Sec. 7.3.2), using an exact linear-time inference subroutine (Sec. 7.3.3), and devise practical techniques for users to incorporate knowledge about the structure of the data (Sec. 7.3.4) and reduce the number of parameters of the system. Our experiments in Sec. 7.4 show that our kernels yield state-of-the-art performance on a standard feature matching benchmark and improve over kernels used in the literature for several descriptors, including one based on metric learning. Our compressed features are very compact, using only 1 to 4 bits per dimension of the original features. For instance, on SIFT descriptors, our QK yields about 10% improvement on matching compared to the dot product, while compressing features by a factor 8.

7.2 Related work

Our work relates to a vast literature on kernel selection and tuning, descriptor, similarity, distance and kernel learning. We present a selection of such works below.

Basic kernels and kernel tuning. A common approach for choosing a kernel is to pick one from the literature: dot product, Gaussian RBF, intersection [Maji *et al.*, 2013], χ^2, Hellinger, etc. These generic kernels have been extensively studied [Vedaldi and Zisserman, 2012] and have properties such as homogeneity or stationarity. These properties may be inadequate for the data of interest and thus the kernels will not yield optimal performance. Efficient yet approximate versions of such kernels [Gong *et al.*, 2012; Scholkopf, 2002; Vedaldi and Zisserman, 2012] are similarly inadequate.

Descriptor learning. Early work on descriptor learning improved SIFT by exploring its parameter space [Winder and Brown, 2007]. Later, automatic parameter selection was proposed with a non-convex objective [Brown *et al.*, 2011]. Recently, significant improvements in local description for matching have been obtained by optimizing feature encoding [Boix *et al.*, 2013] and descriptor pooling [Simonyan *et al.*, 2014; Trzcinski *et al.*, 2012]. These works maximize the matching performance directly via convex optimization [Simonyan *et al.*, 2014] or boosting [Trzcinski *et al.*, 2012]. As we show in our experiments, our approach improves matching even for such optimized descriptors.

Distance, similarity and kernel learning. Mahalanobis metrics (*e.g.* Weinberger *et al.* [2006]) are probably the most widely used family of (dis-)similarities in supervised settings. They extend the Euclidean metric by accounting for correlations between input dimensions and are equivalent to projecting data to a new, potentially smaller, Euclidean space. Learning the projection improves discrimination and compresses feature vectors, but the projection is inherently linear.[1] There are several attempts to learn more powerful non-linear kernels from data. Multiple Kernel Learning (MKL) [Bach *et al.*, 2004] operates on a parametric family of kernels: it learns a convex combination of a few base kernels so as to maximize classification accuracy. Recent advances now allow to combine thousands of kernels in MKL [Orabona and Jie, 2011] or exploit specialized families of kernels to derive faster algorithms [Roig *et al.*, 2013].

[1] Metric learning can be kernelized, but then one has to choose the kernel.

In that work, the authors combine binary base kernels based on randomized indicator functions but restricted them to XNOR-like kernels. Our QK framework can also be seen as an efficient and robust MKL on a specific family of binary base kernels. However, our binary base kernels originate from more general quantizations: they correspond to their regions of constantness. As a consequence, the resulting optimization problem is also more involved and thus calls for approximate solutions.

In parallel to MKL approaches, Non-Parametric Kernel Learning (NPKL) [Hoi *et al.*, 2007] has emerged as a flexible kernel learning alternative. Without any assumption on the form of the kernel, these methods aim at learning the Gram matrix of the data directly. The optimization problem is a semi-definite program whose size is quadratic in the number of samples. Scalability is therefore an issue, and approximation techniques must be used to compute the kernel on unobserved data. Like NPKL, we learn the values of the kernel matrix directly. However, we do it in the index space instead of the original space. Hence, we restrict our family of kernels to piecewise constant ones[2], but, contrary to NPKL, the complexity of the problems we solve does not grow with the number of data points but with the refinement of the quantization and our kernels trivially generalize to unobserved inputs.

7.3 Quantized kernels

In this section, we present the framework of quantized kernels (QK). We start in Sec. 7.3.1 by defining QKs and looking at some of their properties. We then present in Sec. 7.3.2 a general alternating learning algorithm. A key step is to optimize the quantizer itself. We present in Sec. 7.3.3 our scheme for quantization optimization for a single dimensional feature and how to generalize it to higher dimensions in Sec. 7.3.4.

7.3.1 Definition and properties

Formally, *quantized kernels* $\mathcal{QK}_N^D$ are the set of kernels k_q on $\mathbb{R}^D \times \mathbb{R}^D$ such that:

$$\begin{aligned} \exists q : \mathbb{R}^D \mapsto \{1, \dots, N\}, \quad \exists \mathbf{K} \in \mathbb{R}^{N \times N} \succeq 0, \\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^D, \quad k_q(\mathbf{x}, \mathbf{y}) = \mathbf{K}(q(\mathbf{x}), q(\mathbf{y})), \end{aligned} \tag{7.1}$$

[2] As any continuous function on an interval is the uniform limit of a series of piecewise constant functions, this assumption does not inherently limit the flexibility of the family.

where q is a quantization function which projects $\mathbf{x} \in \mathbb{R}^D$ to the finite index space $\{1, \ldots, N\}$, and $\mathbf{K} \succeq 0$ denotes that $\mathbf{K}$ is a positive semi-definite (PSD) matrix. As discussed above, quantized kernels are an efficient parametrization of piecewise constant functions, where q defines the regions of constantness. Moreover, the $N \times N$ matrix $\mathbf{K}$ is unique for a given choice of k_q, as it simply accounts for the $N(N+1)/2$ possible values of the kernel and is the Gram matrix of the N elements of the index space. We can also see q as a 1-of-N coding feature map φ_q, such that:

$$k_q(\mathbf{x}, \mathbf{y}) = \mathbf{K}(q(\mathbf{x}), q(\mathbf{y})) = \varphi_q(\mathbf{x})^\top \mathbf{K} \varphi_q(\mathbf{y}). \tag{7.2}$$

The components of the matrix $\mathbf{K}$ fully parametrize the family of quantized kernels based on q, and it is a PSD matrix if and only if k_q is a PSD kernel. An explicit feature mapping of k_q is easily computed from the Cholesky decomposition of the PSD matrix $\mathbf{K} = \mathbf{P}^\top \mathbf{P}$:

$$k_q(\mathbf{x}, \mathbf{y}) = \varphi_q(\mathbf{x})^\top \mathbf{K} \varphi_q(\mathbf{y}) = \left\langle \psi_q^{\mathbf{P}}(\mathbf{x}), \psi_q^{\mathbf{P}}(\mathbf{y}) \right\rangle, \tag{7.3}$$

where $\psi_q^{\mathbf{P}}(\mathbf{x}) = \mathbf{P} \varphi_q(\mathbf{x})$. It is of particular interest to limit the rank $N' \leq N$ of $\mathbf{K}$, and hence the number of rows in $\mathbf{P}$. In their *compressed form*, vectors require only $\log_2(N)$ bits of memory for storing $q(\mathbf{x})$ and they can be decompressed in $\mathbb{R}^{N'}$ using $\mathbf{P} \varphi_q(\mathbf{x})$. Not only is this decompressed vector smaller than one based on φ_q, but it is also associated with the Euclidean geometry rather than the kernel one. This allows the exploitation of the large literature of efficient methods specialized to Euclidean spaces.

7.3.2 Learning quantized kernels

In this section, we describe a general alternating algorithm to learn a quantized kernel k_q for feature matching. This problem can be formulated as quadruple-wise constraints of the following form:

$$k_q(\mathbf{x}, \mathbf{y}) > k_q(\mathbf{u}, \mathbf{v}), \qquad \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{P}, \quad \forall (\mathbf{u}, \mathbf{v}) \in \mathcal{N}, \tag{7.4}$$

where $\mathcal{P}$ denotes the set of *positive* feature pairs, and $\mathcal{N}$ is the *negative* one. The positive set contains feature pairs that should be visually matched, while the negative pairs are mismatches.

We adopt a large-margin formulation of the above constraints using the trace-norm regularization $\|\cdot\|_*$ on $\mathbf{K}$, which is the tightest convex surrogate to low-

rank regularization [Fazel, 2002]. Using M training pairs $\{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1\ldots M}$, we obtain the following optimization problem:

$$\underset{\mathbf{K}\succeq 0,\ q\in\mathcal{Q}_N^D}{\operatorname{argmin}} \quad E(\mathbf{K}, q) = \frac{\lambda}{2}\|\mathbf{K}\|_* + \sum_{j=1}^{M} \max\left(0, 1 - l_j \varphi_q(\mathbf{x}_j)^\top \mathbf{K} \varphi_q(\mathbf{y}_j)\right), \quad (7.5)$$

where $\mathcal{Q}_N^D$ denotes the set of quantizers $q : \mathbb{R}^D \mapsto \{1, \ldots, N\}$, the pair label $l_j \in \{-1, 1\}$ denotes whether the feature pair $(\mathbf{x}_j, \mathbf{y}_j)$ is in $\mathcal{N}$ or $\mathcal{P}$ respectively. The parameter λ controls the trade-off between the regularization and the empirical loss. Solving Eq. (7.5) directly is intractable. We thus propose to alternate between the optimization of $\mathbf{K}$ and q. We describe the former below, and the latter in the next section.

Optimizing $\mathbf{K}$ with fixed q. When fixing q in Eq. (7.5), the objective function becomes convex in $\mathbf{K}$ but is not differentiable, so we resort to stochastic sub-gradient descent for optimization. Similar to Simonyan *et al.* [2014], we used *Regularised Dual Averaging* (RDA) [Xiao and others, 2010] to optimize $\mathbf{K}$ iteratively. At iteration $t+1$, the kernel matrix $\mathbf{K}_{t+1}$ is updated with the following rule:

$$\mathbf{K}_{t+1} = \Pi\left(-\frac{\sqrt{t}}{\gamma}\left(\overline{\mathbf{G}_t} + \lambda\mathbf{I}\right)\right) \quad (7.6)$$

where $\gamma > 0$ and $\overline{\mathbf{G}_t} = \frac{1}{t}\sum_{t'=1}^{t} \mathbf{G}_{t'}$ is the rolling average of subgradients $\mathbf{G}_{t'}$ of the loss computed at step t' from one sample pair. $\mathbf{I}$ is the identity matrix and Π is the projection onto the PSD cone.

7.3.3 Interval quantization optimization for a single dimension

To optimize an objective like Eq. (7.5) when $\mathbf{K}$ is fixed, we must consider how to design and parametrize the elements of $\mathcal{Q}_N^D$. In this work, we adopt *interval quantizers*, and in this section we assume $D=1$, *i.e.*, restrict the study of quantization to $\mathbb{R}$.

Interval quantizers. An interval quantizer q over $\mathbb{R}$ is defined by a set of $N+1$ boundaries $b_i \in \mathbb{R}$ with $b_0 = -\infty$, $b_N = \infty$ and $q(x) = i$ if and only if $b_{i-1} < x \leq b_i$. Importantly, interval quantizers are monotonous, $x \leq y \Rightarrow q(x) \leq q(y)$, and boundaries b_i can be set to any value between $\max_{q(x)=i} x$ (included) and $\min_{q(x)=i+1} x$ (excluded). Therefore, Eq. (7.5) can be viewed

as a data labelling problem, where each value x_j or y_j takes a label in $[1, N]$, with a monotonicity constraint.

Thus, let us now consider the graph $(\mathcal{V}, \mathcal{E})$ where nodes $\mathcal{V} = \{v_t\}_{t=1\ldots 2M}$ represent the list of all x_j and y_j in a sorted order and the edges $\mathcal{E} = \{(v_s, v_t)\}$ connect all pairs (x_j, y_j). Then Eq. (7.5) with fixed $\mathbf{K}$ is equivalent to the following discrete pairwise energy minimization problem:

$$\underset{\mathbf{q}\in[1,N]^{2M}}{\operatorname{argmin}} \quad E'(\mathbf{q}) = \sum_{(s,t)\in\mathcal{E}} E_{st}(q(v_s), q(v_t)) + \sum_{t=2}^{2M} C_t(q(v_{t-1}), q(v_t)), \tag{7.7}$$

where $E_{st}(q(v_s), q(v_t)) = E_j(q(x_j), q(y_j)) = \max(0, 1 - l_j \mathbf{K}(q(x_j), q(y_j)))$ and C_t is ∞ for $q(v_t) < q(v_{t-1})$ and 0 otherwise (*i.e.*, it encodes the monotonicity of q in the sorted list of v_t).

The optimization of Eq. (7.7) is an NP-hard problem as the energies E_{st} are arbitrary and the graph does not have a bounded treewidth, in general. Hence, we iterate the individual optimization of each of the boundaries using an exact linear-time algorithm, which we present below.

Exact linear-time optimization of a binary interval quantizer. We now consider solving equations of the form of Eq. (7.7) for the binary label case ($N = 2$). The main observation is that the monotonicity constraint means that labels are 1 until a certain node t and then 2 from node $t+1$, and this switch can occur only once on the entire sequence, where $v_t \leq b_1 < v_{t+1}$. This means that there are only $2M + 1$ possible labellings and we can order them from $(1, \ldots, 1)$, $(1, \ldots, 1, 2)$ to $(2, \ldots, 2)$. A naïve algorithm consists in computing the $2M+1$ energies explicitly. Since each energy computation is linear in the number of edges, this results in a quadratic complexity overall.

A linear-time algorithm exists. It stems from the observation that the energies of two consecutive labellings (*e.g.*, switching the label of v_t from 1 to 2) differ only by a constant number of terms:

$$\begin{aligned} E(q(v_{t-1})=1, q(v_t)=2, q(v_{t+1})=2) &= E(q(v_{t-1})=1, q(v_t)=1, q(v_{t+1})=2) \\ +C_t(1,2) - C_t(1,1) + C_{t+1}(2,2) - C_{t+1}(1,2) &+ E_{st}(q(v_s), 2) - E_{st}(q(v_s), 1) \end{aligned} \tag{7.8}$$

where, w.l.o.g., we have assumed $(s, t) \in \mathcal{E}$. After finding the optimal labelling, *i.e.* finding the label change (v_t, v_{t+1}), we set $b_1 = (v_t + v_{t+1})/2$ to obtain the best possible generalization.

Finite spaces. When the input feature space has a finite number of different values (*e.g.*, $x \in [1, T]$), then we can use linear-time sorting and merge all nodes with equal value in Eq. (7.7): this results in considering at most $T + 1$ labellings, which is potentially much smaller than $2M + 1$.

Extension to the multilabel case. Optimizing a single boundary b_i of a multilabel interval quantization is essentially the same binary problem as above, where we limit the optimization to the values currently assigned to i and $i + 1$ and keep the other assignments $\overline{q}$ fixed. We use unaries $E_j(q(x_j), \overline{q}(y_j))$ or $E_j(\overline{q}(x_j), q(y_j))$ to model half-fixed pairs for x_j or y_j, respectively.

7.3.4 Learning higher dimensional quantized kernels

We now want to generalize interval quantizers to higher dimensions. This is readily feasible via product quantization [Jégou *et al.*, 2011b], using interval quantizers for each individual dimension.

Interval product quantization. An interval product quantizer $q(\mathbf{x}) : \mathbb{R}^D \mapsto \{1, \ldots, N\}$ is of the form $q(\mathbf{x}) = (q_1(\mathbf{x}_1), \ldots, q_D(\mathbf{x}_D))$, where $q_1, \ldots, q_D$ are interval quantizers with $N_1, \ldots, N_D$ bins respectively, *i.e.*, $N = \prod_{d=1}^{D} N_d$. The learning algorithm devised above trivially generalizes to interval product quantization by fixing all but one boundary of a single component quantizer q_d. However, learning $\mathbf{K} \in \mathbb{R}^N \times \mathbb{R}^N$ when N is very large becomes problematic: not only does RDA scale unfavourably, but the lack of training data will eventually lead to severe overfitting. To address these issues, we devise below variants of QKs that have practical advantages for robust learning.

Additive quantized kernels (AQK). We can drastically reduce the number of parameters by restricting product quantized kernels to additive ones, which consists in decomposing over dimensions:

$$k_q(\mathbf{x}, \mathbf{y}) = \sum_{d=1}^{D} k_{q_d}(\mathbf{x}_d, \mathbf{y}_d) = \sum_{d=1}^{D} \varphi_{q_d}(\mathbf{x}_d)^\top \mathbf{K}_d \varphi_{q_d}(\mathbf{y}_d) = \varphi_q(\mathbf{x})^\top \mathbf{K} \varphi_q(\mathbf{y}), \tag{7.9}$$

where $q_d \in \mathcal{Q}^1_{N_d}$, φ_{q_d} is the 1-of-N_d coding of dimension d, $\mathbf{K}_d$ is the $N_d \times N_d$ Gram matrix of dimension d, φ_q is the concatenation of the D mappings φ_{q_d}, and $\mathbf{K}$ is a $(\sum_d N_d) \times (\sum_d N_d)$ block-diagonal matrix of $\mathbf{K}_1, \ldots, \mathbf{K}_D$. The benefits of AQK are twofold. First, the explicit feature space is reduced from

$N = \prod_d N_d$ to $N' = \sum_d N_d$. Second, the number of parameters to learn in $\mathbf{K}$ is now only $\sum_d N_d^2$ instead of N^2. The compression ratio is unchanged since $\log_2(N) = \sum_d \log_2(N_d)$. To learn $\mathbf{K}$ in Eq. (7.9), we simply set the off-block-diagonal elements of $G_{t'}$ to zero in each iteration, and iteratively update $\mathbf{K}$ as describe in Sec. 7.3.2. To optimize a product quantizer, we iterate the optimization of each 1d quantizer q_d following Sec. 7.3.3, while fixing q_c for $c \neq d$. This leads to using the following energy E_j for a pair $(\mathbf{x}_j, \mathbf{y}_j)$:

$$E_{j,d}(q_d(\mathbf{x}_{j,d}), q_d(\mathbf{y}_{j,d})) = \max\left(0, \mu_{j,d} - l_j \mathbf{K}_d(q_d(\mathbf{x}_{j,d}), q_d(\mathbf{y}_{j,d}))\right), \tag{7.10}$$

where $\mu_{j,d} = 1 - l_j \sum_{c \neq d} \mathbf{K}_c(q_c(\mathbf{x}_c), q_c(\mathbf{y}_c))$ acts as an adaptive margin.

Block quantized kernels (BQK). Although the additive assumption in AQK greatly reduces the number of parameters, it is also very restrictive, as it assumes independent data dimensions. A simple way to extend additive quantized kernels to model the inter-dependencies of dimensions is to allow the off-diagonal elements of $\mathbf{K}$ in Eq. (7.9) to be nonzero. As a trade-off between a block-diagonal (AQK) and a full matrix, in this work we also consider the grouping of the feature dimensions into B blocks, and only learn off-block-diagonal elements within each block, leading to Block Quantized Kernels (BQK). In this way, assuming $\forall d \quad N_d = n$, the number of parameters in $\mathbf{K}$ is B times smaller than for the full matrix. As a matter of fact, many features such as SIFT descriptors exhibit block structure. SIFT is composed of a 4×4 grid of 8 orientation bins. Components within the same spatial cell correlate more strongly than others and, thus, only modeling those jointly may prove sufficient. The optimization of $\mathbf{K}$ and q are straightforwardly adapted from the AQK case.

Additional parameter sharing. Commonly, the different dimensions of a descriptor are generated by the same procedure and hence share similar properties. This results in block matrices $\mathbf{K}_1, \ldots, \mathbf{K}_D$ in AQK that are quite similar as well. We propose to exploit this observation and share the kernel matrix for groups of dimensions, further reducing the number of parameters. Specifically, we cluster dimensions based on their variances into G equally sized groups and use a single block matrix for each group. During optimization, dimensions sharing the same block matrix can conveniently be merged, *i.e.* $\varphi_q(\mathbf{x}) = [\sum_{d \text{ s.t. } \mathbf{K}_d = \mathbf{K}'_1} \varphi_{q_d}(\mathbf{x}_d), \ldots, \sum_{d \text{ s.t. } \mathbf{K}_d = \mathbf{K}'_G} \varphi_{q_d}(\mathbf{x}_d)]$, and then $\mathbf{K} = \mathrm{diag}(\mathbf{K}'_1, \ldots, \mathbf{K}'_G)$ is learnt following the procedure already described

	Initial	Optimized
Uniform	24.84	21.68
Adaptive	25.99	25.70
Adaptive+	14.62	14.29

Table 7.1: Impact of quantization optimization for different quantization strategies

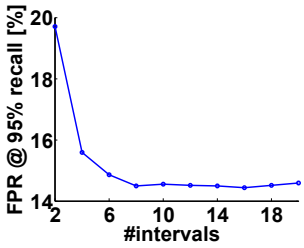


Figure 7.1: Impact of N, the number of quantization intervals

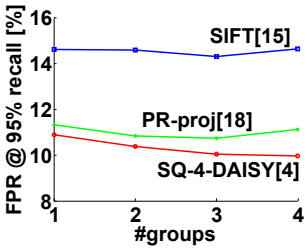


Figure 7.2: Impact of G, the number of dimension groups

for AQK. Notably, the quantizers themselves are not shared, so the kernel still adapts uniquely to every dimension of the data, and the optimization of quantizers is not changed either. This parameter sharing strategy can be readily applied to BQK as well.

7.4 Results

We now present our experimental results, starting with a description of our protocol. We then explore parameters and properties of our kernels (optimization of quantizers, explicit feature maps). Finally, we compare to the state-of-the-art in performance and compactness.

Dataset and evaluation protocol. We evaluate our method using the dataset of Brown *et al.* [2011]. It contains three sets of patches extracted from Liberty, Notre Dame and Yosemite using the Difference of Gaussians (DoG) interest point detector. The patches are rectified with respect to the scale and dominant orientation, and pairwise correspondences are computed using a multi-view stereo algorithm. In our experiments, we use the standard evaluation protocol [Brown *et al.*, 2011] and state-of-the-art descriptors: SIFT [Lowe,

2004], PR-proj [Simonyan *et al.*, 2014] and SQ-4-DAISY [Boix *et al.*, 2013]. M=500k feature pairs are used for training on each dataset, with as many positives as negatives. We report the false positive rate (FPR) at 95% recall on the test set of 100k pairs. A challenge for this dataset is the bias in local patch appearance for each set, so a key factor for performance is the ability to generalize and adapt across sets.

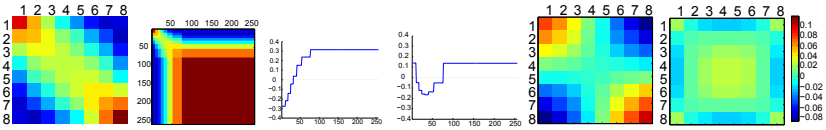
Below, in absence of other mention, AQKs are trained for SIFT on Yosemite and tested on Liberty.

Interval quantization and optimization. We first study the influence of initialization and optimization on the generalization ability of the interval quantizers. For initialization, we have used two different schemes: a) *Uniform quantization*, *i.e.* the quantization with equal intervals; b) *Adaptive quantization*, *i.e.* the quantization with intervals with equal number of samples. In both cases, it allows to learn a first kernel matrix, and we can then iterate with boundary optimization (Sec. 7.3.3). Typically, convergence is very fast (2-3 iterations) and takes less than 5 minutes in total (*i.e.*, about 2s per feature dimension) with 1M nodes. We see in Table 7.1 that uniform binning outperforms the adaptive one and that further optimization benefits the uniform case more. This may seem paradoxical at first, but this is due to the train/test bias problem: intervals with equal number of samples are very different across sets, so refinements will not transfer well. Hence, following Dean *et al.* [2013], we first normalize the features with respect to their rank, separately for the training and test sets. We refer to this process as *Adaptive+*. As Table 7.1 shows, not only does it bring a significant improvement, but further optimization of the quantization boundaries is more beneficial than for the Adaptive case. In the following, we thus adopt this strategy.

Number of quantization intervals. In Fig. 7.1, we show the impact of the number of intervals N of the quantizer on the matching accuracy, using a single shared kernel submatrix ($G = 1$). This number balances the flexibility of the model and its compression ratio. As we can see, using too few intervals limits the performance of QK, and using too many eventually leads to overfitting. The best performance for SIFT is obtained with between 8 and 16 intervals.

Descriptor	Kernel	Dimensionality	Train on Yosemite		Train on Notredame		Mean
			Notredame	Liberty	Yosemite	Liberty	
SIFT [Lowe, 2004]	Euclidean	128	24.02	31.34	27.96	31.34	28.66
SIFT [Lowe, 2004]	χ^2	128	17.65	22.84	23.50	22.84	21.71
SIFT [Lowe, 2004]	AQK(8)	128	10.72	16.90	10.72	16.85	13.80
SIFT [Lowe, 2004]	AQK(8)	256	9.26	14.48	10.16	14.43	12.08
SIFT [Lowe, 2004]	BQK(8)	256	**8.05**	**13.31**	**9.88**	**13.16**	**11.10**
SQ-4-DAISY [Boix *et al.*, 2013]	Euclidean	1360	10.08	16.90	10.47	16.90	13.58
SQ-4-DAISY [Boix *et al.*, 2013]	χ^2	1360	10.61	16.25	12.19	16.25	13.82
SQ-4-DAISY [Boix *et al.*, 2013]	SQ [Boix *et al.*, 2013]	1360	8.42	15.58	9.25	15.58	12.21
SQ-4-DAISY [Boix *et al.*, 2013]	AQK(8)	$\leq$1813	**4.96**	**9.41**	**5.60**	**9.77**	**7.43**
PR-proj [Simonyan *et al.*, 2014]	Euclidean [Simonyan *et al.*, 2014]	$<$64	7.11	14.82	10.54	12.88	11.34
PR-proj [Simonyan *et al.*, 2014]	AQK(16)	$\leq$102	**5.41**	**10.90**	**7.65**	**10.54**	**8.63**

Table 7.2: Performance of kernels on different datasets with different descriptors. AQK(N) denotes the additive quantized kernel with N quantization intervals. Following [Brown and Lowe, 2007], we report the False positive rate (%) at 95% recall. The best results for each descriptor are in bold.



(a) (b) (c) (d) (e) (f)

Figure 7.3: Our learned feature maps and additive quantized kernel of a single dimension. (a) shows the quantized kernel in index space, while (b) is in the original feature space for the first quantizer. (c,d) show the two corresponding feature maps, and (e,f) the related rank-1 kernels.

Explicit feature maps. Fig. 7.3a shows the additive quantized kernel learnt for SIFT with $N=8$ and $G=1$. Interestingly, the kernel has negative values far from the diagonal and positive values near the diagonal. This is typical of stationary kernels: when both features have similar values, they contribute more to the similarity. However, contrary to stationary kernels, diagonal elements are far from being constant. There is a mode on small values and another one on large ones. The second one is stronger: *i.e.*, the co-occurrence of large values yields greater similarity. This is consistent with the voting nature of SIFT descriptors, where strong feature presences are both rarer and more informative than their absences. The negative values far from the diagonal actually penalize inconsistent observations, thus confirming existing results [Jégou and Chum, 2012]. Looking at the values in the original space in Fig. 7.3b, we see that the quantizer has learnt that fine intervals are needed in the lower values, while larger ones are enough for larger values. This is consistent with previous observations that the contribution of large values in SIFT should not grow proportionally [Arandjelovic and Zisserman, 2012; Perronnin *et al.*, 2010; Jégou *et al.*, 2010b].

In this experiment, the learnt kernel has rank 2. We show in Fig. 7.3c, 7.3d, 7.3e and 7.3f the corresponding feature mappings and their associated rank 1 kernels. The map for the largest eigenvalue (Fig. 7.3c) is monotonous but starts with negative values. This impacts dot product significantly, and accounts for the above observation that negative similarities occur when inputs disagree. This rank 1 kernel cannot allot enough contribution to similar mid-range values. This is compensated by the second rank (Fig. 7.3f).

Number of groups. Fig. 7.2 now shows the influence of the number of groups G on performance, for the three different descriptors ($N=8$ for SIFT and SQ-4-DAISY, $N = 16$ for PR-proj). As for intervals, using more groups adds flexibility to the model, but as less data is available to learn each parameter, over-fitting will hurt performance. We choose $G = 3$ for the rest of the experiments.

Comparison to the state of the art. Table 7.2 reports the matching performance of different kernels using different descriptors, for all sets, as well as the dimensionality of the corresponding explicit feature maps. For all three descriptors and on all sets, our quantized kernels significantly and consistently outperform the best reported result in the literature. Indeed, AQK improves the mean error rate at 95% recall from 28.66% to 12.08% for SIFT, from 13.58% to 7.43% for SQ-4-DAISY and from 11.34% to 8.63% for PR-proj compared to the Euclidean distance, and about as much for the χ^2 kernel. Note that PR-proj already integrates metric learning in its design (Simonyan *et al.* [2014] thus recommend using the Euclidean distance): as a consequence our experiments show that modelling non-linearities can bring significant improvements. When comparing to *sparse quantization* (SQ) with hamming distance as done in Boix *et al.* [2013], the error is significantly reduced from 12.21% to 7.43%. This is a notable achievement considering that Boix *et al.* [2013] is the previous state of the art.

The SIFT descriptor has a grid block design which makes it particularly suited for the use of BQK. Hence, we also evaluated our BQK variant for that descriptor. With BQK(8), we observed a relative improvement of 8%, from 12.08% for AQK(8) to 11.1%.

We provide in Fig. 7.4 the ROC curves for the three descriptors when training on Yosemite and testing on Notre Dame and Liberty. These figures show that the improvement in recall is consistent over the full range of false positive rates. For further comparisons, our data and code are available online.[3]

Compactness of our kernels. In many applications of feature matching, the compactness of the descriptor is important. In Table 7.3, we compare to other methods by grouping them according to their memory footprint. As a reference, the best method reported in Table 7.2 (AQK(8) on SQ-4-DAISY) uses 4080 bits per descriptor. As expected, error rates increase as fewer bits are

[3] See: http://www.vision.ee.ethz.ch/~qind/QuantizedKernel.html

used, the original features being significantly altered. Notably, QKs consistently yield the best performance in all groups. Even with a crude binary quantization of SQ-4-DAISY, our quantized kernel outperforms the state-of-the-art SQ of Boix *et al.* [2013] by 3 to 4%. When considering the most compact encodings ($\leq$ 64 bits), our AQK(2) does not improve over BinBoost [Trzcinski *et al.*, 2013], a descriptor designed for extreme compactness, or the product quantization (PQ) [Jégou *et al.*, 2011b] encoding as used in Simonyan *et al.* [2014]. This is because our current framework does not yet allow for joint compression of multiple dimensions. Hence, it is unable to use less than 1 bit per original dimension, and is not optimal in that case. To better understand the potential benefits of decorrelating features and joint compression in future work, we pre-processed the data with PCA, projecting to 32 dimensions and then using AQK(4). This simple procedure obtained state-of-the-art performance with 15% error rate, now outperforming Trzcinski *et al.* [2013] and Simonyan *et al.* [2014].

Although QKs yield very compact descriptors and achieve the best performance across many experimental setups, the computation of similarity values is slower than for competitors: in the binary case, we double the complexity of hamming distance for the 2×2 table look-up.

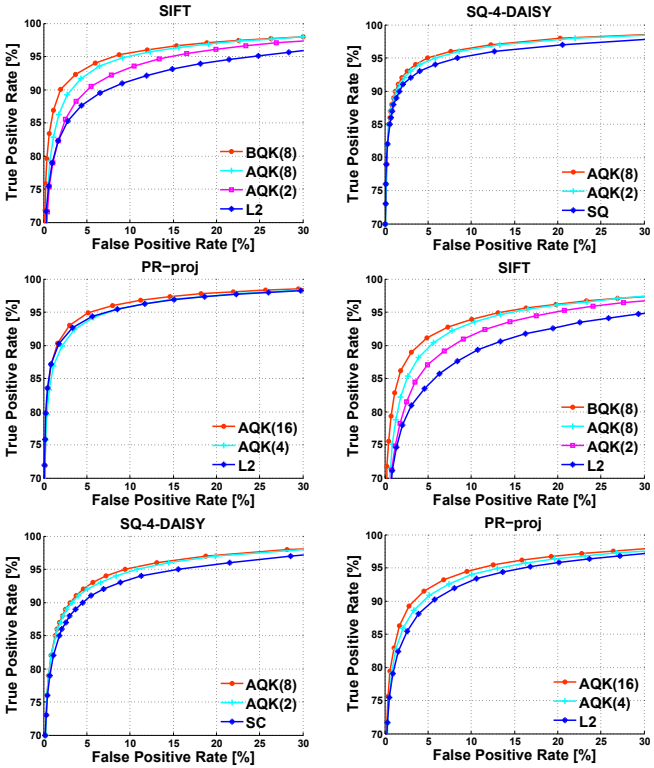


Figure 7.4: ROC curves when evaluating Notre Dame (top) and Liberty (bottom) from Yosemite

Descriptor	Encoding	Memory (bits)	Train on Yosemite		Train on Notredame		Mean
			Notredame	Liberty	Yosemite	Liberty	
SQ-4-DAISY [Boix *et al.*, 2013]	SQ [Boix *et al.*, 2013]	1360	8.42	15.58	9.25	15.58	12.21
SQ-4-DAISY [Boix *et al.*, 2013]	AQK(2)	1360	**5.86**	**10.81**	**6.36**	**10.94**	**8.49**
SIFT [Lowe, 2004]	AQK(8)	384	9.26	14.48	10.16	14.43	12.08
PR-proj [Simonyan *et al.*, 2014]	Bin [Simonyan *et al.*, 2014]	1024	7.09	15.15	8.5	12.16	10.73
PR-proj [Simonyan *et al.*, 2014]	AQK(16)	<256	**5.41**	**10.90**	**7.65**	**10.54**	**8.63**
SIFT [Lowe, 2004]	AQK(2)	128	14.62	19.72	15.65	19.45	17.36
PR-proj [Simonyan *et al.*, 2014]	Bin [Simonyan *et al.*, 2014]	128	10.00	18.64	13.41	16.39	14.61
PR-proj [Simonyan *et al.*, 2014]	AQK(4)	<128	**7.18**	**13.02**	**10.29**	**13.18**	**10.92**
BinBoost [Trzcinski *et al.*, 2013]	BinBoost [Trzcinski *et al.*, 2013]	64	14.54	21.67	18.97	20.49	18.92
PR-proj [Simonyan *et al.*, 2014]	AQK(2)	<64	14.80	20.59	19.38	22.24	19.26
PR-proj [Simonyan *et al.*, 2014]	PQ [Simonyan *et al.*, 2014]	64	12.91	20.15	19.32	17.97	17.59
PR-proj [Simonyan *et al.*, 2014]	PCA+AQK(4)	64	**10.74**	**17.46**	**14.44**	**17.60**	**15.06**

Table 7.3: Performance comparison of different compact feature encoding. The number in the table is reported as False positive rate (%) at 95% recall. The best results for each group are in bold.

7.5 Conclusion

In this paper, we have introduced the simple yet powerful family of quantized kernels (QK), and presented an efficient algorithm to learn its parameters, *i.e.* the kernel matrix and the quantization boundaries. Despite their apparent simplicity, QKs have numerous advantages: they are very flexible, can model non-linearities in the data and provide explicit low-dimensional feature mappings that grant access to the Euclidean geometry. Above all, they achieve state-of-the-art performance on the main visual feature matching benchmark. We think that QKs have a lot of potential for further improvements. In future work, we want to explore new learning algorithms to obtain higher compression ratios – *e.g.* by jointly compressing feature dimensions – and find the weight sharing patterns that would further improve the matching performance automatically.

8

Conclusions

8.1 Contributions

In this thesis we have proposed several methods for improving large scale object retrieval.

In Chapter 4, we present a probabilistic framework for the feature to feature similarity for high-dimensional local features such as SIFT. We then propose a query adaptive feature to feature distance measurement and derive a global image to image similarity function. Despite the simplicity of this approach, it achieves consistently good results on all evaluated datasets, supporting the validity of our model. Furthermore, it does not require parameter tuning to achieve optimal performance.

In Chapter 5, we have presented a novel framework to directly learn the visual similarity that maximizes the accuracy of an object retrieval system. Our model is very flexible and allows us to seamlessly integrate statistical properties of BoW histograms without modelling them explicitly. In our experiments, we have shown the superiority of our similarities compared to those used in state-of-the-art approaches.

In Chapter 6, we have demonstrated that a significant improvement in bag-of-words retrieval can be achieved, without considering the geometric arrangement of features in an image nor by modifying the feature quantization step. Our method uses k-reciprocal nearest neighbors to identify an initial set of highly relevant images in the database which are then used to re-rank the remaining part of the database. On many data sets our approach competes with the state of the art. The memory overhead of our method is linear in the number of documents while the average query time overhead is neglectable.

In Chapter 7, we have introduced the simple yet powerful family of quantized kernels (QK), and presented an efficient algorithm to learn its parameters, i.e. the kernel matrix and the quantization boundaries. Despite their apparent simplicity, QKs have numerous advantages: they are very flexible, can model non-linearities in the data and provide explicit low-dimensional feature mappings that grant access to the Euclidean geometry. Above all, they achieve state-of-the-art performance on the main visual feature matching benchmark.

8.2 Future Work

The field of computer vision has been experiencing a rapid revolution since the "rebirth" of neural nets. Research on generic object recognition is undoubtedly the driving force behind this change. Thanks to the very powerful learning capabilities of Convolutional Neural Nets (CNN), state-of-the-art image classification systems can now outperform humans, which is a remarkable achievement that few researches would have foreseen just a few years ago. Although the features of CNN learnt from generic object recognition yield excellent performance in many other applications, it has been demonstrated by many researchers that fine tuning neural nets to the target application and dataset can be very beneficial. We believe this also applies to specific instance recognition. In addition, due to the highly non-convex nature of CNNs, many experiments have shown a trend that training CNNs in a sequence of easy to hard examples can lead to improved results. Considering that specific instance recognition is generally an easier problem than generic object recognition, it might be a good idea to train a network using samples from the same particular objects first before moving onto generic object categories. In the following, we will extend the discussion about how to improve image retrieval by exploiting the power of CNN and how to push the current CNN to the next level.

Collection of a large scale specific object dataset Without the availability of large scale generic object recognition dataset such as Image-Net, CNN may continue to rust in a dark corner of vision research as CNN has no way to demonstrate its great learning capacity without anything to learn on. This is also true for image retrieval. Over the years, research into image retrieval has been dominated by methods hand-built by a small number of extremely smart researchers. In the era of big data and huge computational resources, rather

than hand-crafting ever-smarter and more sophisticated rules, it is more important to collect rich data, design a good architecture, and let the data tell and the computer work out the detail. However, there is no such kind of large scale specific object dataset available yet. Therefore, we believe it is time to collect such a dataset to enable supervised learning in image retrieval and push research into object recognition to the next level. This dataset should cover a large variety of objects, contain many instances of the same object, and record the spatial correspondence between different instances of the same object, along with the object segmentation mask.

Train an End-to-end retrieval system A first attempt to train an end-to-end CNN for the retrieval task could be to replace the classification lost of Alex-Net or Vgg-Net by pairwise lost, and use the second last layer as image representation. An alternative approach is to use binary classification lost, that is all relevant images of a particular object against a random collection of images.

Considering that spatial verification shows a huge performance boost in non-CNN based image retrieval systems, learning a representation that can optimize the object co-localization could also be beneficial. Using the parameters of the Hough voting space as prediction output, and pairs of images as input, an end-to-end CNN could be trained to maximize the image registration accuracy. Additionally, the spatial coverage of the object of interest can also be learnt in the same model. Such a network could not only provide precise image representations, but could also be used to predict the location of an unknown object if a set of images containing this object are given, which enables automatic object discovery and localization.

Furthermore, it would be interesting to see whether training a network with image pairs with large spatial overlap and small viewpoint changes first, followed by image pairs with much larger variation, would make a big difference in the final performance. If this does prove to be the case, we can optimize the network parameters by collecting a dataset which encodes structured variations in object instances, such as viewpoint angles, scales, area of occlusion, lighting.

Bibliography

S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building Rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 4, 49, 85

R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2911–2918. IEEE, 2012. 19, 41, 46, 59, 85, 97

R. Arandjelovic and A. Zisserman. All about vlad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013. 22

R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. *arXiv preprint arXiv:1511.07247*, 2015. 23

A. Babenko and V. S. Lempitsky. The inverted multi-index. In *CVPR*, pages 3069–3076, 2012. 32

A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1269–1277, 2015. 23

A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *European Conference on Computer Vision*, pages 584–599. Springer, 2014. 23

F. R. Bach, G. R. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the International Conference on Machine learning*. ACM, 2004. 86, 87

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. 2006. 13, 15

X. Boix, M. Gygli, G. Roig, and L. Van Gool. Sparse quantization for patch description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 15, 85, 87, 95, 96, 98, 99, 101

M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007. xiv, 85, 96

M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1):43–57, 2011. 85, 87, 94

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010. 15

Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193. ACM, 2006. 55

O. Chum and J. Matas. Unsupervised discovery of co-occurrence in sparse high dimensional data. In *CVPR10*, 2010. 19, 31, 67, 81

O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 31, 51, 67, 73, 81

O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, 2008. 49, 74

O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall II: Query expansion revisited. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 889–896. IEEE, 2011. 49, 51, 61, 62, 63

M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004. 19

T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *CVPR*, 2013. 95

J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez. Revisiting the vlad image representation. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 653–656. ACM, 2013. 22

M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, page 19. ACM, 2009. 23

M. Douze, A. Ramisa, and C. Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 745–752. IEEE, 2011. 23

M. Fazel. *Matrix rank minimization with applications*. PhD thesis, 2002. 90

B. Fernando and T. Tuytelaars. Mining multiple queries for image retrieval: On-the-fly learning of an object-specific mid-level representation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2544–2551, 2013. 20

M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Comm. of the ACM*, 1981.

S. Gammeter, L. Bossard, T. Quack, and L. V. Gool. I know what you did last summer: object-level auto-annotation of holiday snaps. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 614–621. IEEE, 2009. 49

S. Gammeter. *Applications and theory of large scale image retrieval and large scale mining*. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 21246, 2013, 2013. 13

K. Gao, Y. Zhang, P. Luo, W. Zhang, J. Xia, and S. Lin. Visual stem mapping and geometric tense coding for augmented visual vocabulary. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3234–3241. IEEE, 2012. 19, 50

B. Girod, V. Chandrasekhar, D. M. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham. Mobile visual search. *Signal Processing Magazine, IEEE*, 28(4):61–76, 2011. 49

Y. Gong, S. Kumar, V. Verma, and S. Lazebnik. Angular quantization-based binary codes for fast similarity search. In *NIPS*, pages 1196–1204, 2012. 87

Z. Guo, L. Zhang, and D. Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, 2010. 15

S. C. Hoi, R. Jin, and M. R. Lyu. Learning nonparametric kernel matrices from pairwise constraints. In *Proceedings of the International Conference on Machine learning*. ACM, 2007. 88

G. Hua, M. Brown, and S. Winder. Discriminant embedding for local image descriptors. In *ICCV 2007*. IEEE, 2007. 85

Y. Hwang, B. Han, and H.-K. Ahn. A fast nearest neighbor search algorithm by nonlinear embedding. In *CVPR*, pages 3053–3060, 2012. 32

`http://lear.inrialpes.fr/~jegou/data.php`. 59, 75, 81

M. Jain, H. Jégou, and P. Gros. Asymmetric Hamming Embedding. In *ACM Multimedia*, Scottsdale, United States, October 2011. QUAERO. 46

H. Jégou and O. Chum. Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. In *Computer Vision–ECCV 2012*, pages 774–787. Springer, 2012. 18, 22, 49, 51, 53, 54, 61, 62, 63, 97

H. Jégou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. 2007. 67, 70

H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV08*, 2008. 18, 20, 26, 30, 51, 59, 66, 68, 69, 75, 81, 82

H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1169–1176. IEEE, 2009. 20, 31, 40, 49, 52, 53, 54, 61, 62, 63, 67, 81

H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 87(3):316–336, February 2010. 45

H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311. IEEE, 2010. 22, 97

H. Jégou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. 32(1):2–11, january 2010. 67, 70, 81

H. Jégou, M. Douze, and C. Schmid. Exploiting descriptor distances for precise image search. Research report, INRIA Rennes, June 2011. xiii, 31, 35, 47

H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33(1):117–128, January 2011. x, 18, 19, 30, 32, 40, 44, 45, 47, 92, 99

H. Jégou, R. Tavenard, M. Douze, and L. Amsaleg. Searching in one billion vectors: re-rank with source coding. In *ICASSP*, Prague Czech Republic, 2011. 32

T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002. 50, 55, 56, 59

J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. In *European Conference on Computer Vision*, pages 748–761. Springer, 2010. 21

D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV99*, 1999.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 13, 15, 29, 49, 85, 94, 96, 101

S. Maji, A. C. Berg, and J. Malik. Efficient classification for additive kernel svms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):66–77, 2013. 85, 87

J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, pages 384–393, September 2002. 13

K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1), 2004. 13, 35

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005. 15

K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65:43–72, 2005. 13

A. Mikulík, M. Perdoch, O. Chum, and J. Matas. Learning a fine vocabulary. In *Computer Vision–ECCV 2010*, pages 1–14. Springer, 2010. 14, 19, 30, 35, 45, 46, 50, 66, 69, 81

http://www.vis.uky.edu/~stewe/ukbench/. 59, 75, 81

D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006. 17, 58, 59, 75, 76, 81

A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, 2001. 22

D. Omercevic, O. Drbohlav, and A. Leonardis. High-dimensional feature matching: employing the concept of meaningful nearest neighbors. In *ICCV*, 2007. 31

F. Orabona and L. Jie. Ultra-fast optimization algorithm for sparse multi kernel learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 249–256, 2011. 87

http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/index.html. 58, 68, 75, 81

http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/index.html. 75, 81

M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *Computer Vision and Pattern Recognition (CVPR) Conference*, 2009. 14, 20, 46, 59, 63

F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, June 2007. 22

F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision (ECCV)*. 2010. 22, 85, 97

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 17, 20, 31, 49, 50, 51, 58, 59, 69, 75, 81

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 18, 30, 49, 51, 66, 69, 75, 81

J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *Computer Vision–ECCV 2010*, pages 677–691. Springer, 2010. 18, 31, 35, 50, 66, 69, 81

D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 777–784. IEEE, 2011. 31, 43, 49, 51, 63

D. Qin, C. Wengert, and L. Van Gool. Query adaptive similarity for large scale object retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1610–1617. IEEE, 2013. 19

T. Quack, V. Ferrari, and L. Van Gool. Video mining with frequent itemset configurations. In *CIVR'06*, 2006. 20

`http://www.vision.ee.ethz.ch/~qind/HelloNeighbor.html`. 44

G. Roig, X. Boix, and L. Van Gool. Random binary mappings for kernel learning and efficient SVM. *arXiv preprint arXiv:1307.5161*, 2013. 87

G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007. 49

D. A. F. M. B. Scholkopf. Sampling techniques for kernel methods. In *NIPS 2001*, volume 1, page 335. MIT Press, 2002. 87

X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3013–3020. IEEE, 2012. 21, 31, 51

K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. 16, 50, 62, 63, 85, 87, 90, 95, 96, 98, 99, 101

J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003. 16, 17, 30, 49, 58

Y. Su and F. Jurie. Visual word disambiguation by semantic contexts. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 311–318. IEEE, 2011. 51

E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, May 2010. 15

G. Tolias, T. Furon, and H. Jégou. Orientation covariant aggregation of local descriptors with embeddings. In *European Conference on Computer Vision*, Zurich, Switzerland, September 2014. 22

T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua. Learning image descriptors with the boosting-trick. In *NIPS*, 2012. 85, 87

T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting binary keypoint descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 85, 99, 101

T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, 3(3):177–280, 2008. 12

A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):480–492, 2012. 87

K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. *Advances in neural information processing systems*, 18:1473, 2006. 86, 87

C. Wengert, M. Douze, and H. Jégou. Bag-of-colors for improved image search. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1437–1440. ACM, 2011. 50

S. A. Winder and M. Brown. Learning local image descriptors. In *CVPR*, 2007. 87

Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 25–32. IEEE, 2009. 51

L. Xiao *et al.* Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(2543-2596):4, 2010. 90

Z. Yi, C. Zhiguo, and X. Yang. Multi-spectral remote image registration based on sift. *Electronics Letters*, 44(2):107–108, 2008. 85

G. Yu and J.-M. Morel. ASIFT: An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line*, 2011, 2011. 35

Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 809–816. IEEE, 2011. 21, 31, 51

S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *Computer Vision–ECCV 2012*, pages 660–673. Springer, 2012. 51

L. Zheng, S. Wang, Z. Liu, and Q. Tian. Lp-norm idf for large scale image search. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1626–1633. IEEE, 2013. 20, 49

L. Zheng, S. Wang, Z. Liu, and Q. Tian. Packing and padding: Coupled multi-index for accurate image retrieval. *arXiv preprint arXiv:1402.2681*, 2014. 50

C.-Z. Zhu, H. Jégou, and S. Satoh. Query-adaptive asymmetrical dissimilarities for visual object retrieval. In *ICCV-International Conference on Computer Vision*, 2013. 49, 52, 53, 54, 61, 62, 63

List of Publications

1. D. Qin, X. Chen, M. Guillaumin, L. Van Gool. Quantized Kernel Learning for Feature Matching. In *Neural Information Processing Systems (NIPS)*, 2014.

2. D. Qin, Y. Chen, M. Guillaumin, L. Van Gool. Learning to Rank Bag-of-Word Histograms for Large-scale Object Retrieval. In *Proceedings British Machine Vision Conference (BMVC)*, 2014.

3. D. Qin, C. Wengert, L. Van Gool. Query Adaptive Similarity for Large Scale Object Retrieval. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

4. D. Qin, S. Gammeter, L Bossard, T. Quack, L. Van Gool. Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.