

Improving Instance Search Performance in Video Collections



Zhenxing Zhang

School of Computing
Dublin City University

Supervisor:
Dr. Cathal Gurrin and Prof. Alan Smeaton

This dissertation is submitted for the degree of
Doctor of Philosophy

January, 2017

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of (Doctor of Philosophy) is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____ ID No.: 56125771 Date: _____

Zhenxing Zhang

January, 2017

Table of contents

List of figures	vii
List of tables	x
Abstract	xiii
1 Introduction	1
1.1 Instance Search	2
1.2 Research Motivation and Challenges	5
1.3 Hypothesis and Research Questions	7
1.4 List of Contributions	9
1.5 Thesis Outline	11
2 Literature Review	12
2.1 Video Search Models	13
2.1.1 Content-based search systems	14
2.1.2 Evaluation and Statistical Testing of Video Search Systems	16
2.2 Instance Search Systems	18
2.2.1 Bag-of-Features Approach	18
2.3 Improvement for Visual Search	23

2.3.1	Improving visual vocabulary	23
2.3.2	Spatial Verification	24
2.3.3	Query Expansion	26
2.3.4	Local Convolutional Features	26
2.4	Conclusion	27
3	Research Methodology	30
3.1	Experiment Configurations	31
3.2	Research Methods	35
3.3	Evaluation Procedure	36
3.3.1	Evaluation Metrics	38
3.3.2	Test Collections	40
3.4	Conclusion	44
4	Weak Geometric Correlation Consistency for Instance Search	45
4.1	Introduction	47
4.1.1	Problem Formulation	48
4.1.2	Related work	49
4.2	Weak Geometric Correlation Consistency	51
4.2.1	Motivation	51
4.2.2	Implementation	53
4.2.3	Computational Complexity	56
4.3	Experiment and Discussion	56
4.3.1	Test Collections	57
4.3.2	Evaluation Protocol	58

4.3.3	System Settings	60
4.3.4	Results and Findings	63
4.4	Conclusion	68
5	Improving Instance Search for Video Collections	70
5.1	Improving Feature Matching with Embedded Coding	72
5.1.1	Weaknesses of quantization-based approaches	72
5.1.2	Motivation	74
5.1.3	Implementation	75
5.2	Scalable Retrieval for Large Video Collections	78
5.2.1	Components of a Scalable Instance Search System	79
5.2.2	Improving recall with query expansion	81
5.3	TRECVID Experiments and Discussion of Results	83
5.3.1	Instance Search Task in TRECVID	83
5.3.2	Evaluation Methodology	84
5.3.3	Experimental Results and Findings	85
5.4	Conclusion	95
6	Instance Search for Video Browsing	97
6.1	Video Browsing Task	100
6.1.1	Related Works	100
6.1.2	Video Browser Showdown	102
6.2	Navigating Video Contents with Similarity Graphs	104
6.2.1	Motivation	104
6.2.2	Video Segmentation and Representation	106

Table of contents

6.2.3	Instance Recognition and Attribute Analysis	107
6.2.4	Similarity Graphs Construction	109
6.2.5	Faceted Navigation for Browsing	111
6.3	Evaluate Instance Search for Video Browsing Task	113
6.3.1	Test collection and Participants	115
6.3.2	Experimental Methodology	116
6.3.3	Results and Findings	120
6.4	Conclusion	124
7	Conclusions	126
7.1	Answers to Research Questions	128
7.2	Future Directions	130
7.2.1	Advanced Feature Representation	130
7.2.2	Learning to rank	131
7.2.3	Improving scalability using cloud computing	132
	References	134
	Appendix A The Applications of Instance Search	143

List of figures

1.1	Query example for instance search task	3
2.1	Main components for content-based video retrieval systems	19
2.2	Local feature extraction example	20
2.3	Inverted indexing structure in Bag of Feature framework	22
2.4	Three main types of spatial transformations between object and its estimate	24
3.1	Standard evaluation pipeline for instance retrieval	37
3.2	Evaluation metrics of ranked retrieval results	38
3.3	Image samples from test collections during our experiments	43
4.1	False local feature matching example	48
4.2	Verifying the consistency of feature matches using geometric correlations .	52
4.3	An illustration of applying WGCC on the initial set of feature matches to obtain consistent feature matches.	54
4.4	Sample images from three test collections	59
4.5	Performance comparison in mAP between different approaches on three test collections	65
4.6	Performance improvement in the precision-recall curve plot	66

4.7	Performance comparison for Top-10 results in ranking lists.	67
5.1	Demonstration of “Quantization Error” and motivation of our method . . .	74
5.2	Precision@N curve for local feature matching	78
5.3	Complete instance search system based on the local feature regions	80
5.4	Architecture of query expansion technology	82
5.5	Performance comparison between Bag of Features and embedded coding algorithms	90
5.6	Performance comparison between system (i) and (ii) in TRECvid 2013 . . .	91
5.7	Performance comparison between system (iii) and (iv) in TRECvid 2014 . .	92
5.8	Performance comparison between system system (v) and (vi) in TRECvid 2015	92
5.9	Query topics confusion introduced by standard query expansion	93
5.10	Performance comparison between individual topics for our participation in TRECvid 2015	95
6.1	Photo of video browser showdown event	103
6.2	Architecture overview of the video search system	106
6.3	Object extraction with DCN models	108
6.4	A 14x14 similarity matrix constructed from object instances	110
6.5	Faceted navigation example	112
6.6	Scoreboard for performance comparison during during the evaluation experi- ment	114
6.7	Browsing topics for video browsing evaluation experiment	117
6.8	Interface for concept filtering tool	120
6.9	Interactive interface for faceted navigation tool	121
6.10	Average performance scores and mean submission time for all the three systems	122

6.11 Comparison of user performance score among all tasks	123
A.1 Instance search application one: video search	144
A.2 Instance search application two: SpongeIT	145
A.3 Instance search application three: Mobile search	146

List of tables

3.1	Overview of the Experiment configurations	32
3.2	Statistics on four test collections	40
4.1	Statistics for the three benchmark collections	58
4.2	Performance comparison in mean Average Precision (mAP) score between three approaches	63
5.1	Basic statistics for BBC EastEnders test collection.	84
5.2	Experimental results for three-year participation in TRECVID	88
6.1	Performance comparison between three video browsing tools	121
6.2	Average performance score [0-100] by each browsing task	123

This thesis is dedicated to my wife and my parents, thanks for their endless support and
always being there for me.

Acknowledgements

I extremely grateful to my wonderful supervisors, Dr. Cathal Gurrin and Prof. Alan Smeaton, for their constant guidance and infinite patience during my time producing this work. Thanks to all the current and past members of Insight Centre for Data Analytics who offered advice and support, who generously shared the truly invaluable experience with me. Many thanks to everyone who made PhD work an enjoyable experience.

Then I would like to thank my parents, Jianguo Zhang and Wenju Xi, who deserve all my appreciation for the years of providing good support and motivation. It is your support that lead me to where I am today. And I would like to thank my wonderful wife Lulu Wang, for her constant love, support and encouragement, which help me to produce this work.

Abstract

This thesis presents methods to improve instance search and enhance user performance while browsing unstructured video collections. Through the use of computer vision and information retrieval techniques, we propose novel solutions to analyse visual content and build a search algorithm to address the challenges of visual instance search, while considering the constraints of practical applications.

Firstly, we investigate methods to improve the effectiveness of instance search systems for finding object instances occurring in unstructured video content. Using the bag of features framework, we propose a novel algorithm to use the geometric correlation information between local features to improve the accuracy of local feature matching, and thus improve the performance of instance search systems without introducing significant computation cost.

Secondly, we consider the scenario that the performance of instance search systems may drop due to the volume of visual content in large video collections. We introduce a search algorithm based on embedded coding to increase the effectiveness and efficiency of instance search systems. We participate in the international video evaluation campaign, TREC Video Retrieval Evaluation, to comparatively evaluate the performance of our proposed methods.

Finally, the exploration and navigation of visual content when browsing large unstructured video collections is considered. We propose methods to address such challenges and build an interactive video browsing tool to improve user performance while seeking interesting

content over video collections. We construct a structured content representation with a similarity graph using our proposed instance search technologies. Considering the constraints related to real world usability, we present a flexible interface based on faceted navigation to enhance user performance when completing video browsing tasks.

This thesis shows that user performance can be enhanced by improving the effectiveness of instance search approaches, when seeking information in unstructured video collections. While covering many different aspects of improving instance search in this work, we outline three potential directions for future work: advanced feature representation, data driven rank and cloud-based search algorithms.

Chapter 1

Introduction

Given an image of a particular object, the research objective of this work is to develop video retrieval systems to effectively find video segments, which contain the occurrences of the particular object, from unstructured video collections. In contrast to traditional content-based video search systems which rely on textual annotation or metadata, we focus on searching for particular object instances, such as landmarks, buildings, logos and so on, purely based on the similarity of their visual information. Additionally, our goal is not near-duplicate detection which aims to find out the same video content but to search for the video clips which capture instances of the same objects under various backgrounds, since objects can be captured from different viewpoints, scales, and lighting conditions.

Information retrieval is the process of finding relevant information from a collection of information resources containing a set of documents. There are many different contexts within which people may seek information, such as finding interesting books from book collections in a library, searching for information on the World Wide Web, and so on. However, the task of finding relevant information becomes increasingly difficult as the information collection grows larger. Therefore, information retrieval systems have been

developed to address the needs for effectively seeking relevant information resources from vary large collections. When users enter a query which states their information needs, a successful retrieval system should be able to rank all of the documents in the collection according to their degree of similarity, or relevance. In this way, the retrieval systems could significantly reduce the time spent on finding the most interesting and relevant information, especially for very large collections.

The research reported in this thesis focuses on developing an information retrieval system that supports non-expert users to rapidly explore and find relevant information from within large video collections. More specifically, given an image query of a particular object, the system aims to find those video segments which are most likely to contain a recognizable instance of the object. The objects typically occupy only a part of an image and normally are highlighted with a bounding box in an image-based query. The retrieval task is to locate the appearance of the interesting objects over large video collections by automatically extracting visual information from large volumes of video content. In the literature, this type of retrieval task is called “Instance Search”. This research could help us to effectively seek information from large video collections without applying the time-consuming manual annotation of video content.

1.1 Instance Search

What is Instance Search?

The instance search task belongs to a category of content-based video retrieval tasks and was developed to address information needs in many real-world situations. Here an instance normally represents an object entity that has distinctive visual features, such as shape or



Fig. 1.1 Query examples for instance search task. The first and second rows show building and landmark instances specifically from the Oxford building dataset [69] and Pairs Dataset [70]. The following three rows show query examples from instance search tasks in TRECVID [90] evaluation campaigns.

texture, and exists in the content of video collections. In instance search, a search query would typically be an object example given by an image and normally associated with a bounding box to identify it from the background. The task of instance search is to find the video segment or segments which contain the appearance of the specific object from within a large video collection.

The instance search task in this work is different from object retrieval since it does not contain the detection of category-level objects. For instance, while searching for instances of a special person, we are not interested in human detection [20] from the video collection,

but rather, we are interested to find out where and when the person appeared in the video content. It is also different from semantic concept search which aims to find video segments with similar scenes or events. Good examples of instances include special landmarks (e.g., the Golden Gate Bridge, the Eiffel Tower), specific logos (e.g., company logos, University logos), a specific person (e.g., George W. Bush), and so on. In Figure 1.1, we give a selection of typical query examples for instance search tasks. The query topic is represented as one (or more) example images and for each, a red bounding box is also given to outline the instance of the interesting objects. Note that the instances within the first two rows are focused on building or landmark objects, the last three rows are from the objects and vary in size and appearance.

What is the usage for Instance Search?

Instance search has great potential for solving practical problems and could be used to develop a wide variety of applications. In the context of content-based multimedia retrieval, instance search can be used as a fundamental tool for a large number of computer vision systems. For example, Arandjelovic and Zisserman [2] achieved good results in searching special landmarks, which can be used in location recognition and 3D reconstruction. Instance search methods could also be used for automatic organization of large, unordered image collections in visual object mining tasks. Philbin and Zisserman [71] addressed this problem by applying instance search on image collections and group images that containing the visual similar objects into the same cluster.

In the first appendix section of the thesis, we describe a number of specific applications, which we have worked on to motivate the research of the work in this thesis. These applica-

tions are developed to demonstrate practical usages of instance search. They all benefit users and help them while seeking information from image or video collections.

1.2 Research Motivation and Challenges

With the rapid growth of digital video, various video search applications [67], [49] have been developed to address the enormous challenges of organizing, accessing and searching large video collections. Traditional concept-based approaches are inherited from text information retrieval technologies and built searching functions for metadata such as keywords, tags, or descriptions, that were associated with the videos. These approaches depend heavily on the quality and completeness of their metadata, which is mostly manual annotations from users. For large video collections, such manual annotation could be very time-consuming, and also subjective which may not capture the desired keywords to describe the content. So, video retrieval systems [88] have shifted from traditional meta-based text retrieval approaches to advanced content-based approaches [66].

Recently much research effort [50], [35], [88], [21] has been dedicated to building search algorithms for the visual content based on features such as colors, shapes, textures, or other information that can be derived automatically from the image itself. Such content-based approaches [31] could help to directly interpret user intentions and rank documents based on the visual similarity between low-level features, such as color, texture, and shape information. In this research, we aim to develop a content-based approach for instance search by automatically analyzing visual content in digital videos. As opposed to a traditional approach, there are a number of features in our system which make this research valuable:

1.2 Research Motivation and Challenges

- Our instance search systems rely on the visual content of digital videos, not on any textual annotation or description. So this research would help to avoid extremely time-consuming tasks, such as manual annotation.
- The automated approach of interpreting the visual content of videos provides us with an effective way to extract meaningful information from large collections despite the rapid increase in the amount of video content.

These two features form the core motivation of our work. In the rest of this research work, we are going to investigate the state-of-the-art approaches for instance search tasks and propose novel methods to improve the effectiveness of retrieval performance.

However building instance search systems from visual content is not a simple problem, and is especially difficult for large video collections. There are a number of challenges to address in this work and here we discuss a few major problems to be solved in order to accomplish our research objectives.

The first challenge comes from the variation of imaging condition when capturing objects, such as different viewpoints, changes in lighting or scale, various imaging backgrounds, and so on. In order to build a successful instance search system, we employ a local feature algorithm [51] to describe small image patches which are robust and invariant to image conditions. However different from the use of words in textual documents, noisy local visual features could be introduced which will corrupt or damage the retrieval performance for instance search tasks.

The second challenge is the scalability issue raised when dealing with very large video collections. The performance of some approaches would drop while applying them to instance search tasks for large-scale video collections. This is mainly due to the fact that the discriminative power of local features decreases when the number of videos is growing in the

dataset. This requires us to develop more advanced components to enhance the representation of local features and maintain their performance even when applying to very large collections. In this work we propose and evaluate a set of novel approaches to improve the performance of instance search systems. We also address the challenges of improving the effectiveness of video browsing over large video collections.

1.3 Hypothesis and Research Questions

After discussing the motivation and challenges of instance search, we now present the primary hypothesis and the main research questions which guided our research.

Hypothesis While seeking information in unstructured video collections, the user performance can be enhanced while improving the effectiveness of instance search approaches.

During this research work, we investigated a number of different approaches and explored various ways to primarily prove our hypothesis. We begin our research by asking three main research questions which were derived from the hypothesis. These research questions played an important role in guiding our research, helping us in experimental design, and eventually leading us to incrementally test the hypothesis. We outline three research questions as follows:

RQ 1. How much can we improve the effectiveness of the bag of features framework for instance search, if we make use of the geometric correlation information among local features?

A review of current research on the instance search task reveals that the most successful approach is implemented following the bag of features framework [88]. Since it combined

1.3 Hypothesis and Research Questions

the most advanced visual feature representation with mature text-based information retrieval technologies. To build an instance search system for visual content, we firstly need to extract robust local visual features, and then build a bag of features representation to support an efficient indexing algorithm for fast retrieval. However, this representation model lacks the ability to preserve geometric information among local features. Therefore we investigate the possibilities of applying geometric correlation information among local features to improve the effectiveness of instance search within the bag of features framework. More specifically, we aim to develop an object matching algorithm to improve the reliability of current instance search systems. This research allows us to build successful instance search applications for more challenging tasks in video collections, such as significant changes in viewpoints, heavily different background, small instance objects and so on.

RQ 2. How much can we improve the performance of instance search systems while applying our feature matching method to the video search task?

Currently, most of the research work in the literature focuses on instance search over image collections. In this research question we address the problems raised when applying instance search over large video collections. When applying the standard instance search framework on digital video, the most challenging problem is how to maintain retrieval performance even when dealing with large video collections. We carefully investigated the current approaches and focus on proposed methods to address the requirement of effectiveness for large scale video search tasks. By the end of this research, we aim to build a complete system to provide instance search capability for large video collections.

RQ 3. When browsing unstructured video collections, how much could we enhance user performance by applying our improved instance search approaches?

Finally, we are interested in exploring the ability of scalable instance search approaches in order to enhance user performance in the context of browsing large video collections. In recent years, various video browsing tools were built to help users to interactively navigate video content and locate a short video clip from within a large video collection within a specific time limit. Systems based on traditional concept-based video retrieval systems often failed to complete this task, since it either provides too many results with few concepts or too few results with more concepts. In this research question, we hope to address this problem using our proposed scalable instance approaches and to build interactive video browsing tools to enhance user performance while browsing large video collections.

1.4 List of Contributions

In this section, we list the three key contributions made in this thesis.

1. Weak Geometric Consistency Constraints We introduce a novel algorithm to improve the effectiveness of applying geometric information to perform local feature match verification in the standard instance search framework. This contribution improves the effectiveness of the bag-of-words framework for instance retrieval tasks. Experimental results on three open-source test collections demonstrate the superior performance of our proposed approach compared to other state-of-the-art approaches. Most importantly, our proposed method can be applied at a low computational cost which makes it suitable for large video collections.

2. Scalable Instance Search for Large Video Collections We built a complete instance search system for digital videos and also improved its effectiveness to address the scalability issues for very large collections. During this research, we firstly introduced an embedded coding algorithm to improve the feature matching accuracy thus improving the performance

of the bag of features approach. We then proposed methods to dramatically reduce the complexity of the video collections and implemented an efficient instance search system for large video collections. In addition, we also implemented an effective query expansion technology to further improve the retrieval performance of instance search for video collections. In order to evaluate the effectiveness of our proposed method, we participated in the largest video comparative evaluation campaign, the TREC Video Retrieval Evaluation workshop, also known as TRECVID [90], for three years. The experiment results demonstrate that our proposed methods yield superior performance, offering 94.5% performance improvement, without a notable increase in processing or storage requirements.

3. Instance Search Methods for Video Browsing Tasks We make use of our proposed instance search algorithms and built an interactive video browsing tool to improve user performance when engaging in video browsing tasks under time constraints. We proposed to help users to quickly locate interesting video clips by navigating them via the objects contained in each video segments. We also designed comprehensive user experiments to evaluate the effectiveness of our proposed approaches by comparing its performance over other advanced methods in large video collections. The results suggest that our approach successfully helped users to complete video browsing tasks and outperformed other advanced approaches.

To summarize, we propose the first contribution, Weak Geometric Consistency Constraints, to address research question one and publish a conference paper [113] at the 22nd international conference on multimedia modeling. Then an extended journal paper [114] is published to a special issue on “Good Practices in Multimedia Modeling” in the Neurocomputing article. Our second contribution provides an answer to the second research question. This research work is published in the following papers([110], [54], [55]) in the TRECVID

Video Retrieval Evaluation workshops. Finally, our third contribution provides an answer to the third research question. We also publish the related research works [82],[112], [115] in the International Conference on Multimedia Modeling.

1.5 Thesis Outline

We now present the organization of the dissertation. In this thesis, we begin our research work by outlining currently existing work that relates to instance search tasks in the second chapter. We discuss the bag of features framework, the most widely accepted framework for content-based image retrieval, and its usage in instance retrieval for instance search tasks. In Chapter 3 we present the research methodology applied in our research work. We also discuss several test collections used for experiments and evaluation purposes. Chapter 4 firstly presents our research work which employs geometric information to improve the retrieval performance of the bag of features framework for instance search tasks. Then we propose a scalable approach for instance search tasks in large video collections and discuss several advanced technologies, such as query expansion, to further improve the retrieval performance in chapter 5. Chapter 6 presents our work on building interactive tools to perform video browsing tasks in large video collections. Finally, chapter 7 summarises the work of this thesis, discusses some recent developments and explores a few directions for future research.

Chapter 2

Literature Review

In this chapter, we review the previous research work which investigates novel approaches to the instance search problem. And we also discuss the recent developments relevant to the focus of this work. This chapter serves as a general introduction to the literature relevant to instance search topics, which forms the basis for the fields we are contributing to in this thesis. We begin our literature review by introducing video search systems in Section 2.1. Specifically, we discuss some of the main features in the context of content-based video retrieval tasks and introduce the brief history of instance search tasks. Then in Section 2.2.1, we present one of the most popular algorithms in the field, bag of features framework, which derives from text-based information retrieval. This framework gives superior performance for various computer vision tasks, and is also employed in most of the state-of-the-art approaches to the instance search task. Finally in Section 2.3 we take a deeper look at this framework and discuss the recent developments, such as spatial verification and query expansion technologies, in order to improve the effectiveness of visual retrieval performance. In addition, we also present the issues with these methods especially for large data collections, which motivate the research work in this thesis.

2.1 Video Search Models

In this section, we firstly review the recent development of video search models for the purpose of solving the video retrieval problem. We then review some of the major research topics focused on evaluating the effectiveness of different video search approaches.

Video retrieval [109] is the research area of searching relevant videos from a large video database. A video retrieval process [53] begins when a user enters a query into the system. In the context of video search, queries are formal search topics to describe information needs, for example, search strings or search images. Instead of identifying a single document, a query may match several documents in the collection with different degrees of relevancy. Video retrieval models compute a numeric score reflecting how each document in the database matches the query and rank the documents according to this score value. Finally, the top ranking documents are presented to the users through a user interface. This process may then be iterated if the user wishes to refine the query. For researchers who aim to improve the performance of video retrieval, the process also includes the evaluation of an information retrieval system which means assessing how well a system meets the information needs of its users.

With the rapid growth of video data, there is a huge demand for video retrieval systems since it helps people to find videos relevant to an information need from a collection of video resources. Traditional video retrieval methods [37] [95] are often oriented around text search approaches based on metadata and human-annotated description associated with video content. Those systems, which rely purely on metadata, are quickly becoming impractical for very large video databases that are generated automatically. More advanced search models with the capability of automatic content analysis are needed. Content-based search approaches [79] are introduced to employ computer vision techniques to the video retrieval

problem. "Content-based" means that the search analyzes the contents of the video, such as colors, shapes, textures, or any other information that can be derived from the video visual content itself. In the literature, many research works([26] [29] [96] [111]) have been proposed to address the challenges of the video retrieval problem, and significant advances have been made in various areas. However, the review presented in [48] suggests that the problem of retrieving videos on the basis of their visual content remains largely unsolved. In this work, we focus on improving the performance of video retrieval by proposing content-based approaches.

2.1.1 Content-based search systems

Content-based search systems, as discussed in the work from Hu et al. [31] and Snoek and Worring [97], are the methods that help people to search videos in large digital video archives by their visual content. It has become an active and fast-advancing research area in video retrieval in the last decade. Many researching works [31], [97] have been proposed to improve user performance in finding video resources. The review work from Lai and Chen [46] reveals that the research activities for content-based video retrieval could be grouped into two main categories: semantic-based approaches and feature-based approaches. For the semantic-based approaches, algorithms were developed to build search engines based on automatically extracted semantic concepts from visual content. These systems could accept textual query topics, which are widely used to solve "Query by Natural Language" [31] retrieval tasks. However, identifying the semantic concepts from the visual content in videos is a very challenging task. This leads to many design limitations when applying these algorithms on more general content video databases. Therefore, researchers proposed feature-based retrieval methods to improve the retrieval accuracy. Local image features [107]

are image patterns which could be extracted to describe a certain visual information, such as points, edges or small image patches. In order to search using visual similarities, the query topics for these approaches are image examples, from which the local features can be extracted. This type of approach is also known as “Query by Example” [31] retrieval.

As discussed in this review work [67], there are three main steps to build content-based retrieval systems to search digital videos in large databases. The first step is video segmentation, which aims to reduce the redundancy from raw video content by representing video content with a set of keyframes. In the literature, shot boundary detection technologies are developed for this purpose, and the related research work was reviewed in the paper from Smeaton et al. [89]. The next step is to extract features from keyframes to effectively describe the visual information. Many efficient algorithms were developed to perform feature representation for various video search tasks, including global features, such as HOG, MPEG-7, GIST, and local features, such as SIFT, SURF. The final step is to build retrieval systems to perform video search tasks, which typically include two main components: efficient indexing structures and similarity measurement algorithms. Overall, video retrieval systems need to extract large numbers of features to analyze and query the visual content of video collections, and most features are high dimensional vectors. So it will be always challenging to build effective and efficient video search system without increasing complexity and computational cost, especially for large-scale databases. Patel and Meshram [67] identified the importance of choosing proper features and ranking algorithms according to their potential interest to the user requirement.

The broad range of applications from the research of video search systems has motivated many researchers. For example, near duplicated detection applications [83] were developed to reduce the high volume of redundancy with overlapping or duplicate content for the

growing social video collections. Application for surveillance event detection [65] is another active research area to automatically detect human behaviors efficiently in vast amounts of surveillance video with huge potential benefits, such as improving public safety and security.

2.1.2 Evaluation and Statistical Testing of Video Search Systems

In this section, we present a brief discussion of measuring the effectiveness of video retrieval systems. The main goal of the information retrieval researcher is to make progress by finding better retrieval methods and avoid the promotion of worse methods. Given two information retrieval systems, how can we determine which one is better than the other?

The most standard approach is to select a collection of documents, write a set of topics, and create relevance judgments for each topic and then measure the effectiveness of each system using a metric like the mean average precision (MAP). In recent years, evaluation campaigns which benchmark information retrieval tasks have become very popular for a variety of reasons. The review work from Smeaton et al. [91] highlights several benefits of participating these evaluation campaigns. Firstly, evaluation campaigns allow comparison of the work from different researchers in an open, metrics-based environment. Secondly, they provide shared data, common evaluation metrics and often also offer collaboration and sharing of resources. Lastly, evaluation campaigns are also attractive to non-research agencies or commercial company since they can act as a showcase for research results. It is worth to mention that the annual Text Retrieval Conference (TREC) Video Retrieval Evaluation (TRECVID) conference [90] is introduced to promote progress in video analysis and retrieval since 2001. TRECVID provides well-defined test collection of videos and different video search tasks which allow participants to submit results from their content-

based video retrieval algorithms for evaluation. In this work, we participate in this evaluation campaign to evaluate the effectiveness of our algorithms on test collections.

We want to promote retrieval methods that truly are better for improving retrieval performance, rather than methods that by chance performed better given the set of topics, judgments, and documents used in the evaluation. Statistical significance tests [93] play an important role in helping the researcher achieve this goal. Smucker et al. [93] presents a review work on the statistical significance tests for information retrieval evaluation. As Box et al. [14] explain, a significance test consists of the following essential ingredients:

1. A measurable metric or criterion by which to judge the performance of two systems. In information retrieval evaluation, researchers commonly use the difference in mean average precision (MAP) as the main metric.
2. A distribution of the test statistic given our null hypothesis. A typical null hypothesis is that there is no difference in our two systems.
3. A significance level that is computed by taking the value of the test statistic for our experimental systems and determining how likely a value that large or larger could have occurred under the null hypothesis.

When the significance level is low, the researcher can feel comfortable in rejecting the null hypothesis. If the null hypothesis cannot be rejected, then the difference between the two systems may be the result of the inherent noise in the evaluation. So a powerful test allows the researcher to detect significant improvements even when the improvements are small.

2.2 Instance Search Systems

Instance search belongs to content-based video retrieval systems and draws significant research attention in recent years. In 2006, Sivic and Zisserman [88] firstly introduced the approach to object retrieval which searches for and localizes all the occurrences of an object in a video, given a query image of the object. Then TRECVID [90] created the instance search tasks in 2010, which aimed to promote the research required to meet the needs of finding more video segments of a certain specific person, object, or place from large video collections. Since then, many works [69], [39] have been proposed to improve instance search task. In this work, our research also focuses on developing methods to address the challenges in instance search task. In the following section, we review the state-of-the-art approaches, which obtains high-level of performance for querying object instances from large databases.

2.2.1 Bag-of-Features Approach

Among the research work in the literature, the Bag of Features (BoF) approach, proposed by Sivic and Zisserman [88], was widely studied. This framework is applied to instance search problems by treating local image features as visual words. Since this approach achieves state-of-the-art performance in many evaluation benchmark collections, such as oxford building collection [69], and Pairs6K data collection [18], it is considered to be the standard solution for instance search problem. So in this section, we describe this framework in detail and review the standard components to build an instance search system. Then we diagnosis this approach and discuss its weakness which leads our proposed solution to improve retrieval performance.

Overview

Recent work [44], [18], [70] presents the standard components in BoF approach to build content-based retrieval systems for instance search. To begin with, local features are computed as a high-dimensional vector to describe the “salient” regions from visual content. These descriptor vectors are then clustered into a set of visual words guided by a pre-trained visual vocabulary so that each instance in the database is represented as a bag of features representation, which are built into index algorithm for later querying and retrieval. Figure 2.1 illustrates the main components of the bag of features approach, and we overview each component in the following section.

Feature Extraction

To build reliable search systems, the first step is to extract a set of local feature descriptors from visual content. Comparing to global feature algorithms, local feature algorithms are able to identify the object even under changes in scale, noise, and illumination. They have been applied successfully in many real-world applications, such as object recognition [51], image retrieval [8], video data mining [87], and object category recognition [60]. Wu et al. [104] performed a comparative study for various local feature description algorithms. Their

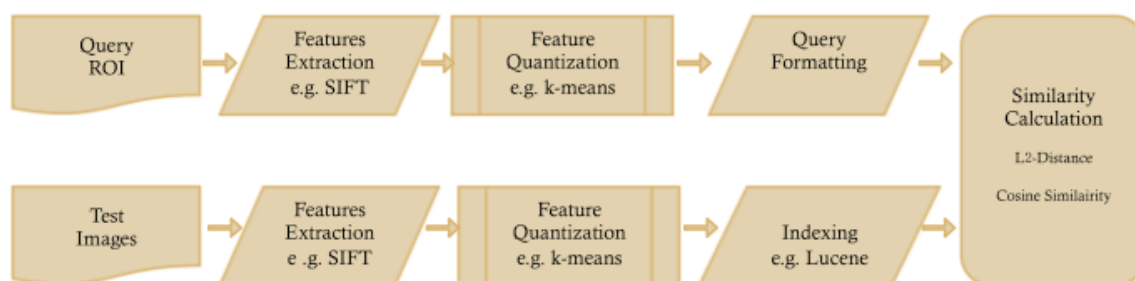


Fig. 2.1 Main components for content-based video retrieval systems

experimental results suggest that the Scale Invariant Feature Transform (SIFT [51]) algorithm performs the best for various imaging condition changes, such as scale and rotation change.

In the bag of features algorithm, the local feature algorithm is employed to extract the stable feature descriptors for every frame image from the video content. Scale-invariant feature transform (SIFT [51]) is a computer vision algorithm to extract and describe local features in images. As demonstrated in figure 2.2, the calculation of SIFT feature mainly includes two steps: 1. local invariant keypoint detection. In this step, local keypoint candidates are identified by finding the extrema values of difference of Gaussians in multiple scale spaces based on applying Gaussian filters at different scales. Low contrast candidate points are discarded to remove the unstable keypoints. 2. feature descriptor computation. A gradient orientation histogram is computed for each keypoint to form a feature vector to represent the local visual pattern. Normally, a 16x16 pixel block around the keypoint is taken, which is further divided into 16 sub-blocks of 4x4 size. An 8 bin orientation histogram is computed within each sub-block, so the final keypoint descriptor is a 128-dimensional vector. Usually, we could extract several hundred or thousand local features from a single image frame.

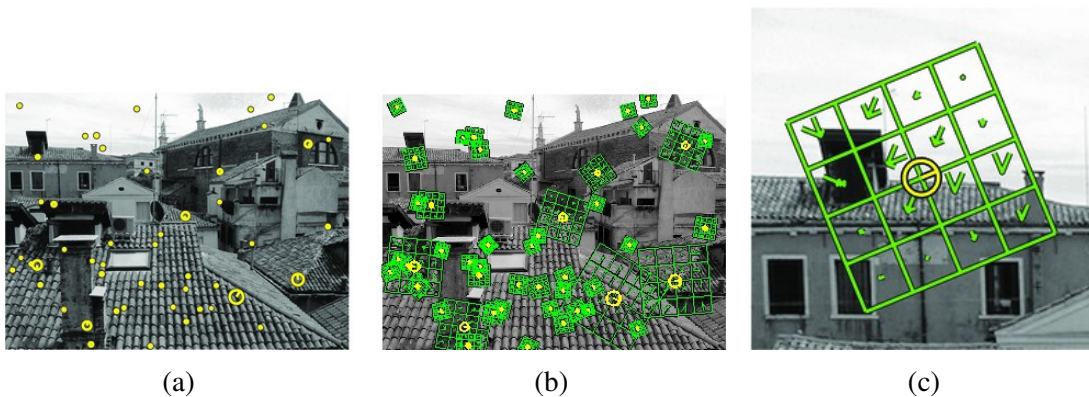


Fig. 2.2 Feature extraction: a: Detected salient regions from SIFT [51] algorithm, b: Feature descriptors calculated from the detected regions, c: A close look at computed oriented gradients

Feature Quantization

Feature quantization is the process of aggregating a set of local descriptors into one vector to represent each image with a fixed-sized feature vector. It is an important step in this framework which helps to build fixed-dimensional vector for instances. Following the previous step, we obtain a large collection of high-dimensional local feature descriptors. To achieve a compact representation, the local features are quantized into the corresponding visual words according to a standard visual vocabulary.

However building visual vocabularies can be a challenging and time-consuming task, especially for large feature collections. The research work from Sivic and Zisserman [88] first proposed to employ the K-means algorithm to generate visual vocabularies. Then an approximate K-means clustering method [69] was developed to improve the increase the vocabulary size which perform as well as the original K-means algorithm, but at a fraction of the computational cost. By now, we quantized local features to a “bag” of visual words which are then used to index the frame images for the video retrieval tasks.

Search Algorithm

Given a query image, the search algorithm focuses on ranking the frame images according to their visual similarity to the query topics. The bag of feature framework uses vector space model [53], a well-studied search algorithm in information retrieval, to calculate the similarity from the normalized euclidean distance between each document vector in feature space. In this model, every document is represented as a sparse vector in the high-dimensional space. The equation 2.1 explains the standard way to compute the cosine similarity score (q, d)

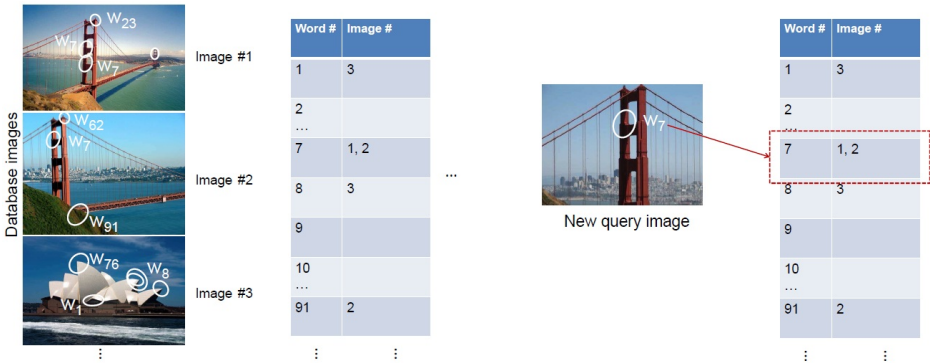


Fig. 2.3 Inverted indexing structure in Bag of Feature framework

between query image q and each database document d .

$$score(q, d) = \frac{\vec{V}(q) \cdot \vec{V}(d)}{|\vec{V}(q)| |\vec{V}(d)|} \tag{2.1}$$

where vector \vec{q} and \vec{d} represent the visual content of query object and object instance from database respectively. It is also standard to combine the advanced *TF-IDF* weighting scheme [53] to improve search accuracy. Inspired by text retrieval methods, Sivic and Zisserman [88] first proposed to merge the quantized visual words into an inverted indexing structure [53] to index all the video frames for the purpose of providing rapid object search capabilities. Figure 2.3 illustrates an example of building the inverted indexing structure from multiple database images. Firstly, images are processed and feature visual words are obtained for different visual objects. Then the inverted indexing table is populated with visual words by adding corresponding document IDs into the indexing table. After that, systems could directly retrieve the database images by only comparing images with share common visual words, which is much faster than brute-force searching every document in the database at query time.

2.3 Improvement for Visual Search

In practice, local feature extraction and quantization are noisy processes, which can introduce error when producing visual words. More importantly, these noisy visual words could cause false feature matches between unrelated object instances during inverted index construction. These wrong feature matches could import errors into the ranking algorithms and lead to corrupted results. In this section, we review advanced technologies which were to improve the retrieval performance for content-based video search systems, especially for large data collections.

2.3.1 Improving visual vocabulary

As discussed in the previous section, approaches inherited from Bag of Feature framework firstly extract high-dimensional local region descriptors from visual content, then build a visual vocabulary to cluster these features into visual words for indexing construction. Mikolajczyk et al. [56] suggested that increasing the size of visual vocabulary could improve the discriminative power of each visual word and then improve the object classification or recognition performance.

Early systems [88] process the feature quantization with straightforward k-means clustering algorithms which are effective but difficult to scale to large vocabularies. Therefore, more recent work [63] has used hierarchical k-means clustering to produce a “vocabulary tree” and greatly increased the size of visual vocabulary size. Their results show that the power of the vocabulary tree approach could be used to handle large data collections, as much as 1 million images. Then Philbin et al. [69] proposed a scalable approach for building similarly large-size vocabulary by the use of approximate nearest neighbor methods. They demonstrated that this method offered superior performance, even with similar complexity as

the vocabulary tree algorithm. The experiment proved that the best performance was obtained at the vocabulary with 1 million visual words. This algorithm has recently been extensively used for other computer vision tasks, such as supervised classification [103] and unsupervised matching [64]. But building large size visual vocabulary with the previous approaches is a very challenging task, since it requires extremely large computational resources. In this research, we address this problem and improve the retrieval performance for visual search tasks.

2.3.2 Spatial Verification

The output from performing a query based on the bag of feature framework described previously is a ranked list of documents. The search algorithm is purely based on the matched visual words extracted from the visual content of each object. Due to the noisy visual words introduced by feature extraction and quantization, we would like to verify that the matched features are from the same object instance region between the target database image and query image.

Hartley and Zisserman [28] presented a comprehensive study on the spatial transformation issues for the feature correspondence between the query objects and their estimate instances. Figure 2.4 demonstrates three main types of spatial transformations: rotation, scaling and location translation. These spatial transformations are introduced to cover object

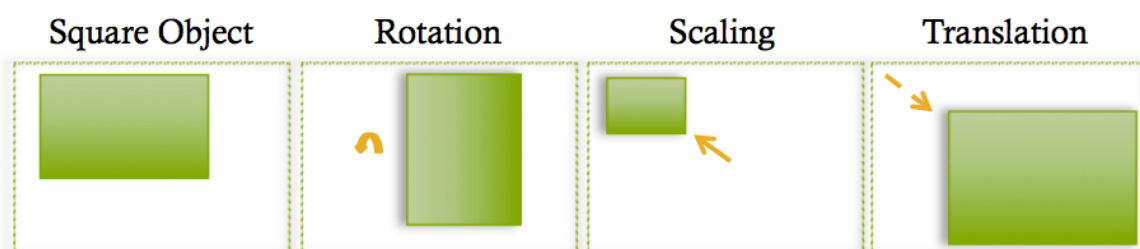


Fig. 2.4 Three main types of spatial transformations between object and its estimate

2.3 Improvement for Visual Search

appearance variations in different images which may be caused by the changing in viewpoints, lighting, etc. In this case, the query object can be mapped to its estimate instance following a combination of related spatial transformations. Such mappings can be estimated from correspondences of local salient regions between the query and database query images.

The work in Sivic and Zisserman [88], Philbin et al. [69], and Zhang et al. [111] shows that these geometric correspondences could be applied to improve the accuracy of various retrieval approaches to visual search tasks. To achieve this improvement, they proposed a procedure to firstly estimate a fast and robust transformation hypothesize and then verify the corresponding feature matches between a query region and target image. A RANSAC [23] scoring algorithm is applied to select the hypothesis with the greatest number of feature correspondences which following the same transformation. This hypothesis could be used to verify the geometric constant for each individual feature correspondence during the retrieval process. After obtaining the initial ranking results, the verification step refines the matched features by filtering out the inconsistent feature correspondences. Then the results with higher number of consistent feature matches are considered as reliable results and moved to higher order in the list.

In the literature, the methods of incorporating spatial information to improve the ranking results are called “Spatial Verification”. Currently, the spatial verification algorithm build complex models to estimate the full spatial transformation, which is very time-consuming and can be applied to few top ranked documents to refine the results. So spatial verification is normally applied up to a maximum of the top 1000 results returned from the search engine. In our work, we propose novel methods to apply spatial verification to cover image variations without explicitly modeling and can be applied to all candidate images at a low computational cost.

2.3.3 Query Expansion

Query expansion is a well-known methodology for improving retrieval performance in text retrieval. The work from Chum et al. [18] illustrated that query expansion can be applied to improve the retrieval results in the context of visual object retrieval systems. In their method, a number of top ranked documents from the initial results are added to the search query and allow the retrieval system to find more challenging documents by using relevant terms that are not included in the original query.

This query expansion procedure can be interpreted as follows: the system extracts a generative model of an object from the query region; then forms the response set from those images in the corpus that are likely to have been generated from that model. The generative model, in this case, is a spatial configuration of visual words extracted from the query region, together with a “background” distribution of words that encodes the overall frequency statistics of the corpus. It is important to note that the performance of query expansion depends heavily on the initial results. In other words, strong initial results could help to form a more general model for the query topic and retrieve richer results by re-querying the database using the expanded query model. However, weak initial results could adversely affect the query model and damage the search results. In this work, we incorporate our spatial verification algorithm and implement an effective approach for query expansion, to automatically enhance our search performance.

2.3.4 Local Convolutional Features

Most recently, Convolutional neural networks (CNNs [45]) have been demonstrated to produce superior performance on various computer vision tasks, such as image classification [19], object recognition [85], face recognition [47], scene recognition [117], and so on.

Naturally, researchers are motivated to propose CNN-representations based systems in order to improve the state-of-the-art for image retrieval tasks. Babenko et al. [5] firstly investigate the use of the top layers of a large convolutional neural network for image retrieval tasks, since it provides an effective representation of the visual content in images. Recent works [4], [108] demonstrate that interpreting the activations from the convolutional layers as local features to describe particular image regions could significantly improve the performance. These approaches aggregate all local CNN-based features from one image into a dense feature vector for similarity comparison in the end, with various encoding methods such as average pooling [19], or VLAD [84]. However, the global dense representation lacks the ability to represent small objects for instance search tasks. Mohedano et al. [58] propose an instance retrieval pipeline based on encoding the convolutional features from CNN network using the bag of feature framework. This bag of feature based sparse visual representation provides a fast composition of object descriptor for any region within the image. Their experiment results prove that their proposed system achieves competitive performance on several evaluation benchmarks [69], [70].

Although these methods perform well in image retrieval benchmarks, it is still a challenging problem for instance search tasks. The main reason is that these approaches are all based on off-the-shelf CNN features, which are trained for completely different purposes.

2.4 Conclusion

This section presents the literature review for our research work in this thesis. We investigate the previous research work related to content-based video search tasks and also studied the real-world applications derived from the research outputs. Especially, we introduce the brief history of instance search tasks and review the various approaches for solving this search

problem. We describe the bag of features framework, the state-of-the-art methods, and review the standard components to build an instance search system. In addition, we review this approach and discuss its weakness that motivates our research work to improve retrieval performance.

Many previous researching works in the literature address instance search tasks and perform well in many retrieval benchmarks. However, instance search is still a challenging problem and remains unsolved, due to the complex nature of the underlying computer vision problem. We conduct our research to continue improving the performance of instance search. In particular, we address three main topics after reviewing the current state of instance search systems.

Firstly, we address the problem of ignoring the geometric information in the bag of features framework [88] and address our first research question to implicitly verify local feature matches and thereby improve the performance of instance search systems. Recently, several approaches have been developed to implicitly verify the feature matches with respect to the consistency of their geometric relations, i.e., scaling, orientation, and location, in the Hough transformation space. Avrithis and Toliás [3] developed a linear algorithm to effectively compute pairwise affinities of correspondences in 4-dimensional transformation space by applying a pyramid matching model constructed from each single feature correspondence. Comparing to the existing work, our proposed method considers the spatial consistency for the geometric correlations between matched feature correspondences, while maintaining the efficiency and increasing the effectiveness of the instance search systems.

Recent work [18],[69] has shown that the standard bag of features approaches can suffer from poor recall when applied to the instance search task in very large collections. Our second research question focuses on the problem of feature quantization in a bag of features

framework, and proposes a scalable method to improve the feature matching. We enhance the feature representation with an embedded coding scheme and build scalable components to handle very large video collections. We also implement a visual query expansion technique to refine the original query topic by aggregating the information from multiple relevant result images.

Lastly, we move on to examine instance search performance in video browsing and propose our third research question to investigate how much we could enhance user performance using scalable instance search technologies. The video browsing task aims to promote progress in content-based analysis and exploration for very large video collections. Several content based video retrieval tools have been proposed to address this research problem. Schoeffmann and Boeszoermyeni [79] proposed an interactive navigation summaries approach to help users to quickly explore the entire video content. In addition, their work provides abstract visualizations of video content at a user-defined level of detail. But the results from the Video Browser Showdown [78] show that the tools for video browsing can be significantly improved. We aim to construct a similarity graph where related video scenes are connected using instance search systems so that abstract representation can be extracted with clustering technologies.

In summary, we introduce the context of content-based video search and present its current research achievements in this review work, which also helps us to identify the main research goals and better understand the impact of our research work.

Chapter 3

Research Methodology

This research focuses on investigating methods to improve the effectiveness of instance search algorithms to enhance user performance while searching over large video collections. During this work, we aim to apply quantitative research to test our hypothesis in the context of instance search task. In order to carry out the research, we implement instance search systems using various different algorithms and design the experiments to precisely measure their performance over appropriate test collections. Consequently, we can demonstrate the effectiveness of our proposed approaches by comparing the results of our proposed system with other state-of-the-art systems.

We begin our work by examining the state-of-the-art approaches for instance search in the context of content-based information retrieval and study the bag of features framework, that is widely used for instance search in the literature [88]. Subsequently, we propose novel methods to improve this framework and design complex experiments to evaluate their performance for improving instance search. For instance search in large video collection, we propose large-scale methods to further improve the retrieval performance. It involves intelligently improving feature representation with embedded coding and employing query

expansion technology from text-based information retrieval to enhance instance search for visual content. To evaluate the effectiveness of our proposed approaches, we participate the instance search tasks in TREC video retrieval evaluation campaign [90] which was designed to demonstrate contributions in the video retrieval research community. In the end, we investigate methods improve the effectiveness of instance search based approaches for helping users to find specific video clips in large video collections. In order to demonstrate the performance improvement of our proposed methods, we participate in the Video Browser Showdown [77] and design user experiments to demonstrate the performance of our proposed approaches.

This chapter presents the research methodology and the experiment settings which help us to achieve our research goals. We firstly discuss our experiment configurations and show how they are designed to address our research questions in Section 3.1. In Section 3.2 we describe the research methods which investigate the current approaches for instance search and proposed solutions to answer the research questions. Then in Section 3.3 we present a standard evaluation pipeline for assessing the performance improvement from the proposed methods in the context of instance search tasks, including all evaluation metrics and test collections used during this work. Finally, we finish this chapter by presenting conclusions from this work in Section 3.4.

3.1 Experiment Configurations

The experiments in this research are designed for the purpose of evaluating the performance of our proposed methods to answer the research questions in section 1.3 and to improve the effectiveness of instance search tasks. There are three main experiments in this research,

3.1 Experiment Configurations

Table 3.1 Overview of the Experiment configurations

Experiment	Chapter	Target	Test Collection	Time		
				2013	2014	2015
WGCC	4	RQ 1.	Oxford5k, Paris6K, FlickrLogo	✓		✓
INS	5	RQ 2.	TRECvid Test Collection	✓	✓	✓
VSS	6	RQ 3.	VBS Test Collection		✓	✓

and each one is designed to address one of the research questions. Table 3.1 presents the overview of our experiment configurations.

1. WGCC Experiment

We started by exploring the Bag of Features framework to build instance retrieval systems; this was inspired by classical text-based information retrieval technologies. As discussed in section 2.2.1, this approach represents the visual information of object instances by a set of local features and builds a search engine based on feature correspondence. However we notice that the spatial layout of local features from object instances was ignored within this framework, hence we suggest that the performance of instance search could be improved with geometric information of local features and investigate our first research question:

RQ 1 How much can we improve the effectiveness of the Bag of Feature framework for instance search, if we make use of the geometric correlation information among local features?

Consequently, we propose a novel algorithm, Weak Geometric Consistency Constraints (WGCC), to address this problem and improve the effectiveness of the bag of features framework.

We design the WGCC experiment to evaluate the performance of our proposed method. In this experiment, we build an instance search approach following the bag of features framework and also implement the WGCC algorithm as an independent component to upgrade the standard bag of features approach. Then we employ three open source test collections, Oxford5K, Paris6K, and FlickrLogo, to perform instance search tasks and evaluate the approach. We choose these three test collections since they are specially designed for evaluating instance search approaches and have been widely used in previous evaluation research([70], [63] [83]). Finally, we demonstrate the effectiveness of our methods by comparing its retrieval performance with other advanced approaches, which are presented in Chapter 4.

2. INS Experiment

In order to address the problems of applying instance search for large video collections, we propose that the retrieval performance of instance search could be improved by increasing the matching quality of local feature correspondences. We propose a novel nearest neighbor search technology to increase the accuracy of local feature matching and build an advanced instance system in order to address our second research question:

RQ 2. How much can we improve the performance of instance search systems while applying our feature matching method to the video search task?

In particular, we implement a scalable instance search system for large video collections, which involves representing video content by sets of keyframes to reduce computational complexity. Moreover, we also investigate other technologies, such as query expansion, to further improve the retrieval performance.

In order to evaluate the retrieval performance for our scalable approach, we participated in the Instance Search Task (INS) in TREC Video Retrieval (TRECVID) campaign for three sequential years from 2013 to 2015. This INS experiment helps us to address the challenges of performing large-scale evaluation by providing large video collections with hundreds of hours of video, well-defined query topics, and a comprehensive metrics-based evaluation. We obtain evaluation results from our participation in TRECVID, which we use to prove the effectiveness of our scalable methods by comparing its retrieval performance with other state-of-the-art approaches for instance search in large video collections. The detailed description of this research work is given in Chapter 5.

3. VSS Experiment

Finally, we focus on improving user performance when browsing large video collections. After conducting a comprehensive literature review, we find that current video browsing tools offer limited abilities for non-expert users to quickly understand the video content, especially for large collections. For this reason, we introduce research question three:

RQ 3. When browsing unstructured video collections, how much could we enhance user performance by applying our improved instance search approaches?

To answer this question, we build an interactive video browsing tool using the scalable instance search approach from our previous research work. In order to evaluate its effectiveness, we design an experiment to examine how the instance search algorithm could help users find interesting video content from large video collections in time-constrained environment.

To assess the effectiveness of our proposed video browsing tool, we participate in the Video Browser Showdown (VSS) event hosted at the annual Multimedia Modeling conference. This event aims to evaluate and demonstrate the efficiency of exploratory video retrieval

tools on a shared data collection. The event provides large video collection along with comprehensive set of query topics to simulate the “known item search” tasks. We design the VSS experiment to measure user performance when performing video browsing tasks with our system. The VSS experiment allows us to evaluate the effectiveness of our retrieval tools in helping users to solve real-world problems. The detailed research work is given in Chapter 6.

3.2 Research Methods

In order to carry out our research, we explore various ways to improve the effectiveness of instance search system and evaluate the performance of these methods while applying them on video browsing tasks. In this section, we discuss the research methods in this research, which are divided into three main parts.

Literature Review We start our research with conducting an extensive literature review of the state-of-the-art approaches for instance search in multimedia retrieval area. And we identify the strengths and weakness of each method for building instance search systems in large video collections.

Research Questions After analysing the existing work, we construct research questions to address the weakness of the current instance technologies and propose novel approaches to improve the performance of instance search systems.

Experiment Design Lastly, we design experiments to evaluate the effectiveness of our proposed approach. In summary, we build complete retrieval systems to implement our proposed improvements, and then design comprehensive instance search experiments on standard test collections. To demonstrate the performance improvement, we can directly compare the results from different advanced approaches.

Quantitative Evaluation For the purpose of quantitative evaluation, we apply the proposed methods to complete instance search tasks and measure the performance results. This work involves participating in different evaluation campaigns, such as the Instance Search tasks (INS) in TRECVID [90] or the Video Browsing Showdown (VBS) [77]. In the end, we use various evaluation metrics for information retrieval to measure the performance and compare with other state-of-the-art systems to demonstrate their effectiveness.

3.3 Evaluation Procedure

In this section, we introduce the evaluation pipeline, evaluation metrics and test collections used to assess the performance of the proposed research work for the instance search task. In the context of instance search task in large video collections, we need a considered and thorough evaluation to demonstrate the superior performance of our proposed techniques on representative test collections. We follow the standard evaluation pipeline in this work to design complex experiments to evaluate the effectiveness of various systems for the instance search task.

Here we explain the details of this pipeline as shown in Figure 3.1, which contains three main steps.

Step 1. The first step involves building instance search systems to implement the proposed solutions. As illustrated in figure 3.1, this step includes feature extraction, object representation from visual content, and most importantly the implementation of a novel ranking algorithm to improve retrieval performance.

Step 2. Then the second step deals with performing instance search tasks using the previously built systems over publicly available test collections and obtaining the ranking results

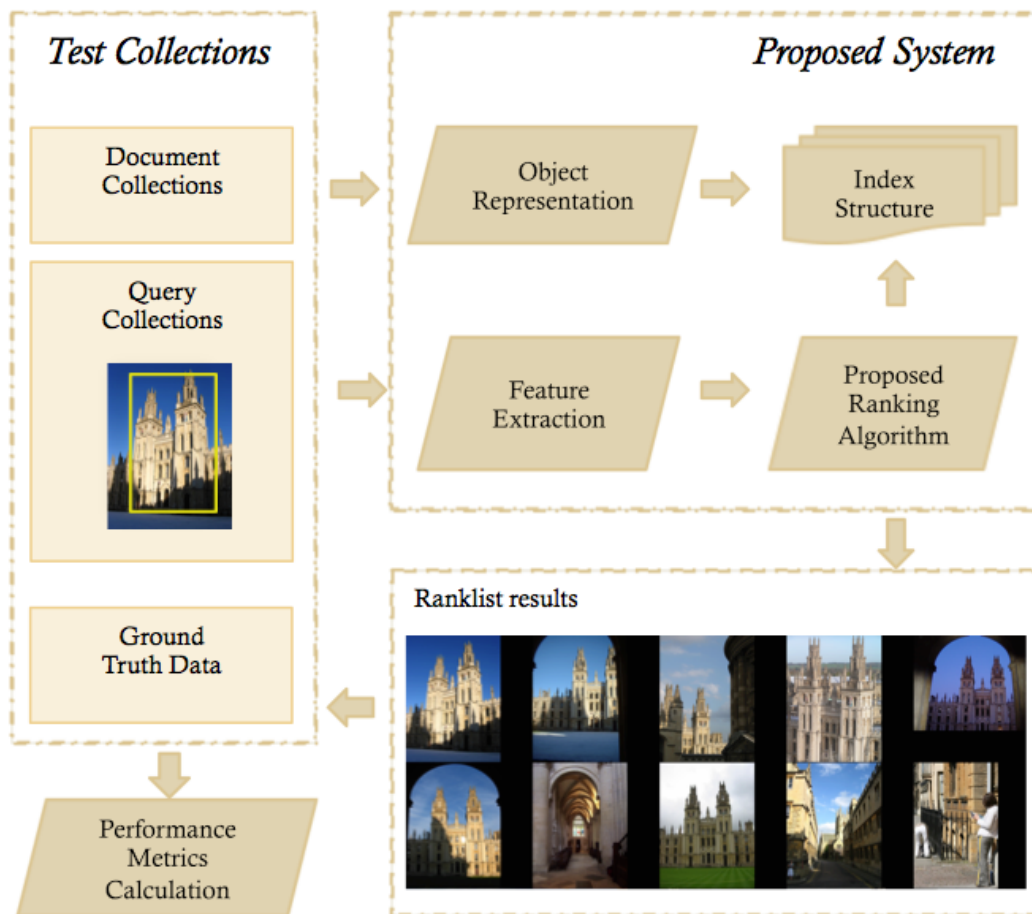


Fig. 3.1 Standard evaluation pipeline for instance search experiments.¹

for all the query topics. This step may include formatting search terms from different query images.

Step 3. Finally, the last step makes use of the output from previous step and measures the retrieval performance with appropriate evaluation metrics during each experiment. We use ground truth data to identify whether the ranking results are either relevant or non-relevant to a specific query topic. We demonstrate the effectiveness of the experimental approach by calculating the metrics between system returned results and ground truth results.

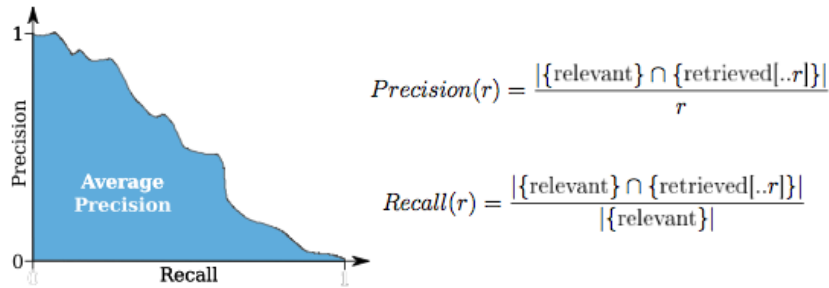


Fig. 3.2 Precision-recall curves. Average precision value is equal to taking the area under the curve

With this evaluation pipeline in place, we can measure the effectiveness of different instance search systems by comparing their performance for the same test collections, same query topics, and same performance measurements.

In the following sections, we present the metrics used to measure system effectiveness. We then describe the publicly available test collections used in this research.

3.3.1 Evaluation Metrics

Among the metrics used to measure the performance of retrieval systems, we focus on those illustrate the effectiveness of instance search methods, and are commonly used in literature and in the collaborative evaluation exercises such as TRECVID [65].

Precision and Recall Precision and recall are two common metrics in information retrieval to measure the effectiveness of algorithms. Precision is calculated as the number of true positives in a collection divided by the number of items labelled as positive (including false positives). A higher precision is desirable in a system because it means the algorithm finds more relevant items than irrelevant results, which help users to find good items. On the other hand, recall is the measure of the fraction of relevant items that are retrieved from

possible relevant results. A higher recall value is better for instance search algorithms, which means the algorithm was better at finding more relevant items from the collection.

Average Precision Average Precision (AP) is another standard metric of performance evaluation for information retrieval. It is a single number to characterize the performance of retrieval systems for a given query, obtained by computing precision averaged across values of recall between 0 and 1.

By computing precision and recall at every position in the ranked sequence of documents, we can plot a precision-recall curve. Figure 3.2 demonstrates the precision-recall curve which plots the precision value at corresponding recall value. The average precision score corresponds to the area under the precision-recall curve. When comparing the performance of different instance search systems, we can quickly compare the single average precision score. A better instance search system will have higher values for average precision as it tends to rank actual object instance images near the top of the ranked list. We can also examine the entire precision-recall curve for in-depth analysis.

Mean Average Precision Mean Average Precision (mAP) is a standard single-number measurement for comparing instance search algorithms across multiple queries. In most of the cases, mAP is the mean of all average precision scores for a set of queries in test collections. In our evaluation experiments, we compute mAP using the following equation,

$$MAP = \frac{\sum_1^N AP(q)}{N} \quad (3.1)$$

where N is the number of queries. The value range of MAP is between 0 and 1, and larger value is desirable for an effective instance search system, since it provides more robust performance across various different topics.

Table 3.2 Statistics on four test collections

Name	Videos(hours)	Images(k)	Size(GB)	Topics/Queries
Oxford5K	N/A	5.062	1.98	55
Paris6K	N/A	6.0	2.6	55
FlickrLogo	N/A	≈ 450.0	≈300	30
BBC EastEnders	≈ 244	≈ 450.0	≈300	28

Precision at N Precision at N measures the precision of the top N items in the ranking list, which effectively indicates the search accuracy of the top N results. For example, P@10 ("Precision at 10") corresponds to the number of relevant results on the first search results page. This metric is very useful since many queries have thousands of relevant documents, which is much more than the results users can interpret. Thus, measuring the precision of the top N list of items could be especially helpful to give us the insight of how the algorithm performs.

3.3.2 Test Collections

In order to effectively compare the performance between different instance search systems, we need to set up a controlled experiment. The experiment should contain a well defined and representative test collection which includes the multimedia dataset, a list of search tasks with necessary information as queries, and ground truth data for judgements. This section describes our selected test collections for performance evaluation. We choose four test collections because they all include a reasonable sized dataset which appropriate a real-world situation, but also a set of well defined query topics. In the context of instance search task, we can use the ground truth data to evaluate the performance of different systems. Table 3.2 shows an overview statistical analysis of the test collections employed in this reserch.

Oxford buildings

Oxford building test collection was introduced by [69] in 2007. It consists of 5,062 high-resolution images automatically crawled from Flickr using queries such as “Oxford Christ Church”, “Oxford Radcliffe Camera” and “Oxford”. There are 11 different landmark instances query topics, each of which represented by 5 possible query images which gives a set of 55 queries. The ground truth data was manually annotated for 11 landmark instance topics. Specifically, images of a certain landmark are labelled as “Good” if they contains clearly visible instances, and “OK” if more than 25% of instances are visible.

Oxford building is a standard benchmark dataset, specially designed for landmarks search evaluation. This dataset captures landmark objects under changes in scale, backgrounds and illumination. It is carefully annotated to obtain the ground truth by manually searching over the entire dataset for each query landmark. We use this dataset in our WGCC Experiment to test both the performance and scalability of different instance search systems.

Paris6K

Paris6K test collection [70] consists of 6,412 images collected by searching for particular Paris landmarks in Flickr. In total, 11 landmarks with 55 image queries are provided with manually annotated ground truth for users to evaluate retrieval performance. The images are considered to be positive if more than 25% of the instances are clearly visible in the image frame.

The motivation for choosing the Paris dataset is to have images of similar scenes to those of the Oxford landmarks (i.e. buildings, often with some similarities in architectural style), but without having identical buildings to those used in the Oxford quantization. We mainly

use this dataset in our WGCC Experiment to test the system reliability of instance search approach.

FlickrLogos-27

This dataset [75] consists of 5,107 images including 810 annotated positive images corresponding to 27 classes of commercial brand logos and 4,207 distraction images that depict their own logo class. This is a very challenging dataset because the positive images share much more visually similar regions with the distraction images and have more noisy background. For each logo, 5 query example images are given for evaluation purposes.

Each image is inspected manually to ensure that the specific logo object is actually shown. The choice of this dataset as an evaluation collection is intended to test the system reliability of instance search approach across different search domain in our WGCC Experiment.

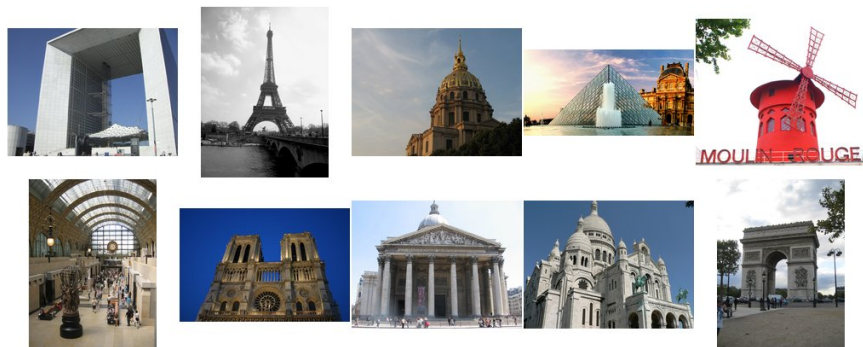
BBC EastEnders

This dataset contains approximately 244 video files (totally 300 GB, 464 hours) with associated metadata, each containing a week's worth of BBC EastEnders programs in MPEG-4/H.264 format. Master shot reference files are given in the data sources to segment videos into elementary shots, each of which contains a scene occurring in continuous time. Therefore we extract the "most middle" frame to represent the content of the shot and formed a target image collections which contains about 488,000 keyframes. Each topic consists of a set of 4 example frame images drawn from test videos containing the item of interest in a variety of sizes to the least possible. For each frame image there is a binary mask of the region of interest to identify the item of interest. The topic targets include mostly small and large rigid objects, logos, and people/animals.

3.3 Evaluation Procedure



(a) Oxford building



(b) Paris6k



(c) Flickr Logo

Fig. 3.3 Image samples from test collections during our experiments

BBC EastEnders [65] dataset is provided for use in benchmarking the performance of video retrieval research systems. It is appropriate for our evaluation purpose, which is to measure the performance of instance search systems in unstructured video collections. We use this test collection in both our INS experiment and VSS experiment.

Figure 3.3 displays some typical image samples from the test collections. It demonstrates some of challenges in order to built successful instance search systems such as the substantial variations in scale, viewpoint and lighting conditions when capturing the instance.

3.4 Conclusion

In this chapter, we present the research methodologies to help us achieve the research goals during this work. We discuss the experiment configuration which is designed to evaluate the effectiveness of our proposed methods for each research question. Finally we describe the test collections used in the experiments during this work, and also discuss the reason and benefits for choosing these collections.

In the next three chapters, we present our research work on improving instance search for video collections and show the experiment results to demonstrate the effectiveness of our proposed work.

Chapter 4

Weak Geometric Correlation

Consistency for Instance Search

A successful application of instance search requires efficient retrieval of instance images with high accuracy, possibly under various imaging conditions, such as rotation, viewpoint, zoom level, occlusion and so on, or even combinations of those conditions.

Most of the state-of-the-art approaches([69], [70], [44]) for this problem are developed based on the Bag-of-Features (BoF) framework first introduced by Sivic and Zisserman [88] in 2006. This framework successfully uses the discriminative power of the local feature descriptors (e.g. SIFT [51], SURF [13]) which are generally robust to the changes in image condition and applied to build a statistical representation of each image in the database to be searched. At query time, the bag of features representation takes advantage of text-based indexing techniques such as inverted files [120] to provide fast retrieval speed, even for large collections. However this representation leads to a loss of the ability to encode the spatial information between local features, so spatial verification [70] technologies were designed to refine the initially returned results, thus improving retrieval accuracy. The general idea

behind the spatial verification technologies was to estimate a full spatial transformation between query object and object instances and then apply this spatial transformation to filter out those false matches which damaged retrieval performance. However those techniques, such as RANSAC [24], are normally computationally expensive and time-consuming to implement, and they can only be applied as a post-processing step to a few top ranked results in the initial ranked list.

In this chapter, we address the challenges of improving the efficiency and robustness by investigating geometric information local features and examining the feature matching consistency to enhance the retrieval performance of instance search applications.

RQ 1 How much can we improve the effectiveness of the Bag of Feature framework for instance search, if we make use of the geometric correlation information among local features?

To improve the effectiveness of such instance retrieval systems while avoiding the computational complexity of the re-ranking stage, we explore the geometric correlations among local features and incorporate these correlations with each individual match to form a transformation consistency in rotation and scale space.

Based on the observations that there can be only one local feature correspondence to any given feature from a query object, we argue that the geometric correlation between reliable feature matches should also be consistent with weak geometric constraints, just like each individual feature match. So in this work, we propose a scheme to incorporate geometric correlations between matched feature correspondences to form a transformation consistency to improve the effectiveness of spatial verification. Different from the existing spatial verification technologies, our proposed approach avoids the most time-consuming step and does not build a full spatial transformation consistency. Instead, we only estimate the

spatial transformation by building a weak geometric correlation consistency in the rotational and scale spaces to effectively eliminate inconsistent feature matches. This weak geometric correlation consistency can be used to effectively eliminate inconsistent feature matches at a low computational cost and can be applied to large-scale instance search.

We propose a novel spatial verification strategy, weak geometric correlation consistency, to incorporate the geometric correlation among local features to improve the retrieval performance for instance search. As we show in our experimental results in Section 4.3, our proposed approach contributes to the Bag of Features approach to instance search by making the spatial verification more tractable and also more reliable when refining the initially returned results, even for large test collections. Experimental results on three standard evaluation benchmarks also suggest that our proposed approach leads to an overall significant improvement in the instance search performance compared to recently proposed methods.

This chapter is organized as follows. In section 4.1, we start our research work by formalizing the research problem and exploring some related work. Then section 4.2 presents the motivation and implementation of our proposed approach, followed by a discussion of its computational complexity. We design comprehensive experiments to evaluate the proposed approach including comparing the retrieval performance to state-of-the-art methods in section 4.3. Finally, we conclude this chapter by answering our research question 1, and we present our discussion for further work in section 4.4.

4.1 Introduction

Recent work by Jegou et al. [39] proposes a novel approach to efficiently apply spatial verification as a post-retrieval filtering stage for instance search and makes it suitable for large video collections. They used the weak geometric constraints, specifically in the

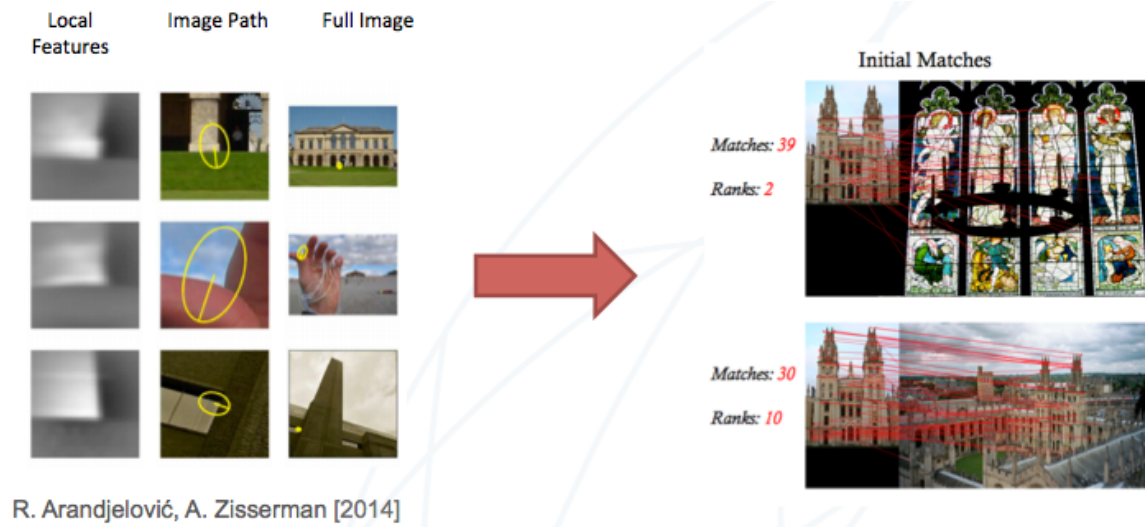


Fig. 4.1 False local feature matching example and how they damage the ranking result

rotational and scale spaces, to examine each individual feature match and to filter out those inconsistent feature matches at a very low computational cost. Although it improved retrieval performance for instance search, we observed that their approach considered feature matches independently and ignored the geometric correlation between local features, thus performing less effectively when searching more challenging datasets like FlickrLogos-27 [75]. In this work, we believe that the geometric correlation between reliable feature matches should also be consistent with the weak geometric constraints, just like each individual feature match. Based on that, we propose a scheme to incorporate the geometric correlations between matched feature correspondences to form a weak geometric correlation consistency to improve the effectiveness of spatial verification.

4.1.1 Problem Formulation

In the bag of feature framework, local features are firstly extracted from each image to encode the invariant visual information into feature vectors. Generally, a feature vector is defined as $\vec{v}(x, y, \theta, \sigma, q)$, where variables $\{x, y, \theta, \sigma\}$ stand for the local salient point's 2-D spatial

location, dominant orientation, and most stable scale respectively, while q represents a 128-D feature vector to describe the local region. For a query image I_q and a candidate image I_c , a set of initial matching features $C_{initial}$ could be established by examining the feature vector q . The task of spatial verification is to eliminate unreliable feature matches and only retain the matches set C_{stable} that linked the patches of the same object. The following equation formalizes this process:

$$C_{stable} = \{m_i \in C_{initial} \quad \text{and} \quad f_{sp}(m_i) = 1\} \quad (4.1)$$

where m_i stands for the i^{th} feature match in the initial match set. f_{sp} stands for the spatial verification function for assessing its geometric consistency. Take the weak geometric consistency [39] for example, verification function in their work could be expressed as following:

$$f_{sp} = \begin{cases} 1 & \text{if } \Delta\theta \in D_\theta \text{ and } \Delta\sigma \in D_\sigma \\ 0 & \text{if otherwise} \end{cases} \quad (4.2)$$

where $\Delta\theta$ and $\Delta\sigma$ is the geometric transformation for an individual feature match and D_θ and D_σ is the dominated transformation in orientation and scale space.

4.1.2 Related work

In this section, we briefly review the development of visual instance retrieval systems and discuss the existing approaches to improve retrieval performance with geometric information.

Sivic and Zisserman [88] firstly addressed the instance search problem using the bag of visual word representation combined with scalable textual retrieval techniques. Subsequently, a number of techniques have been proposed to improve retrieval performance. Philbin

et al. [69] suggested using very high dimensional vocabulary (one million visual words) during the quantization process. This method has improved the retrieval precision with more discriminative visual words, and also increased the retrieval efficiency with more sparse image representations, especially for large scale database. Chum et al. [18] brought query expansion techniques to the visual search domain and improved instance recall by expanding the query information. For further improvement on retrieval performance, both approaches added the spatial verification stage to re-rank the results in order to remove noisy or ambiguous visual words. Recent works in, for example, Wu et al. [105], Zhou et al. [119], Romberg and Lienhart [74] and Albatal et al. [1] extended the bag of visual words approach by encoding the geometric information around the local features into the representation and refine the matching based on visual words. Those methods were very sensitive to the change in imaging condition and made them suitable only for partial-duplicate image search.

Recently, alternative approaches have been developed to implicitly verify the feature matches with respect to the consistency of their geometric relations, i.e., scaling, orientation, and location, in the Hough transformation space. Avrithis and Tolias [3] developed a linear algorithm to effectively compute pairwise affinities of correspondences in 4-dimensional transformation space by applying a pyramid matching model constructed from each single feature correspondence. The work from Jegou et al. [39] increased the reliability of feature matches against imaging condition changes by applying weak constraints to verify the scaling and orientation relations consistency according to the dominant transformation found in the transformation space. Similarly, Jain et al. [38] proposed to represent the feature points geometric information using topology-base graphs and verified the spatial consistency by performing a graph matching.

Our proposed method follows the direction of implicitly verifying the feature matches to reduce the computational cost. Compared to the existing work that focused on individual correspondences, our proposed method also considers the spatial consistency for the geometric correlations between matched feature correspondences, while maintaining the efficiency and increasing the effectiveness of the instance search systems.

4.2 Weak Geometric Correlation Consistency

In this work, we believe that the geometric correlation between reliable feature matches should also be consistent to the weak geometric constraints, just like each individual feature match. Based on that, we propose a scheme to incorporate the geometric correlations between matched feature correspondences to form a weak geometric correlation consistency to improve the effectiveness of spatial verification.

4.2.1 Motivation

We take the geometric correlation among local features into consideration and hypothesize that the pairwise geometric correlation between consistent matches should also be consistent and should follow the same spatial transformation between objects. Instead of verifying the geometric consistency for each match individually, we propose a novel approach to check the consistency between pairwise geometric correlations along with their corresponding local features. Current existing work [3] apply strong constraints on geometric correlations to require transformation followed in 4-dimensional space, which is very computationally inefficient and time-consuming. The main motivation for this work is to form weak constraints to reduce computation and improve performance while keeping the approach sufficiently

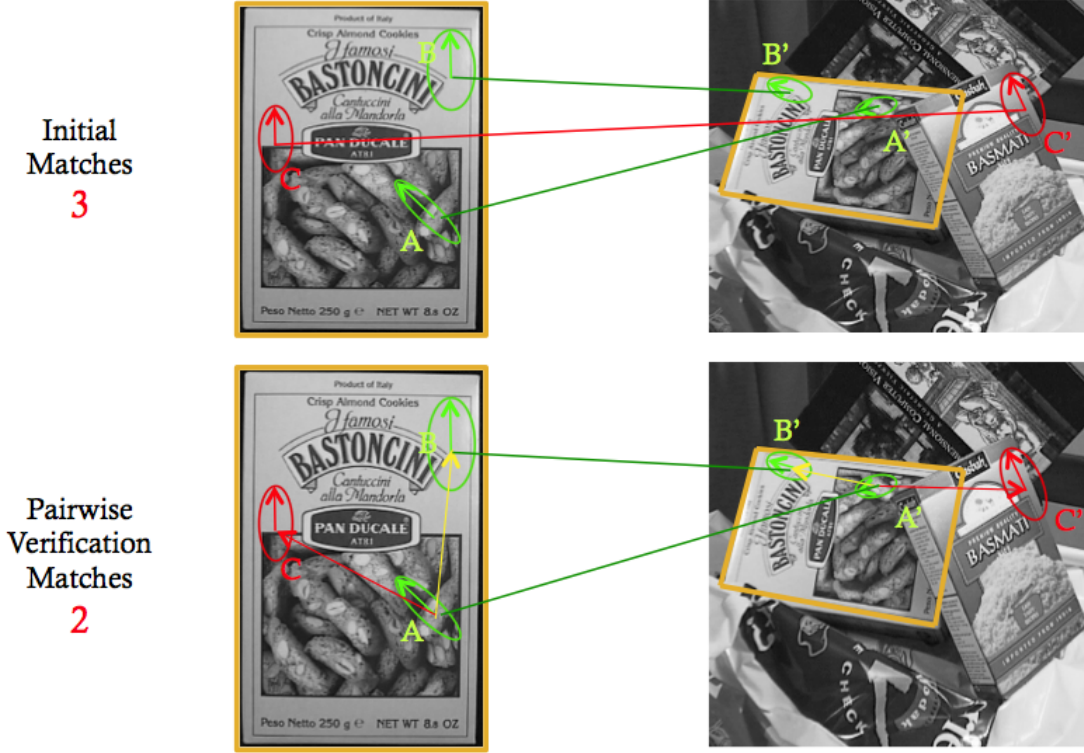


Fig. 4.2 An illustration of verifying the consistency of feature matches using geometric correlations. The green (red) line indicates the consistent (inconsistent) feature matches.

robust to reject inconsistent local feature matching. We name our proposed approach as Weak Geometric Correlation Consistency (WGCC).

For a given pair of feature matches m_l and m_n , we then define the spatial verification function as follows:

$$f_{sp} = \begin{cases} 1 & \text{if } \Delta\theta, \Delta\theta_{l \rightarrow n} \in D_\theta \text{ and } \Delta\sigma, \Delta\sigma_{l \rightarrow n} \in D_\sigma \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

where $\Delta\theta_{l \rightarrow n}$ and $\Delta\sigma_{l \rightarrow n}$ represent the spatial transformation of the geometric correlation from feature match m_l to m_n .

Figure 4.2 demonstrates our idea of using geometric correlations to assess the reliability of feature matches. The object of interest (front cover of a box) is highlighted with a dark yellow box. To begin with, we have three initial feature matches for spatial validation. Matches (A, A') , (B, B') are considered to be consistent because the spatial transformation is consistent between $\text{match}(A, A')$, (B, B') and their correlation $(AB, A'B')$. On the other hand, $\text{match}(C, C')$ is filtered out due to the fact that the geometric correlation between $(AC, A'C')$ is not consistent with the spatial transformation. Hence, we can successfully eliminate the inconsistent feature matches despite the fact that they may obey weak spatial constraints individually.

4.2.2 Implementation

To explicitly examine all the correlations between the initial feature matches is a non-trivial problem. If we take a total number of N initial matches as an example, the potential pairwise correlation could be modeled as $O(C_N^2)$. The initial feature matching number N is usually large in practical systems, and this will cause a high computational cost to verify all the correlations which makes this solution less attractive for large image collections.

In this work, we propose a three-step scheme to reduce the complexity of verifying the geometric correlation consistency and to make it applicable at low cost for large-scale instance search systems. The key idea is to obtain a feature match as a reference point between the initial set of feature matches and then examine only the $O(N)$ correlations between each match and the reference match. These three steps are described in the following paragraphs and an example output for each step is shown in Figure 4.3.

Estimating weak geometric constraints. To begin with, we establish a weak geometric transformation, specifically rotation and scaling, in the spatial space from the initial set of

4.2 Weak Geometric Correlation Consistency

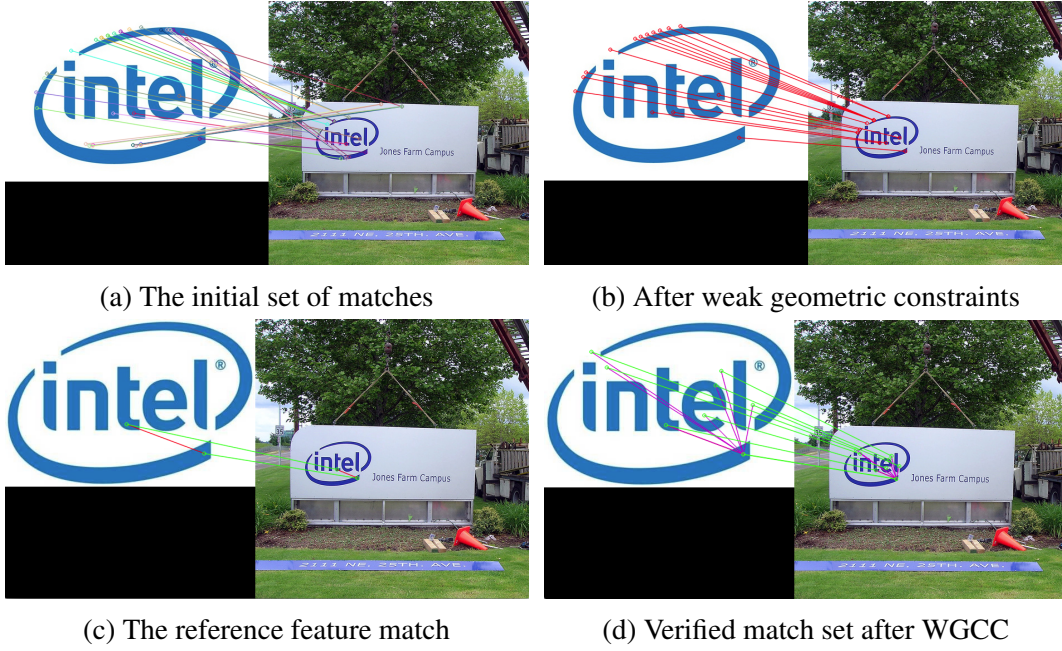


Fig. 4.3 An illustration of applying WGCC on the initial set of feature matches to obtain consistent feature matches.

feature matches. The transformation parameters, rotation angle $\Delta\theta$ and scaling factor $\Delta\sigma$ for each feature matching, were denoted as:

$$\Delta\theta = \theta_m - \theta_i, \Delta\sigma = \sigma_m / \sigma_i \quad (4.4)$$

In order to reduce the sensitivity to non-rigid deformation, we quantize the value of the parameters into bins to estimate an approximated transformation. We use a factor of 30 degrees to divide the rotation range of 360 degrees into 12 bins, and a factor of 0.5 to divide the scale range between 0 to 4 into 8 bins. To avoid any possible bin quantization error, each feature matching votes to the closest two bins in each parameter space. The Hough voting scheme is applied in searching the dominant value D_θ and D_σ to form weak geometric constraints for two purposes. Firstly, we can reduce the computational complexity of the following process by eliminating matches which do not obey the constraints. Secondly,

4.2 Weak Geometric Correlation Consistency

these weak constraints will be used to assess the transformation consistency for geometric correlation to obtain reliable matches.

Identifying the reference matching correspondence. In this step, we aim to determine the strongest feature matches which will serve as a reference match in the step of verifying geometric correlations. We follow the approach of Jain et al. [38] and adopt a topology-based graph match for this purpose. To represent the topology structure for objects, we create Delaunay Triangulation mesh from the geometric layout among the feature points in the object plane. This allows us to then find the strongest feature matches which correspond to the common edges between topology graphs by performing a graph match.

Verifying weak consistency for geometric correlations. The last step focuses on identifying those reliable feature matches by verifying the consistency of the geometric correlations from each feature match to the reference match. Assuming we have a feature match m_l and a reference match m_n between images Q and D , the geometric correlation from m_l to m_n in image Q could be expressed as vector $v_{l \rightarrow n} = (x_l, y_l) - (x_n, y_n)$ where x, y represent the 2D location of corresponding feature points in image Q for matches m_l and m_n respectively. Similarly we can express the geometric correlation between m_l and m_n in image D as vector $v'_{l \rightarrow n} = (x'_l, y'_l) - (x'_n, y'_n)$. Then the transformation parameters in orientation $\Delta\theta_{l \rightarrow n}$ and the scale $\Delta\sigma_{l \rightarrow n}$ between geometric correlations can be defined as:

$$\Delta\theta_{l \rightarrow n} = \arccos \frac{\|v_{l \rightarrow n}\| \|v'_{l \rightarrow n}\|}{v_{l \rightarrow n} \cdot v'_{l \rightarrow n}}, \quad \Delta\sigma_{l \rightarrow n} = \frac{\|v'_{l \rightarrow n}\|}{\|v_{l \rightarrow n}\|} \quad (4.5)$$

It is now possible to assess the spatial consistency by verifying the transformation parameter values with the weak constraints according to the equation 4.3 and to further filter out the inconsistent matches to obtain the final set of reliable feature matches.

4.2.3 Computational Complexity

The major computational cost in the proposed scheme is the second step where we build the triangulation mesh and discover the reference matches by identifying the common edges. These computations are closely related to the total number of feature matches. The benefit is that we already build weak geometric constraints in the first step in order to verify the initial feature matches, so only a subset of a smaller set of feature matches (the cardinality of this set is denoted by n) need to be conducted in this step, which leads to a cost of $O(n \log n)$. In the end, $O(n)$ operations are required to perform the geometric correlation verification which is much less than the $O(C_n^2)$ required for a full verification of all possible geometric correlations.

4.3 Experiment and Discussion

The goal of this experiment is to evaluate the benefits of weak geometric correlation consistency approach at improving the retrieval performance for instance search tasks. In order to do that, we implement a baseline system for instance search based on the classical Bag of Features framework [88]. Moreover, we also take other recent developments into consideration and develop advanced components to support more advanced instance search task. For example, we employ the high dimensional vocabulary approach ([69], [70]) to improve the discriminative power of each individual visual word. Then we apply our proposed strategy to improve the core ranking algorithm and assess its performance improvement considering both retrieval accuracy and response time. We also compare our method against two recent approaches [39] [38], which also aim to improve instance search with spatial verification, to further evaluate the effectiveness of our proposed approach. To assess the robustness of our

proposed approach across different datasets, the experiment was carried out on three standard and publicly available test collections, which are commonly used to evaluate instance search systems. In the following of this section, we introduce the three chosen benchmark datasets, we describe the evaluation protocol and we analyze the experimental results by comparing them to the baseline and to state-of-the-art approaches.

4.3.1 Test Collections

To evaluate the retrieval performance for instance search systems, we employ the following three test collections available for public use. Each of these collections includes standard query topics and an extensive associated manually-annotated ground truth data.

The Oxford buildings: This dataset [69] contains 5,062 high-resolution images crawled from Flickr using texture queries for images of famous Oxford landmarks. 11 building topics with 55 image queries are provided with manually annotated ground truth for users to evaluate the retrieval performance. An image is considered to be positive if it contains an object instance and more than 25% of the instance is clearly visible.

FlickrLogos 27: This dataset [75] consists of 5107 images including 810 annotated positive images corresponding to 27 classes of commercial brand logos and 4,207 distracting images that depict its own logo class. It is a very challenging dataset because the positive images share much more visually similar regions with the distracting images and have more noisy backgrounds. For each logo, 5 query example images are given for evaluation purpose.

Paris6K: This collection [70] consists of 6,412 images collected by searching for particular Paris landmarks from Flickr. In total, 11 landmarks with 55 images queries are provided with manually annotated ground truth for users to evaluate retrieval performance. An image

Table 4.1 Statistics for the three benchmark collections

Dataset	Number of images	Number of features	Number of query topics
Oxford Building	5,062	16,334,970	11
Paris6K	6,300	20,219,488	11
FlickrLogo-27	5,287	2,193,385	27
Total	16,649	38,747,843	49

is considered to be positive if it contains an object instance and more than 25% of the instance is clearly visible.

Table 4.1 presents an overview of the number of images, the number of features and the number of query topics for each collection. In summary, more than 16,000 images and nearly 40,000,000 local features were extracted. Almost 50 query topics were performed to evaluate the retrieval performance for every approach. Figure 4.4 gives some example images from the three test collections. These three datasets cover a wide range of real word instance objects, from landmarks with complex texture to logos with very simple pattern. Apply evaluation experiment over these three datasets could successfully demonstrate robust performance of our proposed system.

4.3.2 Evaluation Protocol

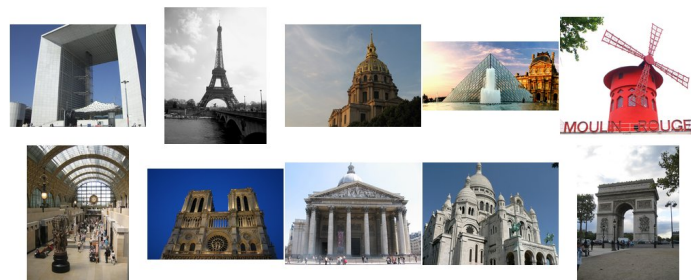
To begin with, we build the baseline instance search system following the standard components in the Bag of Features framework [69]. For all the database images, we use a local feature detector to locate the stable local visual patch and extract a histogram of local gradients to represent the corresponding visual pattern. Subsequently, we apply feature quantization algorithm to each local feature to its closest visual word and build the indexing construction according to a visual vocabulary trained with the approximate K-means algorithm [69]. After that, these visual words (along with auxiliary information, e.g. the



(a) Oxford Buildings



(b) Flickr Logos



(c) Paris Landmarks

Fig. 4.4 Sample images from three test collections

geometric information) are indexed in an inverted structure for supporting the retrieval process.

Secondly, for the instance search task, the system takes a query image for a given object topic and returns a ranked list of images which should contain an instance of the given object. In order to run the query topics on each test collection, we generate Bag of Feature representation for every query image as well. Then the database images will be ranked using an inverted index-based search engine with a weighted algorithm. Spatial verification technologies could then be applied to refine the initial results.

Finally, we measure the retrieval performance for each system and compare their results to assess the benefits. In particular, to measure the effectiveness of a given system, we compute the Average Precision (AP) score of the top 1,000 results for each of the query topics specified in the three test collections, and then we average those scores to obtain a Mean Average Precision (MAP) score for all the query topics. In addition to the MAP, we display precision-recall curves which sometimes can better illustrate the success of our system in improving recall. To evaluate the retrieval efficiency, we record the average response time accurate to one hundredth of a second.

To minimize the effects of variables that may apply to the results, the experiments are carried out on a desktop computer with a 4-core 2.3 GHz CPU and 8G RAM. Only one core was used when performing the query task.

4.3.3 System Settings

In order to apply our proposed approach to improve retrieval performance, we build a search engine to support instance search tasks over thousands of images, which includes the standard components of the classical Bag of Feature framework.

Search Engine

This search engine will take an image as a search query and then sort every image in the corpus according to the likelihood that the image may contain an instance of the topic object. We build the search engine using the vector space model in information retrieval, which is widely used in text search. The Hessian detector and SIFT descriptor implemented in the OpenCV Library [15] are used to extract the local features from database images. We choose this standard library for feature extractions since it provides the most stable performance. In

addition, other researchers can easily repeat our experiment to better understand our proposed method. Subsequently, a visual vocabulary with 1 million visual words is generated using the approximate K-means algorithm [69] to quantize each feature into a visual word to build the bag of features representation. For any given query topic, the search process calculates the similarity between the query vector and the image vector representation. We employ the standard $tf \times idf$ weighting algorithm [53] to calculate the discriminative power of each visual word and then computed the relevance score for images in the collection.

Approaches for Comparison

In order to demonstrate the effectiveness of our WGCC approach, we evaluate the performance improvement of our proposed approach by directly comparing the search results to the baseline approach and two recently created and effective approaches on the shared test collections.

We evaluate the performance improvement of our Weak Geometric Correlation Consistency (WGCC) by comparing the instance search results against the standard RANSAC [69] approach as the baseline, and also against two advanced approaches, namely Weak Geometric Consistency (WGC) [39] and Delaunay Triangulation(DT) [38]. We choose those approaches, since they all aim to use the geometric information to enhance the retrieval performance, but in different ways. Importantly, we implement those approaches with same parameters according to their description in corresponding papers and we find that the results are as expected.

Baseline [69] The baseline approach follows the work described by [69], and estimates a full spatial transformation with the RANSAC algorithm to eliminate false feature matches. We select the top 200 results while considering the trade-off between performance im-

provement and time consumed. We choose this approach as the baseline since it applies strong geometric transformation constraints, which significantly improve the precision of the rank list. However, this method could only be applied to a fixed number of top-ranked results due to the high computational cost. Our proposed approach applies weak geometric transformation constraints to reduce computation and can be applied more effectively.

WGC [39] The weak geometric consistency method builds weak constraints in rotation and scale space to verify the consistency for each matched feature pair by avoiding the full spatial transformation estimation. The constraints for geometric consistency are obtained by converting the parameter values into Hough transformation space. However it does not take the geometric correlation between local features into consideration, so false matching could get passed especially when the texture of the topic object has repeated patterns, such as some logo objects. This approach is selected for performance comparison since it verifies each local feature match with weak geometric consistency. However, it only checks the transformation consistency between individual matching pairs without considering the correlation among them. In contrast, our approach also takes the correlation among local features into consideration to increase the robustness for spatial verification.

DT [38] Delaunay Triangulation (DT) is a most recent development on applying spatial verification in instance search systems. More specifically, this approach makes use of the Delaunay Triangulation (DT) based graph representation to model and matches the layout topology of initially matched feature points. In order to build the topology graph, a hamming embedding signature was used to enforce point-to-point matches and ensure the number of nodes in each graph is identical. We choose this approach to compare against since it aims to reduce the cost of applying spatial verification by examining the geometric layout of local

4.3 Experiment and Discussion

Table 4.2 mAP comparison between our proposed WGCC and the baseline and two other state-of-the-art advanced approaches, on the Oxford, Paris6K and FlickrLogos-27 datasets

Methods	Oxford		Paris6K		FlickrLogos	
	mAP	Time ¹	mAP	Time ¹	mAP	Time ¹
Baseline	0.501	1.76	0.530	1.95	0.137	1.26
WGC	0.530	1.06	0.576	1.12	0.193	0.41
DT	0.542	0.86	0.546	0.89	0.201	0.31
WGCC	0.693	1.07	0.607	1.23	0.231	1.06

Time¹ measures the average response time per query in seconds, excluding feature extraction.

features from object appearance. However, this method only considered the relative spatial position among local features without verifying the spatial transformation.

WGCC [113] This is our proposed approach and the novel contribution of this chapter, as described in section 4.2. For the purpose of reducing computational cost, we also build weak geometric constraints from initial matched feature pairs while avoiding the full geometric transformation estimation. Different from work in WGC [39], we also applied the geometric correlation among matched features to verify their consistency.

4.3.4 Results and Findings

In this section, we present our experimental results on evaluating the performance of the four approaches mentioned previously, on three different test collections. We demonstrate the performance improvement in terms of retrieval accuracy of the proposed WGCC approach compared to the baseline and to two advanced approaches. After that, we evaluate the improvement in efficiency. The experiment results suggest that our approach is scalable for large collections.

Improvement in mean Average Precision (mAP) score Table 4.2 presents the retrieval accuracy in mAP score and average response time in second for all the instance search topics on the three benchmarks.

To study the impact of adopting geometric information for enhancing the retrieval performance in an instance search system, we compare our implemented advanced systems against the baseline system over three test collections. We observe that the advanced approaches for spatial verification consistently improve performance in MAP compared to the baseline. The results also demonstrate that our proposed approach achieved the best results, comparing to the other two advanced systems. Especially on the FlickrLogos dataset, our approach has a 59% relative improvement in MAP performance from the baseline's 0.145 to 0.231 in our method. This proves that our approach is sufficiently robust to reject inconsistent feature matches, while also flexible enough to keep the evidence from locally consistent patches. Figure 4.5 gives a more intuitive comparison in MAP score of the four approaches on the three test collections. The figure clearly suggests that our WGCC approach continuously outperforms the baseline approach and the other state-of-the-art approaches in all three test collections. One interesting observation from the results is that the overall performance in the Flickr-logo dataset is much poorer for all approaches. This dataset is more challenging due to the fact that logos normally take only a small area in an image, which makes it more difficult to capture them within the bag of features framework.

Improvement under Precision-Recall Curve We investigate some of the specific query topics to explore the retrieval accuracy improvement for the WGCC approach in more detail. Figure 4.6 shows the precision-recall curves for three example queries to demonstrate the improvement obtained by the proposed WGCC approach compared to the baseline system. The search topics are delimited in the yellow box from the query image on the left side in each

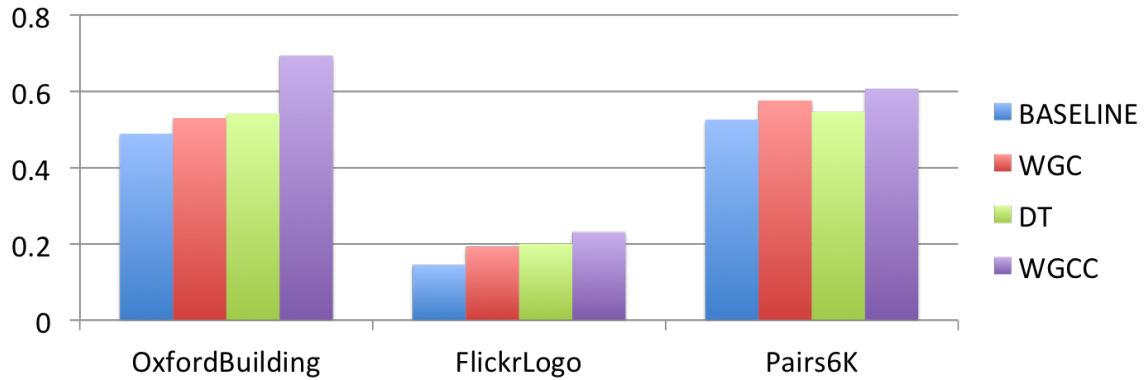
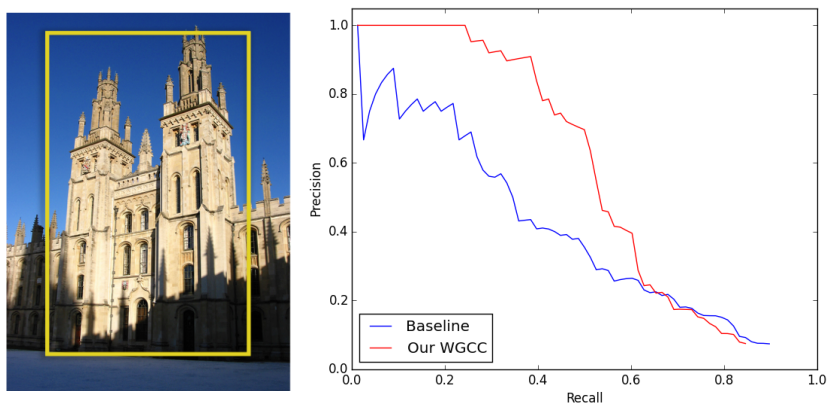


Fig. 4.5 Performance comparison in mAP between different approaches on three test collections

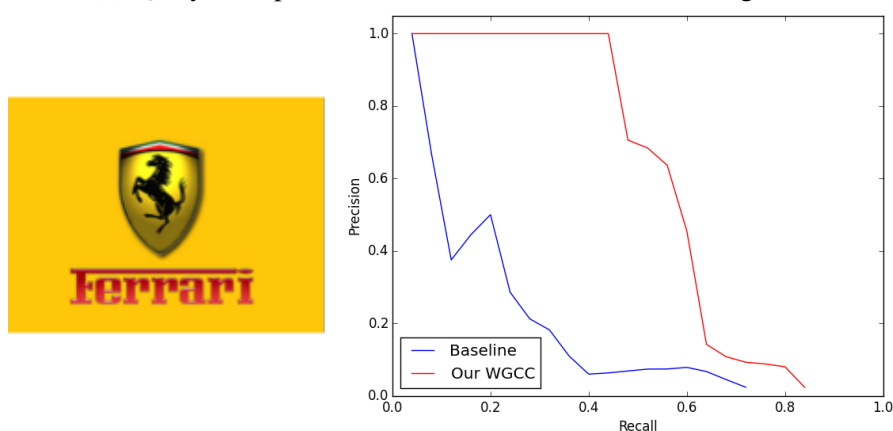
sub-figure and the corresponding precision-recall curve is displayed on the right side with baseline results shown as the blue line and the WGCC method shown as red. The area under the curve indicates the average precision score and the gap area between two lines indicates the performance improvement for our proposed method. In the three cases, the average precision score of our WGCC approach is larger than the baseline approach. (*All_Souls_1* goes from 0.426 AP to 0.774 AP, *FedEx_Logo* increases from 0.133 to 0.613 AP). We believe that our proposed method refined the initial results by successfully re-ranking the relevant images in high order.

Improve in Top-10 results To be even more specific in our analysis, we also compare the top-10 results in the ranked list returned by the baseline approach against our WGCC approach in Figure 4.7. The column on the left-hand side are the query objects. On the right-hand side, the top-10 results are displayed from left to right. For each query topic, the upper rows are the results from baseline system and the lower rows are the results from our WGCC method. The irrelevant result images are indicated with red surrounding boxes and the remainder are relevant results. As shown in the figure, our WGCC method is especially good at returning relevant images among the top-10 results. Since our WGCC approach

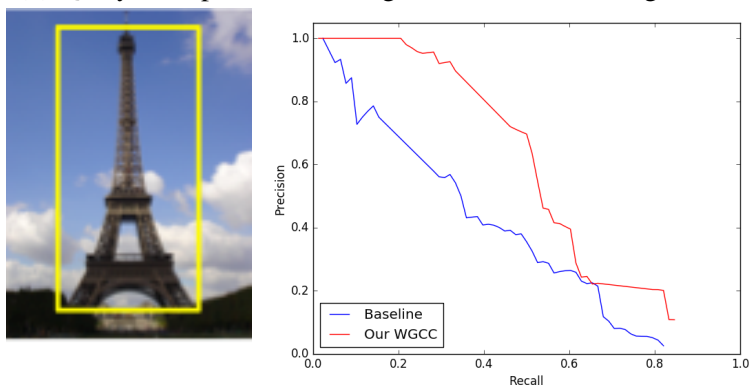
4.3 Experiment and Discussion



(a) Query example: 'All Souls' from the Oxford buildings dataset



(b) Query example: 'Ferrari Logo' from the Flickr Logos 27 dataset



(c) Query example: 'Eiffel tower' from the Paris6K dataset

Fig. 4.6 Examples of the improvement in the precision-recall curve obtained by using our proposed approach compared to the baseline system. The query image was given on the left side of each row, and the precision-recall curve plot was displayed on the right side.

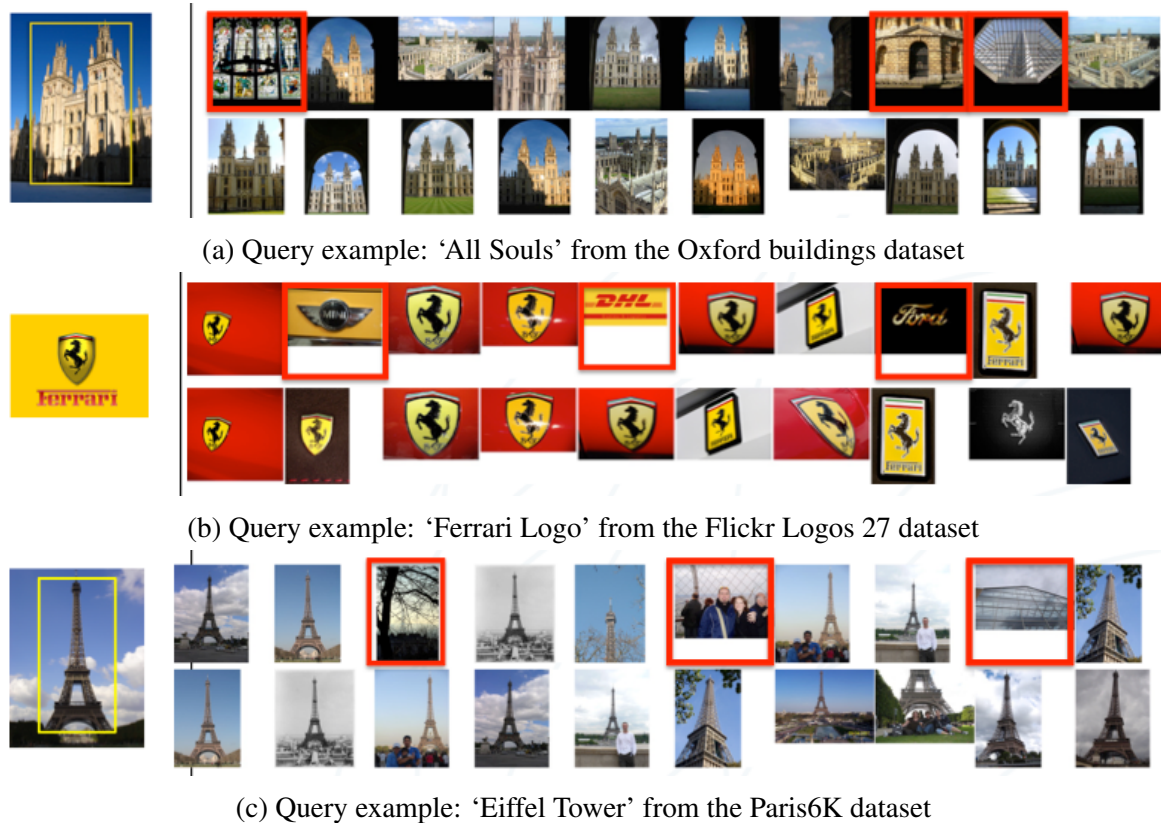


Fig. 4.7 Comparison of Top-10 results in ranking lists. The image on the left column is the query images, and on the right-hand side, the top-10 results are displayed from left to right. For each query, the upper rows are the results from baseline system and the lower rows are the results from our WGCC method. The red box indicates the irrelevant result images.

preserves the geometric layout of the object instance with weak geometric constraints, so our approach would filter out the non-relevant images even if they are considered relevant in initial ranked list.

Improvement in Efficiency We measure the performance of efficiency by recording the response time for all the query topics at run time and average them to obtain an average response time for comparison. Table 4.2 lists the response time for each approach in each dataset. Apparently, three advanced approaches all outperform the baseline approach in terms of efficiency across all three test collection since they all aims to reduce the computation cost. The DT approach obtains the best performance in efficiency, mainly due to the fact

that it compares all the feature matches at once. However, it provides less improvement in accuracy comparing to our proposed approach. In summary, our proposed method achieves the second best retrieval efficiency compared to the three different approaches while providing better retrieval performance. Importantly our approach verifies each local feature matching separately, the efficiency could be further optimized by developing a parallel computing algorithm.

4.4 Conclusion

This chapter introduces a novel approach to improve retrieval performance of standard Bag of Features approach for instance search task. We start the research by diving deep into the Bag of Features framework [88], and find that the geometric information among local features is omitted. So we form weak geometric correlation consistency algorithm [113] to improve local feature match accuracy by combining the pairwise geometric correlations with spatial feature transformations.

The instance search experiments in this chapter prove that our approach is strong enough to eliminate the inconsistent feature matches while keeping the reliable matches using locally spatial correlation. Moreover, it also shows that our approach consistently outperforms the baseline system on three standard image benchmarks and achieved comparable results with two advanced systems. This indicates the effectiveness of our method for spatial verification and answers our first research question. Another positive aspect is that other advanced technologies, such as automatic query expansion [18] and re-ranking based on full spatial verification [69] are compatible with our proposed method and could be used as complementary components to further improve the retrieval performance.

4.4 Conclusion

In future work, there are two problems we would like to address and investigate. Firstly, we want to investigate how to incorporate WGCC methods in large data collections, e.g., a collection with millions of images. And then we are also interested to address the challenges of searching small object instance such as FlickrLogos [75] and improving the retrieval accuracy for these data collection.

Chapter 5

Improving Instance Search for Video Collections

This chapter presents the research work to visually search large-scale video collections for object instances specified by an image query. Like the previous chapter, the topic of this chapter is also improving the performance of instance search. Moreover, this chapter focuses on addressing the specific challenges that arise when organizing and searching large volumes of video data. With this in mind, we explore the visual content in video collections and investigate methods to answer our second research question which guides our research in this work:

RQ 2. How much can we improve the performance of instance search systems while applying our feature matching method on video search task?

To answer this research question, we start our research by improving the feature matching efficiency in the classic Bag of Features (BoF) framework, and we then investigate text-based information retrieval approaches to improve the performance of an instance search system for large video collections.

Recent work [18],[69] has shown that the standard bag of feature approaches can suffer from poor recall when applying to instance search task in large collections. This is mainly because the current feature representation could not precisely describe the different visual pattern and retrieve noisy image patches, which are crucial to instance search from large databases. So in this chapter, we decide to enhance the feature representation with an embedded coding scheme and build scalable components to handle large video collections.

Specifically in section 5.1, we implement a nearest neighbor search algorithm to address the problem of improving feature matching for large datasets. This algorithm helps us to effectively integrate our object matching technology into the retrieval process and enables the ability to refine ranking results, even for very large datasets. Following that, in section 5.2.1 we study several query expansion technologies which could be used to expand and reformulate the initial search query to match additional documents to further improve retrieval performance. We implement a form of query expansion to refine the original query topic by aggregating the information from multiple relevant result images. This technology could help us to capture any missing features of the object query from only one image and could help to boost retrieval performance with more relevant results.

To evaluate retrieval performance, in section 5.3 we build a complete instance search application and participated in the TRECVID Video Retrieval Evaluation [65] (TRECVID), the largest and most accepted evaluation benchmark for content-based analysis of retrieval for digital videos, in order to demonstrate the effectiveness and efficiency of our proposed approach. We are motivated to participate in the TRECVID evaluation workshop since it helps us by providing a standard video data collection with query topics, ground truth data and an open, metrics-based evaluation protocol. For the years 2013, 2014 and 2015, we participated in this evaluation workshop to assess the performance of our proposed approaches.

5.1 Improving Feature Matching with Embedded Coding

The experimental results obtained from TRECVID participation suggest that our approach is reliable and robust, and successfully improve the instance search performance for large video collections. Our first contribution, the embedded coding scheme, improves the scalability and the accuracy of the performance of the bag of features framework. Moreover our second contribution, query expansion implementation, significantly improves the recall of the instance search tasks and found many additional relevant results with the reformulating search queries.

5.1 Improving Feature Matching with Embedded Coding

In this section, we investigate the weakness that the bag of features approach raised when scaled to very large video collections for instance search tasks. We especially focus on improving the feature matching performance by enhancing the visual word representation with embedded coding. As a result, we extend the scalability of the Bag of Features approach to support large video collections, which leads to the improvement of its retrieval performance.

5.1.1 Weaknesses of quantization-based approaches

To address the problem of instance search tasks, Sivic and Zisserman [88] introduced the classic Bag of Feature (BoF) model inspired by text-based search algorithms in the context of image search. This model uses powerful local feature vectors to describe the object instance or scene, even under different imaging conditions. In order to support large data collections, they employed a clustering algorithm, such as K-means, to quantize local features into visual

5.1 Improving Feature Matching with Embedded Coding

words and then they built an inverted file indexing system to provide fast access to the visual words for fast search speed.

Quantization Error During the search process, instead of directly matching individual feature descriptors, two image features are considered identical if they are assigned to the same visual word (cluster center). On the other hand, two features, assigned to different (even very close) clusters, are considered totally different. This could be understood as an approximation of the real matching between individual descriptors and provide fast search speed while sacrificing some feature matching accuracy. In practice, this quantization technology, also known as “hard assignment”, could lead to missing the good matches between local features and introducing noisy feature match correspondences, which could damage the retrieval performance for the instance search system.

In the literature concerning instance search tasks, this error is been called “Quantization Error”. The “quantization error” has existed ever since the Bag of Features model was created and some recent extensions of the approach have been proposed to address this problem. The work of Philbin et al. [69] presents a method of building a large visual vocabulary to improve the discriminative power of the visual words in order to improve the feature matching accuracy. Although this approach outperformed the classic Bag of Features approach, it was a very time-consuming task to build for large video collections and the actual size of the vocabulary always needs to be tuned for the target data collections. In the soft assignment work, Philbin et al. [70] explore technologies to map each image feature to a weighted set of words which allows the reconnection of undiscovered local feature pairs in the quantization stage. But this approach changes the representation of an image feature, which requires changing the ranking algorithm.

5.1 Improving Feature Matching with Embedded Coding

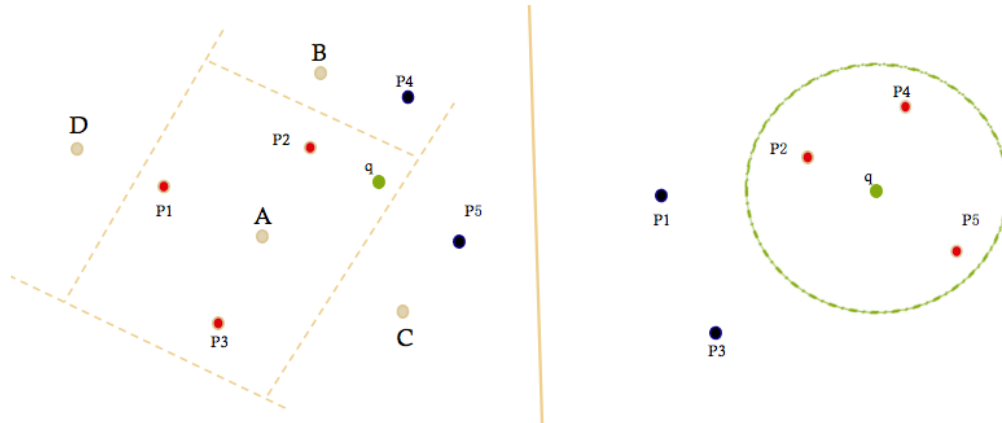


Fig. 5.1 Demonstration of “Quantization Error” and motivation of our method

In our work, we also address this problem by improving the visual matching accuracy with the combination of enhanced representation and nearest neighbor search algorithm. Our enhanced feature representation could help to improve the discriminative power of each visual word and allow feature matching within the same cluster. Then the nearest neighbor search algorithm helps to provide a list of good candidates from nearby clusters for re-ranking.

5.1.2 Motivation

In this section, we present the motivation for the embedded coding algorithm to address the “Quantization Error” problem in the context of instance search for large video collections. We argue that the performance of instance search can be improved by explicitly calculating the distance between local features within the entire feature space. In contrast to the standard bag of feature approach within which false feature correspondences could be introduced by assuming two features are matched if they were assigned to the same visual word, our algorithm could prevent this type of error and improve the quality of the matched feature correspondences.

5.1 Improving Feature Matching with Embedded Coding

Figure 5.1 gives an example of “quantization error” on the left-hand side and illuminates the motivation of our proposed approach on the right side. Suppose we have one local feature q (shown as the green dot) from the query object and five local features $P1, P2, P3, P4$, and $P5$ from object instances in the test collection. The left part of the figure demonstrates the approximate matches from the standard Bag of Features framework. The local features $P1, P2$, and $P3$ are considered to be positive matches to the query feature q because they have all been assigned to cluster center A . In contrast, the right part of the figure demonstrates our proposed approach. The local features $P2, P4$, and $P5$ are considered to be positive matches because they have the closest distance to the query feature q in the vector space. The object instance which contains the local features $P2, P4$ and $P5$ should be considered as good candidates. So in this chapter, our work is specifically interested in investigating a method to search the nearest local features for each query local features and building up an instance search system to improve retrieval performance over a baseline system..

5.1.3 Implementation

Here our primary task is to implement an efficient algorithm for feature matching with a low computational cost which makes it scalable for very large scale instance search tasks. In the instance search task, the target of feature matching is to calculate the distances between the query local features and all the database local features and built up the feature correspondences for a ranking algorithm. So, we convert the feature matching into a nearest neighbor search problem in high-dimensional feature space. For example, the SIFT [51] descriptors, the feature we use in this work, contains a 128-D feature vector. We now present an efficient nearest neighbor search component based on enhanced feature representation with embedded coding.

5.1 Improving Feature Matching with Embedded Coding

Algorithm 1 Feature matching algorithm based on Embedded Coding

```
Ensure: float  $Scores[N] = 0$ 
1: for query descriptor  $q_i \in qxD$  do
2:   calculate the top  $K$  nearest local features
3:   for each positive match pair  $(q_i, l \in d)$  in  $K$  list do
4:      $Scores[d]_+ = f(q_i, l \in d)$ 
5:   end for
6: end for
7: for each  $d$  do
8:    $Scores[d] = Scores[d] / Feature\_length(d)$ 
9: end for
10: return Top  $n$  documents of  $Score[N]$ 
```

Our proposed approach takes advantage of a clustering algorithm, such as K-means, to split the feature space into small groups and to narrow down the choice of candidates by calculating the distance to the clustering centroid. Following that, we employ embedded coding algorithm [39] to enrich the standard bag of feature representation, and use product quantization algorithm [42] to build the nearest neighbor searching algorithm to quickly find a list of local feature candidates for given local feature. Finally, we present the pseudo code to compute the nearest neighbor for any given local feature in our proposed feature matching component in Algorithm 1.

In order to test the effectiveness of our embedded coding algorithm for local feature searching, we carry out an experiment to perform feature matching tasks and compare its performance with other state-of-the-art approaches. In this experiment, we use the ANN_SIFT1M data collection [41] which contains 1 million SIFT local features extracted from the INRIA Holidays database [40]. The ground truth is also provided by calculating the Euclidean distance using a brute-force searching algorithm for 100 queries of local features. We evaluate the performance of different search methods using precision at top N results (Precision@N), where $N = 1, 10, 50, 100, 200, 500, 1000$, for all the 100 queries and

5.1 Improving Feature Matching with Embedded Coding

calculated the average performance across all the queries. For performance comparison, we implement the standard Bag of Features approach as the baseline.

We also implement the hamming embedding approach [40] and compare its performance against our approach. Jegou et al. [40] suggests that this approach improves feature matching in the bag of features framework. This algorithm could help to benefit instance search performance by improving the accuracy of local feature matching. It combines a coarse feature quantization with a low number of centroids and a fine feature quantization with a high number of centroids. To increase efficiency, this approach enhances each visual word by adding a binary signature. During the searching process, this algorithm can refine the local feature matching by comparing the distance between their binary signatures. Although both our algorithm and the hamming embedding algorithm aim to refine feature matching, our algorithm not only refines local features within the same cluster group, but also considers nearby groups to further reduce the “quantization error”. We should expect more robust results from our proposed algorithm.

Figure 5.2 presents the results from our experiment and demonstrates the effectiveness of our proposed approach. Our approach outperforms the standard BoF and Hamming embedding approach (HE) in terms of precision at all levels. For the most important top 100 results, the results of our approach achieved 93% precision which suggests the success of the proposed method in local feature matching. Another important observation from this result is that the maximum precision for the BoF approach and Hamming embedding approach only reached 42%. This is mainly due to the “quantization error” where two closely matching local features are assigned to different visual words and their relationship was not considered anymore. We propose approach to addresses this issue with an improved feature

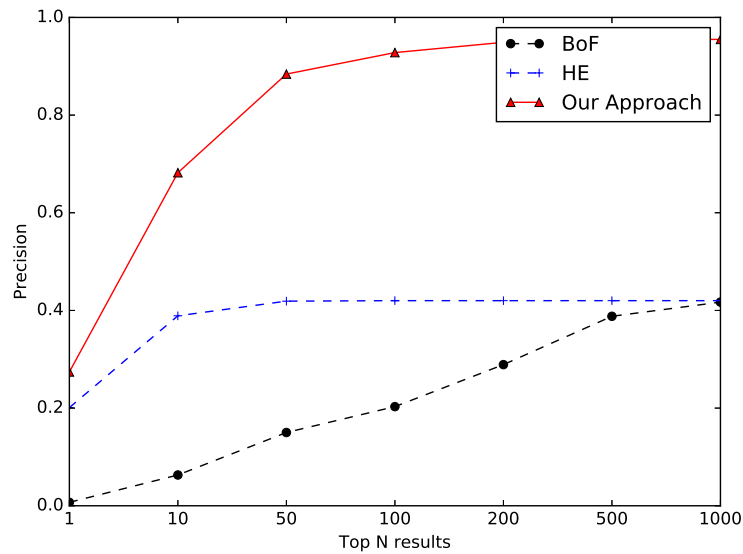


Fig. 5.2 Precision@N curve for local feature matching, results compared between Classical Bof, Hamming Embedding (HE), and our approach

representation and build a re-ranking algorithm to improve local feature matching, which results in almost 98% performance improvement in precision.

In the next section, we take advantage of the our research work and build a scalable instance search system for large video collections. To further improve the retrieval performance, we also present a query expansion technology which refines query topics to find more relevant results.

5.2 Scalable Retrieval for Large Video Collections

In this section, we present our research work on integrating our proposed approach in the bag of feature framework and build instance search systems for large video collections. To further improve the retrieval performance, we also investigate the query expansion technology, which is widely used in text-based information retrieval.

5.2.1 Components of a Scalable Instance Search System

This section overviews the instance search system [111] that we developed for large video collections and describes the main components in order to meet the requirement of scalability.

Video Data Pre-processing

In order to reduce the computing complexity, a common first step in content-based video analysis is to segment videos into elementary shots, each of which contains a scene happening in continuous time, or the same scene. We also extract one frame as a “Keyframe” to represent the visual content for each shot. For each keyframe, the affine invariant interest regions are detected and scale invariant features are extracted. More specifically, we use the ColorDescriptor library [7] and select the Harris-Laplace detector and the SIFT descriptor.

Feature Extraction and Vocabulary Training

The purpose of this step is to aggregate a set of local descriptors into one vector and to represent each keyframe with a fixed-sized feature vector. This work often involves generating a visual vocabulary by using clustering algorithms, such as k-means, and mapping local descriptors into the visual words (i.e. cluster centers) of this vocabulary. The research work [69] shows that a large size vocabulary with more discriminative power is necessary for large scale object retrieval. However, the time complexity of k-means will increase dramatically because there are kxN euclidean distance calculations in each iteration. So we adopt the approximate k-means algorithm [69] and replace the exact distance computation by an approximate nearest neighbor search method.

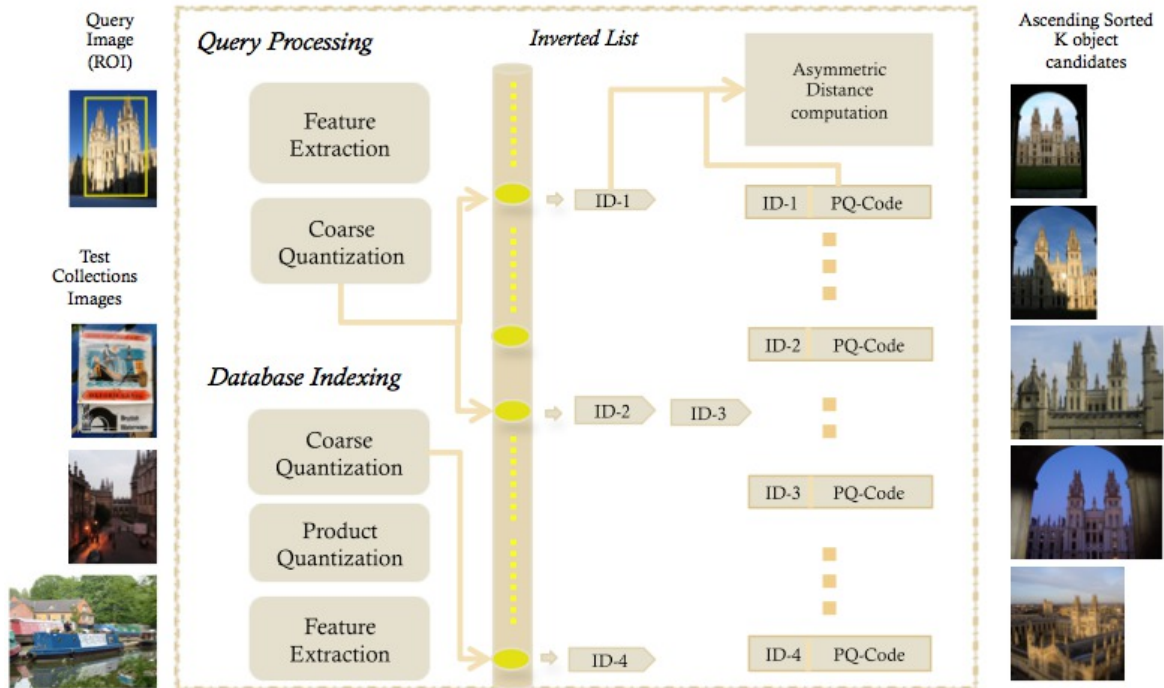


Fig. 5.3 Complete instance search system based on the local feature regions

Search algorithm with Inverted Indexing

The focus of this step is to rank the keyframes according to their visual similarity to the query topics. We use the well-studied vector space model in information retrieval combined with a standard $tf \times idf$ weighting scheme. In this approach, the query topic and each keyframe are represented by a high-dimensional, sparse vector, and the relevance score is calculated using the normalized L2 distance. The open source library Lucene [9] is employed to accomplish the work.

Figure 5.3 illustrates the components of our proposed system for instance search tasks after applying our proposed embedded coding algorithm. We follow the architecture of the work of Sivic and Zisserman [88] and employ an inverted file indexing structure to build the system. For each query image, we apply the same process and then rank the whole corpus to generate a sorted result list according to their visual similarity to the query topics.

5.2.2 Improving recall with query expansion

In the context of information retrieval, query expansion [53] is a technology to refine a search query for improving retrieval performance in text retrieval. In order to increase the quality of search results, this method involves expanding the search query with a number of top ranked documents from the initial results. The refined search query will allow retrieval systems to find more challenging documents by using relevant terms that are not included in the original query. Many works, such as ([106], [100]) have been applied in text-based information retrieval to improve effectiveness of information retrieval. A review of the literature shows that the query expansion technique significantly improves the retrieval performance if the query is not well formalized initially. But it makes little difference in retrieval effectiveness if the original queries are relatively complete descriptions of the information being sought.

The work of Chum et al. [18] illustrates that query expansion methods can be successfully applied in the visual search domain using the bag of features framework. In this work, we explore a novel method to derive a better topic description for any given query image, in order to enhance retrieval performance of our system. We propose a simple but effective approach called average query expansion, to automatically enhance search performance. This method provides the ability to construct a new query by combined the original query with averaging verified results from the initial rank results. Figure 5.4 shows the steps of applying query expansion in our proposed approach. Firstly, we select the top N results returned by instance search engine. A spatial verification step is applied between the query image and original results to filter out any non-consistent results in geometric transformations. The parameter N is normally no greater than 50 since spatial verification methods are very time-consuming. A new query is then formed by averaging the local features from the original query and the top verified results. In contrast to simply extracting query terms from the initially returned

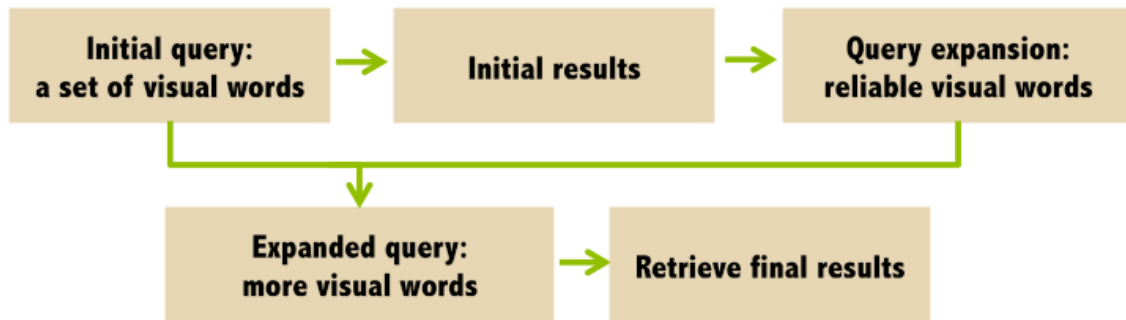


Fig. 5.4 Architecture of query expansion technology

result, our approach studies the consistency of each result before enriching query to ensure that enhanced query could indeed capture additional visual description. We expect to find more matched results with this refined query since it now contains more information about the search instance that may be missed out in the original query. It is important to note that the performance of query expansion depends heavily on the initial results. If the information from the original query image is very comprehensive, then we should expect very little or no performance improvement. If any information that does not belong to the query object has been added into expanded query, we could introduce noisy query features, pollute the original expansion and expect even worse results.

To conclude this section, we present two novel methods to improve the standard bag of features framework retrieval framework for video collection. We firstly propose to enhance local feature representation with embedded coding to improve the feature matching accuracy in the bag of features framework. After that, we implement a query expansion method, averaging query expansion, to further improve the retrieval performance. In the next section, we present our work on evaluating our proposed approaches while participating instance search tasks in TRECVID benchmarking workshop. And we also discuss the performance

improvement by comparing our proposed system with other state-of-the-art approaches for video collections.

5.3 TRECVID Experiments and Discussion of Results

For the purpose of evaluation of the effectiveness of our proposed instance search system for large video collections, we participate in the Instance Search task (INS) in the TRECVID Retrieval Evaluation workshop [90]. In this section, we firstly describe the instance search task in TRECVID. We talk about the evaluation methodologies including the test collection, evaluation measures, and systems for comparison against. Finally, we present the results for our participation (2013–2015) in this evaluation campaign and discuss the key findings after reviewing the results.

5.3.1 Instance Search Task in TRECVID

TRECVID is an international video search benchmarking workshop in which researchers evaluate and demonstrate the retrieval performance of their video retrieval approaches on a shared test collection via an open, metrics-based evaluation. The instance search task focuses on finding video segments which contain a certain specific person, object, or landmark, given a visual query image. During the TRECVID campaigns, 30 topics with ground-truth data were given for each participant to find the top 1,000 shots which are most likely to contain a recognizable instance from a large collection of about 200 hours of video. To evaluate the retrieval performance for each TRECVID submission, the standard evaluation metrics such as Average Precision (AP), mean Average Precision (mAP), Precision at K (P@5, P@10, ...) were computed.

5.3 TRECVID Experiments and Discussion of Results

Table 5.1 Basic statistics for BBC EastEnders test collection.

Number of Videos	Hours	Shots	Keyframes	Size of descriptors
244	464	471,527	471,527	1.47 GB

5.3.2 Evaluation Methodology

In order to evaluate the effectiveness of our proposed approach for finding object instances in large video collections, we firstly build a retrieval system to perform to complete instance search tasks. The system expects image queries as input, and then returns a list of video segments which are most likely to contain a recognizable object instance from the whole video collection. We measure the retrieval performance with evaluation metrics and analyze the performance improvement by directly comparing the results from different approaches.

Test collection

We now introduce the test collection used in the TRECVID evaluation campaigns between the years 2013 and 2015. It contains two main parts: the video data and query topics with ground-truth data. The video collection was selected from the BBC EastEnders TV program and contains approximately 244 video files (totally 300 GB, 464 hours) with associated metadata in MPEG-4/H.264 format. For each video, the master shot reference was also given to segment the videos into shot segments with an average length of 3.5 seconds. Table 5.1 illustrates some basic statistics for this test collection including the number of shots and keyframes. For each year, the INS tasks gives 30 query topics with manually annotated ground-truth data for evaluation. These query topics are carefully selected to represent the possible real-world instance search circumstances and could be divided into three main categories: people, locations, and objects. These topics covered instances with a wide range of imaging conditions, such as various camera positions, different times of year or day.

5.3 TRECVID Experiments and Discussion of Results

To build an instance search system for this test collection, we firstly segment each video into a set of shots according to the master shot file and then extract one keyframe to represent the visual content for each shot. In this case, we generated an abstraction of the video content without losing any information to reduce the computing complexity. For each keyframe image from each shot in the video collection, we extracted the local features from affine-invariant Hessian regions [57]. Typically there are 400-600 affine regions detected on an image of resolution 768×576 . For each of these affine regions, we then computed the 128-D SIFT descriptor [51]. The number of videos and the size of the descriptors for this test collection is also shown in table 5.1. Our experimental systems were built upon those extracted local features.

Evaluation Measures

The target of INS task is to find more video shots which contain the same person, location or object, but not necessarily in the same context, of an interested query topic. So this task is treated as a form of ad-hoc retrieval in which the user specifies a topic query to execute a search and return the documents or video clips which are likely to be relevant to the user's information need. During the evaluation process, each participant is allowed to submit a maximum of 4 prioritized runs for evaluation and the results are evaluated with *average precision* (AP) for each topic in each run and per-run *mean Average Precision* (mAP) over all topics.

5.3.3 Experimental Results and Findings

In the following section, we report the evaluation results for different methods and describe the key findings from our participation in the INS task of the TRECVID campaign. The

5.3 TRECVID Experiments and Discussion of Results

main goal of this experiment is to evaluate the scalability and effectiveness of our proposed approach and we hope our methods could make a contribution for solving the problem of instance search tasks in large video collections.

Experimental Approaches

Here we present the description of our system configuration which was used as the system to participate in the INS tasks in TRECVID.

Bag of Features (BoF) We implement the Bag of Features framework for instance search tasks described in the work of Philbin et al. [69]. To improve the discriminative power of visual words, we train a 1,000,000 dimensional vocabulary for feature quantization using the K-means algorithm implemented by VLfeat library [99].

Embedded Coding The embedded coding algorithm is our contribution to promoting the bag of features approach to address the large-scale instance search problem. It introduces an enhanced visual word representation to improve the feature matching accuracy and then improves the overall performance.

Query Expansion The query expansion technologies are used as a query augmentation technology to further improve retrieval performance. In this work, we implement a query expansion technology in the Bag of Features approach to refine the initial results. Our proposed method could extract a set of additional relevant local features from the top-20 best results to the topic query and help to find more additional relevant results.

WGCC Algorithm Benefiting from the embedded coding algorithm, we also apply our Weak Geometric Consistency Constraints (WGCC) to instance search tasks for large scale video collections. This WGCC algorithm could help us to re-rank the results by using the geometric information among local features to form weak constraints to efficiently filter

5.3 TRECVID Experiments and Discussion of Results

out false feature correspondences. In addition, we also build a system which combined our WGCC algorithm and query expansion technology together. We expect this approach to perform well since the geometric constraints could allow us to accurately verify each returned result, suppressing the false positive results to prevent them negatively impacting on the query expansion technology.

Outline of Experiments and Results

The experiment was arranged into three main parts and implemented as a participation in the TRECVID evaluation campaigns from year 2013 to 2015.

Firstly, we evaluated the retrieval performance of a standard Bag of Features framework [69] in our TRECVID participation in 2013. This result from this state-of-the-art approach served as the baseline and compared to the results from our improved approaches. Following that, we also tested the performance improvement obtained using the query expansion technologies from section 5.2.2.

Then in TRECVID 2014, we focused on evaluating the benefits of the embedded coding algorithm described earlier in section 5.1 for an instance search system. The improved system was built based on the standard Bag of Features framework and we added an embedded coding component to improve the feature matching performance. Furthermore, we also assessed the performance introduced by combining the embedded coding algorithm and query expansion technology.

Finally, during our final TRECVID participation in 2015, we evaluated the performance improvement when combining the embedded coding algorithm and our Geometric Correlation Consistency (WGCC) algorithm together. Our WGCC algorithm considered the geometric information among local features with the aim of helping to refine the initial ranking results.

5.3 TRECVID Experiments and Discussion of Results

Table 5.2 Experimental results for three-year participation in TRECVID

Year		Retrieval System	mAP
2013	i	BoF	0.111
	ii	BoF + Query Expansion	0.120
2014	iii	BoF + Embedded Coding	0.187
	iv	BoF + Embedded Coding + Query Expansion	0.193
2015	v	BoF + Embedded Coding + WGCC	0.191
	vi	BoF + Embedded Coding + WGCC + Query Expansion	0.216

Additionally, we evaluated the system performance after applying all three technologies together.

Table 5.2 shows the experimental results of our participation for the INS task in the TRECVID evaluation campaign between years 2013 and 2015. The measurements in the table are the mean Average Precision score averaging from all the query topics. The results clearly suggest that the approach *vi* which combines our three proposed improvements outperforms all previous approaches. Compared to the standard Bag of Features approach *i*, the approach *vi* has significantly improved the retrieval performance, with about 94.5% improvement from 0.111 to 0.216. These improvements came from our three proposed improvements based on the Bag of Features framework and we discuss their contribution in detail in the following section.

Effect of Embedded Coding

Here we evaluate the effectiveness of the embedded coding algorithm and also discuss the performance improvement for the enhanced retrieval system. As discussed in section 5.1, this work aims to improve the quality of feature matching and to reduce the error feature correspondences introduced by “quantization error” between two images. We compared the classic Bag of Features approach with our proposed Embedded Coding approach in

5.3 TRECVID Experiments and Discussion of Results

performing object matching tasks between a pair of images. We hoped that our approach would out-perform the Bag of Features algorithm in terms of effectiveness which could be used to improve the retrieval performance by robustly filtering out the outliers.

Figure 5.5 demonstrates two object matching examples using the Bag of Features algorithm and with our proposed algorithm. In this figure, the query object is shown on the left side of each matching pair and outlined with a yellow box and the target image is shown on the right side of each image. The first row presents the feature correspondences by connecting the same visual words from two images in the Bag of Features algorithm, the second row presents the refined feature matching after applying the embedded coding algorithm. The comparison of the two object matching examples clearly demonstrates that our proposed algorithm can increase the number of consistent matches and successfully identified the object localization in the target images. To quantitatively measure the effectiveness of our embedding coding approach, we evaluate its performance in the TRECVID experiment. As shown in Table 5.2, with the support of the embedding coding algorithm, system *iii* improved the instance search performance by 68.4% s compared to the standard Bag of Features System *i* on average. So, we can conclude that our approach could help to reduce object matching confusion and boost the retrieval performance because it provides more reliable feature correspondences after applying the embedding coding algorithm on instance search tasks.

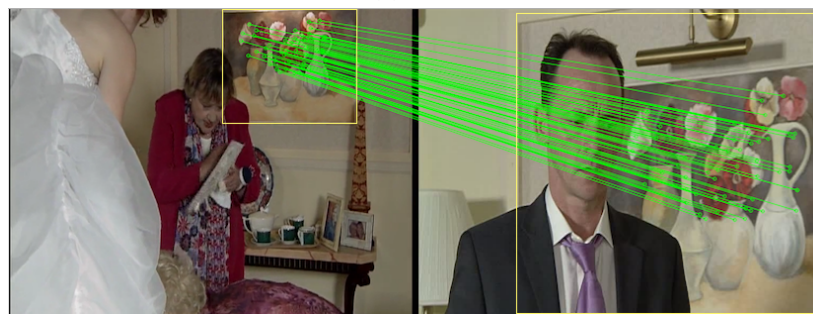
Improvement from Query Expansion

To better understand retrieval improvement from our proposed query expansion technique, we compare retrieval performance under the precision and recall curve for all our participating

5.3 TRECVID Experiments and Discussion of Results



(a) BoF



(b) BoF+Embedded Coding



(c) BoF



(d) BoF+Embedded Coding

Fig. 5.5 Comparing local feature matching performance between standard Bag of Features algorithm (a, c) and our embedded coding algorithm (b, d). The results suggest that feature matches are more consistent and the object is better localized in the target image with our embedded coding algorithm.

5.3 TRECVID Experiments and Discussion of Results

systems between year 2013, 2014, and 2015 in the TREC video evaluation workshop in figure 5.6, 5.7 and 5.8 respectively.

The red curves indicate systems with query expansion technology and in blue systems without query expansion technology. For every year's participation, the figure demonstrates that the query expansion technology has increased the recall value of instance search system when considering the same precision value. This proves that our query expansion technology indeed helps to find more relevant video clips for the instance search task. The results in Table 5.2 also demonstrate that systems (*ii, iv, vi*) with query expansion clearly outperform systems (*i, iii, v*) without query expansion in term of mean average precision score.

The idea behind query expansion is to expand the query terms to return additional relevant results, so the performance requires the top documents from the initial result to be reliable. Otherwise, unreliable documents could add noise terms to query topic representation and

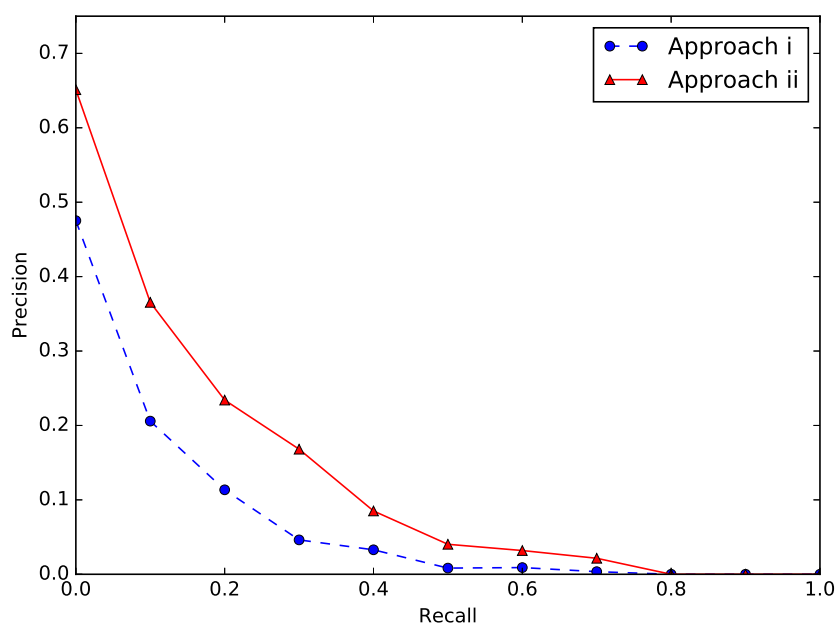


Fig. 5.6 Performance comparison between system (i) and (ii) in TRECvid 2013

5.3 TRECVID Experiments and Discussion of Results

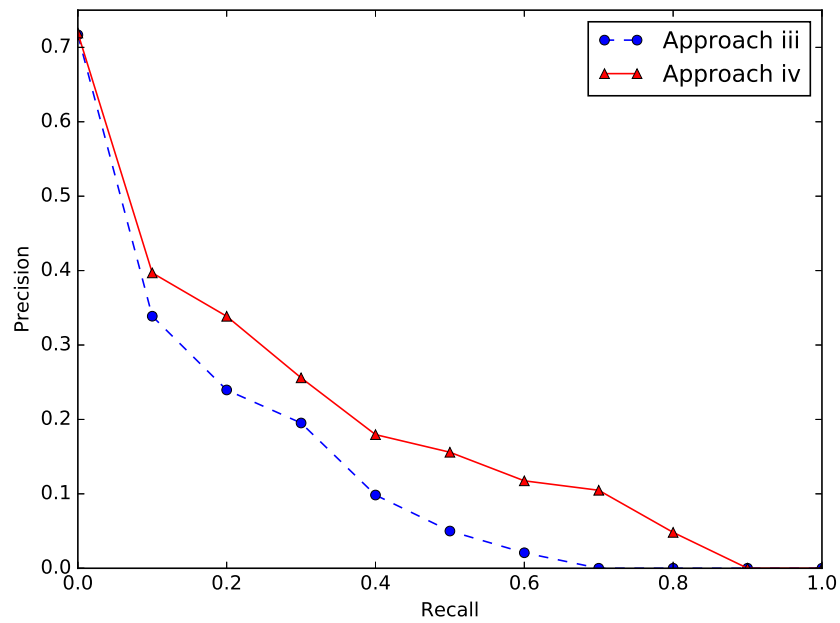


Fig. 5.7 Performance comparison between system (iii) and (iv) in TRECvid 2014

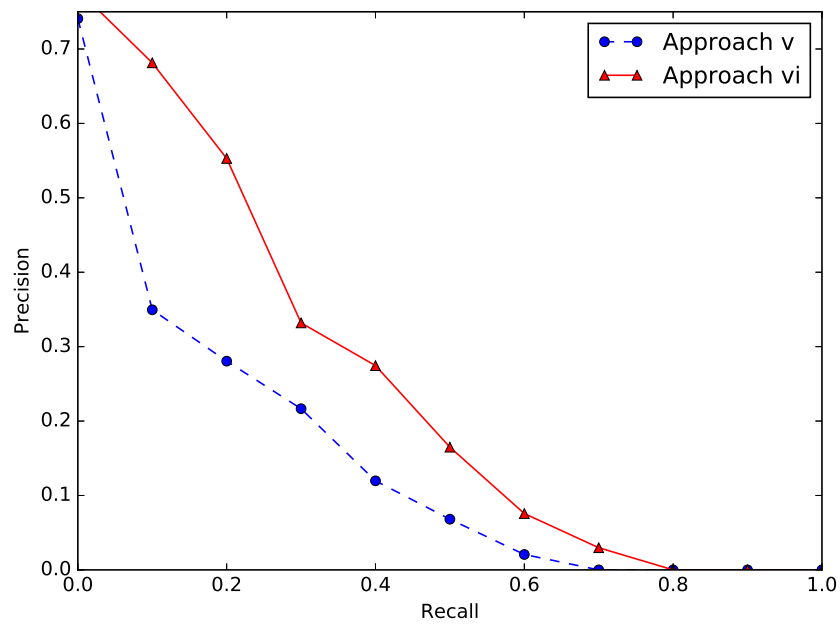


Fig. 5.8 Performance comparison between system (v) and (vi) in TRECvid 2015



Fig. 5.9 Query topics confusion introduced by standard query expansion from TRECVID experiment.

even harm retrieval performance. Figure 5.9 presents an example of query topics polluted by query expansion. Instead of enhancing search terms with relevant feedback, the query expansion algorithm could instead enhances unrelated object or scenes to the original query topic. This result shows that systems perform worse after applying the query expansion algorithm, as illustrated in the second row of figure 5.9. Comparing to the initial results in the first row, the top 10 results in the bottom row are good results to the background scene, but are not relevant to the query topic. The query expansion technology drifts query topic away from the original query object, which is a man wearing a white shirt, to the bar scene in the background.

In this work, we address this problem with our weak geometric consistency constraints algorithm from previous chapter since our algorithm could filter out non-relevant results from the initial ranked list.

Improvement from combination of WGCC, embedded coding and query expansion

Our last evaluation focused on demonstrating the effectiveness of the combination of our Weak Geometric Consistency Constraints (WGCC) algorithm, the embedded coding algorithm, and query expansion technology for instance search tasks in very large video collections.

As discussed previously, the performance of query expansion requires a good initial ranking list with many positive results, especially the top-N results. Our weak geometric consistency constraints algorithm and query expansion technology could be applied to improve the precision and recall of the initial ranking list. Consequently, we build an instance search tool to combine the three technologies and evaluate its performance by participating the TRECVID evaluation in 2015.

Figure 5.10 illustrates the performance improvement from the combination of three methods. Specifically, it lists retrieval performance for each individual query topic for our submissions to the INS task in the year 2015. The red columns are the average precision scores from the combination of three methods including our WGCC algorithm, and the blue columns are the score without the algorithm. The figure shows that the combination approach significantly outperforms in 5 search tasks and the best AP score for the combination approach achieved 0.79, in contrast to only 0.34 for the approach without the WGCC algorithm.

To summary our findings, there are two main contributions from this combination approach when helping performance improvement for instance search task:

- The embedded coding algorithm provides good and reliable feature matching correspondences to improve retrieval performance. Moreover, our WGCC algorithm also benefits from these reliable feature matching, since they provide good input for our

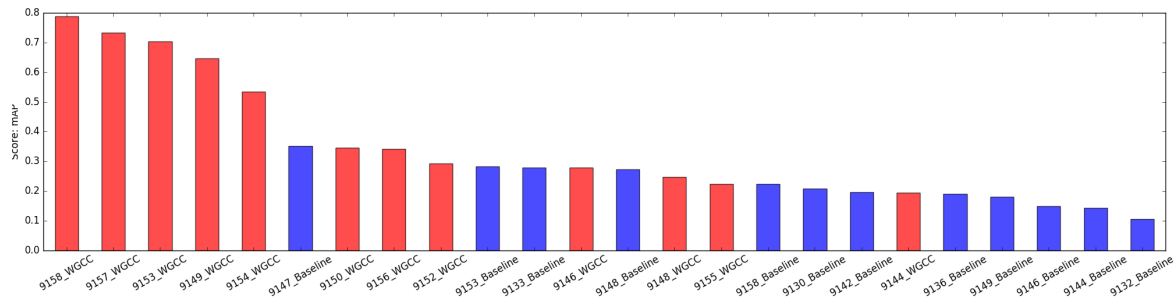


Fig. 5.10 Performance comparison between individual topics for our participation in TRECVID 2015. The red/blue columns are submission results with/without our WGCC algorithm

WGCC algorithm. Therefore combining them further improves the performance of instance search task.

- Then our WGCC algorithm could help to improve precision performance by re-ranking the initial results, so it provides a list of positive results for query expansion technology to enrich the search terms. This could prevent query expansion technology to introduce confusion to query topic and ruin the retrieval performance.

5.4 Conclusion

In this chapter, we address the challenges of building scalable and effective instance search systems for large video collections. We are motivated to carry out this research, since the apparent lack of tools to help non-expert users to access large video content. At the end of this research, we aim to come up with an answer to our second research question which was defined earlier.

We start our work by investigating the weakness of current approaches and find that local feature matching performance for instance search task could be improved with a more accurate nearest neighbor search algorithm based on an embedded coding algorithm. Secondly,

we present our work on building a complete instance search system and implement a query expansion technology to further improve the retrieval results for large video collections. To demonstrate the performance of our proposed approach, we evaluate the effectiveness of our proposed improvements while participating in the TREC video evaluation campaign [90] between years 2013 to 2015.

The experimental results strongly suggest that our proposed methods improve the retrieval performance for instance tasks in large video collections in both measurements of precision and recall. In contrast to the standard Bag of Features approach, our proposed methods improve the performance for about 94.5% in mean average precision, from 0.111 to 0.216, on instance search task in a test collection of 464 hours of video content.

After examining these approaches and finding them to be useful, we then investigate methods to answer our second research question. We demonstrate that we can address the requirement of effectiveness with scalable and accurate local feature matching methods while solving instance search problem on a large video collection. In order to build scalable instance search engines, our further work will focus on explore opportunities to employ the cloud computing technologies to build parallel and distributed application Basirat and Khan [11], especially with real-time data processing capabilities.

Chapter 6

Instance Search for Video Browsing

In the previous chapters, we have looked at ways to improve the efficiency and scalability of instance search algorithms. Now we turn our attention to examine the video browsing problem and address our third research question. Video browsing task [79] is the interactive search process to help users quickly explore large video content and find videos of interest by inspecting visual content. The increasing amount of video content leads this task into more interesting, yet challenging problem of finding particular video clips. So in this chapter, we focus on building instance search tools to address the challenges of video browsing task. Especially we would like to guide our research by answering the following research questions:

RQ 3 When browsing unstructured video collections, how much could we enhance user performance by applying our improved instance search approaches?

Answering this research question requires us to understand related research, and propose novel approach to improve users performance when performing video browsing task in large video collections. We begin our research by conducting a literature review to understand the strength and weakness of current approaches. After that, we propose to use instance

search methods to build content-based video retrieval and navigation tools, with the purpose of addressing the video browsing task. Finally, we develop a proof-of-concept application to evaluate the performance of our approach while asking users to explore the large video content and locate the targeted video segment.

The video browsing task aims to promote progress in content-based analysis and exploration for very large video collections. This task requires researchers to design video browsing tools which can help users to quickly seek and filter video segments from the large video collection within a specific time limit. In the literature, several content-based video retrieval tools have been proposed to address this research problem. Schoeffmann and Boeszoermyi [79] proposed an interactive navigation summaries approach to help users to quickly explore the whole video content. In addition, their work provides abstract visualizations of video content at a user-defined level of detail. Then Scott et al. [81] investigated the performance of building a concept-based visual search engine to provide search ability for interactive video browsing.

However, the results from Video Browser Showdown [78], a live video search competition at the multimedia model conference, shows that the tools for video browsing can be significantly improved. So in this chapter, we aim to propose a content-based approach to improve user performance for video browsing task. After studying the existing video browsing tools([34], [76], [61], [25] [7]), we found that the difficulty for designing such video browsing tools is that human perception is only able to process 20-50 unstructured visual content at a time. So this research investigates the possibility of visually browsing the entire video collection with structured abstraction and representation of visual content. More specially, we want to construct a similarity graph where related video scenes are connected with each other, so that abstract representation can be extracted with clustering technologies

In combination with visual features, we prove that a similarity graph could be built with our instance search technologies proposed in previous chapters. After that, similar areas in the graph can be reached by following the most promising path of edges. For efficient content exploration, we propose a faceted navigation method which projects visual instances onto a hierarchical structure to preserve their complex inter-image relationships. When performing the search process, users may either choose from a selection of typical videos scenes or use interactive interfaces. The retrieved video frames are arranged on a canvas and the view of the graph is directed to a location where matching frames can be found.

Moreover, in order to demonstrate the ability of our proposed exploratory method, we build a proof-of-concept video retrieval tools, and carefully design user experiments to evaluate its performance. The user experiment follows the settings of the video browser showdown event, which aims to promote the development of video browsing tools, by providing a well-defined video browsing task on a shared dataset. Researchers could then evaluate the effectiveness of their tool in direct comparison with other systems. We implement a user experiment with nine users and score their performance for a set of video browsing tasks using our proposed tool, as well as two comparison tools. This experiment result shows that our instance-based browsing tool improves user performance in general by 30%, and significantly outperforms the other two advanced tools, especially in three tasks.

Chapter 6 is organized as follows. We firstly introduce video browsing task in section 6.1 and discuss the related research in the literature. Then we review the video browser showdown as an event to evaluate the performance for video browsing tools. After that, we give the details of our proposed approach based on a similarity graph and faceted navigation. Finally, to evaluate the effectiveness of the proposed approach, we implement and discuss the result from user experiment in section 6.3, which follows the same setting as video browser

showdown. We conclude this chapter with answering the research question and discuss future research directions in section 6.4.

6.1 Video Browsing Task

In this section, we conduct a literature survey of related research on video browsing and present the recent approaches for designing interactive browsing tools. We focus on related work that addresses problems of video browsing with content-based analysis and interactive designs. Specifically, we review papers that use techniques to improve content consumption, such as through novel content presentation, visual content navigation, or interactive search capabilities.

6.1.1 Related Works

Video browsing approaches are hugely motivated by the fact that producing digital videos has become a simple task, thanks to the continuous improvement of capabilities to record and store digital content. However browsing and searching video collections to find the interesting videos remains a challenging and time-consuming task, especially as the volume of digital videos has kept increasing. The results of a user study from Schoeffmann and Boeszoermenyi [79] suggest that users are not satisfied with the video retrieval systems that provide ordinary meta-data based search functionality. Therefore, the need for tools that can manipulate video content in the same way as traditional databases manage numeric and textual data is significant. However, the complexity of video content introduces enormous challenges in the design of video browsing systems in comparing with the traditional text retrieval engine. One of the most difficult challenges is to build an effective interface to handle

the interaction between the user and retrieval engine to support their information-seeking process.

In 2010, Schoeffmann et al. [80] presented a comprehensive review of the state-of-the-art in video browsing and retrieval systems. They reviewed more than 40 different video browsing and retrieval interfaces, which aimed to help users with better interaction interface and various underlying video retrieval engine. They classified the different approaches into three main categories: applications with advanced video-player-like interaction([9], [68], [33]), applications with video retrieval and filter abilities([17], [22], [30]), and applications with exploring capability based on optimized video representations([102], [59], [98]). For each category, they summarize the strengths and weaknesses of each solution by comparing them against each other.

To summary the literature review, we find that the problem of video browsing task is not yet been solved, although the previous approaches made great success to address some of the challenges. We believe that tools for video browsing task can be significantly improved. Considering the diverse nature of video content and various retrieval technologies, we can conclude that successful video browsing systems should meet the following three requirements:

- Firstly, systems should have video content analysis and search functionality, so that they can provide content abstraction or representation to support users who wish to seek interesting video segments from large video collections.
- Secondly, systems should provide easy-to-use user interfaces and assist users to quickly construct search queries to enable users to interactively explore visual content with the help of underlying search engines.

- Finally, systems need to provide an intuitive visualization interface, which should allow users to interactively explore and understand the complex video content.

While there has been a lot of effort to improve video content analysis and retrieval, there is not much research in video content visualization for video browsing. So in this chapter, we study the system requirements for user-friendly content visualization and design systems that support the video browsing task.

6.1.2 Video Browser Showdown

Although many tools have been developed for interactive video search [80], there is a lack of evidence on how well these tools perform for video browsing tasks. Besides, it is also impractical to compare the performance of various tools without a common user study. For example, in content-based video retrieval tasks, we can perform a fair comparison by providing a common data set, common queries, and standard evaluation metrics. But in video browsing tasks, we also need to consider the highly interactive nature while asking users to explore large video content. So here we introduce the Video Browser Showdown event [78] for evaluating the performance of different video browsing tools.

The Video Browser Showdown is an annual live video browsing competition, held as a special session at the international conference on Multimedia Modeling. It was originally created by Schoeffmann et al. [78] with the aim of evaluating video browsing tools for their performance at video browsing task, with a well-defined video collection. Researchers could evaluate and demonstrate the efficiency of their tools by participating in this event and performing various tasks in a live competition environment. In order to directly compare their performance, it is necessary to create user studies with exactly the same setup, such as same video collections, same browsing tasks, same users, same time limit, and so on.



Fig. 6.1 The moderated public session of the Video Browser Showdown at the International Conference on MultiMedia Modeling holding at Miami, USA in 2016. Copyright@VBS

Different from the TRECVID [90] mentioned in the previous chapter, this event focuses on the interactive video browsing rather than video retrieval task. Therefore, instead of evaluating the automatic retrieval performance in a video collection using standard metrics such as recall and precision, the video browser showdown event focuses on user-centric video search evaluation. More specifically, it evaluates the abilities of each video browsing tool on helping users to interactively find the target segments by navigating, filtering, browsing, or using other interactive query methods.

Figure 6.1 shows a photo captured the event from the video browser showdown event at the 21st international multimedia modeling conference. This event is an interactive open session and all the participants are required to compete with each other in front of live audiences. Each video browsing task is to find a video clip, with a time frame of about 20 seconds long, which is shown on a projector lively for all the participants to see, from a large video collection. The performance of each participating tool is measured with a score calculated after a successful submission. A false submission will receive a penalize and results in reducing the performance score. For our evaluation, we follow the same setting as

video browser showdown event to set up the user experiment to evaluate the efficiency of our proposed approach.

6.2 Navigating Video Contents with Similarity Graphs

In this section, we present a novel approach to enhance user performance during video browsing. This proposed approach aims to help users to quickly navigate and browse large collections. In comparing to the existing approaches, we can identify two novel contributions for the video browsing task. Firstly, our approach builds a similarity graph based on object instances to provide content abstraction and summarization. This similarity graph provides users an overview of whole video content and helps them to quickly understand the video structure through content exploration and navigation. Secondly, we also develop a faceted navigation interface to allow users quickly exploring a very large video collections. This interface enables users to form on-demand filter queries based on their own understanding of the search topics. In the following work, we discuss our motivation and present the implementation of the proposed approach.

6.2.1 Motivation

As we discussed in the previous section, various tools have been developed to help users to address this challenge of video browsing. They either present advanced interactive interfaces to help users to quickly explore the video content, or they provide powerful content-based search engines to narrow the content down into a list of candidate video segments. In both cases, users are required to further inspect the content and make the final decision based on their understanding of the videos. According to the research from Barthel et al. [10], human

6.2 Navigating Video Contents with Similarity Graphs

perception is limited to 20 to 50 unsorted visual entities at a time. Their understanding of the video content will be quickly lost if more unstructured content is displayed. The current systems often provide too many videos for user inspection, whereas search-based systems with more keywords may offer too few or no results at all. The results of the video browser showdown event [80] have shown an overall performance for video browsing tools is limited. And there is still a lot of research space to build advanced tools to enhance user performance.

The main motivation of our approach is to provide a structured representation to allow users to visually explore the entire video collection. If videos are clustered by their visual similarities, users could inspect up to several hundred video clips simultaneously. Automatic organization of large, unordered image collections is an extremely challenging problem [94]. So in this work, we propose to use a similarity graph to automatically organize large, unordered video collections. Our approach centers around the concept of grouping the visual content that describes the similar object instances into the same cluster. Here, the instance could mean the same object, the same group of people, or the same landmarks in the same place. The ability to group similar instance together in such large collections has many potential benefits. For example, the objects with frequent occurrences can be quickly organized together into one cluster to abstract the visual content in large video collections. Then we could abstract a large chunk of content into one representative example. Besides, we could also use these object clusters to build a structured graph, so that users can explore the visual content, and drill down to intercept object of interests even for very large video collections.

In order to implement this proposal, the main idea of our approach is to efficiently generate a similarity graph using instance search over the entire video collection. Within this graph, each node represents a video scene and the graph edges represent the visual content

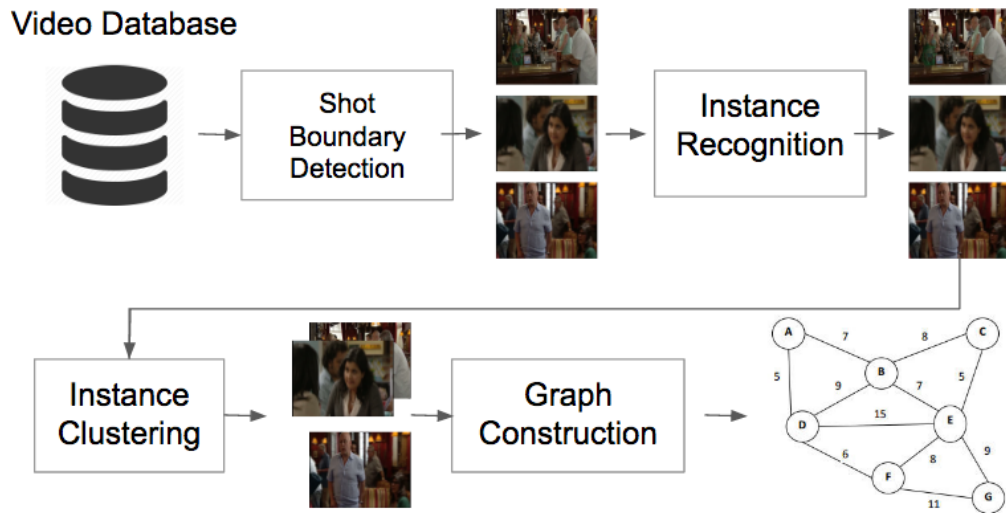


Fig. 6.2 Architecture overview of the video search system

similarity of object instances between pairs of video scenes. For example, if both video scenes contain a common object, the edge linked their nodes should be strong to reflect this connection.

Figure 6.2 shows the architecture overview of our proposed implementation, which consists of multiple stages. In the first stage, we apply video segmentation methods to divide a large video into small short-length video scenes. Then we employ content-based visual analytic methods to extract object instances from all the video scenes. Following that, a similarity graph is constructed using clustering technologies. In a final step, we design an interactive user interface with faceted navigation to help users quickly explore the video content and navigate them to the interested video scenes.

6.2.2 Video Segmentation and Representation

To reduce computing complexity and remove near-duplicate frames, a common method used in content-based video analysis is to segment a video into constituent scenes, each of which

6.2 Navigating Video Contents with Similarity Graphs

is a temporally sequential set of video frames, with an appropriate keyframe to represent the content of each scene. In this work, we segment the videos into a set of scenes based on the shot boundaries detection technologies outlined by Pickering and R uger [72]. Following this, the middle frame was extracted to provide a quick overview storyboard of the video content; this strategy has proved successful in the TRECVID instance search task [110]. The average duration of a scene in our system is around 10 seconds which means that a 100-hour video collection would abstract to about 36,000 shots. In addition, we also extract dense frames (one frame per second) to allow users to inspect the detailed context of each video shot whenever necessary.

6.2.3 Instance Recognition and Attribute Analysis

Image classification and object recognition using Deep Convolutional Networks (DCNs) have been shown to perform well on various evaluation tasks in recent years. The learning framework, Caffe [43], along with learned models is open source, which encourages researchers to use and contribute to the framework. In our work, we employ DCNs to extract meaningful information, such as semantic concepts, object labels and object attributes, to describe the visual content of video shots. More specifically, we choose two pre-trained models to cover the wide range of possible topics in the BBC programming videos, these two models are:

R-CNN ILSVRC-2013 [27] This model includes 200 object categories, such as person, dog, chair and so on. We choose this model to describe desired object information that could be easily captured by users while interpreting the search topics.

Places-CNNs [118] The second model introduces 205 scene categories to describe the overall background context for any given images, such as office, restaurant, valley, desert and

6.2 Navigating Video Contents with Similarity Graphs



Fig. 6.3 An example of visual content analysis using DCNs models. The Place-CNNs model labeled the image with overall background information, and the R-CNN model focused on label image with object categories.

so on. We choose this model to allow users to navigate the video content with straightforward environment information.

We use the Caffe Library¹ running on a machine equipped with a GeForce GTX 970 Graphic card and 16 GB RAM to do the visual processing tasks. Thanks to the efficient algorithm implementation and GPU computational power, it takes about two seconds to extract the visual information and label a frame into a set of textual terms.

Figure 6.3 illustrates an example result after applying the semantic visual analysis process for one random frame from the video collection. When using the R-CNN model, the bottom-up region proposals approach [27] was employed to provide the localization of labeled objects from frames with complex background. For background place recognition with Places-CNN model, the frame image was used as one input to label the most likely place categories from various informative regions.

¹<http://caffe.berkeleyvision.org/>

6.2.4 Similarity Graphs Construction

Once we extract the object instances from video scenes, we can now build the similarity graph for the large video content. In order to build this graph, we can first produce a similarity matrix S over all pairs of object instances. Then this matrix S can be used with similarity-based clustering algorithm to identify object clusters, each of which contains a set of object instance which presumably should be grouped together. Then during content exploring, we can interpret each cluster as one entry which refers to the same underlying object.

Computing the Similarity Matrix

The objective of this step is to compute an $m \times m$ similarity matrix between all pairs of objects. However, we note that for a large number of records m , computing the full similarity matrix, an $O(m^2)$ operation, may be impractical since the vast majority of object pairs are not similar at all. Therefore, it is necessary to select a subset of candidate pairs between which similarity is computed.

So we employ our proposed instance search technologies to build this matrix. For each object candidate extracted from video scenes, we query over the whole visual contents and only keep the top-100 results which are spatially verified as described in the chapter 4. We recognize the verified local feature correspondence between two object instances as their visual similarity measurement, which will be used a new edge to the graph linking stage. Especially, we also apply a normalization to the similarity measurement to reduce the effect of varying local feature lengths extracted from different objects. We repeat this process for each object across the whole object extracted from the previous section.

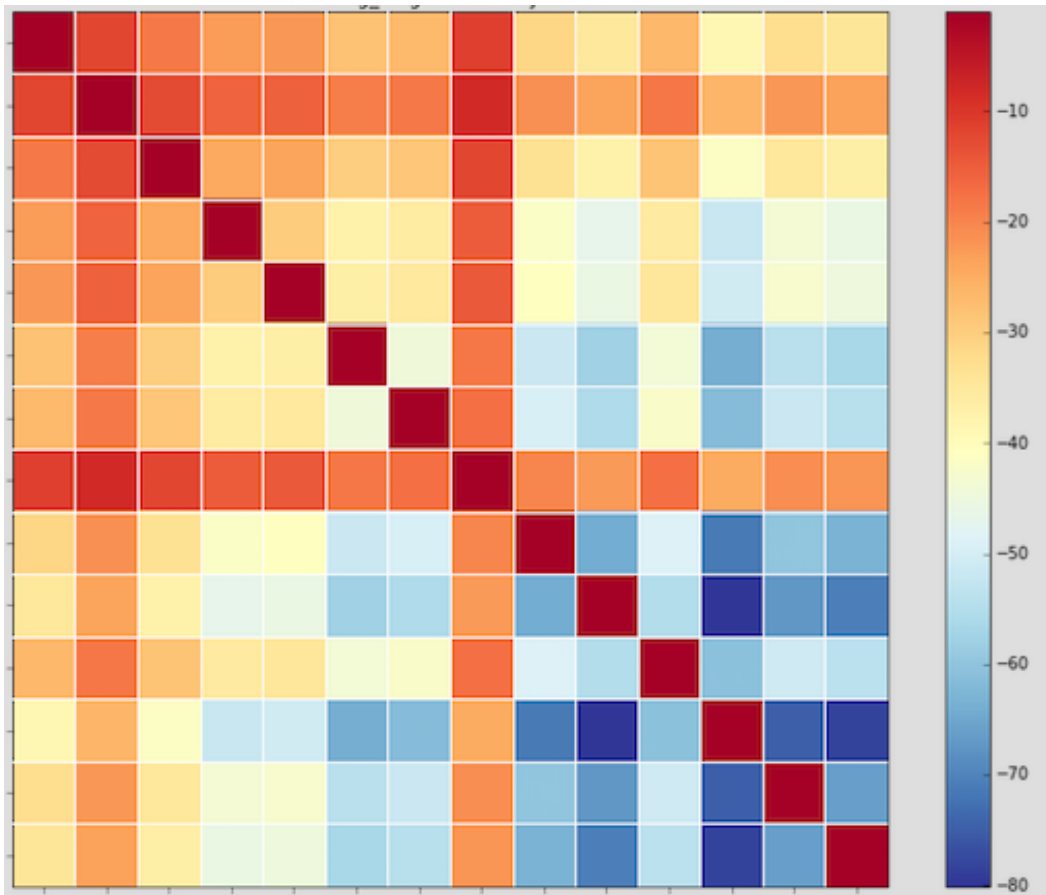


Fig. 6.4 A 14x14 similarity matrix constructed from object instances

Figure 6.4 demonstrates a similarity matrix built from a set of 14 objects. It is worth notice that this similarity matrix is a symmetry matrix, and the diagonal have the same value due to normalization effect.

Instance Clustering

Armed with a similarity matrix, we face the subsequent problem of identifying the cluster of object instances. In previous research work, two primary approaches have been used for the actual task of building similarity graphs: (1) pairwise linkage [86] followed by transitive closure, and (2) collective approaches [86] where linkage decisions between multiple pairs are made in conjunction. In this work, we choose to implement instance cluster algorithm

6.2 Navigating Video Contents with Similarity Graphs

following the Hierarchical Agglomerative Clustering (HAC) algorithm [36], which can be understood as a bottom-up greedy collective approach.

The similarity graph is empty at the first place and hierarchical agglomerative clustering initially places every node in its own singleton cluster. Then we iteratively merge the most similar clusters using the similarity matrix computed as described above. At every iteration, the similarity between clusters is defined as the lowest similarity of any individual cluster members. Thus, this clustering implementation is a greedy but efficient cluster method since the merge decision for every node pair depends on similarities of other pairs from the clusters to which the node under consideration belongs. More importantly, this hierarchical cluster structure abstracts the video content at various levels, which gives us the opportunity to navigate users through multiple layers. The higher layer provides more abstract overview and help users to understand the whole video content. On the other hand, the lower layers provide a more detailed view of same objects with the different surrounding.

In the next subsection, we introduce the methods to build an interactive interface to make use of this similarity graph in order to help users whiling performing video browsing tasks.

6.2.5 Faceted Navigation for Browsing

The review work on video retrieval from Snoek and Worring [96] indicates that text-based search or filtering interfaces are unsatisfactory for user experience while browsing video collections. One of the main challenges is the semantic gap [92] introduced by representing low-level rich video content with limited high-level semantic concepts. The lack of sufficiently rich description of high-level concepts extracted from visual content may not fulfill the interpretation from users while interpreting the same video data. Therefore, we implement an alternative solution, faceted navigation, to support exploratory search and

6.2 Navigating Video Contents with Similarity Graphs

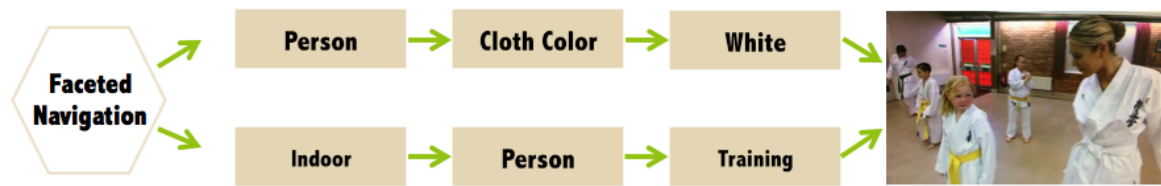


Fig. 6.5 This example demonstrates two possible routes to find the target video clip with text query: A group of mostly kids practicing Karate moves indoors (in white clothes), including close-ups of a blond young woman talking to a girl, and shots showing the instructor, a bald man with glasses.

navigation This approach aims to help users to find the required video clips quickly and effectively. Faceted navigation reflects the fact that users may seek information in a number of different ways, based on their own understanding of the query topics. In our standard faceted navigation system, the facets are built from different object labels or scene categories which are generated during the video content analysis process. These facets contain attributes by which the list of the same category can be further filtered. Compared to the previously proposed tools for this task, the main advantages of our faceted navigation system could be summarized into the following three points:

- It does not require users to manually input search query to match the high-level semantic concepts or low-level feature descriptors. This frees users from manually interpret the complex visual or text topics.
- It could take the full advantage of the advanced visual analysis approaches. Since the facets are organized and present as filters for users to narrow down the range of search results, it could benefit from more precise content descriptions.
- Finally, it provides multiple navigation routes to help users to access the same video.

Our faceted navigation relies on an underlying infrastructure that enables associations among elements of multiple types and allow users to drill down through categories and

6.3 Evaluate Instance Search for Video Browsing Task

attributes naturally. The taxonomy structure of our system is constructed from the visual content of keyframes and specifically addressed on two aspects: similarity clusters and semantic labels. To explore the content of video collection, users could choose different facets and allow the system to narrow down the candidate collection for inspection. Figure 6.5 demonstrates an example of using our faceted navigation to find the interesting video clips from two different routes. After the general understanding of example text query topic in the first step, users can very easily to locate the target video clips by following the provided various facets.

6.3 Evaluate Instance Search for Video Browsing Task

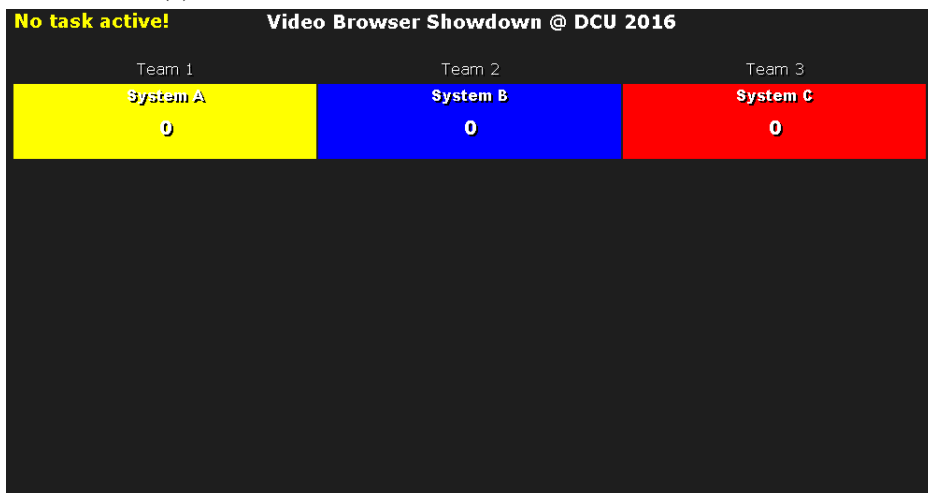
In this section, we evaluate the proposed instance-based video search and navigation approach for the video browsing task in the context of large video collections. The video browsing task is an interactive process within which users inspect the video content in order to find particular scenes from a video collection. To accomplish the task, users may need to use video browsing tools to quickly explore the video content or perform automatic content retrieval or filtering to find relative content. The aim of this evaluation is to demonstrate the efficiency of our instance based video search tools with a well-defined data set in direct comparison with other appropriate methods. We expect that our proposed approach could outperform compared systems in completing the video browsing tasks.

During this experiment, we compare the proposed instance-based search and navigation approach with a video browsing application with exploring interface as a baseline system. Additionally, we also compare our system to a more advanced system with concept filtering capabilities. We build web-based applications for the three approaches and then recruited nine users to complete a number of interactive video browsing tasks using each of the three

6.3 Evaluate Instance Search for Video Browsing Task



(a) Scoreboard at one video browser showdown event



(b) Close look at scoreboard

Fig. 6.6 Scoreboard for performance comparison during during the evaluation experiment. This interface shows performance score for all submissions.

6.3 Evaluate Instance Search for Video Browsing Task

applications. As in video browser showdown event [80], users need to interactively find a short video clip (20 secs) from large video collection, about 200 hours of video content, within a 5 minutes time constraint. Figure 6.6 illustrates the scoreboard for performance comparison during the evaluation experiment. To demonstrate the efficiency of each video browsing tool, we compared their performance score obtained when users performing video browsing tasks.

6.3.1 Test collection and Participants

In our evaluation experiment, we use the same video dataset as the Video Browser Showdown competitions, which includes 244 video files (totally 80.1 GB, approximately 200.0 h) with associated metadata. Each video file with an average duration of 1 hour in PAL resolution, The video content contains a week's worth of BBC EastEnders programs [65], which is available one month before the competition for all participating teams to download. We pick this test collection since it provides us a number of unique features, and is appropriate for our evaluation purpose. The visual content of each video is quite diverse since they all are describing a wide range of real life scenes. And this large number of videos make it impossible if users try to memorize the structure of the video collection and use this knowledge to corrupt the evaluation results.

In this evaluation experiment, we focus on the evaluation of the browsing tools rather than the impact of the users. So we recruit nine novice users who have no experience of development or research on the topic of content based video search as unbiased users for the interactive experiment. All users are either postgraduate students in computing science or researchers in computing faculty in Dublin City University. As such, they could be

considered as technically competent users who will have no difficulty in understanding the evaluation tasks and the various tools, yet not experts in video browsing tasks.

6.3.2 Experimental Methodology

In this work, we aim to examine the improvement of user performance by applying our proposed approaches while browsing unstructured video collections. Therefore instead of doing qualitative user evaluation, we demonstrate the improvement in the effectiveness of our proposed approaches in helping users to complete the video browsing tasks. We present users with the scenarios in which they are required to complete a number of video browsing tasks with different video browsing tools, and compare the performance of different systems to measure user performance improvement. Each task requires the user to find the specific scene from a large video collection within a specified time limit. At the video browser showdown event [80], the performance score is related to the submission time. We ask users to finish nine video browsing tasks to provide unbiased measures for each tool. An HTTP server is set up for the purpose to host the scoreboard interface and compute performance scores for each tool. Finally, we measure the system performance and compare their results to evaluate the effectiveness on helping users to complete the video browsing tasks.

Browsing tasks

For the performance evaluation experiment, we choose the nine browsing topics from the video browser showdown [78] in 2015. These topics are short video clips, randomly selected from the whole corpus to form the queries for the search tasks. These topics are chosen to cover a wide range of scenes to describe the practical information needs. Each of topic contains about 20 seconds of video content, which are displayed in figure 6.7.

6.3 Evaluate Instance Search for Video Browsing Task





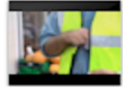




1		Indoor scene happened in a kitchen, two women talking	6		Indoor scene in studio with a group of people debating
2		A television studio scene A female commentator wearing a red shirt	7		Outdoor scene where a few people trying paragliding
3		Outdoor scene in a construction site, a male wearing safety vest	8		A stage that a band playing live music with crowded audience
4		Outdoor scene, a man standing in street and speaking	9		Indoor scene, an orchestra layout
5		A male commentator in studio with and big screen in background			

Fig. 6.7 The nine browsing tasks for video browser evaluation experiment, which were selected from the 2015 video browser showdown.

Evaluation Measures

We employ the same evaluation measurement from the Video Browser Showdown event [80] in multimedia modeling conference. For each browsing task, a user could get a maximum performance score of 100 points. The number of points is dependent on the number of wrong submissions for the current task and also the task solve time. The score linearly decreases from 100 to 50 over the given task solve time, which is 5 minutes in our experiment. Therefore, if you submit the correct segment in the last second, you will still get 50 points for it. However, wrong submissions could cause a penalty to be applied, except the first two wrong submissions for each task.

A scoring server with graph interface is setup to check the correctness of every submission, and compute a score for the corresponding tool and task based on the submission time and the number of false submissions. To be precise, we use the following formulas to compute

6.3 Evaluate Instance Search for Video Browsing Task

system performance score s_k^i for tool k and topic i .

$$s_k^i = \frac{100 - 50 \frac{t}{T_{max}}}{p_k^i} \quad \text{where} \quad p_k^i = \begin{cases} 1 & \text{if } m_i^k \leq 1 \\ m_i^k - 1 & \text{otherwise} \end{cases} \quad (6.1)$$

where m_k^i is the number of submissions by tool k for task i and p_k^i is the penalty due to wrong submissions. Then we measure the overall performance for tool k using score S_k , which is simply the sum of the scores across all the tasks. The final score for each system can be formed as follow:

$$S^K = \sum_{i=1}^9 s_i^k \quad (6.2)$$

These two are originally designed in Schoeffmann et al. [78] such that participants submitting several wrong results get significantly fewer points than participants submitting just one correct result. This penalty is intended to limit trial-and-error approaches. On the other hand, the linear decrement of performance score over time should motivate the users to find the target sequence as fast as possible. The hardware for the competition is normalized since we use the same machine for all the three systems.

Experiment Procedures

We evaluate the performance improvement of our proposed approach by comparing the search results to a baseline approach and one advanced approach from content-based video search tasks. We list the key features for all three approaches as follows:

Baseline tool This system summarizes the content and presents users with a browsing interface to assist them exploring video content. We choose to implement this tool to demonstrate user performance with only simple exploration and browsing capabilities while

6.3 Evaluate Instance Search for Video Browsing Task

solving video browsing tasks. Comparing to our proposed approach, this system lacks the abilities of advanced content analysis methods. Our motivation is to use this tool as the baseline tool so that we can compare its performance with other systems to evaluate the performance improvement from our proposed methods.

Concept filtering tool Concept filtering tool is an extension of the baseline tool with the ability of concept-based filter using semantic concepts. These semantic concepts are automatically extracted from video content to represent their visual information. Users are able to filter video content using a single concept or the combination of concepts. This tool could help users to speed up the validation process by providing a list of related video candidates. But this tool suffers the semantic gap problem [29], which is to successfully describe visual content from low-level feature analysis to semantic content representation of videos. Our proposed approach addresses the semantic gap problem with faceted navigation and we choose this tool for evaluation the performance improvement by comparing it with this tool. Figure 6.8 shows the interface design for concept filtering tool. The user could use this interactive user interface to filter video by select one or more semantic concepts.

Faceted navigation tool The third system implements our proposed approach with similarity graph and faceted navigation interface to enhance users performance while performing video browsing tasks. Different from the other two systems, we provide an extensive abstraction of large video collection and help users to quickly explore the content in an interactive manner. Figure 6.9 illustrates the interface design of our interactive interface for faceted navigation system. The user interface allows the users to trigger the search process by selecting the interesting concepts or object clusters. Then it navigates users to explore visual content by continuously ask them to contribute input to further narrow down the search range by suggesting useful facets.

6.3 Evaluate Instance Search for Video Browsing Task



Fig. 6.8 Interface for concept filtering tool

In order to evaluate the effectiveness of each video browsing tool, we ask users to complete a number of video browsing tasks. We firstly introduce the three evaluation tools to all the participants. To minimize the impact to the results introduced by the users or topics, every user needs to complete all nine browsing tasks with all three tools. The results were averaged for each tool over all nine users and nine topics. In this experiment, we randomize the query topics to remove the bias introduced by tools for special topics.

We build a web-based application to provide an easy access to each tool so that the experiment would be carried out in their day-to-day working environment for each user. This application is used to collect the results from the users and compute their performance score after each submission.

6.3.3 Results and Findings

In this subsection, we present the result of the evaluation experiments and discuss the findings from the quantitative results measuring the effectiveness of each tool. We expect

6.3 Evaluate Instance Search for Video Browsing Task

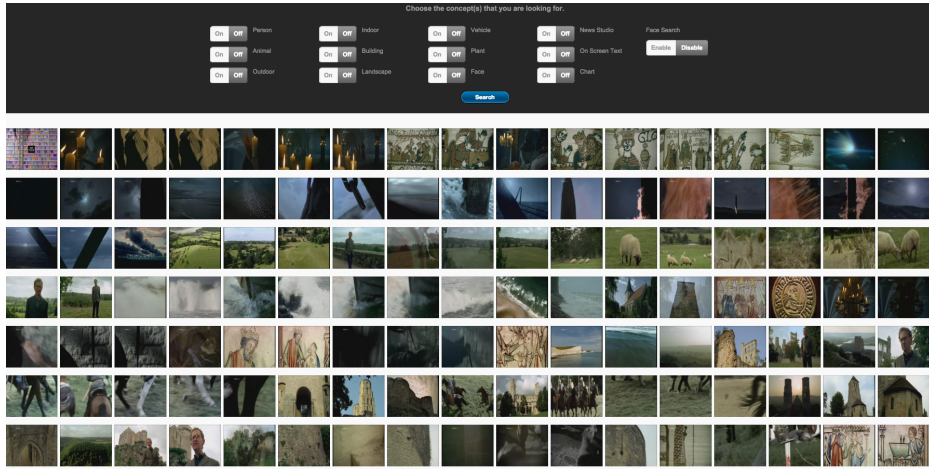


Fig. 6.9 Interactive interface for faceted navigation tool

Table 6.1 Performance comparison between three video browsing tools

Tools	Total Score [0-900]	Submission Time [0-300 seconds]
baseline	264	161
concept filtering	444	83
faceted navigation	689	46

the evaluation results could help us in answering the third research question and give some insights on our future designs for video browsing tools.

As we discussed in the previous section, we design our experiment in a constrained environment, with same topics on a shared video dataset, so that we could evaluate the efficiency of different video browsing tools by directly comparing their performance scores. So in the rest of this work, we demonstrate the efficiency of our proposed approach by comparing the performance scoring with two other tools.

Overall Performance

To begin with, we present the overall performance for the three tools from our experiment. Table 6.1 shows some details for the average performance across all the visual browsing tasks performed in the experiment. The value range for the total score for 9 video browsing tasks

6.3 Evaluate Instance Search for Video Browsing Task

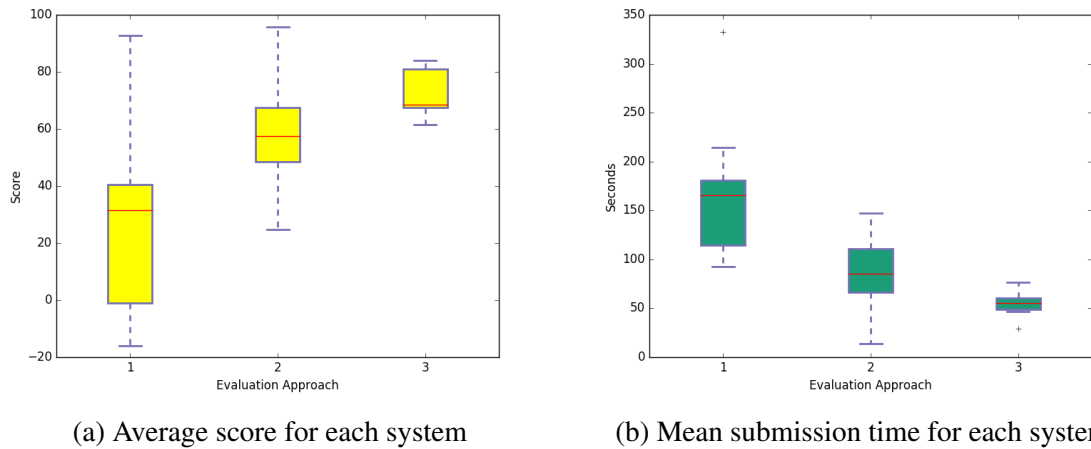


Fig. 6.10 Average performance scores and mean submission time for all the three systems

is between 0 and 900. When looking at the overall score of all systems in the experiment, we can see that users using the faceted navigation tool clearly gained higher performance score than the other two tools. Moreover, users with faceted navigation tool spent much less time to complete the video browsing tasks as well.

In addition, figure 6.10 illustrates the box plot the average score and mean submission time across all the browsing tasks for the three systems. From this figure, we can conclude that user performance in faceted navigation tool is much more stable than baseline tool and concept filtering tool, in both performance score and submission time. This result reflects that the third tool with similarity graph and faceted navigation interface are more efficient than baseline tool and concept filtering tool.

Comparing performance score per each task

Next, we present the average performance scores for all the systems per task while user completing each browsing task in table 6.2. The value range for the average score is between 0 and 100. And the results strongly suggest that faceted navigation tool continuously offered superior performance. The score for baseline tool is relatively low, which demonstrate that

6.3 Evaluate Instance Search for Video Browsing Task

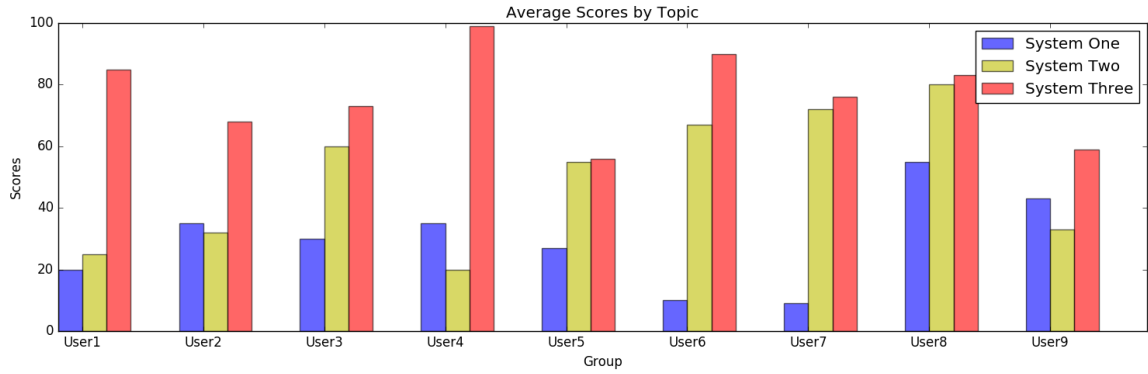


Fig. 6.11 Comparison of user performance score among all tasks

approaches purely based on content exploring suffers for very large video collections. Other than that, performance scores for concept filtering tool have a big variation with the lowest score at 20 and the highest score at 80. The reason is the performance for tools based on filtering really depended on the right concept that user extracted from the browsing topic.

Comparing performance score per each user

Finally, we show the performance scores across all browsing tasks for each user using three systems in figure 6.11. The value range for user performance score is also between 0 and 100 per each topic. Users gain highest performance score for all the tasks with the help of faceted navigation tool.

Table 6.2 Average performance score [0-100] by each browsing task

Tools	Average score for each task								
	I	II	III	IV	V	VI	VII	VIII	IX
baseline	20	35	30	35	27	10	9	55	43
concept filtering	25	32	60	20	55	67	72	80	33
faceted navigation	85	68	73	99	56	90	76	83	59

6.4 Conclusion

In this chapter, we present a graph-based representation for large video collection based on the clusters of object instance within video scenes, and propose a faceted navigation interface to help users to improve their efficiency while solving video browsing tasks.

We begin our research by investigating the importance of addressing video browsing task and conduct a literature review to study requirement of video browsing tools from related work. We find that video browsing tools should offer users an overview of large video content and provide a structured content representation to help me exploring video collections. So we propose our third research question and examine the ability scalable instance search technologies for enhancing user performance for video browsing task. This research question guides us through this research. In comparison to recent development in content-based video search, our approach not only provides a structured description of large collection of videos but also allow users to quickly navigate content.

To demonstrate the abilities of the proposed approach, we implement a web-based application and design comprehensive user experiments over very large video collections. Our experiment follows the format of the video browser showdown event from the international multimedia modeling conference, and aims to evaluate the efficiency of our tools by directly comparing its performance with other browsing tools on a shared dataset. The experiment results show that our proposed system clearly outperforms the approaches compared against and proves to be more efficient in helping users while solving video browsing tasks. To answer the research question, our scalable instance search technologies helps us to group the videos into various clusters, so that the similarity graph could be built to present large video content.

Future work could investigate more advanced interactive interface to assist users in video browsing and exploration task. These user interfaces should not only provide visualization for visual content, but also adapt to the user. For example, in a video retrieval scenario, we could employ relevance feedback techniques to suggest more accurate content for user inspection. Moreover, considering the diversity of potential video browsing applications, another challenge is how to build effective tools to provide user service for the different use cases, such as for small mobile screens.

Chapter 7

Conclusions

This thesis presents methods to improve instance search and enhance user performance while browsing unstructured video collections. Through the use of computer vision and information retrieval techniques, we propose novel solutions to analyse visual content and build a search algorithm to address the challenges of visual instance search, while considering the constraints of practical applications.

Firstly, we investigate methods to improve the effectiveness of instance search systems for finding object instances occurring in unstructured video content. Using the bag of features framework, we propose a novel algorithm to use the geometric correlation information between local features to improve the accuracy of local feature matching, and thus improve the performance of instance search systems without introducing significant computation cost.

Secondly, we consider the scenario that the performance of instance search systems may drop due to the volume of visual content in large video collections. We introduce a search algorithm based on embedded coding to increase the effectiveness and efficiency of instance search systems. We participate in the international video evaluation campaign, TREC Video Retrieval Evaluation, to comparatively evaluate the performance of our proposed methods.

Finally, the exploration and navigation of visual content when browsing large unstructured video collections is considered. We propose methods to address such challenges and build an interactive video browsing tool to improve user performance while seeking interesting content over video collections. We construct a structured content representation with a similarity graph using our proposed instance search technologies. Considering the constraints related to real world usability, we present a flexible interface based on faceted navigation to enhance user performance when completing video browsing tasks.

In summary, we investigate the state-of-the-art solutions and motivate our research work by studying the weakness of current approaches for instance search and video browsing task, which leads us to prove the following hypothesis for our thesis:

Hypothesis While seeking information in unstructured video collection, then user performance can be enhanced while improving the effectiveness of instance search approaches.

We ask three research questions and propose various novel methods to improve the performance of instance search over video collections. The answer to the research questions shows that user performance can be successfully enhanced by improving the effectiveness of instance search approaches when seeking information in unstructured video collection.

This section is organized as follow. In Section 7.1, we start with answering our three main research questions which are outlined previously in Chapter 1. Then in Section 7.2, we conclude this chapter after listing possible future research directions.

7.1 Answers to Research Questions

In this chapter, we revisit the three research questions which guide us through this research work.

RQ 1. How much can we improve the effectiveness of the Bag of Features framework for instance search, if we make use of the geometric correlation information among local features?

In investigating ways to build content-based search approaches to address the problem of instance search tasks, we notice that most state-of-the-art approaches following the Bag of Features framework [88]. After reviewing these approaches, we found out that false feature matching between the query object and object instances could adversely affect the performance of approaches derived from this framework. In order to improve its performance, we decide to use the geometric correlation information among local features which was originally ignored in the Bag of Features framework.

Therefore we investigate the methods of applying geometric correlation information to improve the effectiveness of instance search within the bag of feature framework. To improve the performance of instance search, we develop an object matching algorithm to improve the reliability of feature matching by filtering false matching with geometric validation. More specifically, we implement weak geometric correlation constraints which perform feature validation in a very efficient way. This research allows us to build successful instance search application for more challenging tasks in video collections.

RQ 2. How much can we improve the performance of instance search systems while applying our feature matching method on video search task?

When addressing the problem of instance search over large video collections, we face the challenges of maintaining good retrieval performance, including search accuracy and response speed. Answering this question requires us to investigate methods to offer effective and scalable solutions over large video content.

We proposed an approach to enhance the feature representation with embedded coding scheme and implemented a nearest neighbor search algorithm to address the problem of improving feature matching for large datasets. This algorithm helps us to effectively integrate our object matching technology into the retrieval process and enabled the ability to refine ranking results even for very large datasets. By the end of this research, we successfully build a complete retrieval system to provide instance search capability for large video collections.

RQ 3. When browsing unstructured video collections, how much could we enhance user performance by applying our improved instance search approaches?

In the third question, we address the challenges of enhancing user performance in the context of browsing large video collections. After reviewing the current video browsing tools, we learn that user could easily get lost with the lack of the ability to provide an overall understanding of large video content. So we focus our research on providing a content abstraction and build video browsing tools to support video browsing over large collections.

To answer this research question, we applied the scalable instance search technologies to build a similarity graph from the object instances occurring in the video content. This similarity graph not only provides an overview of the video collection but also offers a multilayer representation of large video content. Further, we also designed an interactive interface with faceted navigation to help users quickly explore very large video content. This tool allows users to quickly locate the short video clip in large video collection within a

specific time limit. This research question helps us to build an interactive video browsing tools to enhance user performance while browsing large video collections.

Having answered our three research questions, we find that improving the performance of content based instance search approaches could help users to browse large video collections and quickly find interesting information. Our research on instance search indeed contributes to addressing the challenges of organizing, exploring and searching large video collections. Therefore this lead us to draw the conclusion that our hypothesis has been confirmed, improving the effectiveness of instance search approaches could help us to design powerful applications to hugely enhance user performance while browsing large video content.

7.2 Future Directions

There is always room for improvement and many potential avenues to explore to further improve the performance of instance search systems. While we have covered many different aspects of improvement for instance search systems in this thesis, we believe that the research we have done is but the tip of the proverbial iceberg. In this section, we identify three research directions that show significant potential for scientific discovery in the future.

7.2.1 Advanced Feature Representation

The most recent research work on Convolutional Neural Networks (CNN) has shown that feature representations from deep learning outperform handcrafted features in different image retrieval benchmarks [73]. Activations from the fully connected layers of a pre-trained CNN network designed for image classification can be used as a global image representation for retrieval [101], [6]. Their experiment results prove that their proposed system achieves

competitive performance on the several evaluation benchmarks [69], [70]. However, those representations tend to be too very high-level and sometimes fail to capture sufficient local image detail for instance search.

Recently, Razavian et al. [73] showed that convolutional layers outperform fully connected ones in retrieval tasks and Ng et al. [62] provide a guidance of how to extract local image representations from convolutional layers. These works inspire researchers to investigate the possibilities to design a bag of visual words system based on local CNN descriptors. Mohedano et al. [58] propose an instance retrieval pipeline based on encoding the convolutional features from CNN network using the bag of feature framework. This bag of feature based sparse visual representation provides a fast composition of object descriptor for any region within the image. The main difference with existing instance approaches is that the features from well-trained lower convolutional layers to keep the good local details of objects, which is critical for instance search tasks.

Seeking more advanced feature representation will help us to overcome the information loss when encoding local information from object instances, thus aid us to build practical interactive retrieval systems.

7.2.2 Learning to rank

The rank function is at the core of information search engine which provides the ability to rank the candidate documents to their relevance to the query. Learning to rank, a relatively new field, aims to tune more advanced rank functions for more accurate search experience by employing machine learning algorithms. Recent work from Malisiewicz et al. [52] demonstrated that a single positive object classifier can perform well on detecting visually similar instances with a linear SVM [16] algorithm even with a single positive example.

A typical setting in learning to rank is that feature vectors describing query-document pairs are constructed and relevance judgments of the documents to the query are employed to learn a rank function. After the training process, a hyperplane, $\vec{w}^T \vec{x} + b = 0$, could be applied to sort candidate documents in data collections according to their distance from this decision hyperplane. Comparing against statistical models [53], data-driven similarity functions consider the most important visual pattern from queries and improve the search performance with various settings such as supervised, semi-supervised or reinforcement learning.

However, these algorithms require training the rank function on-the-fly for each query, which could be very inefficient and computationally expensive. In addition, ranking a list of large objects with machine learning algorithm could be time-consuming and offer non-optimal user experience. Further researches to solve these problems will help us to offer most advanced service for instance search tasks.

7.2.3 Improving scalability using cloud computing

The new surge of interest in cloud computing is accompanied with the exponential growth of data sizes generated by the democratization of digital video production [12]. Cloud computing provides services over the Internet in a scalable manner, which are essentially designed to support data-intensive application, for instance search applications over large video collections.

We look forward to exploring the opportunities to build parallel and distributed application in the cloud to offer scalable instance search engines, especially with real-time data processing capabilities. However dynamic and distributed nature of cloud computing environment presents us with many challenges when building successful cloud-native instance search

applications, such as data management, data security, data workload balancing, and so on. Further research on addressing these challenges could benefit us on effectively processing the immense large video data collections.

This chapter concludes our research work for this thesis by revisiting the research question, addressing the hypothesis, and listing the future research directions. We hope that the lessons learned and methods developed in this work can benefit the research community and contribute towards advancing the field of instance search.

References

- [1] Albatal, R., Mulhem, P., and Chiamarella, Y. (2010). Visual phrases for automatic images annotation. In *Content-Based Multimedia Indexing (CBMI), 2010 International Workshop on*, pages 1–6.
- [2] Arandjelovic, R. and Zisserman, A. (2012). Multiple queries for large scale specific object retrieval. In *BMVC*, pages 1–11.
- [3] Avrithis, Y. and Tolias, G. (2014). Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval. *International journal of computer vision*, 107(1):1–19.
- [4] Babenko, A. and Lempitsky, V. (2015). Aggregating local deep features for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1269–1277.
- [5] Babenko, A., Slesarev, A., Chigorin, A., and Lempitsky, V. (2014a). Neural codes for image retrieval. In *European Conference on Computer Vision*, pages 584–599. Springer.
- [6] Babenko, A., Slesarev, A., Chigorin, A., and Lempitsky, V. S. (2014b). Neural codes for image retrieval. *CoRR*, abs/1404.1777.
- [7] Bai, H., Wang, L., Dong, Y., and Tao, K. (2013). Interactive video retrieval using combination of semantic index and instance search. In *International Conference on Multimedia Modeling*, pages 554–556. Springer.
- [8] Bakar, S. A., Hitam, M. S., and Yussof, W. N. J. H. W. (2013). Content-based image retrieval using sift for binary and greyscale images. In *Signal and Image Processing Applications (ICSIPA), 2013 IEEE International Conference on*, pages 83–88. IEEE.
- [9] Barbieri, M., Mekenkamp, G., Ceccarelli, M., and Nesvadba, J. (2001). The color browser: A content driven linear video browsing tool. *2012 IEEE International Conference on Multimedia and Expo*, 0:160.
- [10] Barthel, K. U., Hezel, N., and Mackowiak, R. (2015). Imagemap-visually browsing millions of images. In *International Conference on Multimedia Modeling*, pages 287–290. Springer.
- [11] Basirat, A. H. and Khan, A. I. (2010a). Evolution of information retrieval in cloud computing by redesigning data management architecture from a scalable associative computing perspective. In *Proceedings of the 17th International Conference on Neural Information Processing: Models and Applications - Volume Part II, ICONIP'10*, pages 275–282, Berlin, Heidelberg. Springer-Verlag.

-
- [12] Basirat, A. H. and Khan, A. I. (2010b). Evolution of information retrieval in cloud computing by redesigning data management architecture from a scalable associative computing perspective. In *International Conference on Neural Information Processing*, pages 275–282. Springer.
- [13] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359.
- [14] Box, G. E., Hunter, W. G., Hunter, J. S., et al. (1978). *Statistics for experimenters*.
- [15] Bradski, G. and Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc."
- [16] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27.
- [17] Christel, M. G. (2008). Supporting video library exploratory search: When storyboards are not enough. In *Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval, CIVR '08*, pages 447–456, New York, NY, USA. ACM.
- [18] Chum, O., Philbin, J., Sivic, J., Isard, M., and Zisserman, A. (2007). Total recall: Automatic query expansion with a generative feature model for object retrieval. In *IEEE International Conference on Computer Vision*.
- [19] Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L., and Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1237.
- [20] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE.
- [21] Datta, R., Li, J., and Wang, J. Z. (2005). Content-based image retrieval: Approaches and trends of the new age. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval, MIR '05*, pages 253–262, New York, NY, USA. ACM.
- [22] Deng, Y. and Manjunath, B. S. (1997). Content-based search of video using color texture and motion. In *IEEE International Conference on Image Processing*, volume 2, pages 534–537.
- [23] Fischler, M. A. and Bolles, R. C. (1981a). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.
- [24] Fischler, M. A. and Bolles, R. C. (1981b). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

-
- [25] Frisson, C., Dupont, S., Moinet, A., Picard-Limpens, C., Ravet, T., Siebert, X., and Dutoit, T. (2013). Videocycle: user-friendly navigation by similarity in video databases. In *International Conference on Multimedia Modeling*, pages 550–553. Springer.
- [26] Gibbon, D. C. and Liu, Z. (2008). *Introduction to video search engines*. Springer Science & Business Media.
- [27] Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524.
- [28] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.
- [29] Hauptmann, A., Yan, R., and Lin, W.-H. (2007). How many high-level concepts will fill the semantic gap in news video retrieval? In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 627–634. ACM.
- [30] Heesch, D., Howarth, P., Magalhães, J., May, A., Pickering, M., Yavlinsky, A., and Rüger, S. (2004). Video retrieval using search and browsing. In *TREC2004 – TEXT RETRIEVAL CONFERENCE*, pages 15–19. Publications.
- [31] Hu, W., Xie, N., Li, L., Zeng, X., and Maybank, S. (2011). A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):797–819.
- [32] Hughes, M., Zhang, Z., Newman, E., and Smeaton, A. F. (2012). A lifelogging approach to automated market research. In *SenseCam Symposium 2012*.
- [33] Hürst, W., Götz, G., and Lauer, T. (2004). New methods for visual information seeking through video browsing. In *8th International Conference on Information Visualisation, IV 2004, 14-16 July 2004, London, UK*, pages 450–455.
- [34] Hürst, W., van de Werken, R., and Hoet, M. (2015). A storyboard-based interface for mobile video browsing. In *International Conference on Multimedia Modeling*, pages 261–265. Springer.
- [35] Hussain, S., Hashmani, M., Moinuddin, M., Yoshida, M., and Kanjo, H. (2012). Image retrieval based on color and texture feature using artificial neural network. In *International Multi Topic Conference*, pages 501–511. Springer.
- [36] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323.
- [37] Jain, R. and Hampapur, A. (1994). Metadata in video databases. *ACM Sigmod Record*, 23(4):27–33.
- [38] Jain, R., Prabhakaran, B., Worring, M., Smith, J. R., and Chua, T., editors (2013). *International Conference on Multimedia Retrieval, ICMR’13, Dallas, TX, USA, April 16-19, 2013*. ACM.

-
- [39] Jegou, H., Douze, M., and Schmid, C. (2008a). Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision: Part I, ECCV '08*, pages 304–317, Berlin, Heidelberg. Springer-Verlag.
- [40] Jegou, H., Douze, M., and Schmid, C. (2008b). Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision: Part I, ECCV '08*, pages 304–317, Berlin, Heidelberg. Springer-Verlag.
- [41] Jégou, H., Douze, M., and Schmid, C. (2009). Searching with quantization: approximate nearest neighbor search using short codes and distance estimators. Technical Report RR-7020, INRIA.
- [42] Jegou, H., Douze, M., and Schmid, C. (2011). Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128.
- [43] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia, MM '14*, pages 675–678, New York, NY, USA. ACM.
- [44] Kalantidis, Y., Pueyo, L. G., Trevisiol, M., van Zwol, R., and Avrithis, Y. (2011). Scalable triangulation-based logo recognition. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ICMR '11*, pages 20:1–20:7, New York, NY, USA. ACM.
- [45] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- [46] Lai, C. and Chen, Y. (2011). A user-oriented image retrieval system based on interactive genetic algorithm. *IEEE Trans. Instrumentation and Measurement*, 60(10):3318–3325.
- [47] Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113.
- [48] Lew, M. S., Sebe, N., Djeraba, C., and Jain, R. (2006). Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2(1):1–19.
- [49] Lin, D., Fidler, S., Kong, C., and Urtasun, R. (2014). Visual semantic search: Retrieving videos via complex textual queries. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [50] Liu, G.-H. and Yang, J.-Y. (2013). Content-based image retrieval using color difference histogram. *Pattern Recogn.*, 46(1):188–198.
- [51] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

- [52] Malisiewicz, T., Gupta, A., and Efros, A. A. (2011). Ensemble of exemplar-svms for object detection and beyond. In *ICCV*.
- [53] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- [54] McGuinness, K., Mohedano, E., Zhang, Z., Hu, F., Albatal, R., Gurrin, C., O'Connor, N. E., Smeaton, A. F., Salvador, A., Giro-i Nieto, X., and Ventura (2014). Insight centre for data analytics (dcu) at trecvid 2014: instance search and semantic indexing tasks. In *2014 TREC Video Retrieval Evaluation*.
- [55] McGuinness, K., Mohedano, E., Zhang, Z., O'Connor, N. E., and Smeaton, A. F. (2015). Insight dcu at trecvid 2015. In *2015 TREC Video Retrieval Evaluation*.
- [56] Mikolajczyk, K., Leibe, B., and Schiele, B. (2006). Multiple object class detection with a generative model. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1, CVPR '06*, pages 26–36, Washington, DC, USA. IEEE Computer Society.
- [57] Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86.
- [58] Mohedano, E., Salvador, A., McGuinness, K., Marques, F., O'Connor, N. E., and Giro-i Nieto, X. (2016). Bags of local convolutional features for scalable instance search. *arXiv preprint arXiv:1604.04653*.
- [59] Monserrat, T.-J. K. P., Zhao, S., McGee, K., and Pandey, A. V. (2013). Notevideo: Facilitating navigation of blackboard-style lecture videos. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13*, pages 1139–1148, New York, NY, USA. ACM.
- [60] Moreno, P., Marín-Jiménez, M. J., Bernardino, A., Santos-Victor, J., and de la Blanca, N. P. (2007). A comparative study of local descriptors for object category recognition: Sift vs hmax. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 515–522. Springer.
- [61] Moutzidou, A., Mironidis, T., Apostolidis, E., Markatopoulou, F., Ioannidou, A., Gialampoukidis, I., Avgerinakis, K., Vrochidis, S., Mezaris, V., Kompatsiaris, I., et al. (2016). Verge: A multimodal interactive search engine for video browsing and retrieval. In *International Conference on Multimedia Modeling*, pages 394–399. Springer.
- [62] Ng, J. Y., Yang, F., and Davis, L. S. (2015). Exploiting local features from deep networks for image retrieval. *CoRR*, abs/1504.05133.
- [63] Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, CVPR '06*, pages 2161–2168, Washington, DC, USA. IEEE Computer Society.
- [64] Obdrzalek, S. and Matas, J. (2005). Sub-linear Indexing for Large Scale Object Recognition. In *BMVC 2005*.

- [65] Over, P., Awad, G., Michel, M., Fiscus, J., Kraaij, W., Smeaton, A. F., Quénot, G., and Ordelman, R. (2015). Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2015*. NIST, USA.
- [66] Patel, B. and Meshram, B. (2012b). Content based video retrieval systems. *arXiv preprint arXiv:1205.1641*.
- [67] Patel, B. and Meshram, B. (April 2012a). Content based video retrieval systems. In *International Journal of UbiComp (IJU)*.
- [68] Peker, K. A. and Divakaran, A. (2004). Adaptive fast playback-based video skimming using a compressed-domain visual complexity measure. In *ICME*, pages 2055–2058. IEEE.
- [69] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- [70] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [71] Philbin, J. and Zisserman, A. (2008). Object mining using a matching graph on very large image collections. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 738–745. IEEE.
- [72] Pickering, M. J. and Rüger, S. (2003). Evaluation of key frame-based retrieval techniques for video. *Comput. Vis. Image Underst.*, 92(2-3):217–235.
- [73] Razavian, A. S., Sullivan, J., Maki, A., and Carlsson, S. (2014). Visual instance retrieval with deep convolutional networks. *CoRR*, abs/1412.6574.
- [74] Romberg, S. and Lienhart, R. (2013). Bundle min-hashing for logo recognition. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval*, ICMR '13, pages 113–120, New York, NY, USA. ACM.
- [75] Romberg, S., Pueyo, L. G., Lienhart, R., and van Zwol, R. (2011). Scalable logo recognition in real-world images. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, pages 25:1–25:8, New York, NY, USA. ACM.
- [76] Rossetto, L., Giangreco, I., Schuldt, H., Dupont, S., Seddati, O., Sezgin, M., and Sahillioğlu, Y. (2015). Imotion—a content-based video retrieval engine. In *International Conference on Multimedia Modeling*, pages 255–260. Springer.
- [77] Schoeffmann, K. (2014). A user-centric media retrieval competition: The video browser showdown 2012-2014. *MultiMedia, IEEE*, 21(4):8–13.
- [78] Schoeffmann, K., Ahlström, D., Bailer, W., Cobarzan, C., Hopfgartner, F., McGuinness, K., Gurrin, C., Frisson, C., Le, D.-D., Del Fabro, M., Bai, H., and Weiss, W. (2013). The video browser showdown: a live evaluation of interactive video search tools. *International Journal of Multimedia Information Retrieval*, pages 1–15.

- [79] Schoeffmann, K. and Boeszoermenyi, L. (2009). Video browsing using interactive navigation summaries. In *Content-Based Multimedia Indexing, 2009. CBMI'09. Seventh International Workshop on*, pages 243–248. IEEE.
- [80] Schoeffmann, K., Hopfgartner, F., Marques, O., Boeszoermenyi, L., and Jose, J. M. (2010). Video browsing interfaces and applications: a review. *Journal of Photonics for Energy*, pages 018004–018004.
- [81] Scott, D., Guo, J., Gurrin, C., Hopfgartner, F., McGuinness, K., O'Connor, N. E., Smeaton, A. F., Yang, Y., and Zhang, Z. (2013). Dcu at mmm 2013 video browser show-down. In *International Conference on Multimedia Modeling*, pages 541–543. Springer.
- [82] Scott, D., Zhang, Z., Albatal, R., McGuinness, K., Acar, E., Hopfgartner, F., Gurrin, C., O'Connor, N. E., and Smeaton, A. F. (2014). Audio-visual classification video browser. In *International Conference on Multimedia Modeling*, pages 398–401. Springer.
- [83] Shang, L., Yang, L., Wang, F., Chan, K.-P., and Hua, X.-S. (2010). Real-time large scale near-duplicate web video retrieval. In *Proceedings of the 18th ACM International Conference on Multimedia, MM '10*, pages 531–540, New York, NY, USA. ACM.
- [84] Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813.
- [85] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [86] Singla, P. and Domingos, P. (2005). Object identification with attribute-mediated dependences. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 297–308. Springer.
- [87] Sivic, J. and Zisserman, A. (2004). Video data mining using configurations of viewpoint invariant regions. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–488. IEEE.
- [88] Sivic, J. and Zisserman, A. (2006). Video google: Efficient visual search of videos. In *Toward category-level object recognition*, pages 127–144. Springer.
- [89] Smeaton, A. F., Over, P., and Doherty, A. R. (2010). Video shot boundary detection: Seven years of trecvid activity. *Computer Vision and Image Understanding*, 114(4):411–418.
- [90] Smeaton, A. F., Over, P., and Kraaij, W. (2006a). Evaluation campaigns and trecvid. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, MIR '06*, pages 321–330, New York, NY, USA. ACM.
- [91] Smeaton, A. F., Over, P., and Kraaij, W. (2006b). Evaluation campaigns and trecvid. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 321–330. ACM.

- [92] Smeulders, A. W., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12):1349–1380.
- [93] Smucker, M. D., Allan, J., and Carterette, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632. ACM.
- [94] Snavely, N., Simon, I., Goesele, M., Szeliski, R., and Seitz, S. M. (2010). Scene reconstruction and visualization from community photo collections. *Proceedings of the IEEE*, 98(8):1370–1390.
- [95] Snoek, C. G. and Worring, M. (2005a). Multimodal video indexing: A review of the state-of-the-art. *Multimedia tools and applications*, 25(1):5–35.
- [96] Snoek, C. G. and Worring, M. (2008). Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2(4):215–322.
- [97] Snoek, C. G. M. and Worring, M. (2005b). Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools Appl.*, 25(1):5–35.
- [98] Song, Y. and Marchionini, G. (2007). Effects of audio and visual surrogates for making sense of digital video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 867–876, New York, NY, USA. ACM.
- [99] Vedaldi, A. and Fulkerson, B. (2010). Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1469–1472. ACM.
- [100] Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *SIGIR'94*, pages 61–69. Springer.
- [101] Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., and Li, J. (2014). Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 157–166, New York, NY, USA. ACM.
- [102] Wildemuth, B., Marchionini, G., Yang, M., Geisler, G., Wilkens, T., Hughes, A., and Gruss, R. (2003). *How fast is too fast? evaluating fast forward surrogates for digital video*, volume 2003-January, pages 221–230. Institute of Electrical and Electronics Engineers Inc.
- [103] Winn, J. and Criminisi, A. (2006). Object class recognition at a glance. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR) – Video Track*.
- [104] Wu, J., Cui, Z., Sheng, V. S., Zhao, P., Su, D., and Gong, S. (2013). A comparative study of sift and its variants. *Measurement Science Review*, 13(3):122–131.
- [105] Wu, Z., Ke, Q., Isard, M., and Sun, J. (2009). 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.

- [106] Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11. ACM.
- [107] Yang, H., Shao, L., Zheng, F., Wang, L., and Song, Z. (2011). Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831.
- [108] Yue-Hei Ng, J., Yang, F., and Davis, L. S. (2015). Exploiting local features from deep networks for image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 53–61.
- [109] Zhang, H. J., Wu, J., Zhong, D., and Smoliar, S. W. (1997). An integrated system for content-based video retrieval and browsing. *Pattern recognition*, 30(4):643–658.
- [110] Zhang, Z., Albatal, R., Gurrin, C., and Smeaton, A. F. (2013a). Trecvid 2013 experiments at dublin city university. In *2013 TREC Video Retrieval Evaluation*.
- [111] Zhang, Z., Albatal, R., Gurrin, C., and Smeaton, A. F. (2013b). Trecvid 2013 experiments at dublin city university. In *2013 TREC Video Retrieval Evaluation*.
- [112] Zhang, Z., Albatal, R., Gurrin, C., and Smeaton, A. F. (2015). Interactive known-item search using semantic textual and colour modalities. In *International Conference on Multimedia Modeling*, pages 282–286. Springer.
- [113] Zhang, Z., Albatal, R., Gurrin, C., and Smeaton, A. F. (2016b). Instance search with weak geometric correlation consistency. In *The 22nd International Conference on Multimedia Modelling (MMM'16)*.
- [114] Zhang, Z., Albatal, R., Gurrin, C., and Smeaton, A. F. (in press 2016a). Enhancing instance search with weak geometric correlation consistency. *Journal of Neurocomputing*.
- [115] Zhang, Z., Li, W., Gurrin, C., and Smeaton, A. F. (2016c). Faceted navigation for browsing large video collection. In *International Conference on Multimedia Modeling*, pages 412–417. Springer.
- [116] Zhang, Z., Yang, Y., Cui, R., and Gurrin, C. (2014). Eolas: video retrieval application for helping tourists. In *The 20th Anniversary International Conference on MultiMedia Modeling*.
- [117] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014a). Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495.
- [118] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014b). Learning deep features for scene recognition using places database. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 487–495. Curran Associates, Inc.
- [119] Zhou, W., Lu, Y., Li, H., Song, Y., and Tian, Q. (2010). Spatial coding for large scale partial-duplicate web image search. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 511–520. ACM.
- [120] Zobel, J., Moffat, A., and Ramamohanarao, K. (1998). Inverted files versus signature files for text indexing. *ACM Trans. Database Syst.*, 23(4):453–490.

Appendix A

The Applications of Instance Search

Video Search Engine

The most widely used application for instance search is to provide video search or browsing functions to address the challenges of locating interesting video clips from large collections, given a query image of an interesting instance. Figure A.1 shows an example application developed during this research to search for locations within the Oxford building [69] test collection. Users are allowed to upload their own query image and select the interesting instance by cropping the image. They can also quickly view the result images as they were listed in grid layout just behind the query panel. This kind of video retrieval system is useful, for example, to easily find clips of a certain person or location from large surveillance video archives in crime investigation.

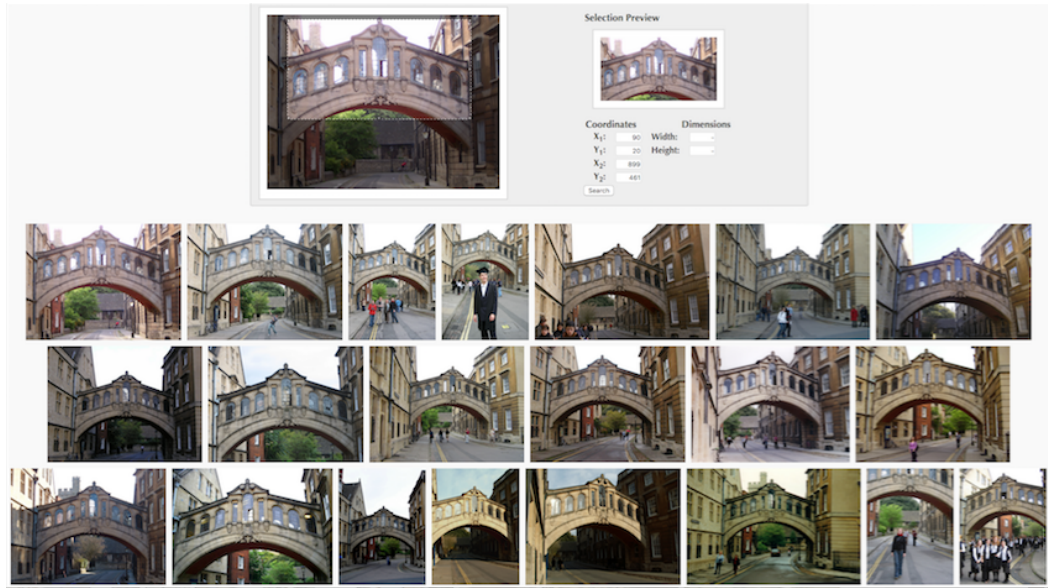


Fig. A.1 Video search: searching landmarks within the Oxford building [69] test collection.

Qualitative Market Study

As an application of instance search, we have developed an analysis tool [32] to support qualitative market research, such as evaluating the effectiveness of advertisement campaigns. This system can measure audience exposure to specific advertising campaigns, using instance search algorithms to automatically detect the presence of known logos in large real-life photo collections. Figure A.2 presents the system interface to the software tool. Data is presented to users based on temporal layout (left figure) and spatial layout (right figure). The temporal based interface shows the numbers of detected logo instance over the course of several different time frames, such as hourly, daily, weekly etc. The Spatial based interface displays the location information as an overlay on a map, where each detected instance was found. The map interface is divided into a grid with colour coding of each square relating to the frequency of each detected instance at the corresponding location.

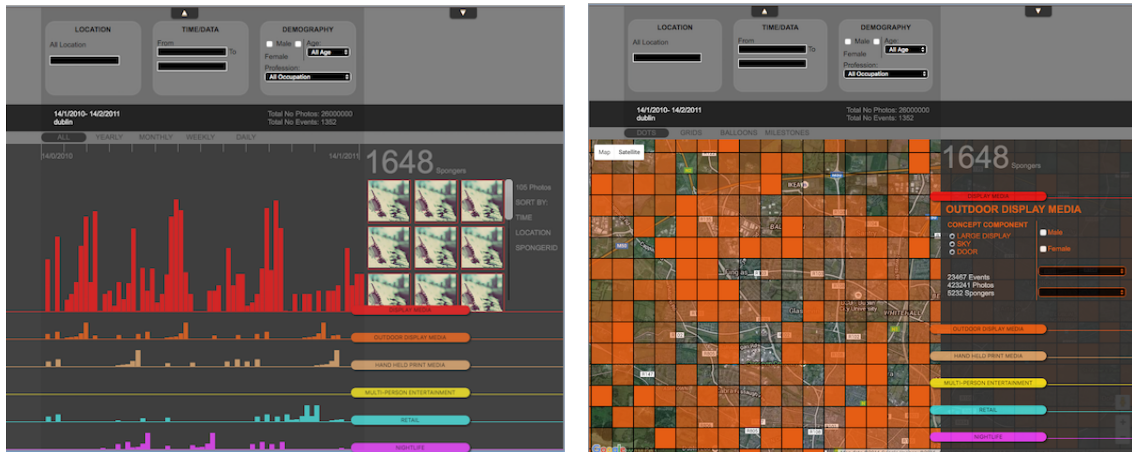


Fig. A.2 User Interface for SpongeIT [32]: a research tool for qualitative market study

Mobile Search

By taking a picture with a camera, mobile phones have become an ideal interactive media for people to access the massive video collections. In the work of the Eolas [116] mobile application, we built an instance search system to help tourists find documentary videos to understand more about a landmark of interest. While traveling abroad, this application was particularly useful to overcome the language barrier when seeking tourist information by simply taking a snapshot of interesting landmarks. Additionally, it also made a contribution to bringing state-of-the-art video retrieval technologies beyond a laboratory environment into a real-time mobile device usage. Fig A.3 shows the interactive interface for the Eolas mobile application.

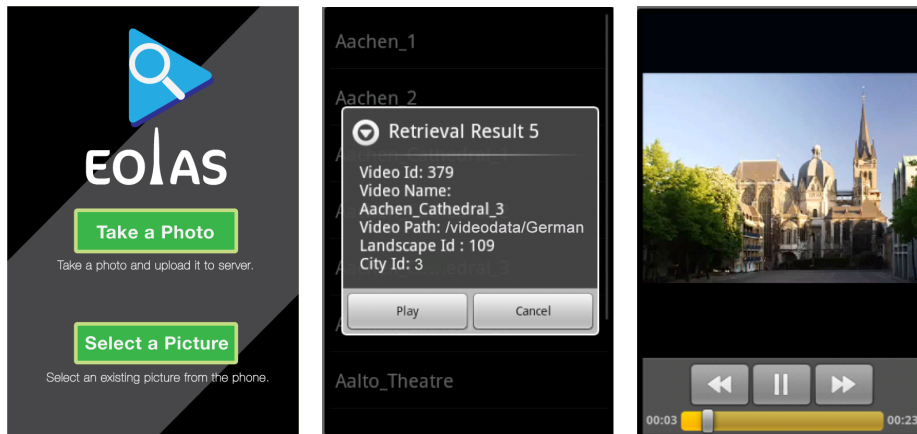


Fig. A.3 Snapshot of Eolas mobile application GUI