

**GEO-REFERENCED VIDEO RETRIEVAL:
TEXT ANNOTATION AND SIMILARITY SEARCH**

YIN Yifang

(B.Sc., Northeastern University, 2011)

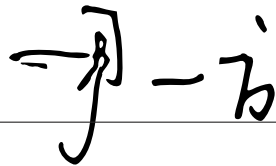
**A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE**

2016

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.



YIN Yifang

18 Jan. 2016

ACKNOWLEDGEMENTS

This dissertation is a summary of my four years research work. I would like to thank all the people who have made this thesis possible. Also thanks to the school for the wonderful research resources and services it provides that help me through the whole PhD programme.

First and foremost I would like to express my sincerest gratitude to my supervisor, Prof. Roger Zimmermann, for his persistent guidance and support throughout my PhD study. His encouragement and patience always help me overcome the difficulties that I have encountered in life. Thanks for helping me with paper writing and polishing. Without his inspiration and advices, this thesis would not have been completed.

I am deeply grateful to all my group members and labmates in NUS: Beomjoo Seo, Yu Yi, Liu Zhenguang, Hao Jia, Shen Zhijie, Ma He, Zhang Ying, Ma Haiyang, Zhang Lingyan, Cui Weiwei, Rajiv Ratn Shah, Abhay Sharma, Subhasree Basu, Bayan Ta'ani, Abdelhak Betaleb, for their support and accompany. I would also like to thank my friends in Singapore, for the wonderful years that we have spent together.

Last but not the least, I would like to express my deepest gratitude to my family. A special love goes to my late father, who had been a great mentor in my life and had constantly encouraged me to be a better person.

LIST OF PUBLICATIONS

- **Yifang Yin**, Yi Yu, Roger Zimmermann, “On Generating Content-Oriented Geo Features for Sensor-Rich Outdoor Video Search”. In *IEEE Transactions on Multimedia*, Volume 17 Issue 10, pages 1760-1772, 2015.
- **Yifang Yin**, Luming Zhang, Roger Zimmermann, “Exploiting Spatial Relationship between Scenes for Hierarchical Video Geotagging”. In *Proceedings of ACM International Conference on Multimedia Retrieval*, pages 363-370, 2015.
- **Yifang Yin**, Beomjoo Seo, Roger Zimmermann, “Content vs. Context: Visual and Geographic Information Use in Video Landmark Retrieval”. In *ACM Transactions on Multimedia Computing, Communications, and Applications*, Volume 11 Issue 2, Article No. 39, pages 39:1-39:21, 2015.
- **Yifang Yin**, Zhijie Shen, Luming Zhang, Roger Zimmermann. “Spatial-Temporal Tag Mining for Automatic Geospatial Video Annotation”. In *ACM Transactions on Multimedia Computing, Communications, and Applications*, Volume 11 Issue 2, Article No. 29, pages 29:1-29:21, 2014.
- Guanfeng Wang, **Yifang Yin**, Beomjoo Seo, Roger Zimmermann, Zhijie Shen, “Orientation data correction with georeferenced mobile videos”. In *Proceedings of ACM International Conference on Advances in Geographic Information Systems*, pages 390-393, 2013.
- Guanfeng Wang, Beomjoo Seo, **Yifang Yin**, Roger Zimmermann, Zhijie Shen, “OSCOR: an orientation sensor data correction system for mobile generated contents”. In *Proceedings of ACM International Conference on Multimedia*, pages 439-440, 2013.

CONTENTS

Summary	v
List of Figures	vii
List of Tables	xi
1 Introduction	1
1.1 Background and Motivation	1
1.2 Overview	9
1.3 Roadmap	12
2 Literature Review	14
2.1 Geo-metadata Correction	14
2.2 Geo-tagged Image and Video Annotation	16
2.3 Landmark Recognition	20
2.4 Video Similarity Search	23

3	Preliminaries	26
3.1	Viewable Scene Model	26
3.2	Automatic Geo-tagged Video Annotation	28
3.3	Datasets	29
3.4	Notations	30
4	Automatic Geographic Metadata Correction	33
4.1	Introduction	33
4.2	System Overview	36
4.2.1	Design Principles	36
4.2.2	Problem Description	37
4.3	Video Georegistration	38
4.3.1	Camera Model	39
4.3.2	Energy Definition	40
4.3.3	Energy Minimization	46
4.4	Evaluation	47
4.4.1	Experimental Setup	47
4.4.2	Geographic Metadata Correction	48
4.5	Summary	53
5	Spatial-Temporal Tag Mining	55
5.1	Introduction	55
5.2	Review of the Automatic Tag Generation System	57
5.2.1	System Overview	57
5.2.2	Data Source Limitations	57
5.2.3	Seeking More Varied Data Sources	59
5.3	Positioning Social Tags in Spatial-Temporal Domain	61

5.3.1	Geographically Positioning Social Tags	61
5.3.2	Temporally Positioning Social Tags	69
5.4	Extension of the Auto-annotation Approach	72
5.5	Evaluation	74
5.5.1	Accuracy of Positionable Tag Classification	75
5.5.2	Accuracy of Tag Positioning	80
5.5.3	Tag Expansion and Ranking	82
5.6	Summary	86
6	Visual and Geographic Information Use in Video Landmark	
	Retrieval	87
6.1	Introduction	87
6.2	Landmark Retrieval Methods	90
6.2.1	Landmark Retrieval from Visual Cues and Features	90
6.2.2	Landmark Retrieval from Geographic Information	93
6.3	Evaluation	97
6.3.1	Experimental Settings and Datasets	98
6.3.2	Frame Retrieval Evaluation	99
6.3.3	Content-based Method Robustness Analysis	105
6.3.4	Geo-based Method Robustness Analysis	108
6.3.5	Hybrid Integrated Content and Context Analysis	112
6.4	Summary	116
7	Hybrid Video Similarity Search	119
7.1	Introduction	119
7.2	Hybrid Model for Video Representation	122
7.2.1	L1: Geographic Coverage Calculation	123

CONTENTS

7.2.2	L2: Representative Visual Features Selection	126
7.2.3	Video Similarity Measure	127
7.3	Geo-Codebook Generation	129
7.3.1	Problem Formulation	131
7.3.2	Clustering Cells into Coherent Regions	131
7.3.3	Region Saliency Estimation	133
7.4	Evaluation	134
7.4.1	Experimental Setup	134
7.4.2	Geo-Codebook Generation	135
7.4.3	Evaluation on Video Retrieval	137
7.5	Summary	145
8	Conclusions and Future Work	147
	Bibliography	150

SUMMARY

Advanced technologies in consumer electronic products have enabled individual users to record, share and view videos with mobile devices. With the volume of videos increasing tremendously on the Internet, fast and accurate video search and annotation have become urgent tasks and have attracted much research attention. However, video search and management operations are typically supported by either the low-level visual features or the manual textual annotations. Those approaches often suffer from low recall as they are highly susceptible to changes in viewpoint, illumination, and noisy tags. By leveraging geo-metadata, more reliable and precise search results can be obtained. The geographic metadata is one of the important kinds of contextual information. Due to the ubiquity of sensor-equipped smartphones, it has become increasingly feasible for users to capture videos together with the geographic information, for example the location and the orientation of the camera. Such contexts create new opportunities for the organization and retrieval of geo-referenced videos.

This dissertation studies the geographic information use in video annotation and retrieval. Since raw sensor data collected is often noisy, we first preprocess the geo-metadata by building a comprehensive model to reduce the errors in GPS and compass readings. The proposed approach can effectively provide more accurate geo-metadata for downstream applications such as tagging and search. For video annotation, we propose to leverage crowdsourced data from social multimedia applications that host tags of diverse semantics to build a spatial-temporal tag repository. In particular, we retrieve the necessary data from several social multimedia applications, mine both the spatial and temporal features of the tags, and then refine and index them accordingly. Consequently,

the tag repository we built acts as the input to our previous auto-annotation approach which we extend in several ways for better integration with the new vocabulary. For video landmark retrieval, we present the Geo Landmark Visibility Determination (GeoLVD) approach which computes the visibility of a landmark based on intersections of a camera’s Field-of-View (FOV) and the landmark’s geometric information available from geographic information systems and services. We compare our method with the content-based spatial pyramid matching approach combined with two advanced coding methods: sparse coding and locality-constrained linear coding. By analyzing their strength and weakness, we further integrate the visual and geographic information to achieve improvements. For video similarity search, we propose a novel video description which consists of (a) determining the geographic coverage of a video based on the camera’s FOV and a pre-constructed geo-codebook, and (b) fusing video spatial relevance and region-aware visual similarities to achieve a robust video similarity measure. Toward a better encoding of a video’s geo-coverage, we also construct a geo-codebook by segmenting a map into a collection of coherent regions. The experimental results show that the proposed techniques achieved significant improvements over its competitors, especially with fine-grained and accuracy-enhanced geographic metadata.

LIST OF FIGURES

1.1	Applications that benefit from geographic information in multi-media.	2
1.2	Key frame registration to a digital 3D world. [134]	3
1.3	Georeferenced video search architecture. [7]	4
1.4	An overview of the geo-tagged video annotation and retrieval system.	10
3.1	Illustration of the 3D <i>FOVScene</i> model.	27
3.2	Illustration of a sample <i>FOVScene</i> and the visible objects which are supplied by Google Earth and determined by conducting geometry computations (Copyright © 2013 Google).	29
4.1	The overall architecture of the proposed automatic geo-metadata correction framework. Raw sensor data is enhanced to provide more accurate geographic information to downstream applications.	35
4.2	Illustrations of the coordinate systems used in our framework.	37
4.3	Scene understanding by semantic pixel labeling and 3D projection based on camera pose and OSM data.	44
4.4	Raw and processed camera orientation error comparison for individual videos.	49

LIST OF FIGURES

4.5	Effectiveness analysis of sensor data correction algorithms with or without geographic context and its connections to the error patterns in the camera orientation readings.	52
5.1	(a) The architecture of the automatic tag generation framework for sensor-rich outdoor videos, and (b) the process of building a positionable tag repository and interfacing it with the remaining framework.	58
5.2	Conceptual illustration of the placement of tags in the spatial-temporal domain. The dashed lines show the durations of tag usage while the projected circles are the related places of the tags.	59
5.3	Illustration of the global distribution of the geo-coordinates of tag <i>f1</i>	63
5.4	Illustration of the temporal distribution of the timestamps of tag <i>f1</i>	70
5.5	(a) precision, (b) recall and (c) F1 score under different combinations of the number of centers and the sum of priors thresholds.	77
5.6	Cumulative distribution function (CDF) of the distances between the estimated and the real positions.	81
5.7	Illustration of snapshots of sample videos. The top tags are generated with the proposed auto-tagging system based on different datasets.	84
5.8	Comparison of (a) relevance and (b) diversity of the tags generated based on different datasets.	85
6.1	(a) An illustration of a camera’s field of view in Google Earth (Copyright © 2013 Google [36]). (b) The corresponding scene projection on the 2D plane.	96
6.2	Illustration of frames with fully and partially visible landmarks in the test set.	99
6.3	Two typical images of the Singapore Flyer landmark in the experimental dataset: (a) a representative of the Flickr training images (Copyright © 2013 Flickr [33]) and (b) a representative of the frames from the video dataset.	103

6.4	Details of the Google StreetView training images collected near Marina Bay. Right: image location distribution, left: an example of four side views per location. Copyright © 2013 Google [36].	107
6.5	Evaluations with two training sets (Flickr only or Flickr with Google StreetView images, <i>Flickr+StV</i>) and two test sets (partial or entire view of landmarks) for the content-based <i>ScSPM</i> method.	108
6.6	<i>GeoLVD</i> method results with retrieval queries based on Google StreetView images.	110
6.7	<i>GeoLVD</i> method results with retrieval queries based on video frames.	110
6.8	Estimated Gaussian Function of F1 Score over Distance for different landmarks.	113
6.9	Precision-recall curves comparison of methods based on content only, context only, and their hybrid integration.	115
7.1	Illustration of key techniques for geographic and visual feature fusion in our proposed video retrieval system.	121
7.2	Illustration of the proposed hybrid model for video representation.	122
7.3	Illustration of <i>FOVScene</i> model in 2D.	123
7.4	An example of similarity calculation between two videos.	128
7.5	Limitations of a grid-based codebook that cannot satisfactorily capture the diverse granularity of geographic objects.	130
7.6	Examples of the generated geo-codebook in different areas around the world.	136
7.7	Ten geotagged Flickr images used as queries.	137
7.8	P@n comparison of the proposed and the existing fusion methods.	139
7.9	P@n comparison based on different availability of geo-metadata.	141

LIST OF FIGURES

LIST OF TABLES

2.1	A comparison with the previous work.	16
3.1	Notations used in this dissertation.	30
4.1	Georeferenced video dataset description.	48
4.2	Precision comparison of raw and processed GPS data.	48
5.1	30 most popular Flickr tags in the Marina Bay area of Singapore and their corresponding semantics.	62
5.2	Precision, recall and F1 score statistics using the SVM classifier.	76
5.3	Precision, recall and F1 score statistics using the proposed clas- sifier by thresholding.	78
5.4	F1 scores based on different settings of NP and α	80
5.5	Illustrations of the estimated social tags' temporal visibility in- tervals.	80
5.6	Precision comparison of tag expansion based on the true posi- tionable tags $geotags_t$ and the automatically detected position- able tags $geotags_d$	83

LIST OF TABLES

5.7	Illustrations of tag expansion. The tag detected is listed together with its nearest positionable neighbor and the <i>Jensen-Shannon divergence</i> between them.	83
6.1	Retrieval technique comparison over different landmarks and video conditions.	100
6.2	Retrieval technique comparison over supplementary landmarks among day-time videos.	101
6.3	Execution time per query frame of the content- and geo-based methods.	101
6.4	Details of landmark visibilities in Google StreetView images. . .	109
6.5	F1 scores of methods based on content only, context only, and their hybrid integration.	114
7.1	MAP comparison of the proposed and the existing fusion methods.	140
7.2	MAP comparison of fusion with OB and BoS.	141
7.3	MAP comparison based on different availability of geo-metadata.	141
7.4	Mean average precision decrement.	143
7.5	The comparison of the execution time for feature extraction per image.	144
7.6	The comparison of the average retrieval latency.	144

CHAPTER 1

Introduction

1.1 Background and Motivation

With advances in the technology of mobile device manufacturing and network engineering, user-generated videos have become very popular in recent years. The most popular video sharing website YouTube¹ has more than one billion users. According to the official statistics, 300 hours of video are uploaded every minute [2]. This fast growing trend in video volume challenges the traditional media organization schemes. Issues arise such as the high cost for visual feature extraction and matching in content-based video retrieval systems, and the well-known semantic gap between the low-level visual features and the high level semantic concepts. Therefore, automatic understanding and efficient retrieval of the growing user-generated videos are highly desired.

While the traditional content-based methodologies sometimes struggle to

¹<https://www.youtube.com/>

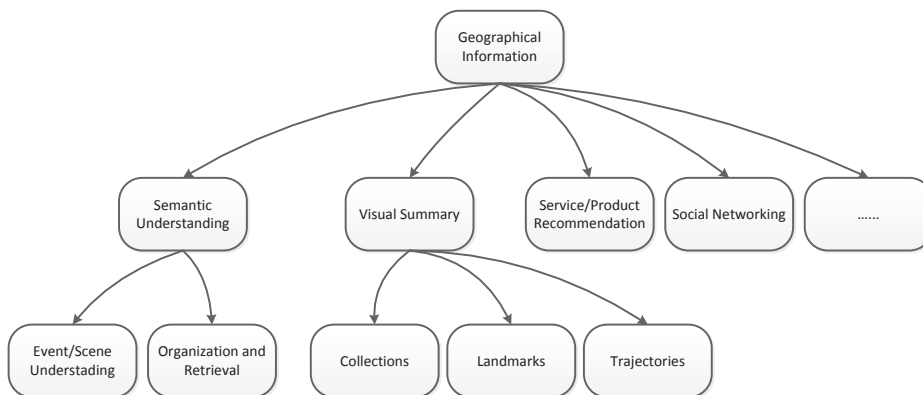


Figure 1.1: Applications that benefit from geographic information in multimedia.

achieve satisfying results, people have begun to realize the importance of using the contextual information. The geographic metadata is just one important type of such information that helps a lot in a variety of research domains [79]. Figure 1.1 shows the applications that can benefit from geographic information in the multimedia domain. As can be seen, the presence of geographically relevant metadata with images and videos has opened up interesting research avenues within the multimedia domain. Nowadays the geotagged images are pervasive in people’s life from community photo collections to the worldwide Google Street View Service [36]. With today’s sensor-equipped smartphones, it is also common to tag recorded videos with a continuous stream of extended geographic properties that relate to the camera scenes. In our group’s prior work [8], Arslan Ay *et al.* proposed a sensor-based description of video scenes. To the best of our knowledge, this is the first work that addresses the issue of modelling the actual geographic information of video scenes and utilizing it for an efficient video search. The initial sensor rich video recording system incorporates three devices: a video camera, a 3D digital compass, and a Global

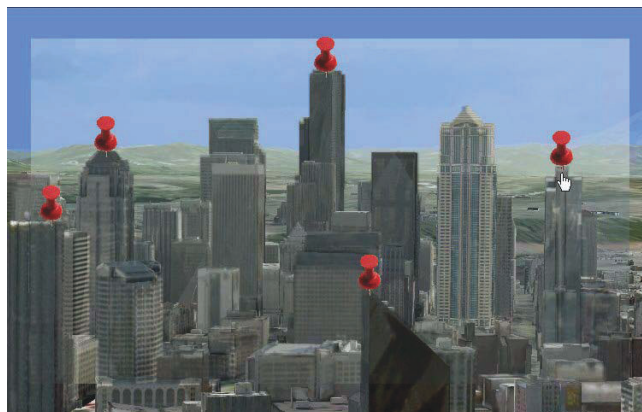


Figure 1.2: Key frame registration to a digital 3D world. [134]

Positioning System (GPS) device. GPS records the latitude, longitude, and altitude, while a compass records the heading angle together with the pitch and roll values. Considering the popularity of the sensor-equipped smartphones, we developed geographic video recording applications for both Android- and iOS-based mobile phones as well.

The media geographic coordinates can also be obtained by analyzing content and context such as title, tags, and text description [97, 55, 134]. This is the so-called *geotagging*, whose goal is to determine the unknown location of an image or a video. Though the state-of-the-art methods have reported promising results, the accuracy is still far beyond satisfaction. In the annotation and navigation system for tourist videos proposed by Zhang *et al.* [134], not only the position but also the orientation of the camera can be obtained by registering videos to a digital 3D world. For each key frame specified, they use an interactive registration tool to align the image to 3D terrain and building models as shown in Figure 1.2. However, they neglect to report the detailed accuracy of their video registration method. In this study, we only focus on the geographic information obtained from sensors such as GPS and digital compass,

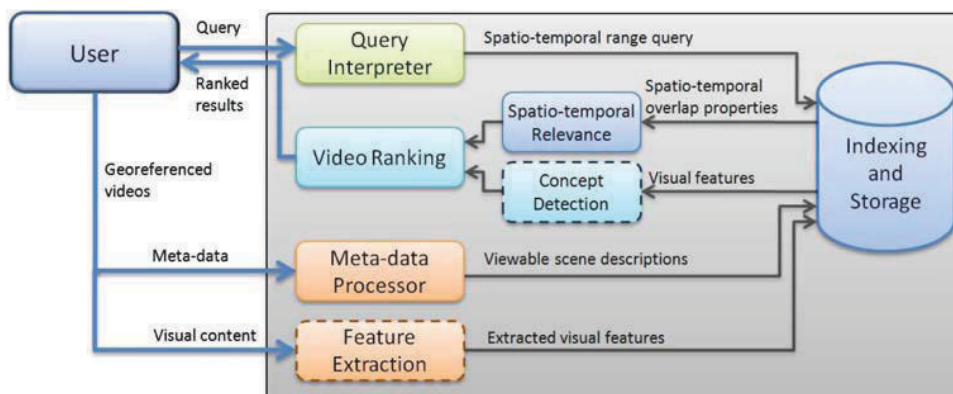


Figure 1.3: Georeferenced video search architecture. [7]

and geotagging techniques hardly fall within the scope of this discussion.

Based on the camera’s field-of-view model, Kim *et al.* designed and implemented a geo-tagged video search framework [57]. Figure 1.3 shows the architecture of the search engine. Queries supported in the system include: (1) point query, (2) point query with bounded distance r , (3) directional point query, and (4) rectangular range query. The *Video Ranking* module in Figure 1.3 rates search results according to the spatiotemporal overlap properties. Three metrics have been proposed to describe the relevance of a video [7]: (1) Total Overlap Area, (2) Overlap Duration, and (3) Summed Area of Overlap Regions. The *Concept Detection* module provides information about the semantic content of the video segments to aid the ranking process, which is currently not implemented and left for the future work.

As can be seen that the geographic information helps in understanding the semantics of image and video contents, *e.g.*, the type of event or scene that the user has captured. It is especially useful for tourist image and video collections, in terms of media organization, retrieval and navigation. However, most of the previous work only focuses on the management of images. Though geo-based video retrieval techniques have been proposed, there is a lack of thorough

comparison between the rising geo-based techniques and the traditional content-based techniques. The integration of visual and geographic information is still in the initial stage which is of great necessity to be further developed and evaluated. In this dissertation, we focus on the geographic information use in video annotation and search. Before we introduce the details of the proposed techniques, let us first have a look at the major issues existing in the current geo-referenced video management systems.

- The geo-metadata collected from sensors can be noisy and inaccurate. While for GPS this issue has been extensively studied [17, 20], only a few efforts have been made on the correction of compass data [81, 89]. The difficulty level of this problem is especially high since (a) unlike GPS, a compass sensor does not provide any accuracy bounds, and (b) from our empirical observations compass errors can sometimes be very high (up to 180°). Considering the large amount of geo-metadata, a fully automatic approach is preferred.
- The world is not a sensor-rich environment where every object has a geotag. Note that the geographic context of the world can greatly help people understand the video content based on the camera location and orientation. Fortunately, nowadays spatial data have become increasingly available from mapping services on the Internet. Apart from the physical entities such as buildings and landmarks, it is also possible to determine the geotag of events or other concepts of interest from crowdsourced data available online (*e.g.*, social multimedia applications such as Flickr). A rich spatial-temporal tag repository is highly desired.
- Most of the geo-tagged videos are only associated with a single GPS loca-

tion. This is not always helpful since a camera can move in both location and viewing direction. We believe that it is important and indispensable to propose effective techniques with fine-grained contextual information where every video frame is tagged. Per-frame camera geographic properties can be either processed individually or compressed to video-level features for efficiency concerns.

To solve the issues outlined above, we try to maximize the use of the geographic metadata in automatic annotation and retrieval of geo-referenced video collections. Raw sensor data collected is often noisy, resulting in subsequent inaccurate geospatial analysis. Therefore, we first focus on the challenging correction of compass data and present an automatic approach to reduce the errors. Given the small geo-distance between consecutive video frames, image-based localization does not work due to the high ambiguity in the depth reconstruction of the scene. As an alternative, we collect geographic context from OpenStreetMap and estimate the absolute viewing direction by comparing the image scene to world projections obtained with different external camera parameters. To design a comprehensive model, we further incorporate smooth approximation and feature-based rotation estimation when formulating the error terms. Experimental results show that our proposed pyramid-based method outperforms its competitors and reduces orientation errors by an average of 58.8%. Hence, for downstream applications, improved results can be obtained with the accuracy-enhanced geo-metadata.

In our prior work, we determine the geographic objects that are visible in a video based on the viewable scene descriptions, by querying the geographic information systems and services. However, the performance of this approach

significantly depends on the quality of the adopted data sources. Previously we built our prototype with the OpenStreetMap. However, its completeness varies in different regions. To enrich the vocabulary for video tagging, we began to seek for more diverse data sources. One of the promising information sources could be the crowdsourced data available from social multimedia applications, such as Flickr, Picasa and YouTube, where the semantics of multimedia documents can be acquired by analysing the user-generated tags. The geo-coordinates of a tag are likely to be unevenly distributed. To identify where the peaks are, we construct Gaussian mixture models to describe the distribution of geo-coordinates. By doing this, the geo-coordinates are replaced with continuous kernel functions to create summary statistics that are less sensitive to high-frequency noise in the data. Given the distribution characteristics, classifiers are built to determine if it is positionable tag that can be added to our spatial-temporal tag repository.

Next, we study and compare the visual and geographic information use in video landmark retrieval. In recent years the bag-of-words (*BoW*) model [28], which is inspired by the success of text-based retrieval, has been extremely popular in a variety of visual retrieval and categorization tasks. The basic idea behind the model is to view an image as a document comprised of unordered visual words. Then a classification tool such as a support vector machine (*SVM*) classifier can be trained to perform landmark categorization based on the labelled *BoW* representations of a training set. Once the *SVM* classifier is trained, it can be used to retrieve images of a certain landmark because it is capable of distinguishing to which category an image belongs. The original *BoW* approach has been improved in various ways in its performance [126, 119]. Such content-based retrieval, however, has a drawback that hinders its scalability:

high computational complexity due to extensive image processing. Moreover, it is highly susceptible to environmental conditions of the image, *e.g.*, the illumination and shooting angle. Additionally, we present a lightweight geo-based approach termed Geo Landmark Visibility Determination (GeoLVD). We analyzed the factors that may affect the retrieval performance. For the content-based method, we analyzed the influence brought by the representativeness of the training set and the diversity of the video frames. We also seek for better image sources to select training images, and propose to use Google Street View as a supplement to Flickr, which has been shown to be effective in improving the retrieval performance by the experiment. For the geo-based method, we analyzed the influence brought by the accuracy of the video’s geographic metadata and the detail level of the information we collected from the geographic information systems. Finally we propose a hybrid retrieval method based on the integration of the visual and geographic information. Experiments show that it achieves great improvements in terms of precision and recall.

Additionally, we study the problem of video similarity search. A good similarity measure is a key component in such a retrieval system. While the viewable scene model [8] has been adopted for many geo-referenced video applications [7, 134, 99], one fundamental issue is it describes the camera properties rather than the video content. We argue that content-oriented geo features are highly desired because their consistency with visual clues can make the fusion more seamlessly. Therefore, we propose a novel two-layer model in which frames are indexed by the regions they capture instead of the camera location. Subsequently, geo and visual features are directly connected via regions. Based on this model, a novel video similarity measure is proposed by summing up local similarity scores on a region-by-region basis. Toward a better encoding

of the geographic features, we present the *Geo-Codebook Generation* module, which segments a map into a collection of coherent regions as a geo-codebook. Compared with a grid-based geo-codebook, our approach is shown to be more descriptive and thus leads to a better similarity measure.

In order to evaluate the proposed methods, ground-truth annotations were required and manually labeled with the help of my colleagues and labmates. We first labeled the ground-truth individually, and then discussed together to make an agreement. It is worth mentioning that this dissertation introduces an effectiveness hybrid video retrieval system, which consists of multiple components and parameters. Thus, within the limited time we only focus on several key components and design our algorithms in a way that can be easily integrated with other existing techniques in the rest of the modules. Currently for the modules that are not our research focus, we only choose one of the popular off-the-shelf techniques for illustration, but please keep in mind that these modules can be easily replaced by more advanced techniques for further improvements. Additionally, the optimal setting of some parameters is related to the data characteristics. We would recommend users to tune such parameters with a subset of the data, as how we proceed in our experiments.

1.2 Overview

As illustrated in Figure 1.4, this study concentrates on the annotation and retrieval of fine-grained geo-tagged videos. Four major components in such a system are presented in this dissertation. We first preprocess the geo-metadata to reduce the errors in the raw sensor data. Next, we mine the spatial-temporal tags from social multimedia applications such as Flickr to enrich our tag repos-

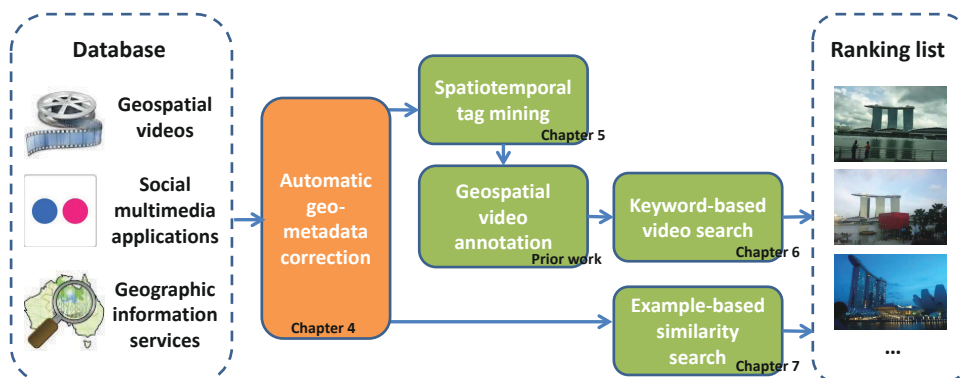


Figure 1.4: An overview of the geo-tagged video annotation and retrieval system.

itory for automatic video annotation. Thereafter, we study two types of video queries, namely keyword-based video search and example-based similarity search, and propose a hybrid model for each problem by fusing geo and visual features to improve the retrieval effectiveness. Experimental results have shown that our proposed approaches achieved significant improvements. The contributions of each work are listed below:

Automatic geo-metadata correction. To reduce the errors in the geographic metadata, we focus on the challenging compass data correction and formulate the task as an optimization problem by leveraging a set of complementary data sources. First, constrained by the geographic priors, the optimized camera parameters should not drift too far away from the sensor readings. Next, we estimate relative rotation between consecutive frames by performing local feature matching with SIFT descriptors. To determine the absolute viewing direction, we quantify the distance between the pixel semantic labels of the frame and the 3D projections of the scene. Two distance metrics have been designed and implemented, namely the pixel-based and the pyramid-based measure that encode the spatial information of the semantic labels with different granularity.

By minimizing the formulated objectives, we are able to reduce the errors in the raw sensor data and provide more accurate geo-information as an input for downstream applications.

Spatial-temporal tag mining. In order to enrich the candidate tag repository for geo-tagged video annotation, we concentrate on how to screen raw tags from social multimedia websites. We mathematically model the geographic distribution of tags, extract meaningful features from the model, and build both simple and SVM-based classifiers to discover positionable tags. Furthermore, we demonstrate that the simple classifier which does not require manual input can achieve equally good performance compared to the SVM-based approach. Similarly, we model the temporal distribution of positionable tags to mine the duration when they are appropriate to be used. To better coalesce with the repository of tags indexed in the spatial-temporal domain, we extend our prior space-only visibility computation algorithm to the spatial-temporally combined domain, mine more information from social multimedia applications to compute tag similarities and popularities, and re-score tags' relevances to videos, achieving a better quality of the generated tags. This work has been published in the TOMM [129] journal.

Landmark retrieval from geo-tagged videos. We compare two state-of-the-art content-based methods with a geo-based method which we refer to as Geo Landmark Visibility Determination (*GeoLVD*) in terms of precision, recall and execution time, respectively, analyze the strength and weakness of each method, and discuss how to select the most suitable retrieval method according to video conditions and system requirements. We investigate the factors that

affect the retrieval effectiveness, measure and compare their influence through experiments, and propose methods to reduce their adverse influence when it is possible. Finally, we propose a hybrid retrieval method by integrating visual and geographic information, which has been shown to achieve significant improvements in terms of precision and recall. This work has been published in the TOMM [128] journal.

Hybrid video similarity search. This work concentrates on the following two challenges in geo-tagged video management systems: (1) how to quantify the spatial relevance of videos with the visual similarity to generate a pertinent ranking of results according to users' needs, and (2) how to design a compact video representation that supports efficient indexing for fast video retrieval. To solve the above issues, we propose a novel hybrid model for video representation which generates content-oriented geographic features that can be effectively fused with visual cues to improve the precision of video similarity search. Additionally, we utilize the information available from the geo-information sources to semantically segment an area into a set of coherent regions, based on which the geographic coverage of a video can be better encoded. This work has been published in the TMM [130] journal.

1.3 Roadmap

The rest of this dissertation is organized as follows. Chapter 2 reports the important related work to this study. Chapter 3 introduces the preliminaries of the research. Chapter 4 introduces an automatic approach to reduce the errors in the geographic metadata. Chapter 5 presents the construction of a rich

spatial-temporal tag repository for automatic geospatial video annotation. In Chapter 6, we demonstrate the visual and geographic information use in video landmark retrieval. Chapter 7 presents a novel hybrid video representation for similarity search. Chapter 8 concludes and suggests the potential future work.

CHAPTER 2

Literature Review

This chapter looks into existing research work that is highly relevant to our study. This review mainly focuses on four parts: geo-metadata correction, geo-tagged image and video annotation, landmark recognition, and video similarity search.

2.1 Geo-metadata Correction

In multimedia, a significant number of techniques benefit from the presence of geographic metadata associated with images and videos [105, 8]. However, such solutions may sometimes face performance issues due to the occurrence of GPS and compass errors. Traditionally, raw GPS trajectories are usually processed by standard smoothing techniques [20] and map matching algorithms [17]. To produce more precise geographic context, the determination of camera viewing direction has attracted much research attention in recent years. Several

content-based computer vision techniques have been proposed based on local feature extraction and matching. Luo *et al.* [80, 81] estimated the viewing directions of world’s photos by reconstructing the scenes using a normalized 8-point algorithm. Based on the assumption that the camera location extracted from the geographic metadata is correct, they further geo-registered the photos on Google Maps to assist users in exploring places of interests around the world. Park *et al.* [89] proposed to utilize both Google Street View and Google Earth satellite images to determine the camera orientation of a geotagged image. Kroepfl *et al.* [61] presented a method to geo-locate a photo and then estimate the viewing direction by registering the image onto street level panoramas. However, these methods usually require a large image database to perform reliable object matching. Their effectiveness can sometimes be influenced by the limitations of the data sources, *e.g.*, Street Views are only applicable for photos taken on or near road networks [118].

It is one of the central problems in photogrammetry to determine the relative position and orientation among a set of images. Horn [47] presented an iterative method to solve the least-squares problem with more than five correspondences. Snavely *et al.* [105] computed sparse 3D model of a scene and determined the relative camera viewpoints of photographs for interactive 3D browsing. However, these approaches have not dealt with the geo-registration of camera poses with respect to world maps. Benefiting from the developed Structure from Motion (SfM) reconstruction approaches, image-based localization using 3D models of urban scenes has been extensively studied in recent years [73, 123]. Sattler *et al.* [96] utilized 3D scenes reconstructed from Flickr images, and showed that direct 2D-to-3D matching offered considerable potential for accurate image localization. Similarly, Li *et al.* [70] estimated camera

Table 2.1: A comparison with the previous work.

Work	Geo Metadata	Visual Features	Auxiliary Images	Geo Context
SfM reconstruction [70, 96, 105]		✓	✓	
Image-based matching [61, 89]		✓	✓	
Location-constrained geo-registration [80, 81]	✓	✓		
Geocontext-aware sensor data correction, proposed	✓	✓		✓

poses with respect to a large geo-registered 3D point cloud. Aided by advanced matching techniques, system reliability and efficiency have been further improved. However, such methods might sometimes be limited by their feasibility as the 3D reconstruction step usually requires extensive image collections with large baselines and sufficient overlaps.

As illustrated in Table 2.1, we have compared our method with the related work in terms of feature sources. To the best of our knowledge, there are basically no algorithms designed for efficient compass sequence correction. The existing techniques mostly focus on camera orientation determination where good accuracies rely on the robust feature matching with extensive computational costs. Moreover, it can be easily seen that the proposed method is the first attempt to consider the geographic context derived from OpenStreetMap (OSM) for fine-grained video geo-registration.

2.2 Geo-tagged Image and Video Annotation

Automatic or semi-automatic semantic annotation has greatly facilitated image and video search online. A number of studies have proposed state-of-the-art content analysis methods to understand the semantics of multimedia content [83, 32, 93]. Alternatively, other studies proposed to leverage crowdsourced

web data, or combine it with visual features [101, 100, 122]. Social media content, such as videos and images uploaded to YouTube and Flickr, is widely exploited recently. In general, the candidate tags for an image or a video can be suggested by its nearest neighbors. Siersdorfer *et al.* [100] proposed to capture the connections between videos using their content redundancy. Ballan *et al.* [11] presented a system for video tag suggestions and temporal localization based on the collective knowledge and visual similarity of frames. Several annotation techniques based on relevance models, which are used to estimate the joint distribution of words and images, have also been proposed and have achieved encouraging performance [52, 83]. Liu *et al.* [75] argued that the performance and scalability of traditional relevance-model-based methods can be limited by the semantic gap and the dependence on training data, and further proposed a dual cross-media relevance model which estimates a joint probability from the expectation over words in a pre-defined lexicon.

Recently, researchers have investigated the relationship between tags and geo-contexts of multimedia content, and used it to suggest tags. Moxley *et al.* [84] proposed a tag suggestion method exploiting both content-based analysis and geo-referenced information. Given an image to suggest tags, their system queries a number of geographically closeby images, extracts their tags as candidates, and scores them based on their local popularity and the visual similarity between the target image and its neighbors. Abdollahian *et al.* [3] proposed a similar method, but it was aimed at video annotation instead. To conduct a visual comparison between the target video and geographically selected images, they segment the video and extract key frames to represent it. These two methods have two limitations compared to ours. First, it is computationally challenging to require a k -nearest neighbor computation for each image/video

to suggest tags. Second, without investigating the global distribution of a tag, it cannot reliably be judged whether the tag carries distinguishable semantics in some place even if it frequently appears. For example, *tourism* and *travel* may be popular in places of interest all over the world, to the point where they cannot help users to recall where the image/video was taken.

Larson *et al.* [66] presented three tasks devoted to tagging and geo-tagging at the MediaEval 2010 benchmarking initiative [65]. MediaEval brings multimedia researchers together to pool research resources and focus efforts on developing solutions for challenging issues facing multimedia indexing and retrieval. Recently, several techniques have been proposed to uncover the relationship between word concepts and geographic regions. Yanai *et al.* [125] proposed to use both image region and geo-location entropy to analyze relations between location and visual features. Intagorn and Lerman [49] proposed that the boundaries of places can be learnt from noisy social annotations. Thomee and Rae [110] uncovered the colloquial boundaries of locally characterizing regions by innovatively modeling the data using scale-space theory. In the geographic information systems literature, methods for smoothing raw data points to create continuous distributions have been proposed, with the advantage of creating summary statistics that are less sensitive to high-frequency noise in the data [15]. The basic idea is to replace the data points with continuous kernel functions, *e.g.*, Gaussian probability distributions are usually used. Sizov [104] built a framework named GeoFolk for multi-modal characterization of social media by combining text features with spatial knowledge in order to construct better algorithms for content management, retrieval, and sharing. The method captured the correlations between coordinates and tags by a mixture of latent topics, where a mixture of per-topic Gaussian distributions was adopted.

There exist two studies that are most closely related to ours. Rattenbury *et al.* [95] proposed a method for finding the tags that represent places or events. In their method the domain of study is partitioned into segments of some pre-defined scales, then the tag usage in each segment is analyzed, and the significant segments where the tag is used are identified and judged whether to indicate a place/event or not. Compared to this study, our method does not need to partition the domain, but focuses on street-level positioning and considers the global distribution. Moreover, we analyze the tag similarity to increase the semantic diversity of the generated tags. The other relevant study was proposed by Zhang *et al.* [135]. They also investigated the distribution of tags over the temporal and spatial domains, but they used the distributions as features to mine the similarity among tags. Another important difference is that our study demonstrates a novel scenario of using the correlation model of tags and locations, that is, fertilizing the vocabulary for sensor-rich video annotations.

The geo-context of multimedia objects may be used for innovative applications. For example, some studies demonstrated the usage of photos with geo-coordinates to create tourism plans [78, 34]. Others used geo-coordinates to place the content on a map to facilitate browsing and navigation of images/videos [114, 4]. Yin *et al.* studied the problem of discovering and comparing geographic topics from GPS-associated documents [132] and investigated the problem of mining and ranking trajectory patterns from the uploaded photos with geotags and timestamps [131]. Besides tag annotation and video search, such geographic mining based applications can benefit from the spatial-temporal tag repository we aim to build in this work as well.

2.3 Landmark Recognition

Geographic location tags help users to localise videos, allowing the media to be anchored to real world locations [94]. The *MediaEval Placing Task* is held annually for participants to attempt to automatically assign latitude and longitude coordinates to each of the provided test videos [55]. Nowadays, with an increasing number of devices being available that can automatically encode geotags, it has become easier and more efficient to record the geo-metadata at the time when videos are taken. A variety of methods and solutions benefit from the presence of geographically relevant metadata. Liu *et al.* [76] presented a sensor enhanced video annotation system (referred to as *SEVA*) which enables searching videos for the presence of particular objects. However this approach requires a controlled environment where a sensor is attached to every object. Simon *et al.* [103] presented an application framework that retrieves the visible objects within the user’s viewable scene in the real world. Arslan Ay *et al.* [8] proposed a viewable scene model, based on camera location and orientation, to describe the viewable region within a video. This viewable scene model was further extended for efficient tagging and searching in other work [99, 7, 58]. One challenge is the occurrence of GPS and compass errors and therefore techniques based on geographic information may sometimes face performance issues. Zhang *et al.* [134] proposed an annotation and navigation system for tourist videos based on video tracks and orientation. The method can calibrate, or even obtain, position and orientation information by registering videos to geo-referenced 3D models. It brought awareness to the importance of geographic metadata, especially for tourist videos.

In the computer vision domain, the bag-of-words method is the current

state-of-the-art approach for landmark image retrieval [28]. The most popular choice for feature extraction in the *BoW* model is the Scale-Invariant Feature Transform (*SIFT*) descriptor [77]. It has been reported that, in terms of landmark recognition, *SIFT* outperforms not only global features such as color and texture, but also other local features such as Speeded Up Robust Features (*SURF*) and Multi-Scale Oriented Patches (*MSOP*) [5, 127]. Yap *et al.* [127] also showed that *dense-SIFT* works better than *sparse-SIFT*, and an enhanced *BoW* integrated with multiresolution patches and *dense-SIFT* achieves the best performance.

As the traditional *BoW* approach discards the spatial information of local descriptors, the descriptive power of its image representation is severely limited. Subsequently, efforts have been made to encode the spatial information into image content descriptions [45, 90]. Lazebnik *et al.* [67] proposed a spatial pyramid matching technique for natural scene categorization. Advanced coding techniques have also been proposed, which better encode the original feature descriptors based on the vocabulary basis to yield significant performance improvements [126, 119]. Endeavors to enrich the *BoW* model with spatial information from other perspectives have been tried as well, such as using homography mappings that geometrically connect pairs of images [25, 35]. The idea of expanding 2D images into 3D landmark models for the task of landmark recognition has also been studied [39]. However, performance improvements are achieved by adopting more complex spatial models with a larger vocabulary, at the expense of high memory and computational costs.

In the last several years, an important trend has emerged within the multimedia and computer vision communities in an increasing emphasis on modeling and use of contextual information. Researchers began to utilize geographic

data as supplement to visual information. Several methods have recently been proposed for landmark recognition and retrieval that integrate geographic information with content analysis, but with different goals. Zheng *et al.* [136] presented a web-scale landmark recognition engine that organizes, models and recognizes landmarks on the scale of the entire planet Earth. Avrithis *et al.* [10] proposed a system that can retrieve not only landmarks but also non-landmark images in collections of community photos by constructing a 2D scene map for each view cluster and preserving details from all the reference images while discarding repeated visual features. Chen *et al.* [21] addressed the problem of city-scale landmark recognition from cell phone images. More advanced content and context integration techniques for mobile landmark recognition have been proposed to achieve better performance as well [22, 71].

Most of the recent approaches on landmark recognition and retrieval focus either on the landmark organization and modeling from large community photo collections, or on the real-time landmark recognition within an image taken from a mobile phone. Our work addresses a different aspect of landmark retrieval from geo-referenced video collections. It also differs from previous video retrieval techniques [107, 31] in that it has a more specific focus on landmark retrieval and proposes landmark recognition techniques suitable for videos, not just images. Recently, Penatti *et al.* [91] proposed a novel video representation model, called Bag-of-Scenes, which uses scenes as the basic elements to represent a video. The method has shown promising results in video geocoding, but its performance in video retrieval still remains unknown. Moreover, the dictionary of scenes is predefined, so issues may arise when retrieving relevant segments of an arbitrary landmark that differs from any of the scenes in the dictionary.

2.4 Video Similarity Search

Great efforts have been made in multimedia search research in recent years. In the following we will first report the related work on multimodal similarity measures based on text and visual features, then move on to describe methods that incorporate geographic features. Lastly we will describe the limitations of the video descriptions in the existing approaches.

Many of the previous text-based video retrieval techniques perform unsatisfactorily due to the mismatch between textual information and video content. To solve this problem, a number of fusion strategies have been developed to improve video retrieval from different modalities [42]. Campbell *et al.* presented a fully automatic retrieval system for speech, visual and semantic modalities [16]. Different types of visual features extracted from keyframes (*e.g.*, color and texture) and text features extracted from speech transcripts were empirically evaluated by experiments for concept detection and video search. To better exploit the underlying relationship between video shots, Liu *et al.* proposed a *PageRank-like* graph-based approach which simultaneously leveraged textual relevancy, semantic concept relevancy, and low-level-feature-based visual similarity in video ranking [74]. Additionally, several multimodal reranking methods have been proposed to improve the initial text search results. Hsu *et al.* proposed a context reranking method by leveraging the contextual information associated with recurrent images or videos over distributed sources [48]. A context graph was constructed where the nodes are videos and the edges are weighted by multimodal contextual similarities, then the video reranking problem was solved through a random walk on this context graph. Tian *et al.* proposed a content-based reranking technique by formulating video search

reranking as a global optimization problem within a Bayesian framework [112]. The conditional prior indicates the ranking score consistency between visually similar samples, and the likelihood reflects the disagreement between the reranked list and the initial one returned by text-based search. However, it is worth emphasizing that none of these methods utilize the geographic metadata which is one of the important kinds of contextual information.

In recent years, the geographic metadata has been widely utilized in image mining, annotation and retrieval. Kennedy and Naaman proposed a system that can generate diverse and representative sets of images for landmarks by combining context and content [56]. Crandall *et al.* investigated the problem of organizing a large collection of geotagged photos [26]. Kamahara *et al.* proposed a conjunctive ranking function using both geographic distance and image distance for image retrieval [54]. Liao *et al.* [72] studied geo-aware tag features for image classification. They built tag features by tag propagation from both visual and geo neighbors. For video, Arslan Ay *et al.* proposed to model a camera's field-of-view based on camera position, orientation, viewable angle, and the far visible distance [8]. This viewable scene model was further utilized for efficient video tagging and searching by other work [7, 134, 99, 58]. Arslan Ay *et al.* proposed to rank geo-referenced videos based on three fundamental metrics related to the search area, *i.e.*, the total overlap area, the overlap duration and the accumulation of overlap regions [7]. Zhang *et al.* proposed to calibrate camera location and orientation by registering videos to a mirror 3D world [134], but it requires interactive registration and accurate 3D terrain and building models. Without leveraging the visual features, it is difficult to detect occlusions as this world is not a static world and we do not have the geo-information of dynamic obstacles such as vehicles.

Unfortunately, little efforts have been put on fusing the visual content and the geo-context for sophisticated video similarity measure. Many of the content-based video retrieval solutions decompose videos into a set of keyframes and define the video similarity based on the pairwise keyframe distances [23, 98]. This prior work usually suffers from low recall rates as the search relies on visual duplication. To better describe video content, modern approaches utilize a set of concepts as intermediate descriptors to facilitate video search [133, 18, 69]. The concept set is usually general and frequent so as to answer as many queries as possible [121], yet this results in difficulties for the precise interpretation of queries (*e.g.*, queries for a specific building). To overcome the limitations, this work presents a hybrid video representation, based on which precise delimited search results can be obtained. It conjunctively leverages video spatial relevance and local visual similarities in video ranking, so it provides excellent support for query-by-example in geospatial video search systems. Experiments show that, based on a geo-referenced video clip or a geotagged image, our proposed system can effectively retrieve the most relevant video clips compared with existing methods.

CHAPTER 3

Preliminaries

3.1 Viewable Scene Model

Video sensors are becoming ubiquitous and the volume of captured video material is very large. Therefore, in our prior work, Arslan Ay *et al.* [8] proposed to collect and fuse multiple sensor streams such as the camera location, direction, *etc.*, to provide a comprehensive model of the viewable scene. As illustrated in Figure 3.1, the *camera viewable scene* describes the visible scene based on a camera’s field-of-view (*FOV*). The 3-dimensional $FOVScene(P, \vec{d}, \theta, \phi, R)$ model is formulated by the following parameters: (1) the camera position P , (2) the camera direction (*i.e.*, compass) vector \vec{d} , (3) the horizontal and vertical camera viewable angles θ and ϕ which describe the angular extent of the scene filmed by the camera, and (4) the far visible distance R which is the maximum distance at which a large object within the camera’s field-of-view can be rec-

ognized. The parameters θ , ϕ and R are constants that can be estimated from the optics of the camera used for video recording [43].

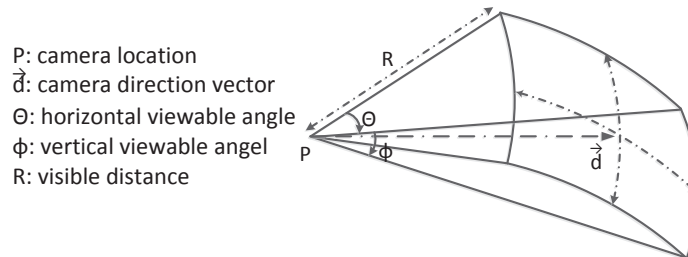


Figure 3.1: Illustration of the 3D *FOVScene* model.

We created special geospatial video recording applications for both the Android- and iOS-based smartphones. They acquire, process and record location and orientation metadata along with the video streams. To obtain the camera orientation, the apps employ the heading, pitch and roll acquired from the compass and accelerometer sensors. Camera location coordinates are acquired from the embedded GPS sensor. The collected geographic metadata is recorded in JSON format. Each metadata item in the JSON data corresponds to the viewable scene of a particular video frame. For synchronization purpose, each metadata item is associated with an accurate timestamp. The users can choose to upload their recorded geo-tagged videos to our server, where people can submit geo-queries to search and view the videos via a web interface. Note that compared to other geo-referenced video management systems, which usually assign a single geo-coordinate to a whole video [46, 111], ours provides the viewable scenes at frame-level granularity, such that it can enhance the accuracy of video processing based on geo-context.

3.2 Automatic Geo-tagged Video Annotation

Based on the *camera viewable scene* model introduced above, we leveraged the geospatial properties of videos and proposed a sensor-rich and data-driven approach to automatically generate tags for them [8, 99]. Here we briefly review the key features of this approach.

The annotation process is automated by querying proper data sources using the viewable scene descriptions [99]. The method has two major stages. In stage one, the data sources are queried for visible objects in the videos where the objects' visibility is calculated through spatial computations. Occlusion detection is performed to remove hidden objects. The system then generates descriptive textual tags based on the object information retrieved from the geo-information services, such as names, types, locations, dimensions, *etc.*. Figure 3.1 shows the 3D viewable scene model we adopted and Figure 3.2 illustrates how the visible objects are retrieved for each frame (the visible area is highlighted in blue while the occluded area is highlighted in yellow). In stage two, six relevance criteria are introduced to rank the tags based on their relevance to the videos, which are the *closeness to the FOVScene center*, the *distance to the camera location*, the *horizontally and vertically visible angle ranges*, and the *horizontally and vertically visible percentages*.

After ranking tag candidates based on their relevance, the video segments for which the tags are relevant to are determined. Unlike many other video annotation techniques, this approach can associate tags precisely with the video segments in which they appear, rather than the whole video clip. Therefore, given a query of a certain tag, we are able to only return those relevant video sections to the user.



Figure 3.2: Illustration of a sample *FOVScene* and the visible objects which are supplied by Google Earth and determined by conducting geometry computations (Copyright © 2013 Google).

3.3 Datasets

The geo-referenced video datasets we used in this dissertation were collected from the GeoVid ¹ website maintained by our group. Users can record and share videos using the GeoVid smartphone applications, or explore the world by watching videos via a web browser. Moreover, the GeoVid project also provides APIs ² for users to obtain public videos together with their corresponding geographic metadata. The location and orientation metadata is recorded along with each video stream by sampling the GPS sensor every second and the compass sensor every 200 milliseconds.

There are other two auxiliary data sources that are frequently used in this dissertation. For online mapping services, we chose the OpenStreetMap (OSM), which is a crowdsourcing project that provides editable maps of the world.

¹<http://geovid.org/>

²<http://api.geovid.org>

Useful properties, such as name, type, footprint, and height, can be easily extracted from the map data for a great number of geographic objects. In the early stage of its development, issues such as important landmarks missing or height information of buildings unavailable hindered its utilization in geo-based applications. Therefore, in our early work we also collected building heights from other sources such as EMPORIS³, or estimated based on other clues, *e.g.*, the number of storeys. But as the data contribution growth has continued to rise quickly [37], the map data has been greatly enriched. Nowadays, users can even build three dimensional city models from it easily [120].

For social sharing services, we collected Flickr images together with their associated timestamps, textual annotations, geo-coordinates, geotag accuracy levels, *etc.*. Flickr is one of the best online photo management and sharing applications. We used Flickr images as the training data for content-based or hybrid analysis, and tried to mine positionable tags including event names from Flickr in order to enrich the spatial tag repository of OpenStreetMap.

3.4 Notations

The important notations used in this dissertation are listed in Table 3.1.

Table 3.1: Notations used in this dissertation.

Symbols	Meanings
FOV model	
P	Camera location with geo-coordinates
\vec{d}	Camera direction vector
θ	Camera horizontal viewable angle
ϕ	Camera vertical viewable angle
R	Camera visible distance

³<http://www.emporis.com/>

Table 3.1-continued from previous page

Symbols	Meanings
Automatic geo-metadata correction	
S	A sequence of video frames
L	The associated location sequence
D	The associated viewing direction sequence
$[x, y, z]$	The UTM coordinates and altitude associated with a location
$[\alpha, \beta, \gamma]$	The yaw, pitch and roll associated with rotation
K	The intrinsic parameter matrix of a camera
R_i	The rotation matrix of the i-th frame
T_i	The translation vector of the i-th frame
E_{approx}	The error term of smooth approximation
$E_{rotation}$	The error term of relative rotation
$E_{direction}$	The error term of absolute viewing direction
μ_1, μ_2, μ_3	The balancing factors of the error terms
S^r	A set of virtual scenes
$Label_c(s)$	The semantic label matrix of a frame s
$Label_p(s^r)$	The projection label matrix of a scene s^r
$Dist(L_1, L_2)$	The defined distance between two label matrices L_1 and L_2
Spatial-temporal tag mining	
\mathcal{T}	Tag collection
$\mathcal{G}^{(\tau)}$	The geo-coordinates related to a tag τ
$T^{(\tau)}$	The timestamps related to a tag τ
$\vec{\gamma}, \vec{\mu}, \vec{\Sigma}$	The parameters in Gaussian mixture models for tag geographic distribution modeling
R_{cr}	The geographic confidence region
$f_1(\tau)$	The number of positioning locations in the AOI
$f_2(\tau)$	The prior sum of the positioning locations in the AOI
$minPts$	The minimum number of points required to form a cluster
NP	The noise control for DBSCAN
α	The standard deviation control for DBSCAN
$CNum, IC_{\mathcal{I}}$	Threshold parameters for estimating a tag's temporal visible intervals
$\mathcal{I}_{vis}^{(\tau)}$	The temporal visible intervals for tag τ
$S_b(\tau)$	The visual relevance score of a tag τ
$S_p(\tau)$	The popularity promotion score of a tag τ
ω	The scaling factor between $S_b(\tau)$ and $S_p(\tau)$
$S(\tau)$	The overall saliency score of a tag τ for ranking

Table 3.1-continued from previous page

Symbols	Meanings
Video landmark retrieval	
X	The image local descriptors
B	The codebook for feature coding
C	The sparse codings for X
G_{frame}	The geometry of a frame viewable scene
$G_{landmark}$	The geometry of a queried landmark
G_o	The geometry set of the relevant geographic objects
$VisibleR$	The visible angle ranges of the queried landmark
Hybrid similarity search	
r	A geographic region
f	A video frame
ol	The overlap between a $FOVScene$ and a geographic region
P^c	The centroid of the overlap
\vec{d}^c	The vector pointing from the camera location to the overlap centroid
$A(ol)$	The area of overlap ol
$A(r)$	The area of geographic region r
$\hat{A}(ol)$	The normalized area of overlap ol
$D(P^c, P)$	Euclidean distance between P^c and P
$D_\theta(\vec{d}^c, \vec{d})$	Angular distance between \vec{d}^c and \vec{d}
$K_{\sigma, \sigma_\theta}$	A 2D Gaussian kernel
σ, σ_θ	The Gaussian parameters used in generating the geo-histograms in the proposed video representation
$hist^{geo}(v)$	The proposed geo-feature for video v
$VS(r)$	Visual saliency of geographic region r
$SS(r)$	Social saliency of geographic region r
λ	The balancing factor between $VS(r)$ and $SS(r)$
$saliency(r)$	The overall saliency score of geographic region r
$w_r^{vis}(v_i, v_j)$	The local visual similarity in region r between videos v_i and v_j
$Sim(v_i, v_j)$	The proposed similarity measure between videos v_i and v_j

CHAPTER 4

Automatic Geographic Metadata Correction

4.1 Introduction

Online video content is continuing to experience rapid growth. Uploading, sharing, and viewing videos on the web have become an everyday activity in people's lives. With the ubiquity of sensor-equipped smartphones and tablets, it is increasingly common for users to take images or record videos together with the geographic properties of the camera (*e.g.*, location and viewing direction). The presence of the geospatial contextual information has opened up new opportunities in video management systems. This is especially the case with fine-grained contextual information where every video frame is tagged. A great number of applications, such as navigation systems [134], travel recom-

mendation [78], and video tagging [99], can benefit from the geo-metadata by utilizing it as an alternative or supplement to the traditional content analysis approaches. However, the use of geographic information is sometimes hampered by the presence of inaccuracies in the raw sensor data. While for GPS this issue has been extensively studied [17, 20], only a few efforts have been made on the correction of orientation data acquired from digital compasses and accelerometers [81, 89]. The difficulty level of this problem is especially high since (a) unlike GPS, a compass sensor does not provide any accuracy bounds, and (b) from our empirical observations compass errors can sometimes be very high (up to 180°). Although the Structure from Motion (SfM) technique can be applied for camera pose determination, robust estimation results usually rely on the significant overlap and the large baseline (geo-distance between camera locations) among the images to perform 3D reconstruction [70, 96, 123]. Moreover, such methods do not make full use of the geographic priors in the metadata while reconstructing the scenes and therefore result in high computational costs. In this study we argue that, with the rapid growth of spatial data available online, web images are no longer the only data source that may be utilized. Buildings and other objects within a scene can be efficiently collected from geographic information services (GIS). Thus, we propose to use the scene context obtained from GIS instead of the 3D models reconstructed from large scale images to geo-register video frames to world maps.

In recent years, spatial data have become increasingly available on the Internet. Online mapping services enable users not only to consume but also contribute geospatial information voluntarily. For instance, OpenStreetMap (OSM) is an open project that provides user-generated maps of the world. In the early stage of its development, issues such as important landmarks missing

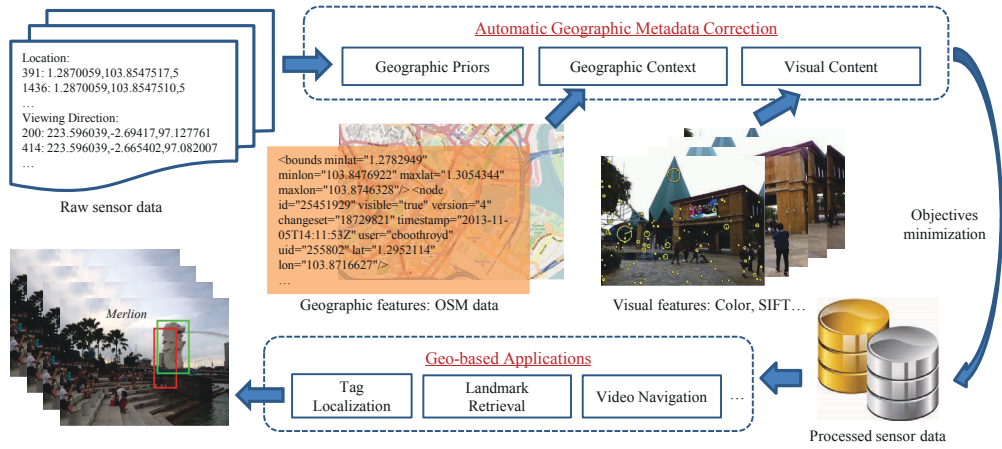


Figure 4.1: The overall architecture of the proposed automatic geo-metadata correction framework. Raw sensor data is enhanced to provide more accurate geographic information to downstream applications.

or height information of buildings unavailable hindered its utilization in geo-based applications. But as the data contribution growth has continued to rise quickly [37], the map data has been greatly enriched. Nowadays, users can even build three dimensional city models from it easily [120]. It is reasonable to assume that the quality of the spatial data will continue to improve over time. Numerous techniques and solutions can benefit from the valuable information that geo-information services provide about the world.

Figure 4.1 illustrates the overall architecture of our proposed automatic geo-metadata correction framework. In this study, we mainly focus on the camera orientation correction and formulate the task as an optimization problem by leveraging a set of complementary data sources. First of all, constrained by the geographic priors of the sensor readings, the optimized camera parameters should be near the corresponding input data. Next, we extract local visual descriptors such as SIFT to perform feature matching between consecutive frames for relative rotation estimation. According to Olsson and Enqvist [88], although

frames have short baselines that increase the ambiguity in depth determination, it does not have much impact on the rotation estimates. Finally, with the geographic context derived from OSM, we geo-register the frames to the world coordinate system by quantifying the distance between the pixel semantic labels and the 3D projection of the scene. Two distance metrics have been designed and implemented in our system, namely the pixel-based and the pyramid-based measure that encode the spatial information of the semantic labels with different granularity. By minimizing the formulated objectives, we process the raw sensor data to provide more accurate geographic information as an input for downstream applications.

4.2 System Overview

4.2.1 Design Principles

Our objective is to minimize the errors in the geo-metadata recorded by sensors. To achieve this goal, we have formulated design principles by utilizing a set of complementary data sources as follows:

Prior knowledge:

The geographic metadata recorded by GPS, compass and accelerometer.
Goal: the optimized locations and orientations should not drift too far away from the input priors.

Visual content:

The visual clues extracted from frames. Goal: the relative orientation between frames should be consistent with the rotation matrix estimated by keypoint matching.

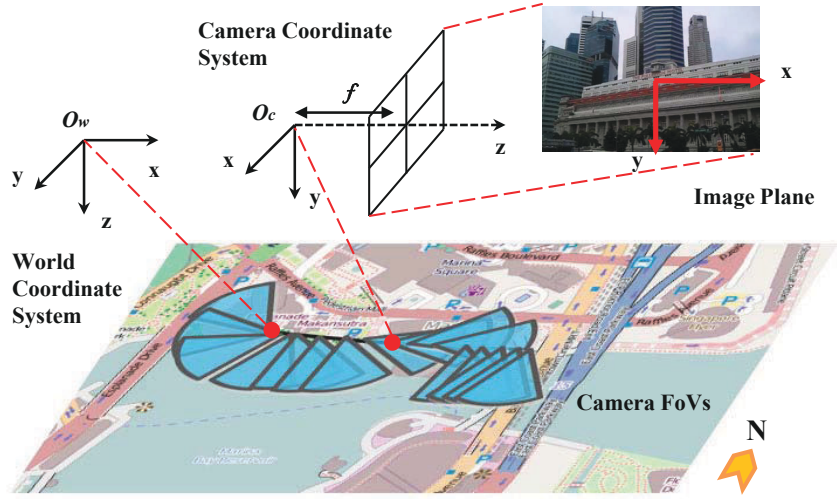


Figure 4.2: Illustrations of the coordinate systems used in our framework.

Geographic context:

The 3D scene built from OpenStreetMap. Goal: video content should be aligned with the 3D scene in respect of the corresponding external camera parameters.

To follow the above criteria, we begin by describing the problem formally.

4.2.2 Problem Description

Given a sequence of video frames, $S = \{s_1, s_2, \dots, s_n\}$, and its associated sensor readings. The video geo-metadata correction problem is formulated as finding the optimal location $L = \{l_1, l_2, \dots, l_n\}$ and viewing direction $D = \{d_1, d_2, \dots, d_n\}$ sequences that simultaneously satisfy the aforementioned design principles. Note that L and D have the same form with the input priors L^p and D^p derived from the raw sensor data, both of which are formatted as introduced below.

In our framework, we use three coordinate systems to describe the location of a point as shown in Figure 4.2. The image coordinate system is defined to

be located at the centre of the image with x and y axes pointing to right and down, respectively. The origin of the camera coordinate system is located f units before the image plane along the z axis where f is the focal length. The world coordinate system is placed at the geo-coordinates of the first input frame s_1 with x axis pointing to the east and y axis pointing to the south. Subsequently, we interpret the raw sensor readings associated with a frame s_i into the location l_i^p and the viewing direction d_i^p with respect to the world coordinate system. The camera location prior $L^p = \{l_1^p, l_2^p, \dots, l_n^p\}$ is given by $l_i^p = [x_i^p, y_i^p, z_i^p]^\top$ where x_i^p and y_i^p are the UTM coordinates converted from latitude and longitude tuples and z_i^p is related to altitude setting to 1.5 by default. The camera orientation prior $D^p = \{d_1^p, d_2^p, \dots, d_n^p\}$ is presented by $d_i^p = [\alpha_i^p, \beta_i^p, \gamma_i^p]^\top$ which are the angles of yaw (also known as heading), pitch and roll that describe the rotations of the coordinate system around z, y, and x axis, respectively. For example, a positive yaw rotates the camera to the right, the angle of which always equals to the compass reading.

4.3 Video Georegistration

We start with the introduction of the camera model that we adopt in the framework. To describe the relations between different coordinate systems, we introduce how to compute the external camera parameters based on the raw sensor data and present the formulas for coordinate transformations between different systems. With the above preliminary knowledge, we describe the formulated objectives for error minimization in the raw geo-metadata.

4.3.1 Camera Model

Without loss of generality, we assume the intrinsic parameter matrix of a camera to be $K = \text{diag}([f, f, 1])$. The focal length f is either known for calibrated cameras or can be effectively estimated by content-based approaches [40, 14]. For a 3D point p in the world coordinate system, its corresponding image projection q can be computed based on a rotation matrix R and a translation vector T using the pinhole camera model:

$$\lambda \begin{bmatrix} q \\ 1 \end{bmatrix} = K (Rp + T) \quad (4.1)$$

where λ denotes the depth factor. The rotation R and translation T can be derived from L and D , which are the location and viewing direction sequences that need to be optimized. In linear algebra, a rotation matrix is a matrix that is used to perform a rotation in Euclidean space. Using the right hand rule, the three basic rotation matrices that rotate a vector around x, y, or z axis by an angle of θ are given by

$$\begin{aligned} R_x(\theta) = & & R_y(\theta) = & & R_z(\theta) = \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} & & \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} & & \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Recall that for the i -th input frame s_i , the camera viewing direction $d_i = [\alpha_i, \beta_i, \gamma_i]^\top$ is given by yaw, pitch, and roll, which are the Tait-Bryan angles representing intrinsic rotations about $z - y' - x''$. Subsequently, the rotation of the camera coordinate system with respect to the world coordinate system can

be obtained from the above three elemental intrinsic rotations using matrix multiplication: $R_x(\gamma_i)R_y(\beta_i)R_z(\alpha_i)$. According to this change in the coordinate system (also known as passive transformation), the rotation matrix R_i is computed as

$$R_i = (R_x(\gamma_i)R_y(\beta_i)R_z(\alpha_i))^{\top} \quad (4.2)$$

Comparatively, the calculation of translation T_i is quite straightforward, which is simply $T_i = l_1 - l_i$.

4.3.2 Energy Definition

Given a video sequence associated with geographic metadata, we are interested in finding the optimal locations L and viewing directions D that minimize the following energy function:

$$E = \mu_1 E_{approx} + \mu_2 E_{rotation} + \mu_3 E_{direction} \quad (4.3)$$

where E_{approx} keeps the outputs from drifting away from the priors too much. $E_{rotation}$ and $E_{direction}$ control the errors of relative rotation and absolute viewing direction, respectively. Parameters μ_1 , μ_2 and μ_3 are balancing factors that control the weights assigned to different objectives.

Smooth Approximation

We formulate the approximation requirement as $E_{approx} = E_{approx}^{loc} + E_{approx}^{direc}$. The smoothing cubic spline algorithm [92] is adopted to process the locations

$E_{approx}^{loc} = L(t, x) + L(t, y)$ and function $L(\cdot)$ is given by

$$L(t, x) = \rho \sum_{i=1}^n \left(\frac{x_i^p - S_x(t_i)}{\sigma_i} \right)^2 + (1 - \rho) \int_{t_1}^{t_n} (S_x''(t))^2 dt \quad (4.4)$$

where t is a sequence of timestamps and $S_x(t)$ is a set of cubic polynomials to fit the observations t and x . The parameters σ_i can be used to change the weight of each point in the error term. We set it to the accuracy measure associated with GPS that indicates the degree of closeness between the GPS reading and the true location. For the approximation of camera viewing direction, we try to minimize the distance between the target D and the input prior D^p described by the sum of L^2 norms, which is

$$E_{approx}^{direc} = \sum_{i=1}^n \|d_i - d_i^p\|_2 \quad (4.5)$$

Relative Rotation

Next we discuss how to estimate the error of relative rotations, $E_{rotation}$. For a 3D point p in the world coordinate system, let q^{s_i} and $q^{s_{i+1}}$ denote its projections on two consecutive frames s_i and s_{i+1} , respectively. If the frames are sampled at a relatively high frequency (*e.g.*, 5 fps), it is reasonable for us to assume that frames s_i and s_{i+1} are taken at the same location. Therefore, according to Eq. 4.1 we have

$$\lambda_{i+1} \begin{bmatrix} q^{s_{i+1}} \\ 1 \end{bmatrix} = K R_{i+1} R_i^{-1} K^{-1} \cdot \lambda_i \begin{bmatrix} q^{s_i} \\ 1 \end{bmatrix} \quad (4.6)$$

Given a set of matched keypoints q^{s_i} and $q^{s_{i+1}}$ by feature matching, we are able to rewrite Eq. 4.6 into a set of linear equations of the form $A_i e_i = 0$,

where e_i is a vector consisting of the entries of matrix $KR_{i+1}R_i^{-1}K^{-1}$. Recall that $R_i = (R_x(\gamma_i)R_y(\beta_i)R_z(\alpha_i))^\top$, so vector e_i can be written in the form of the camera focal length f and the target viewing direction d_i . Therefore, we seek to optimize the sequence of camera orientations D by minimizing the sum of $\|A_i e_i\|_2$ over the input frames

$$E_{rotation} = \sum_{i=1}^{n-1} \|A_i e_i\|_2 \quad (4.7)$$

For the keypoint detection and matching, we use SIFT as the visual feature [77]. It provides a local descriptor for each keypoint including its location, scale and orientation. Thereafter, we match the keypoints between consecutive frames by first querying for the nearest neighbors, followed by using a minimal solver in conjunction with RANSAC to filter out possible outliers. The set of geometrically consistent matches that have been found as described above is used to construct matrices A_i in Eq. 4.7.

Absolute Viewing Direction

To quantify the error of the absolute viewing direction of a camera is less straightforward and requires additional information of the scene where the video is taken. Recently, image-based localization techniques [70, 96] have been proposed that match photos to pre-built 3D models of the world. Although promising performance gains have been reported, the construction of 3D scenes usually relies on large amounts of high quality input images. Here we argue that photos are no longer the only data source that can be utilized. Nowadays, the information about a scene can be easily collected from mapping services that are freely available (*e.g.*, OSM). Additionally, to facilitate solving

problems in computer vision, efforts have also been made on building 3D world by extending OSM [120]. Aided by the pre-built 3D world, the scene captured in an image can be well estimated based on the camera parameters derived from the geo-metadata.

Alternatively, we can also try to understand an image scene based on the content by semantic pixel labeling, *e.g.*, we adopt the SuperParsing method [113] in our experiments. As shown in Figure 4.3, it annotates every pixel with a semantic label (*e.g.*, building, water, road, and *etc.*), which provides a good outline of the semantic classes and their distributions in the image. On the other hand, as we mentioned before a scene can be labeled based on 3D projection techniques. OSM uses tags, such as building, road, and *etc.*, to indicate the category of an object. Therefore, the semantic labels can be derived from the 2D projections of the world on the image plane. We illustrate this idea in Figure 4.3 by giving two examples, namely the Marina Bay Sands hotel and the Marina Bay Reservoir. If the input of camera location and orientation is close to the ground truth, the 3D projection results should be well aligned with the semantics derived from the content. This observation provides us a simple but effective solution to estimate the absolute viewing direction of a camera.

For a frame s_i , let $Label_c(s_i)$ and $Label_p(s_i)$ denote the semantic labels derived from the image content and the world projection, respectively. Considering the orientation of a camera is a continuous variable that has the form of $d_i = [\alpha_i, \beta_i, \gamma_i]^\top$, it may not be feasible to compute the distance between $Label_c(s_i)$ and $Label_p(s_i)$ every time we change the camera parameters for optimization. Therefore, we alternatively choose to sample a set of virtual scenes $S^r = \{s_1^r, s_2^r, \dots, s_m^r\}$ with fixed camera parameters as references, based on which the absolute viewing direction error of frame s_i , denoted by $E_{direction}^{s_i}$, can be

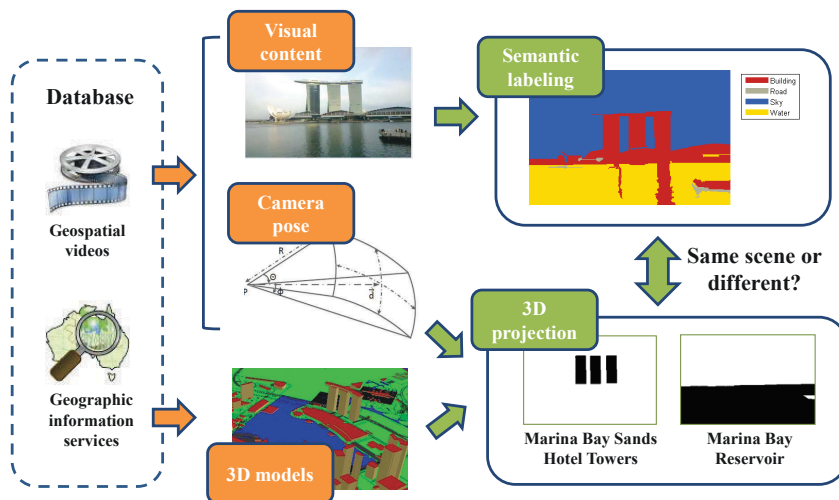


Figure 4.3: Scene understanding by semantic pixel labeling and 3D projection based on camera pose and OSM data.

estimated as a weighted sum using the following equation

$$E_{direction}^{s_i} = \sum_{j=1}^m w_{ij} \cdot Dist (Label_c(s_i), Label_p(s_j^r)) \quad (4.8)$$

where w_{ij} denotes the weight of the j -th reference scene s_j^r with respect to frame s_i . Without loss of generality, the reference scenes S^r can be selected by sampling uniformly in each of the six dimensions of camera pose. w_{ij} should be defined based on the similarity of camera parameters between s_i and s_j^r , as scenes that are taken within a small area pointing to similar directions can be considered as good representatives for each other. The details about how to decide w_{ij} will be discussed in Section 4.3.3, as it is related to the selection of S^r and the optimization strategy.

The next task for us is to compute the difference between $Label_c(s_i)$ and $Label_p(s_j^r)$. We select a list of concepts including building, water, road, sky and pedestrian to annotate pixels. Both $Label_c(s_i)$ and $Label_p(s_j^r)$ are matrices

whose entries are integer numbers that serve as the index of the pixel labels. Thus, they have the same size as the input frame s_i , denoted by $height(s_i) \times width(s_i)$. In our framework, two distance measures are analyzed. The first one is pixel-based. We count the number of pixels that are labeled with the same concept in both $Label_c(s_i)$ and $Label_p(s_j^r)$ and normalize the value as follows

$$\begin{aligned} &Dist (Label_c(s_i), Label_p(s_j^r)) \\ &= 1 - \frac{numofzeros(Label_c(s_i) - Label_p(s_j^r))}{height(s_i) \cdot width(s_i)} \end{aligned} \quad (4.9)$$

where function $numofzeros(M)$ returns the number of zero entries in matrix M . This measure estimates the pixel-wise distance between two label matrices, but the results can be sometimes susceptible to the small changes in camera pose. Inspired by the spatial pyramid matching designed for scene recognition based on local features [67], we also implement a pyramid-based distance measure by partitioning the label matrix into increasingly fine cells and computing histograms of concepts for each cell. More specifically, we construct a spatial pyramid that has a total of L_{pyr} levels. At level $l_{pyr} = 1, 2, \dots, L_{pyr}$, the label matrix is partitioned into $2^{l_{pyr}-1}$ sub-regions. For each sub-region, a histogram of concepts is generated by counting the number of times that each label appears. Let $hist^{l_{pyr}}(M)$ be the vector formed by concatenating the histograms generated on level l_{pyr} for a label matrix M . Intuitively, we would like to penalize the features of larger cells because they preserve decreasing spatial information. Therefore, we assign weights $\frac{1}{2^{L_{pyr}-l_{pyr}}}$ to histograms $hist^{l_{pyr}}(M)$ and concatenate the weighted histograms into a feature vector which is $hist(M) = [\frac{hist^1(M)^\top}{2^{L_{pyr}-1}}, \frac{hist^2(M)^\top}{2^{L_{pyr}-2}}, \dots, \frac{hist^{l_{pyr}}(M)^\top}{2^{L_{pyr}-l_{pyr}}}, \dots]^\top$. Subsequently, the distance between two label matrices can be measured based on this pyramid-

based feature as

$$\begin{aligned} & \text{Dist} (Label_c(s_i), Label_p(s_j^r)) \\ &= 1 - \frac{\text{hist}(Label_c(s_i))^\top \text{hist}(Label_p(s_j^r))}{\|\text{hist}(Label_c(s_i))\|_2 \cdot \|\text{hist}(Label_p(s_j^r))\|_2} \end{aligned} \quad (4.10)$$

We compare the performance of the above two distance measures and the analysis results are discussed later in Section 4.4. Finally, the absolute viewing direction error $E_{direction}$ in the energy function (see Eq. 4.3) is estimated by the sum, $E_{direction} = \sum_{i=1}^n E_{direction}^{s_i}$.

4.3.3 Energy Minimization

Inference in our model can be conducted by adopting an efficient two-stage optimization strategy [60]. First, we optimize the location L by minimizing the energy term E_{approx}^{loc} . Next we optimize the viewing direction D by keeping the previously estimated location L fixed.

According to Eq. 4.4, we smooth the GPS trajectories with cubic splines. As it is a traditional method, here we focus on discussing the optimization of the viewing direction D while keeping the location L fixed. In order to simplify the calculation of w_{ij} in Eq. 4.8, we sample the virtual scenes S^r at the optimized locations in L instead of a uniform sampling in the 3D space. As discussed before, w_{ij} should be formulated based on the similarity between the camera poses of input frame s_i and reference scene s_j^r . Given the above sampling strategy of S^r , only the virtual scenes that are located at l_i will be considered while computing $E_{direction}^{s_i}$. In other words, let l_j^r and d_j^r denote the location and orientation associated with scene s_j^r . The weight before normalization $\tilde{w}_{ij} = 0$ if $l_j^r \neq l_i$. Otherwise, we define the orientation difference between d_i and d_j^r ,

$Dist(d_i, d_j^r)$, to be the degrees that the unit vector along the z axis $[0, 0, 1]^T$ rotates from one camera coordinate system to the other. Thereafter, we convert distance to similarity using equation $\tilde{w}_{ij} = 180 - Dist(d_i, d_j^r)$, and normalize the weights by the softmax function,

$$w_{ij} = softmax_j(\tilde{w}_{ij}) = \frac{\exp \tilde{w}_{ij}}{\sum_j \exp \tilde{w}_{ij}} \quad (4.11)$$

The softmax function reduces the influence of reference scenes whose camera pose greatly differs from the input frame, and limits the weights to have a sum of one. After the normalization, we use the simplex search algorithm [64] to optimize the camera viewing directions D with the initial point setting to the geographic priors D^p derived from the geographic metadata.

4.4 Evaluation

4.4.1 Experimental Setup

We evaluated our proposed algorithm on the publicly available geo-referenced video dataset from the GeoVid ¹ website. To evaluate our approach, we manually annotated the ground truth of camera poses based on map services (*e.g.*, Google maps and Google Street View). We randomly selected ten sensor-rich videos taken in Singapore to carry out the experiments. The description of the dataset is illustrated in Table 4.1. The average video duration of this dataset is 28 seconds. We sampled frames every three seconds to let users perform the ground truth annotation and interpolated the camera parameters between the sampled frames for later comparisons.

¹<http://geovid.org/>

Table 4.1: Georeferenced video dataset description.

Video duration	Shortest	Longest	Average
	20 sec	62 sec	28 sec
No. of videos	10 videos with 83 ground truth labels		

The dataset might still be small mostly because of the effort needed to obtain the ground truth annotations, but its size is comparable to other camera orientation determination papers [89, 81]. Moreover, to the best of our knowledge, this work is among the early efforts that have been made on solving the problem of automatic geo-metadata correction for video sequences.

4.4.2 Geographic Metadata Correction

We processed the raw geo-metadata and present the error reduction results. The GPS accuracy of our test dataset is good, as all the accuracy measures associated with GPS (σ_i in Eq. 4.4) are less than or equal to five. Since this work focuses on the correction of the orientation data, at the current stage we simply processed locations by the traditional smoothing technique with cubic splines. Here we report the smoothing result on a more challenging dataset (the accuracy value $\max(\sigma_i) > 50$) [117] in Table 4.2. The parameter ρ in Eq. 4.4 was set to 0.6. We show the precision before and after processing at different geographic margins of error.

Table 4.2: Precision comparison of raw and processed GPS data.

Radius	10 m	20 m	30 m	40 m	50 m
Raw Data	68.2%	91.0%	92.0%	92.9%	93.4%
Processed	70.9%	92.0%	93.4%	94.5%	95.2%

As can be seen, there was an improvement on location accuracy within all error margins. On average, the smoothing splines were able to reduce the error

per frame by 27.32%. Further improvements can be obtained by applying more advanced techniques, such as mapping GPS traces to road maps [17]. Those approaches can be integrated into our framework easily as we adopt a two-stage optimization strategy by processing camera location and camera viewing direction separately in different modules.

Next, we compared the camera orientation errors and report the results in Figure 4.4. Recall that the viewing direction of a camera has the form of $d = [\alpha, \beta, \gamma]^T$. As most of the users hold the camera perpendicular to the ground while taking a video, the variations in pitch and roll are usually very small (*i.e.*, $\beta \approx 0^\circ$ and $\gamma \approx 90^\circ$). Therefore, we focus on evaluating the correction of yaw, α , and define the error to be the absolute angle difference between the measured and the true values in degrees. In other words, let α_i^t and α_i^e denote the true and the estimated camera heading for frame f_i . The error δ_i is computed as $\delta_i = \min(\|\alpha_i^e - \alpha_i^t\|, 360 - \|\alpha_i^e - \alpha_i^t\|)$. For an input video, the orientation error is computed as the average of its frames, *i.e.*, $E = \frac{1}{n} \sum_{i=1}^n \delta_i$.

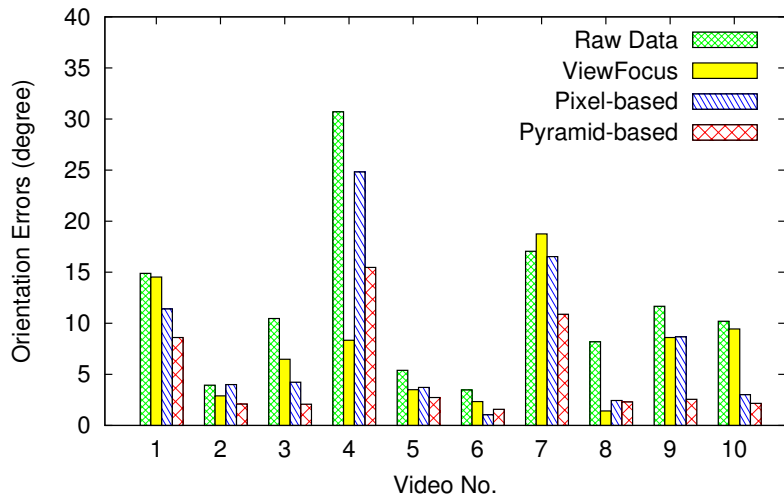


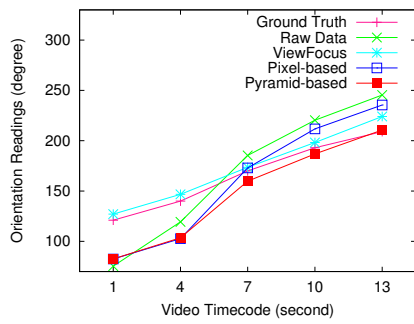
Figure 4.4: Raw and processed camera orientation error comparison for individual videos.

Based on the measurement above, we compared our proposed method with ViewFocus, which is the most related to our work that determines the camera direction with the existence of geo-metadata [80, 81]. Considering the baseline between frames is usually small, we further optimized the result of ViewFocus by conjunctively minimizing the distance to both the estimated external camera parameters and the raw geographic priors (both location and orientation). For the image-based methods discussed in the related work (see table 2.1), it is difficult to perform a fair comparison due to the lack of third-party auxiliary images. Moreover, such techniques are not always applicable, as the appearance of at least one geo-object in the content is required to perform robust feature matching and 3D reconstruction. As shown in Figure 4.4, raw data represents the geographic priors derived from the input sensor readings. Pixel-based (see Eq. 4.9) and Pyramid-based (see Eq. 4.10) indicate the distance measure we used to quantify the difference between two label matrices. The reference scenes S^r were sampled at the optimized locations of the input frames with viewing directions sampled uniformly every 10 degrees. The balancing coefficients in Eq 4.3 were set to $\mu_1 = 1$, $\mu_2 = 0.02$, and $\mu_3 = 1000$.

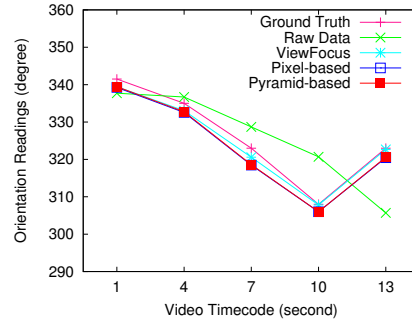
The average error reduction obtained by ViewFocus was 31.4%. Without considering the geo-context derived from OSM, it was only able to work well on certain videos (*e.g.*, video 4), while being less effective for the rest of the cases. Actually the correction effectiveness is related to the error patterns of the geo-metadata. We will discuss this in the next paragraph by showing some examples. Among the three approaches, the pyramid-based method is the most effective and outperforms its competitors in eight out of the ten cases. It obtained an average error reduction of 58.8%, where the best and the worst cases were an 80.1% and 36.2% error decrease, respectively. Compared with the

pixel-based distance measure, the pyramid-based approach achieved an average of 27.6% improvement over the former. This is mostly because the value of the pixel-based measure is susceptible to the changes in camera pose. Even a small shift in camera orientation may have a big impact on the result of the pixel-based distance measure. This might cause some issues as we sampled the reference scenes S^r with a relatively coarse granularity. It is possible to further improve the effectiveness by adopting a more fine-grained sampling approach, but this will also increase the computational complexity. Comparatively, the pyramid-based measure achieved better results as it is less sensitive to camera changes while encoding part of the spatial information of the semantic labels into the distance calculation.

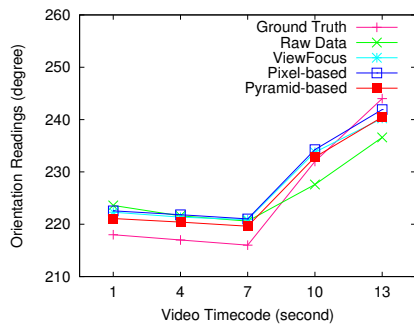
To better understand real world effects, we further examined the raw, the processed and the ground truth camera orientation sequences in our test dataset. For the eight videos where the average orientation error of the raw geo-metadata was larger than five degrees (videos 2 and 6 were excluded), we plotted the compass readings in the beginning 13 seconds of each video in Figure 4.5. The graphs were sorted ascendantly according to $E_{ViewFocus} - E_{Pyramid}$, which is the difference between the orientation errors obtained by the pyramid-based and the ViewFocus approach. In other words, we show the plots with increasing effectiveness of the former method *w.r.t.* the latter from Figure 4.5(a) to 4.5(h). As can be seen, interestingly the videos in different rows exhibited different inaccuracy patterns of the raw geo-metadata. While ViewFocus worked well on cases where the raw compass readings were distributed around the truth values and the inaccuracy mostly came from the relative rotation errors (*e.g.*, Figures 4.5(a)), it became highly difficult to handle the camera orientation shift without considering the geographic context of the world. As shown in the last



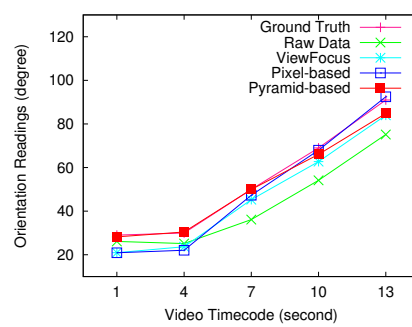
(a) Video No. 4



(b) Video No. 8

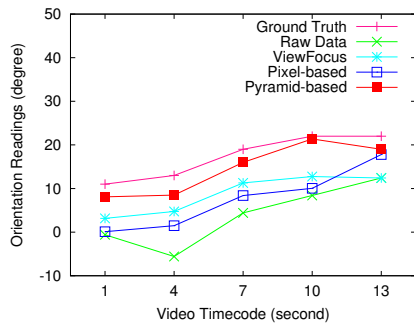


(c) Video No. 5

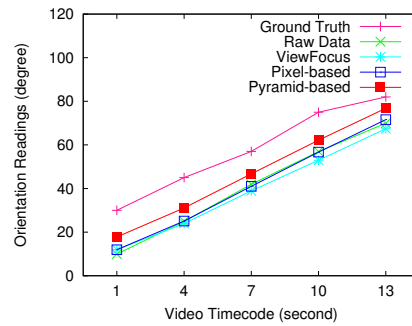


(d) Video No. 3

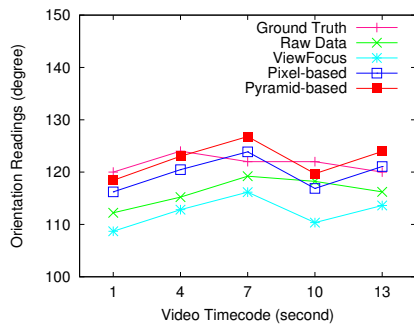
Sudden delay that causes corrupted relative rotations



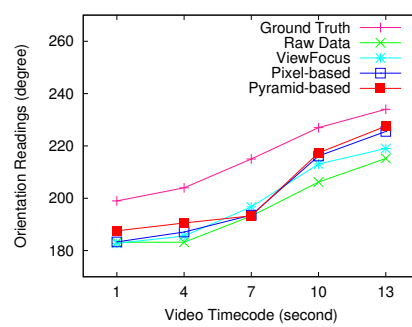
(e) Video No. 9



(f) Video No. 7



(g) Video No. 10



(h) Video No. 1

Orientation shift where the relative rotations stay approximately accurate

Figure 4.5: Effectiveness analysis of sensor data correction algorithms with or without geographic context and its connections to the error patterns in the camera orientation readings.

row, the orientation shift resulted in the incorrectness of the absolute orientation values while the relative rotations stayed approximately accurate. By applying our proposed optimization strategy, this kind of error can be effectively reduced by the third energy term, $E_{direction}$, in Eq. 4.3, which matches the image scene to the projections of the world. Moreover, the second energy term $E_{rotation}$ limits the error of the relative rotation between consecutive frames. This part is similar to ViewFocus, which is capable of correcting the corrupted compass readings caused by sudden delays or outliers.

4.5 Summary

We formulated the sensor data correction as an optimization problem. To improve the efficiency and the feasibility of the framework, we built 3D scenes based on OSM data. Next, we projected the 3D models onto the image plane and compared it to the image scene analyzed by pixel labeling. This technique provided us with an efficient way to quantify the absolute viewing direction error of a camera. By analyzing the real-world data, we draw a number of interesting observations that we summarize as follows:

(i) The geo-metadata errors can be roughly divided into two categories by checking if there are serious corruptions in terms of the relative rotation. Content-based approaches can effectively reduce rotation errors between consecutive frames, but without the context of the scene it becomes highly difficult to correct orientation shift where the relative rotations are approximately accurate.

(ii) Most of the existing image-based methods are only applicable to photos that clearly capture at least one object in order to perform robust keypoint

matching and reconstruction. Comparatively, we geo-register cameras by conjunctively considering the distribution of geo-objects and the rotation consistency in the temporal domain. Good estimation can be obtained as long as the landscape, where the video was taken, is fairly diverse towards different directions.

(iii) One factor that may have an impact on our approach is the detail level of the spatial data available from mapping services, *e.g.*, the label matrix generated by 3D projection can be imprecise due to missing buildings. Fortunately, with the rapid growing collection of map data, it is reasonable to expect that the proposed method will be able to geo-register video sequences with increasing accuracies.

Please note that this work has been conducted in the last year of my PhD, so the geo-metadata used in the following three chapters refer to the original sensor data. At the current stage, the geo-based 3D projection and the content-based semantic pixel labeling are regarded as two separate modules in our framework. As part of the future work, we are interested in developing a joint camera geo-registration and image scene understanding algorithm to further improve the results in both of the subtasks.

CHAPTER 5

Spatial-Temporal Tag Mining

5.1 Introduction

To search videos from a large corpus, annotation (or tagging) is still one of the most practical and powerful tools [6]. However, manual annotations are laborious, often ambiguous, and their uneven quality has been well documented [124, 109]. In particular, annotating a video is more challenging than annotating an image, because it consists of multiple scenes, where some are easily overlooked. Therefore, researchers have investigated solutions to automate or semi-automate the annotation process. Principally, candidate tags for an image or a video can be inferred from its nearest neighbors based on certain similarity measurements. Some prior solutions only analyzed the visual features of multimedia content, which is very challenging for open domains and usually very compute-intensive [51]. In recent years, data-driven methods

have been suggested which leverage the collective knowledge that resides in some social multimedia applications [101, 100, 122]. The annotation task can also be addressed by employing relevance models, which are used to estimate the joint distribution of words and images based on a high quality training dataset [52, 83, 32]. With the increasing availability of geo-tagged images from social sites such as Flickr, geo-aware tag suggestion tools that consider both the geographic context and multimedia content have also been proposed [3, 84]. While most of the existing work focuses on entire-video tag suggestions, several techniques have been proposed to localize tags at the shot- or even frame-level granularity [11, 12, 99].

In our prior work, we leveraged the geospatial properties of videos and proposed a sensor-rich and data-driven approach to automatically generate tags for them [8, 99]. This approach does not analyze the visual features, and therefore is particularly effective specifically for geography-oriented videos. This method first models the viewable scenes of the camera as geometric shapes by means of its accompanied sensor data, and then determines the geographic objects that are visible in the video by querying geo-information databases through the viewable scene descriptions. Subsequently textual information about the visible objects is extracted to serve as tags. However, the data-driven nature implies that the performance of the aforementioned approach significantly depends on the quality of the geo-information databases used. Previously we built our prototype using geographic information system (GIS) sources, but they can currently still be incomplete. Details are discussed later in Section 5.2.2. In order to enrich the candidate tag repository in our system, this study concentrates on how to screen raw tags from social multimedia websites, build a tag repository, and integrate it with our auto-annotation system.

5.2 Review of the Automatic Tag Generation System

Recall that a brief review of this approach has been presented in Chapter 3.2. Here we will discuss the limitations of the data source we used before.

5.2.1 System Overview

Figure 5.1(a) illustrates the framework of our previous auto-annotation approach. In our framework, the term *object* is abstract, and can be instantiated in many ways, depending on what the data source is. The only requirement is that an object must be accurately located in some place, such that its relevance to the video can be determined by our viewable scene model. As illustrated in Figure 5.1(b), this work studies the problem of how to build a rich positionable tag repository that can be directly applied in the aforementioned annotation system. The basic idea is to mine spatiotemporal tags from social multimedia applications. In the rest of this section, we will first discuss the limitations of the data source we previously used, and then introduce the proposed approaches to incorporate more varied data sources.

5.2.2 Data Source Limitations

The data-driven nature of the aforementioned approach implies that its performance significantly depends on the quality of the adopted data sources. Previously we built our prototype with the OpenStreetMap¹ used as the data source. OSM is a community based map application that can supply detailed

¹www.openstreetmap.org

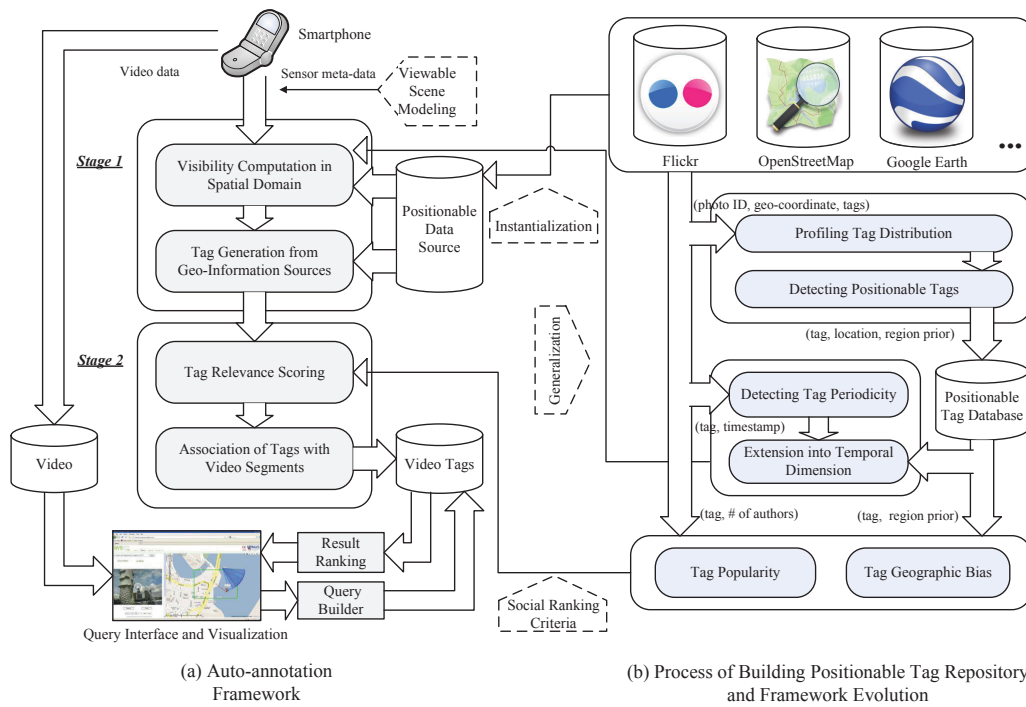


Figure 5.1: (a) The architecture of the automatic tag generation framework for sensor-rich outdoor videos, and (b) the process of building a positionable tag repository and interfacing it with the remaining framework.

information (*e.g.*, names, types, outlines) of numerous geographic objects (or landmarks). However, its completeness varies in different regions. For instance, the *Merlion* is a popular landmark in the Marina Bay area of Singapore and it was featured in our previous testing videos, but our prototype was unable to recognize it because it is missing in OSM. A more severe problem is that OSM only records landmarks in the physical world, such that the semantics of the generated tags are all within the geospatial domain. In contrast, though we require objects to be associated with some place, they do not necessarily have to be landmarks. Events may also be strongly correlated with a location. For example, the *national day parade*, which is an event, is held in the Marina Bay once a year. Summarily, video tags may miss some important semantics if a

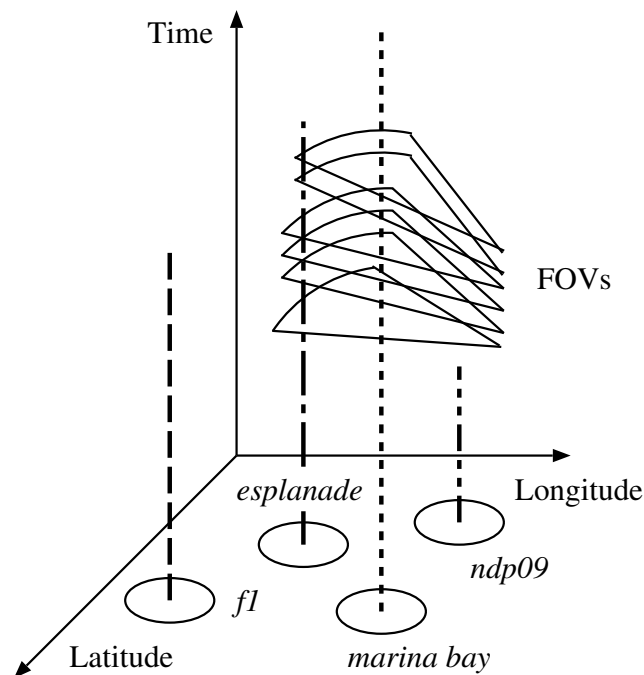


Figure 5.2: Conceptual illustration of the placement of tags in the spatial-temporal domain. The dashed lines show the durations of tag usage while the projected circles are the related places of the tags.

system only relies on the data sources of geographic objects. This motivated us to seek more diverse data sources.

5.2.3 Seeking More Varied Data Sources

We desire that the data sources provide comprehensive information and diverse semantics. However, the objects we investigated in our prior work were only physical entities such as geographic landmarks. In this study we extend the scope and objects can be landmarks, events or other concepts of interest that are positionable in a specific place. One promising source of information is the crowdsourced data available from social multimedia applications, such as Flickr, Picasa and YouTube, where the semantics of images/videos can be acquired by analysing the user-generated tags. Helpfully, the semantics extend beyond the

geospatial domain. For example, we retrieved the first 20,000 images sorted by popularity in the Marina Bay area of Singapore from Flickr and collected their associated tags. Table 5.1 lists the top 30 tags and their corresponding semantics, including place, time, event, camera parameters, *etc.* Meanwhile, these applications support multimedia positioning, that is, images/videos can be assigned a geo-coordinate (or geo-tag). Hence, with images/videos acting as the intermediary, tags and geo-coordinates are correlated. This raises the potential that we can discover some tags which are strongly correlated with a specific place. Moreover, the visibility of social tags can be sensitive to time as well (*e.g.*, event tags), which means they are not applicable to videos that recorded the same place but at different times. This raises the need for us to consider the coverage of a tag in both the spatial and temporal domains.

The data from social multimedia websites is not as organized as that from geo-information systems, and much of the data are not relevant. To solve this problem, we propose to build a spatiotemporal tag repository that can be directly applied to our auto-annotation system, by utilizing the data available from social multimedia applications. As illustrated in Figure 5.1(b), we collect the tags, the geo-location, and the timestamp associated with multimedia objects. To determine whether a tag is positionable or not, we describe its geographic distribution by a Gaussian mixture model, based on which a classifier is built. Next, we extend the repository into the temporal dimension by predicting the periodicity of each tag. Lastly, we estimate the tag popularity and geographic bias, and integrate these two criteria into the tag relevance ranking. In the next section, we will introduce the methods we adopted to build such a tag repository which is both spatially and temporally indexed (*e.g.*, see Figure 5.2) by making use of social multimedia applications.

5.3 Positioning Social Tags in Spatial-Temporal Domain

We introduce our approach to make use of social multimedia applications to build a data source of positionable tags, and determine their effective period. First we need to retrieve data from a particular social multimedia information source. In this study, we demonstrate the approach with Flickr. Nevertheless, the method can easily be extended to other similar applications such as Picasa and YouTube, assuming that the applications contain multimedia content associated with tags and geo-coordinates. The retrieved data is a collection of multimedia objects, which is formally described as $\mathcal{M} = \{m_i | i = 1, 2, \dots, k\}$. We let $tags(m)$, $geo(m)$ and $time(m)$, respectively, represent the associated tags, the geo-coordinates and the recording time of the object m .

Next, we denote the tag collection of the photos as $\mathcal{T} = \bigcup_{m \in \mathcal{M}} tags(m)$, and all the images where a tag $\tau \in \mathcal{T}$ appears as $\mathcal{M}^{(\tau)} = \{m | \tau \in tags(m), \forall m \in \mathcal{M}\}$. Consequently, all the geo-coordinates related to a tag can be expressed as $\mathcal{G}^{(\tau)} = \bigcup_{m \in \mathcal{M}^{(\tau)}} geo(m)$, and all the recording times can be similarly formulated as $T^{(\tau)} = \bigcup_{m \in \mathcal{M}^{(\tau)}} time(m)$.

5.3.1 Geographically Positioning Social Tags

Importantly, we need to formally define the concept of a *positionable tag*, which is a tag that is *strongly correlated* to some location at *street level* accuracy. There are two requirements for this. Being *strongly correlated* indicates that the tag needs to frequently occur in some places but not elsewhere, while reaching *street level* resolution makes sure that the accuracy level of the location of the

Table 5.1: 30 most popular Flickr tags in the Marina Bay area of Singapore and their corresponding semantics.

1 – 15		16 – 30	
tag	semantics	tag	semantics
singapore	place [†]	film	other
f1	event	ndp09	event
marina bay	place	ndpeeps	event
night	time	bw	other
asia	place [†]	2008	time
canon	camera	skyline	other
esplanade	place	formula 1	event
city	place [‡]	kodak	camera
marina bay sands	place	analogue	other
marina	place [‡]	travel	other
geotagged	other	analog	other
bay	place [‡]	black	other
nikon	camera	architecture	other
street	place [‡]	2009	time
2010	time	river	place [‡]

tag matches that of our viewable scene model, which is on the order of hundreds of meters. However, not all the tags can meet these two requirements. In Table 5.1, the place tags with a “[†]” mark are so general that the distributions of their geo-coordinates tend to be relatively uniform. On the other hand, the place tags with a “[‡]” mark are sure to occur more frequently in some places, but the granularity of the places is too coarse to be comparable with our viewable scene model. Note that not just place tags can be positionable. For example, the street course of *f1*, which means the Formula One automobile race, is well defined. Therefore, the first challenge is to determine whether a tag is positionable, and if it is, where the tag is positioned.

To solve this problem, we build a model to describe the distribution of the geo-coordinates of a tag, and leverage the expectation maximization algorithm [29] to estimate its parameters. This step is considered as a dimension

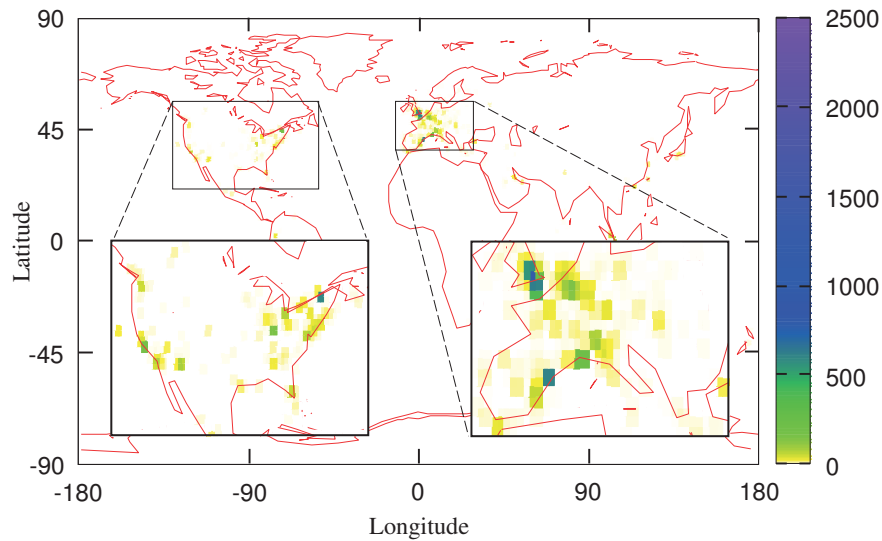


Figure 5.3: Illustration of the global distribution of the geo-coordinates of tag *f1*.

reduction to some extent. Next, we extract two features from the distribution model and use them to build a classifier to determine whether the tag can be positioned into our area of interest (AOI). Note that since a tag can be positioned anywhere, it is not easy to build a world-wide ground truth to evaluate the performance of our method. Moreover, in many cases, applications may be only interested in some specific places. Hence we properly adapt the original challenge to detect a positionable tag in our pre-defined AOI. In the remainder of this section, we explain the method in detail.

Profiling Tag Distribution

The geo-coordinates of a tag are likely to be unevenly distributed. Figure 5.3 shows an example of the tag *f1*, where we can observe a number of hot spots (the points in color), indicating the frequent usage of this tag in these regions. To identify where the hot spots are, we construct a high-level mathematical model

to describe the distribution of geo-coordinates. The basic idea is to replace the geo-coordinates with continuous kernel functions to create summary statistics that are less sensitive to high-frequency noise in the data. Intuitively, for a certain tag τ , each hot spot can be modeled with a bivariate normal distribution $\mathcal{N}(\mu, \Sigma)$, where the mean $\mu = \mathbb{E}[\vec{g}] = (\mathbb{E}[lon], \mathbb{E}[lat])^\top$ and the covariance matrix $\Sigma = \mathbb{E}[(\vec{g} - \mathbb{E}[\vec{g}])(\vec{g} - \mathbb{E}[\vec{g}])^\top]$ (superscript (τ) is omitted for simplicity). Note that a hot spot is not necessary to be as pronounced as shown in Figure 5.3. Assume there are n such normal distributions, and each single geo-coordinate \vec{g} follows either one with the probability γ , where $\sum_{i=1}^n \gamma_i = 1$. Hence we can model the distribution of all the geo-coordinates as the weighted composite of the n normal distributions, that is,

$$\mathcal{P}_g(\vec{g}|\vec{\gamma}, \vec{\mu}, \vec{\Sigma}) = \sum_{i=1}^n \gamma_i \mathcal{N}_i(\vec{g}|\mu_i, \Sigma_i). \quad (5.1)$$

However, $\vec{\gamma}$, $\vec{\mu}$ and $\vec{\Sigma}$ in Equation (5.1) are actually unknown variables. We need to estimate them from the set of geo-coordinates \mathcal{G} that we obtained. From the probability function, we can derive the likelihood function as

$$\mathcal{L}(\vec{\gamma}, \vec{\mu}, \vec{\Sigma}|\mathcal{G}) = \mathcal{P}_g(\mathcal{G}|\vec{\gamma}, \vec{\mu}, \vec{\Sigma}) = \prod_{i=1}^{|\mathcal{G}|} \mathcal{P}_g(\vec{g}_i|\vec{\gamma}, \vec{\mu}, \vec{\Sigma}) \quad (5.2)$$

or the more convenient log-likelihood function as

$$\hat{l}(\vec{\gamma}, \vec{\mu}, \vec{\Sigma}|\mathcal{G}) = \frac{1}{|\mathcal{G}|} \sum_{i=1}^{|\mathcal{G}|} \ln \mathcal{P}_g(\vec{g}_i|\vec{\gamma}, \vec{\mu}, \vec{\Sigma}). \quad (5.3)$$

Consequently, our target can be formalized as

$$\arg \max_{\vec{\gamma}, \vec{\mu}, \vec{\Sigma}} \hat{l}(\vec{\gamma}, \vec{\mu}, \vec{\Sigma} | \mathcal{G}). \quad (5.4)$$

To find the parameters of our geo-coordinate distribution model that maximize the likelihood function, we make use of the well established expectation maximization (EM) algorithm [29]. The EM algorithm is an iterative method: it alternately performs an expectation (E) step, where the expectation of the log-likelihood is evaluated with the current estimations of the parameters, and a maximization (M) step, where parameters is computed to maximize the expected log-likelihood found on the previous E step. One of the issue that has not been clarified is the number of confederate normal distribution n , which needs to be specified during the execution of the EM algorithm. Therefore, we additionally recruit a v -fold cross-validation algorithm [1] to automatically determine how many normal distributions are required to model the distribution of geo-coordinates. The general idea is to divide the observed data (or \mathcal{G} here) into v folds. The EM algorithm is respectively applied to the v folds of the training data. The log-likelihood values for all the v folds are averaged into a single metric to measure the stability of our model. At the beginning, the number of normal distributions is set to 1. If the average log-likelihood has been increased, we will correspondingly increment the number of normal distributions by 1 and invoke a new round of cross-validation.

Building a Positionable Tag Classifier

With the distribution characteristics highlighted by the aforementioned model, it is possible to determine whether the tag is positionable in our pre-defined area

of interest (AOI). Intuitively, a tag is considered positioned at the place where a hot spot emerges, and the mean vector $\vec{\mu}$ is consequently regarded as the set of candidate positioning locations. However, not all the hot spots qualify. As is mentioned earlier, the accuracy of the tag position should reach street level, thus the area of the bell-shape of a normal distribution (or the confidence region R_{cr}) should be small enough, such that each mean μ decisively approximates a specific location of the tag. The area can be estimated through the covariance matrix Σ , that is, $R_{cr} = var(lon) + 2cov(lon, lat) + var(lat)$. Hence we define the positioning locations of a tag, denoted by $\vec{\mu}'$, as the ones that are subject to $R_{cr} \leq \pi r_0^2$, where r_0 is the threshold of the street-level granularity.

However due to data noise and incompleteness, we found that having one or more positioning locations can not ensure that a tag is positionable. To address this problem, we build a binary classifier \mathcal{C} , which takes the information of a tag's positioning locations as input and outputs 1 if it considers the tag to be positionable in the AOI, and 0 otherwise. We employ two features to build the classifier. The first feature $f_1(\tau)$ is the number of positioning locations in the AOI. By definition, $f_1(\tau) = \|\vec{\mu}' \cap \text{AOI}\|$. The second feature $f_2(\tau)$ is the sum of the priors of the positioning locations in the AOI. The prior p is estimated by the Gaussian mixture model. By definition as well, $f_2(\tau) = \sum_{\mu_i \in \vec{\mu}' \cap \text{AOI}} p_i$. We observe that some tags have a hot spot in the AOI but are not widely considered as strongly correlated to the AOI. The reason is that these tags happened to be frequently used by a small number of users in the AOI, such that placing the tags there may not make sense to a majority of users. According to the distribution model, we expect this phenomenon to produce some hot spots with relatively low priors in the AOI. Therefore, we involve a filter to eliminate this

hazard. Finally, we can obtain a classifier that is formalized as

$$\mathcal{C}_{\vec{\mu}', \vec{p}}(\tau) = \begin{cases} 1 & \text{if } f_1(\tau) \geq c_0 \wedge f_2(\tau) \geq p_0 \\ 0 & \text{else} \end{cases} \quad (5.5)$$

where c_0 and p_0 are pre-defined thresholds.

One drawback of the above methodology is the need to heuristically assigned thresholds to both features in Equation (5.5). To overcome this problem, we can leverage a supervised learning algorithm such as SVM [27]. First, we select a small set of tags and ask experts to determine whether they are positionable in the AOI. Furthermore, the values of $f_1(\tau)$ and $f_2(\tau)$ of this tag set are computed. Then, we leverage the SVM algorithm to train the classifier $\mathcal{C}_{svm}(\tau)$. One pre-requisite of this method is the availability of an annotated training set. For one or a few AOIs, the manual effort is probably manageable, however, for hundreds of AOIs or more, it is too laborious. As a result, if $\mathcal{C}_{\vec{\mu}', \vec{p}}(\tau)$ is not obviously inferior to $\mathcal{C}_{svm}(\tau)$, we prefer the former. This comparison will be further discussed in the evaluation section.

Applying the classification, we now retain a set of tags that are considered as being positionable in the AOI, and denoted as $\mathcal{T}_p = \mathcal{C}(T)$. For each retained tag, we store a tuple $\langle \text{tag, spike center(s), area(s) of the confidence regions, location prior(s)} \rangle$ into a database. It is noteworthy that, (1) a tag may have multiple positioning locations in the AOI according to our classification algorithm, and (2) the database issues are out of the scope of this study, *e.g.*, how to properly index the tuples to accelerate range query processing.

Tag Expansion based on Geo-spatial Feature Similarity

As mentioned at the beginning of Section 5.3.1, there exist tags that are related to places but are difficult to detect because of their uniform distribution or their coarse granularity (*e.g.*, *bay* and *garden*). Fortunately, the meaning of a tag is usually delimited by its geo-location. For example, the tag *garden* is most likely referring to the Gardens by the Bay if we know that it was published near Marina Bay, and thus the location distribution of the tags *garden* and *gardens by the bay* should be highly similar in the AOI of Marina Bay. Based on this observation, we can find the tags that implicitly refer to a specific place by comparing their geo-spatial distributions in the AOI with the ones of the positionable tags detected by our classifier. Those tags are considered to be geographically positionable as well, and our tag collections are thus further enriched.

Zhang *et al.* [135] proposed to compute tag geo-spatial similarity by aggregating tags into geo-spatial buckets. Here, since we have modeled the distribution of a tag by a mixture of Gaussians, we adopt the *Jensen-Shannon divergence* (JSD) which is a popular method of measuring the similarity between two probability distributions. It is a symmetrized and smoothed version of the *Kullback-Leibler divergence* (KLD), and is defined as:

$$D_{JS}(\mathcal{P} \parallel \mathcal{Q}) = \frac{1}{2}D_{KL}(\mathcal{P} \parallel \mathcal{M}) + \frac{1}{2}D_{KL}(\mathcal{Q} \parallel \mathcal{M}) \quad (5.6)$$

where \mathcal{P} and \mathcal{Q} are two distributions and $\mathcal{M} = \frac{1}{2}(\mathcal{P} + \mathcal{Q})$. For distributions \mathcal{P} and \mathcal{Q} of a continuous random variable, the KLD is defined to be the integral:

$$D_{KL}(\mathcal{P} \parallel \mathcal{Q}) = \int_{-\infty}^{\infty} \ln \left(\frac{p(x)}{q(x)} \right) p(x) dx \quad (5.7)$$

where $p(x)$ and $q(x)$ denote the densities of \mathcal{P} and \mathcal{Q} . Unfortunately, the KLD between two Gaussian mixture models is not analytically tractable. Here we estimate the KLD between two Gaussian mixture models by the *Monte-Carlo* algorithm [44]. In our system, we utilized the Java library *jMEF*² that can create and manage mixtures of exponential families.

5.3.2 Temporally Positioning Social Tags

A positionable tag may still not be relevant to some video, even if it is in the coverage area of the video, because its semantics are not valid for the time when the video was captured. It is noteworthy that the semantics of such a tag probably refer to an event. For instance, the tag *ndp09* indicates the National Day Parade held in the area of the Marina Bay on 9 August 2009. While the tag *ndp09* is non-repeatable, the usage of tag *f1* spikes once a year, each time when the Formula One Grand Prix is held in Singapore (*e.g.*, see Figure 5.4). Therefore, we must estimate the coverage of a tag not only in the spatial but also in the temporal domain.

Currently we only consider the recording times of the photos that are located in the AOI and denote the time set with $T_p^{(\tau)} \subseteq T^{(\tau)}$. Though the data model in the temporal domain is similar to that in the spatial domain, we prefer to use DBSCAN [30] instead of EM because the density is known beforehand. A repeatable event is expected to occur at a similar hour of different days, or at a similar date/month of different months/years. Therefore, it is very effective to use DBSCAN, which is a density-based clustering algorithm, to discover the time intervals $\mathcal{I}^{(\tau)} = \{i = [t_{begin}, t_{end}]\}$ during which the tag τ is visible in the AOI.

²<http://vincentfgarcia.github.io/jMEF/>

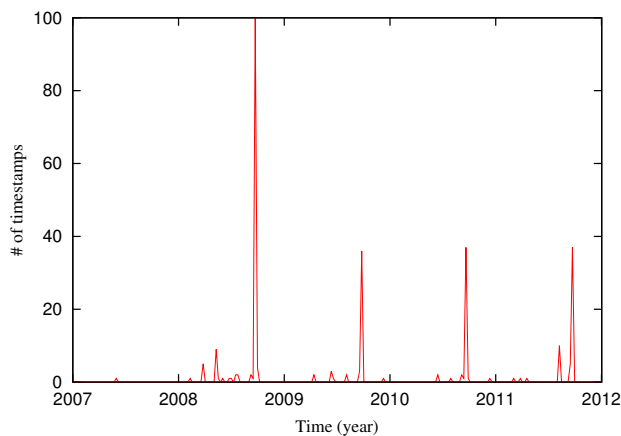


Figure 5.4: Illustration of the temporal distribution of the timestamps of tag $f1$.

Algorithm 1 sketches the overall procedure to determine a tag’s temporal visible intervals. Specifically, we set the level of density reachability ϵ to hour, day and month, respectively, and limit the minimal number of timestamps required to form a cluster to filter small hazard intervals. Next we execute DBSCAN to generate the cluster centers and the standard deviations based on which we further compute the time intervals at different granularity $\mathcal{I}_h^{(\tau)}$, $\mathcal{I}_d^{(\tau)}$, and $\mathcal{I}_m^{(\tau)}$. Subsequently, we analyze the statistics of each tag from the fine-grained to the coarse-grained level to see if a tag’s visibility is sensitive to time. We first skip the situations where the timestamps are not well clustered, *i.e.*, where the percent of the points that are marked as noise is greater than a threshold NP or where the average standard deviation is α times larger than the density parameter ϵ . Then, we review the number of clusters generated. If there is only a single time interval (*i.e.*, $\|\mathcal{I}^{(\tau)}\| = 1$), we consider that the tag is representing a single event that is only visible during this time. Otherwise, if the number of clusters generated is greater than a threshold $CNum$, we fit $\mathcal{I}^{(\tau)}$ into an arithmetic progression $\mathcal{I}(n)$. If the fitting achieves a pre-defined

```

Input:
1   The collection of social tags,  $\mathcal{T}$ ;
2   The density-based neighbor's reachability parameters,  $\epsilon_1 := hour$ ,
     $\epsilon_2 := day$  and  $\epsilon_3 := month$ ;
3   The minimum number of points required to form a cluster,  $minPts$ ;
4   The threshold parameters,  $NP$ ,  $\alpha$ ,  $CNum$  and  $IC_{\mathcal{I}}$ ;
Output:
5   The estimated temporal visible intervals for each tag  $\tau$ ,  $\{\mathcal{I}_{vis}^{(\tau)}\}$ ;
6 for each  $\tau \in \mathcal{T}$  do
7   for  $i := 1$  to 3 do
8        $Center, Stddev, noisePerc := DBSCAN(T_p^{(\tau)}, \epsilon_i, MinPts)$ ;
9       if  $noisePerc > NP$  or  $average(Stddev) > \alpha\epsilon_i$  then continue;
10      for  $j := 1$  to  $\|Center\|$  do
11           $t_{begin}^j = center_j - stddev_j$ ;
12           $t_{end}^j = center_j + stddev_j$ ;
13      end
14       $\mathcal{I}_i^{(\tau)} = \{[t_{begin}^j, t_{end}^j] | j = 1, 2, \dots, \|Center\|\}$ 
15      if  $\|\mathcal{I}_i^{(\tau)}\| = 1$  then
16          /* detect events that happened only once */
17          mark tag  $\tau$  as a single event
18           $\mathcal{I}_{vis}^{(\tau)} := \mathcal{I}_i^{(\tau)}$ ;
19      else if  $\|\mathcal{I}_i^{(\tau)}\| \geq CNum$  then
20          /* detect periodic events */
21           $\mathcal{I}(n), prob := arithProgressionFitting(\mathcal{I}_i^{(\tau)})$ ;
22          if  $prob \geq IC_{\mathcal{I}}$  then
23              mark tag  $\tau$  as a periodic event;
24               $\mathcal{I}_{vis}^{(\tau)} := \mathcal{I}(n)$ ;
25          end
26      end
27      if  $\tau$  has been marked as an event then break;
28  end
29  if  $\tau$  is not marked as any event then
30       $\mathcal{I}_{vis}^{(\tau)} := any\ time$ ;
31  end
32  return  $\{\mathcal{I}_{vis}^{(\tau)}\}$ ;
    
```

ALGORITHM 1: Social tags' temporal visible intervals estimation.

confidence interval $CI_{\mathcal{I}}$, we determine the tag to represent a periodic event that is visible during $\mathcal{I}(0) + k(\mathcal{I}(n) - \mathcal{I}(n - 1)), k \in \mathbb{N}$. If a tag is not marked as an event at any granularity, it is considered to be visible at any time.

5.4 Extension of the Auto-annotation Approach

Our auto-annotation approach can freely incorporate the positionable tag repository. We can compute whether the tags are covered by the viewable scenes of a certain video as used to do it for landmarks. However, we need to extend the visibility computation by adding one more dimension (*i.e.*, time). We compare the timestamp of our *FOVScene* with the temporal visible intervals of the tags. Since determining the visibility of a tag in the time domain is not very computationally complex, we invoke it before performing spatial domain testing, where sophisticated geometry computations are more intense. We make use of the principle location of a tag and assume that its outline is a circle that is congruent with the confidence region. Afterwards, we search for and score any qualified tags for the videos. Finally, an ordered list of tags for each sensor-rich video is obtained. Note that some refinement of the auto-annotation approach may lead to a better use of a new data source. Since the tags are obtained from social multimedia applications, crowdsourced data can be leveraged as metrics for tags. These metrics, such as tag popularity and geographic bias, can serve as the criteria to re-score the tags. The popularity of a tag can be estimated by the number of authors who use it. In practice, we select all the multimedia objects in our retrieved data set that are annotated by a specific tag, count the number of unique author IDs, and store them in the database. The priors of the positioning locations of a tag, which is computed when building the Gaussian

mixture model, can indicate the tag geographic bias. Next, we present how to use these two measures to re-score the tag relevance.

Our auto-annotation system first scores the candidate objects based on their visual relevance to a video. We refer to it as the *baseline score*, $S_b(\tau)$. However, some inherent characteristics of tags are likely to be missing. For instance, the *Esplanade* is a famous landmark in the Marina Bay area of Singapore and one would expect that it attracts more video captures than other, less known structures. However, our experimental system did initially not promote the rank of this tag. Fortunately, social multimedia applications can help to judge the importance of tags. Hence, starting from the baseline score, we propose a promotion score $S_p(\tau)$ to give more credit to important tags.

Recall that the visual relevance of a tag is computed based on the following six criteria: the *closeness to the FOVScene center*, the *distance to the camera location*, the *horizontally and vertically visible angle ranges*, and the *horizontally and vertically visible percentages*. Since a tag can have multiple positioning locations in the spatial-temporal repository we built, we compute the visual relevance score for each of the positioning locations based on the above six criteria. The *baseline score* for a tag is subsequently modified to $S_b(\tau) = \sum_i p_i S_b^i(\tau)$, where $S_b^i(\tau)$ represents the visual relevance of the i -th positioning location in the AOI and p_i is the corresponding location prior. Next, we compute the *promotion score* based on the tag popularity which is set to be proportional to the number of authors. Here we prefer widely used tags because they agree with the majority of users' perception and people may be more inclined to use them to search for images/videos as well. Lastly, we linearly

combine these two scores for tag relevance ranking:

$$S(\tau) = S_b(\tau) + \omega S_p(\tau) \quad (5.8)$$

where ω scales the promotion score against the baseline score. As a result, the distinguishable and important tags are promoted, leading to a more appropriate tag ranking mechanism.

5.5 Evaluation

We choose Flickr to evaluate the performance of the approach for building a positionable tag repository. The following five AOIs were selected: the Marina Bay in Singapore, the James R. Thompson Center and the Grant Park in Chicago, the Humble Administrator's Garden in China and the Todaiji Temple in Japan. Each of the AOIs was defined as a region of a circle with a radius of 1 km. We compiled the data set from Flickr with the following steps. First, we used the range search API to retrieve the first 20,000 photos taken from 2007 to 2011 in each of the selected AOIs, and ranked according to their popularity. Then, we extracted all the tags used by these seed photos. Thereafter, for each tag, we retrieved at most 20,000 popular photos using it (some tags may not be used by that many photos), and recorded the photo ID, the author ID, the geo-coordinates, its accuracy, the recording time and the co-occurrent tags, which make up the data set. Considering the data noise, we detected and merged duplicate tags by calculating the Levenshtein distance between tags. In the remaining of this section, we demonstrate the accuracy of our positionable tag classification, the accuracy of tag positioning, and the quality of the generated

tags by our auto-annotation prototype.

5.5.1 Accuracy of Positionable Tag Classification

To evaluate the performance of our classifier, we selected the 2,500 most frequently used tags (500 per AOI) and invited users to judge whether the tags are associated with a specific place. The tag distributions are modeled as a mixture of Gaussians using Weka [38]. Based on this manually annotated ground truth, we first trained and evaluated the performance of the SVM-based classifier \mathcal{C}_{svm} . In our implementation, we used LIBSVM [19] to train the classifier, using the number of positioning locations in the AOI (*i.e.*, $f_1(\tau)$), the sum of the priors of the positioning locations in the AOI (*i.e.*, $f_2(\tau)$) and both features, respectively. Only the tags' positioned locations whose confidence region was no larger than the AOI (*i.e.*, $r_0 \leq 1\text{km}$) were considered. The tags with the ground truth were randomly divided into two partitions, that is, a training set and validation set at a rate of 4:1. We ran 40 rounds of classifier training and validation, and in each round, we randomly re-selected the training tags to minimize the bias resulting from the training data selection. We use precision and recall as the metrics to evaluate the effectiveness of the classifier. We also report the F1 score, $F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$, as it considers both precision and recall.

Table 5.2 illustrates the performance of the SVM-based classifier \mathcal{C}_{svm} . In general, either the number of positioning locations in the AOI or the sum of the location priors in the AOI is an effective feature, which achieves impressive precision and recall. Using the two features together achieves the best performance in terms of the F1 score. Additionally, we observe that the standard

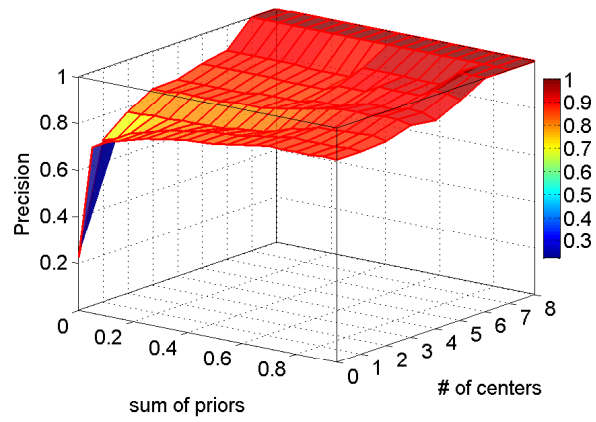
Table 5.2: Precision, recall and F1 score statistics using the SVM classifier.

	$f_1(\tau)$	$f_2(\tau)$	$f_1(\tau) + f_2(\tau)$
Precision mean	0.735	0.862	0.845
Precision std.	0.032	0.028	0.033
Recall mean	0.826	0.709	0.724
Recall std.	0.028	0.034	0.027
F1 Score mean	0.777	0.778	0.780

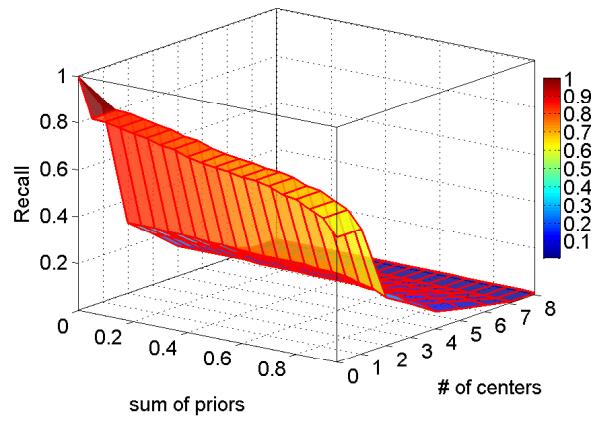
deviations of precision and recall are small, indicating that the performance of the classifiers trained by different data sets is rather stable.

Next, we evaluate the classifier $\mathcal{C}_{\bar{\mu}', \bar{p}}$ based on heuristics. The thresholds are $f_1(\tau) \geq 1$ and $f_2(\tau) \geq 0.6$, with which we obtain a classifier that achieves 0.846 precision and 0.707 recall (see Table 5.3). This indicates that the performance of $\mathcal{C}_{\bar{\mu}', \bar{p}}$ is as good as that of \mathcal{C}_{svm} , considering the precision-recall metric. Furthermore, we are interested whether the threshold choice based on our intuition is optimal. Figures 5.5(a)–(c) describe the performance with respect to the precision-recall metrics over different combinations of the thresholds of $f_1(\tau)$ and $f_2(\tau)$. Clearly, with an increase of the thresholds, tags are less likely to be considered positionable, such that the precision increases while the recall declines. Considering both the precision and the recall, we observe that the sweet spot is zero or one centers for $f_1(\tau)$ and a not too large percentage for $f_2(\tau)$, where our threshold choices lie. In summary, we can achieve good results with the simple classifier, and need not rely on the SVM-based one that requires manual input.

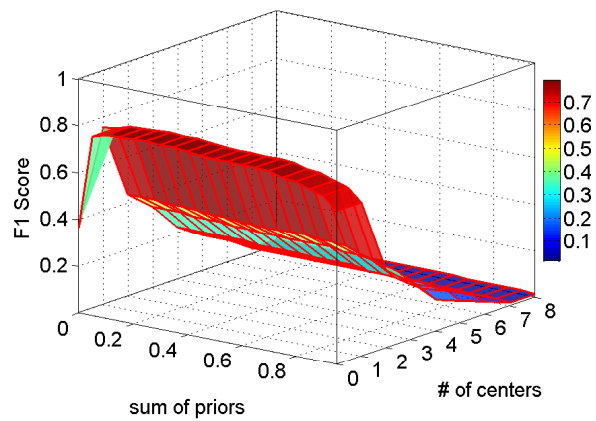
Additionally, we study the impact of the performance of the geo-coordinates on the classification. In practice, the geo-coordinates associated with a photo in Flickr may originate from human annotation, or positioning via GPS, cellular base stations or Wi-Fi access points, *etc.* Different positioning methods



(a)



(b)



(c)

Figure 5.5: (a) precision, (b) recall and (c) F1 score under different combinations of the number of centers and the sum of priors thresholds.

Table 5.3: Precision, recall and F1 score statistics using the proposed classifier by thresholding.

	<i>accuracy level</i> ≥ 1	<i>accuracy level</i> ≥ 14
Precision	0.846	0.866
Recall	0.707	0.772
F1 Score	0.770	0.816

have varying accuracy levels. However, we restrictively require each tag to be positionable at some place at street-level accuracy. Therefore, we would expect the accuracy of our classification to be encumbered by inaccurate geo-coordinates. Our generic classification approach is blind to the accuracy level of geo-coordinates, because the information cannot be assumed to be universally available. Fortunately, Flickr quantifies the accuracy level (from world ~ 1 to street ~ 16) and supplies it to API users. Hence, for a subsequent experiment we filtered out the geo-coordinates whose accuracy level is below 14 to form the input of our algorithm, and reported the statistics in Table 5.3. By doing so, the classifier achieved 0.866 precision and 0.772 recall with the same threshold settings.

In general, as geotags are collected from crowdsourced media, it is reasonable to assume that the accuracy level of their majority is relatively high. Moreover, the good classification results shown in Figure 5.5 indicate that our method is capable of filtering out inaccurate data to a certain extent and reflecting the properties of the majority. As pointed out by Hauff [41], the positional accuracy of the geotag information of Flickr images is highly dependent on the popularity of a landmark. The average distance to the ground truth location is between 11 to 13 meters for images taken at popular landmarks, which is small compared to the size of the viewable scene model we consider.

Next we evaluate the estimation of tags' temporal visibility intervals. We

manually annotated tags based on whether they are temporally sensitive or not, and evaluated the effectiveness of Algorithm 1 as a two-class classifier. The dataset was divided into two subsets of equal size, working as the training set and test set, respectively. Now let us recall the input parameters required by the algorithm. $minPts$ denotes the minimum number of points to form a cluster. NP and α are thresholds to skip the situations where the timestamps are not well clustered. $CNum$ and $IC_{\mathcal{I}}$ are parameters for periodic events detection. We set $minPts = 10$, $CNum = 4$, and $IC_{\mathcal{I}} = 0.9$ heuristically, and then tuned NP and α through experiments with the training set. Table 5.4 lists the F1 scores based on different combinations of NP and α values. As shown, the F1 score reaches its maximum when $NP = 10\%$ and $\alpha = 3$, and then decreases on all sides. Therefore, we selected this point as the optimal setting and achieved 0.863 precision and 0.704 recall on the test set. Table 5.5 shows some examples of the temporally sensitive tags detected by our algorithm together with the estimated center and standard deviation of their visibility intervals. In general the results are promising. As illustrated, the method is capable of detecting not only the names of single/annual events, but also the tags indicating the time (*e.g.*, month, season, or even year). The last two tags marked by “†” in Table 5.5 are examples of false positives generated by our algorithm. Though such tags are usually considered to be visible at all times, on occasion they can be closely related to an event as well. The tag *wall street* is associated with the *Occupy Wall Street* movement which staged a protest event that happened in New York City’s Wall Street financial district³ and the tag *transformers 3* is associated with the filming of the movie “Transformers 3” in Chicago, in 2010.

³The actual event that triggered the hot spot of the tag *wall street* in Chicago is the *Occupy Chicago* collaboration which began on 24 September 2011, in solidarity with the *Occupy Wall Street* protests.

Table 5.4: F1 scores based on different settings of NP and α .

$NP \backslash \alpha$	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
5%	0.55	0.55	0.585	0.585	0.622	0.549	0.536	0.508	0.455
10%	0.651	0.651	0.682	0.682	0.708	0.63	0.61	0.581	0.522
15%	0.651	0.651	0.667	0.667	0.694	0.618	0.6	0.571	0.522
20%	0.636	0.636	0.652	0.652	0.68	0.618	0.6	0.571	0.522

Table 5.5: Illustrations of the estimated social tags' temporal visibility intervals.

Tags	Center	Std. Deviation	Period	isGeo-Positionable
f1	Sep. 28	7 days	Every Year	Yes
2010	Jul. 10, 2010	90 days	—	No
spring	May 1	23 days	Every Year	No
october	Oct. 11	8 days	Every Year	No
christmas	Dec. 29	10 days	Every Year	No
lollapalooza	Aug. 6	2 days	Every Year	Yes
occupy wall street	Oct. 17, 2011	28 days	—	Yes
wall street [†]	Oct. 15, 2011	7 days	—	Yes
transformers 3 [†]	Aug. 4, 2010	78 days	—	No

It is difficult to recognize such situations and therefore the algorithm marked them as events as well. We further examined the temporal sensitive tags that were not easily detected and found they mainly included two types: the ones whose deviation was much larger than the density (*e.g.*, *day*, *evening* and *2011*) and the ones that are ambiguous (*e.g.*, *march*).

5.5.2 Accuracy of Tag Positioning

In the tag classification step, our classifier selected 412 positionable tags that were used for the following evaluation. Recall that we adopt the location of the hot spot covering the highest percentage of geo-coordinates as the principle loca-

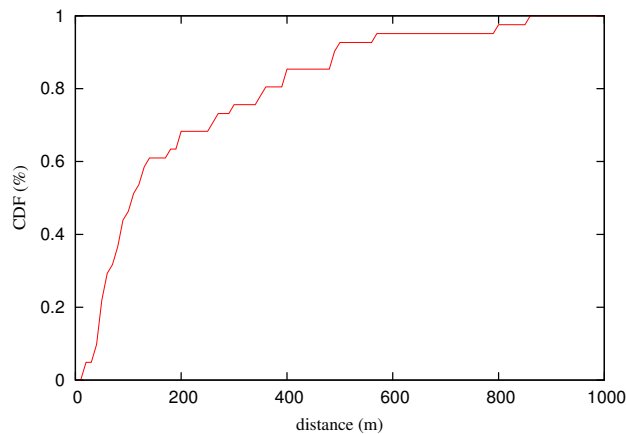


Figure 5.6: Cumulative distribution function (CDF) of the distances between the estimated and the real positions.

tion of a positionable tag. This location is further used by our auto-annotation approach to conduct geometry computations and determine the coverage of the tag by a specific video. Therefore, we need to determine whether the estimated locations are accurate enough. Though the classification step ensures that tags are positioned at some locations with street-level accuracy, we need to check whether they are positioned at the locations where they are semantically supposed to belong.

Usually, it is difficult to decide what the correct location of a tag is, except when the tag represents a landmark. We obtained the locations of 41 such positionable tags from Google Map, Wikipedia, *etc.*, to serve as the ground truth, and then computed the distance between the ground truth and our estimated locations. In general, the mean distance is 202 m while the standard deviation is 207 m. In detail, Figure 5.6 shows the cumulative distribution function of the distances, which are not uniformly distributed. More than 50% of the distances are shorter than 100 m. The absolute values would seem to be still acceptable since the scale of these landmarks is usually at the level of hundreds of meters,

and the camera may not be still, but pan across a region.

5.5.3 Tag Expansion and Ranking

Based on the positionable tags detected by the classifier, we first carried out tag expansion and then supplied these positionable tags to our auto-annotation framework, which was equipped with the new features introduced in Section 5. To verify the tag expansion approach, we compute the precisions under different threshold settings and report the statistics in Table 5.6. The first row lists the results computed based on the true positionable tags that were manually labeled as the ground truth. To eliminate manual work, we carried out the tag expansion based on the positionable tags that were automatically detected, and report the precisions in the second row. As can be seen, both of them achieve the highest precision when the threshold is set to 0.1. Due to error accumulations, the precisions decreased slightly when we utilized the automatically detected positionable tags. Fortunately, the probability that two random tags are similar in geospatial distribution is low. Compared with the precision of the positionable tag classifier which is 0.846 as reported in Section 5.5.1, the tag expansion precision 0.778 is compatible and thus can be integrated into our system. Table 5.7 shows some examples of the tags that were expanded.

Figure 5.7 shows two canonical sensor-rich videos we previously captured and the generated tags for each based on different datasets. The recording locations of the video clips were the Marina Bay in Singapore and the Grant Park in Chicago, respectively. For comparison purposes, the first row lists the tags generated using the information extracted from OSM only. The second row of results are generated from the geographically positionable tags that we

Table 5.6: Precision comparison of tag expansion based on the true positionable tags $geotags_t$ and the automatically detected positionable tags $geotags_d$.

JSD	0.05	0.1	0.15	0.2
Tag expansion based on $geotags_t$	0.815	0.829	0.714	0.632
Tag expansion based on $geotags_d$	0.714	0.778	0.654	0.633

Table 5.7: Illustrations of tag expansion. The tag detected is listed together with its nearest positionable neighbor and the *Jensen-Shannon divergence* between them.

Tag	NN	JSD
occupy	nato summit	0.0106
protests	occupy chicago	0.0161
sands	mbs (marina bay sands)	0.0351
skyscraper	downtown chicago	0.0521
bay	marina bay sands	0.0537
downtown	downtown chicago	0.0700
skyway	supertree	0.0949
fountain	grant park	0.0989
bean	attplaza	0.1376
cloud forest	gardens by the bay	0.1507

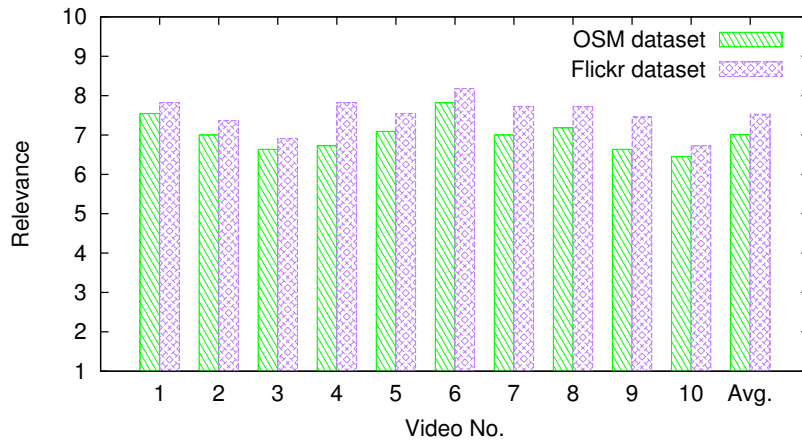
detected by applying tag classification and expansion. We can observe that the tags in the first row look long, formal and are completely spelled out. In contrast, tags in the second row originate from the Flickr dataset and are more concise and casual. By taking the tags' temporal visibility into consideration, we were able to remove the tags of the National Day Parade and the F1 Grand Prix from the video clip taken near Marina Bay while keeping the tags of the NATO Summit and the Chicago NATO protests for the one taken in Chicago.

To evaluate the effectiveness of our proposed technique, we carried out a user study to capture user preferences regarding the annotation results. We selected ten video clips from different regions around the world. Without loss of generality, we used only the top ten tags generated based on different datasets.

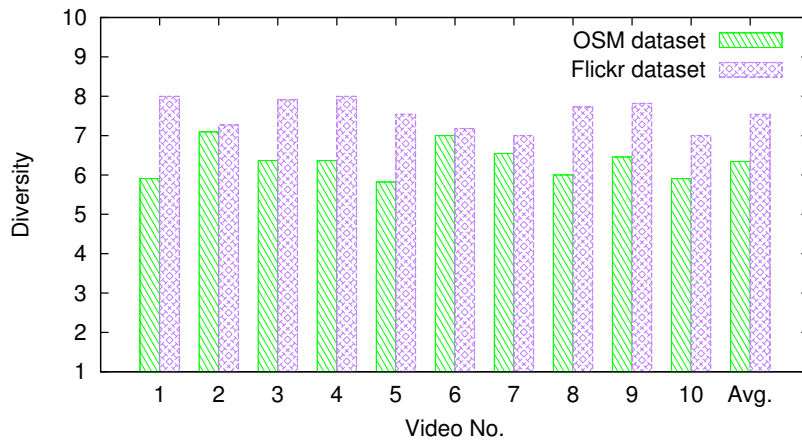


Figure 5.7: Illustration of snapshots of sample videos. The top tags are generated with the proposed auto-tagging system based on different datasets.

22 volunteers who are familiar with the regions where the videos were taken participated in this user study. They were requested to watch each video carefully and score the tag set based on the following two criteria: (1) the relevance of the generated tags (1 – least, 10 – most), and (2) the diversity of the generated tags (1 – least, 10 – most). Figure 5.8 shows the results of this user study. As can be seen, the relevance of the tags generated based on either of the datasets is high. The average relevance score achieved by using the Flickr dataset is 7.53, which is higher than the score of 7.01 achieved by using the OSM dataset. The results demonstrate the effectiveness of our proposed techniques to build the spatial-temporal tag repository. In terms of tag diversity, the improvement achieved by using the tag repository we built is even higher. The average diversity scores are 6.35 and 7.55, respectively. As the OpenStreetMap only records



(a) Relevance



(b) Diversity

Figure 5.8: Comparison of (a) relevance and (b) diversity of the tags generated based on different datasets.

landmarks in the physical world, the semantics of the generated tags are all within the geospatial domain. Comparatively, the tags in the spatial-temporal repository we built are not limited to the names of geographic objects but can be any tag that is strongly correlated with a specific place (*e.g.*, the name of an event). Additionally, by applying the tag expansion approach, the semantics of the tags are further enriched. Overall, there is strong evidence that our adaptation algorithms are effective in generating accurate tags with more diverse

semantics.

5.6 Summary

We presented an innovative auto-annotation approach for sensor-rich videos, and showed how a positionable tag repository extracted from social multimedia applications can be beneficial. To setup such a repository, we estimated the geographic distribution model of tags, extracted two features from the model, and built two classifiers to detect positionable tags. Furthermore, we profiled their temporal distributions to determine their effective durations. To make better use of the repository, we extended the visibility computation algorithm to the temporal domain, and computed tag similarity, popularity and geographic bias to re-order the tag list. The excellent quality of the generated tags with this overall approach has been confirmed through our evaluation.

In our future work we plan to investigate how to combine tags supplied from heterogeneous data sources, extend our approach to Internet-scale, and popularize our mobile video capturing applications to obtain more sensor-rich videos for evaluation.

CHAPTER 6

Visual and Geographic Information Use in Video Landmark Retrieval

6.1 Introduction

The retrieval of landmark sequences from video collections still remains a very challenging task. While the previous chapter focuses on efficient geo-based video annotation in support of keyword-based search, this work introduces a more effective hybrid retrieval method by leveraging the content and the context conjunctively. Content-based visual information retrieval offers a promising approach for landmark search. In recent years the bag-of-words (*BoW*) model [28], which was inspired by the success of text-based retrieval, has been extremely popular in a variety of visual retrieval and categorization tasks. The original *BoW* approach has subsequently been improved in its performance in

various ways [67, 126, 119]. Such content-based retrieval, however, has one drawback that hinders scalability, namely high computational complexity due to extensive signal-level processing. Moreover, it is susceptible to environmental conditions associated with an image, for example its illumination and camera recording angle [63, 51].

Since content-based retrieval is sometimes struggling to achieve satisfactory results, researchers have begun to utilize *contextual information* as an alternative or supplement to visual information. For outdoor videos and images, geographic information is especially useful. When performing a landmark retrieval task among a set of GPS-tagged images, geo-clustering is usually applied at an early stage [56, 10]. For geo-referenced videos, the associated geo-metadata has been utilized for auto-tagging and searching large collections of community-generated videos [7, 99]. The principles of the geo-based technique can also be applied to landmark retrieval. One challenge is that its performance is influenced by the accuracy of the sensor data.

In this study, we evaluate, compare and finally integrate two major types of landmark retrieval techniques: (a) the content-based and (b) the geo-based approaches. Note that we do not utilize textual metadata as it differs from visual content and geo-context in terms of granularity, *i.e.*, textual annotations such as titles and tags are usually not localized to frames. Moreover, textual annotations can be ambiguous and noisy and hence their accuracy is difficult to assess. The video collection we use in our experiments is *geo-referenced*, meaning the location and orientation of the cameras were recorded and associated with the video streams as metadata that can subsequently be used for geo-based retrieval. For content-based retrieval, videos need to be preprocessed such that the visual feature information is extracted, coded, and stored.

We first compare two state-of-the-art content-based methods, namely Spatial Pyramid Matching with Sparse Coding (*ScSPM*) and Locality-constrained Linear Coding (*LLC*) [126, 119], with a geo-based method which we refer to as Geo Landmark Visibility Determination (*GeoLVD*), in terms of precision, recall and execution time, respectively. We selected these methods as representatives because they currently exemplify the state-of-the-art and are superior in their own fields. Both *ScSPM* and *LLC* enrich the traditional *BoW* with spatial information and the advanced coding techniques they adopt not only accelerate the processing speed but also significantly improve the effectiveness. *GeoLVD* computes the visibility of a landmark based on intersections of a camera’s field-of-view and the landmark’s geometric information available from GIS. It utilizes the state-of-the-art *FOVScene* [8] to model the camera’s field-of-view and its effectiveness is well supported by experimental results. Second, we analyze the detailed factors that can affect the retrieval effectiveness. For the content-based method, we investigate the influence of selecting a representative training set and the impact of the diversity of the video frames. We also seek better sources for training images and propose to use Google StreetView as a supplement to Flickr, a combination which is shown to be effective in our experiments. For the geo-based *GeoLVD* method we analyze the influence of the accuracy of a video’s geographic metadata and the level of detail of the information we can obtain from geographic information system sources. Finally, we propose a hybrid retrieval method based on the integration of visual and geographic information. Experiments show that such a combination is compelling and achieves the best performance in the landmark retrieval task.

6.2 Landmark Retrieval Methods

The initial step for video landmark retrieval is to determine the landmark’s visibility in any given frame. We describe the algorithmic fundamentals for two different retrieval paradigms to recognize landmarks from a collection of community contributed videos. Section 6.2.1 first describes two existing, state-of-the-art content-based methods and then Section 6.2.2 introduces a context-based method that utilizes geographic properties.

6.2.1 Landmark Retrieval from Visual Cues and Features

Recently the *BoW* model [28] has been extremely popular for use in a variety of visual retrieval and categorization tasks because of its high classification quality. The method treats an image as a collection of orderless appearance descriptors extracted from local patches, quantizes them into discrete visual words, and then computes a compact histogram representation for semantic image classification. Subsequently *SVM* classifiers are constructed from the labelled *BoW* representations. Here we model the landmark retrieval task as a two-class (positive vs. negative) classification problem. In a retrieval session, our system selects the *SVM* trained for the target landmark, and then scans and ranks the frames based on the probability scores output by the selected *SVM*.

One limitation, however, is that the *BoW* approach discards the spatial information of local descriptors, which severely limits the descriptive power of the image representation. Therefore the *SPM* technique has been proposed to overcome this issue by [67] for natural scene categorization. The approach

creates a spatial pyramid representation by partitioning the image into increasingly fine sub-regions and computing histograms of local features found inside each sub-region. *SPM* shows significantly improved performance on challenging scene categorization tasks. However, researchers have empirically found that it only works well with a particular type of nonlinear Mercer kernels, *e.g.*, the *Chi-square* kernel [126]. As a consequence, the nonlinear classifier imposes additional computational complexity, namely $\mathcal{O}(n^3)$ in training and $\mathcal{O}(n)$ in testing, where n is the number of support vectors. New coding techniques have been proposed to make it work well with simple linear *SVMs*, which can dramatically improve the scalability of the training phase and the speed of testing. Two linear *SPMs* with advanced coding techniques are tested in our experiments. The algorithmic details are described in the two following Sections 6.2.1 and 6.2.1.

Linear Sparse Coding

A method called *ScSPM* [126] has been proposed to compute the spatial-pyramid image representation based on Sparse Coding (*SC*), instead of the *K*-means vector quantization (*VQ*) in the traditional *SPM*. The approach is naturally derived by relaxing the restrictive cardinality constraint of *VQ*. Furthermore, it uses max pooling, which is more robust to local spatial translations and more biologically plausible, rather than the average pooling adopted in the original *SPM*.

Let X be a set of D -dimensional local descriptors extracted from an image, *i.e.*, $X = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{D \times N}$, and let $B = [b_1, b_2, \dots, b_M] \in \mathbb{R}^{D \times M}$ be a codebook with M entries. The *SC* in *ScSPM* can be described as solving the

following problem:

$$\operatorname{argmin}_C = \sum_{i=1}^N \|x_i - Bc_i\|^2 + \lambda \|c_i\|_l \quad (6.1)$$

where $C = [c_1, c_2, \dots, c_N] \in \mathbb{R}^{M \times N}$ is the set of sparse codings for X . The restrictive cardinality constraint of VQ is relaxed by using a sparsity regularization term, which is the l norm of c_i in this case. The codebook training is equivalent to finding the optimization of problem Eqn. 6.1, which can be solved by algorithms such as the *feature-sign search* algorithm.

It has been found that *ScSPM* works well with simple linear *SVMs*, which means it remarkably reduces the complexity of *SVMs* to $\mathcal{O}(n)$ in training and a constant in testing, and even improves the classification accuracy. However, the coding speed is relatively slow.

Locality-constrained Linear Coding

Wang *et al.* [119] proposed an approach that enables *SPM* to work with locality-constrained linear coding, referred to as *LLC*. This approach also adopts max pooling, and it utilizes the locality constraints to project each descriptor into its local-coordinate system. The *LLC* code uses the following criteria:

$$\begin{aligned} \operatorname{argmin}_C &= \sum_{i=1}^N \|x_i - Bc_i\|^2 + \lambda \|d_i \odot c_i\|^2 \\ \text{s.t.} \quad &1^T c_i = 1, \quad \forall i \end{aligned} \quad (6.2)$$

where \odot denotes element-wise multiplication, and $d_i \in \mathbb{R}^M$ is the locality adaptor that allocates different freedom for each basis vector proportional to its similarity to the input descriptor x_i .

Just like with *ScSPM*, the codebook can be trained using *LLC* coding cri-

teria shown in Eqn. 6.2. In effect, it can also be trained using a simple K -means clustering method, since experiments have shown that the codebook generated by K -means clustering can produce satisfactory accuracy. More extensive experiments on the selection of codebooks were carried out by Viitaniemi and Laaksonen [116]. A fast approximated LLC method has been proposed as well to further speed up the encoding process. This efficiency significantly adds to the practical value of LLC for real-world applications.

6.2.2 Landmark Retrieval from Geographic Information

Due to technological advances, sensor-equipped smartphones have made it easy to record geo-referenced videos through the use of popular apps. The location and direction of the camera can be obtained from the built-in GPS and compass sensors of a smartphone. Furthermore, geographic information about landmarks can be retrieved from map sources such as OpenStreetMap (<http://www.openstreetmap.org/>), which is a free service that provides editable maps of the world. Next, we will introduce the details of a landmark retrieval technique based on geographic information.

Viewable Scene Model

Presented with the geographic information associated with a video frame, here we utilize the viewable scene model (referred to as $FOVScene$) proposed by Arslan Ay *et al.* [8] to describe the visible scene based on a camera's field-of-view (FOV). Recall that the 3-dimensional $FOVScene(P, \vec{d}, \theta, \phi, R)$ model is illustrated in Figure 3.1, with the following parameters: (1) the camera position P , (2) the camera direction (*i.e.*, compass) vector \vec{d} , (3) the horizontal and

vertical camera viewable angles θ and ϕ which describe the angular extent of the scene filmed by the camera, and (4) the far visible distance R which is the maximum distance at which a large object within the camera’s field-of-view can be recognized. In this study, the position (latitude/longitude) and orientation information of the camera are associated with video streams at a fine-granular level as metadata, *i.e.*, at each – or every few – frames. For simplicity, we assume that the camera is always level, *i.e.*, the direction vector \vec{d} only stays on the horizontal plane. The parameters θ , ϕ and R are constants that can be estimated from the optics of the camera used for video recording [43].

Determination of Landmark Visibility

Given the name of a landmark and a video frame, the task here is to determine the landmark’s visibility in the frame. The geometry of the viewable scene of the frame G_{frame} is modeled as $FOVScene(P, \vec{d}, \theta, \phi, R)$. The geometry of the landmark $G_{landmark}$ can be retrieved from, for example, OpenStreetMap. We also need to consider the situation where sometimes the existence of other buildings may hide the landmark from the camera’s sight, and it is therefore necessary to retrieve the geometry of all relevant objects within the same region as the landmark for occlusion checks. Let $\{o_1, o_2, \dots, o_k\}$ be the set of all relevant objects, then $\{G_{o_1}, G_{o_2}, \dots, G_{o_k}\}$ represents their corresponding geometry set.

Given the geometry information above we introduce an algorithm termed *GeoLVD* to determine the visibility of a certain landmark within a frame. This algorithm is inspired by the 3D visibility query processor proposed by Shen *et al.* [99], but differs mainly in the following three aspects:

- The *GeoLVD* algorithm determines the visibility of a given landmark,

while Shen’s method aims to find all the visible objects within the viewable scene.

- The *GeoLVD* algorithm considers the landmark to be either visible or invisible, while Shen *et al.* further classify the visible objects into two sub-categories: front objects and vertically visible objects.
- The *GeoLVD* algorithm uses an *R*-tree* index structure to organize the geometry of all the relevant objects, while Shen’s method does not mention any advanced data structure adopted for spatial indexing.

Instead of assigning scores to frames based on the ratio of viewable angles of the target landmark, we define the output of *GeoLVD* to be binary, *i.e.*, the output is either zero or one, where zero (one) indicates that the landmark is invisible (visible). Our rationale for using a binary output is that in our experience the effectiveness of the geo-based method highly depends on the accuracy of the geographic metadata. We found that the use of more elaborate geometry calculations provides no significant benefit for the correction of the variations induced by noise which the geographic metadata intrinsically possesses. Figure 6.1 shows an example of the *GeoLVD* method in (a) Google Earth together with its corresponding (b) projection on the 2D plane. Assume that the landmark to be retrieved is the Marina Bay Sands hotel. The main steps of the algorithm are as follows. First it computes the field-of-view towards the landmark within the *FOV* of the camera, which is the region colored in yellow. Next, the shape’s Minimum Bounding Rectangle (*MBR*) is calculated – the geometry colored in green – and used to perform an *R*-tree* spatial search among the GIS source objects. The objects labeled with letters are returned because they overlap with the *MBR*. Among the returned objects, A, D, and E appear

in between the camera and the landmark, so their height needs to be checked to see if an occlusion occurs. Finally, *GeoLVD* determines the visibility of the landmark after removing all the occlusion situations.

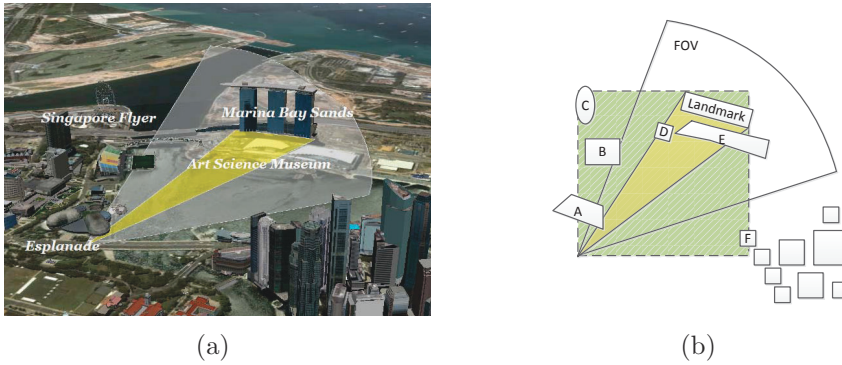


Figure 6.1: (a) An illustration of a camera's field of view in Google Earth (Copyright © 2013 Google [36]). (b) The corresponding scene projection on the 2D plane.

To describe the problem formally, let $r = [\mu, \nu]$ denote a horizontal angle range. Then the visible angle ranges of the landmark within the *FOV* can be denoted as a set $VisibleR = \{r_1, r_2, \dots, r_k\}$. The goal of the algorithm is to compute the visible angle ranges of the queried landmark. A value of $VisibleR = \emptyset$ indicates that the landmark is invisible, otherwise, it is considered visible. Algorithm 2 sketches the overall procedure to determine a landmark's visibility. *InitVisibleRanges()* initializes the horizontal visible angle ranges of the landmark within the FOV without considering occlusions. *GeometryFilter()* filters out the objects that will not cause occlusions by executing an *R*-tree* spatial search with the query area as the MBR of $VisibleR$. All the returned objects are examined one by one to see if an actual occlusion occurred. *ComputeOccludedRC()* computes the horizontal occluded angle ranges, which are considered as candidates and will be further checked by *isVerticallyOccluded()* to see if any of them are vertically occluded as well. If not, the object needs to be

<p>Input:</p> <p>1 The geometry of the viewable scene $G_{frame}: FOVScene$;</p> <p>2 The geometry of the queried landmark $G_{landmark}$;</p> <p>3 The geometry set of all the relevant objects $G_o = \{G_{o_1}, G_{o_2}, \dots, G_{o_k}\}$;</p> <p>Output:</p> <p>4 The visible angle ranges of the queried landmark $VisibleR$;</p> <p>5 $VisibleR = InitVisibleRanges(FOVScene, G_{landmark})$;</p> <p>6 $G'_o = GeometryFilter(VisibleR, G_o)$;</p> <p>7 for each $G_{o_i} \in G'_o$ do</p> <p>8 $OccludedRC = ComputeOccludedRC(VisibleR, G_{o_i})$;</p> <p>9 for each $r \in OccludedRC$ do</p> <p>10 if $!isVerticalOccluded(G_{landmark}, G_{o_i}, r)$ then</p> <p>11 $OccludedRC = OccludedRC - \{r\}$;</p> <p>12 end</p> <p>13 end</p> <p>14 $UpdateVR(VisibleR, OccludedRC)$;</p> <p>15 if $VisibleR = \emptyset$ then</p> <p>16 break;</p> <p>17 end</p> <p>18 end</p> <p>19 return $VisibleR$;</p>

ALGORITHM 2: *GeoLVD* — Geo Landmark Visibility Determination Query Processor

removed from the candidate set. $UpdateVR()$ updates the visible angle ranges by subtracting the occluded angle ranges at the end of each iteration. The algorithm terminates when $VisibleR$ becomes empty or after all the relevant objects have been checked for potential occlusions.

6.3 Evaluation

Next we compare the retrieval performance of the content- and geo-based methods. For this purpose, we first introduce the datasets and the experimental settings, and then we report the statistics and summarize the strength and weakness of each approach.

6.3.1 Experimental Settings and Datasets

We selected eight popular landmarks in Singapore as our retrieval targets as follows: the Marina Bay Sands hotel, the Esplanade, the Singapore Flyer, the Art Science Museum, the Gardens by the Bay¹, the Merlion¹, the Universal Studios Globe¹ and the Ngee Ann City¹. For each of these landmarks, we collected images from Flickr by posting queries while setting both text and location restrictions. Next, the image sets were manually filtered, keeping only 250 images for each landmark with considerations for both high quality and good diversity. We found that some landmarks might co-occur in the same image (*e.g.*, the Marina Bay Sands hotel and the Art Science Museum). Thus, to reduce complexity, we prepared a common negative examples set for all the landmarks. The images in the negative set were collected from Flickr consisting of other landmark images around the world and images taken in Singapore, and again applying a manual filter and retaining 750 images in the end. As a result, a 1,000-image training set was formed for each of the landmarks including 250 positive and 750 negative instances.

The video collection on which we performed the landmark search consists of 131 geo-referenced videos taken in Singapore. To understand the impact of illumination on any content-based retrieval we further divided the videos into two groups of 114 day-time and 17 night-time videos. The groundtruth of a landmark’s visibility was annotated manually frame by frame at a sample rate of five per second. The groundtruth annotations distinguish the following three situations: (1) landmark entirely visible, (2) partially visible, and (3)

¹These final four landmarks were mainly used in testing the scalability of the proposed hybrid method due to some dataset restrictions – we had limited night-time videos in the test set and there was a lack of Google StreetView images.



Figure 6.2: Illustration of frames with fully and partially visible landmarks in the test set.

invisible. Several examples of frames with fully visible and only partially visible landmarks in the test set are displayed in Figure 6.2.

The local feature we used were *SIFT* descriptors of 16×16 pixel patches computed over a grid with a spacing of 8 pixels. We adopted a three-level pyramid matching with a vocabulary size of 1,024 for both *ScSPM* and *LLC*.

6.3.2 Frame Retrieval Evaluation

Each video was treated as a collection of frames and we evaluated the different methods on the task of frame retrieval. The test set for each run was formed by randomly choosing 1,000 frames for querying from the video collection. The proportion of positive to negative samples in the test set was set to 3:7, considering the fact that a landmark usually appears only in a small portion of a video. The retrieval techniques were evaluated on the criteria of precision, recall, and execution time. Ten experimental runs were carried out and the average results are reported in Tables 6.1, 6.2 and 6.3. Our tests were performed on a desktop computer with a 3.20 GHz dual core CPU and 4 GB of main memory. Both the entirely and partially visible landmarks were regarded as positive instances. The classification threshold of the *SVMs* is set to the mid-value of the output range, which is 0.5 when the output is a probability score varying

Table 6.1: Retrieval technique comparison over different landmarks and video conditions.

(a) Marina Bay Sands hotel

Illumination	Day		Night	
Criterion	Precision	Recall	Precision	Recall
<i>ScSPM</i>	0.8972	0.5410	0.7393	0.4547
<i>LLC</i>	0.8838	0.4927	0.6868	0.4553
<i>GeoLVD</i>	0.7144	0.8910	0.6545	0.6640

(b) Esplanade

Illumination	Day		Night	
Criterion	Precision	Recall	Precision	Recall
<i>ScSPM</i>	0.9396	0.3273	0.6151	0.1347
<i>LLC</i>	0.9099	0.3173	0.6688	0.2113
<i>GeoLVD</i>	0.6991	0.8450	0.6706	0.9223

(c) Singapore Flyer

Illumination	Day		Night	
Criterion	Precision	Recall	Precision	Recall
<i>ScSPM</i>	0.8845	0.1100	0.4181	0.0247
<i>LLC</i>	0.7308	0.0587	0.3150	0.0557
<i>GeoLVD</i>	0.7910	0.8307	0.6143	0.7343

(d) Art Science Museum

Illumination	Day		Night	
Criterion	Precision	Recall	Precision	Recall
<i>ScSPM</i>	0.8936	0.3223	0.5735	0.3500
<i>LLC</i>	0.8614	0.3113	0.5288	0.4927
<i>GeoLVD</i>	0.7314	0.8213	0.7417	0.7450

between zero and one. For the spatial index structure we used the Java R^* -tree implementation available from <https://code.google.com/p/spatialindex>.

Result Discussion

We first report our main observations from Tables 6.1, 6.2 and 6.3 and then discuss interesting phenomena and their potential reasons. Finally we summarize the strengths and weaknesses of the content-based and geo-based methods, respectively.

Table 6.2: Retrieval technique comparison over supplementary landmarks among day-time videos.

Landmark	Gardens by the Bay		The Merlion		Universal Studios		Ngee Ann City	
Criterion	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
<i>ScSPM</i>	0.8165	0.5037	0.9874	0.2810	0.8550	0.4577	0.9112	0.1930
<i>LLC</i>	0.7714	0.8597	0.9598	0.5630	0.7760	0.4330	0.6641	0.1630
<i>GeoLVD</i>	0.7070	0.9740	0.8183	0.8967	0.8367	0.5870	0.8612	0.8537

Table 6.3: Execution time per query frame of the content- and geo-based methods.

Step	Preprocessing	Retrieval
<i>ScSPM</i>	2.4 s	0.17 ms
<i>LLC</i>	1.1 s	0.17 ms
<i>GeoLVD</i>	-	0.30 ms

- **Observation 1: Dependence on Illumination.** Among day-time videos, the geo-based method always achieves a better recall while the content-based methods achieve better precisions. However, among the night-time videos the geo-based method outperforms the content-based methods in both recall and precision, except for the one case of the Marina Bay Sands hotel.
- **Observation 2: Dependence on Landmark.** The performance of both *ScSPM* and *LLC* varies significantly among the different landmarks we tested, while the *GeoLVD* method performs relatively stably.
- **Observation 3: Execution Time.** The three methods exhibit comparable, high retrieval speeds, but the content-based methods involve an extra preprocessing step for visual feature extraction and coding, which is time-consuming. Though *LLC* is much faster than *ScSPM* in the preprocessing step, it performs less stable and mostly worse in terms of precision and recall.

It is obvious that the illumination has a great impact on the content-based methods. This is not unexpected because objects become less distinguishable in low light even for human eyes, and therefore also for *SIFT* feature descriptors. On the other hand, illumination should have little impact on the geo-based method, so ideally *GeoLVD*'s performance should be similar on videos taken under various conditions. However, this is not always the case as is illustrated in Table 6.1. For example, the precision of *GeoLVD* for the Singapore Flyer has a gap of 18% between day and night. We consider this to be caused by the variations in the video sets' geographic distributions which also varies, besides the illumination. The location where a video is recorded is a crucial factor for a geo-based method because smaller obstacles that may not be stored in a GIS database (*e.g.*, OpenStreetMap), such as trees and unimportant buildings, are more likely to exist in some places than others. Therefore, the retrieval precision could be significantly reduced under such circumstances.

The second observation concerns the impact of the diversity of landmarks. For the content-based methods – though the *SIFT* feature descriptor we used is invariant to uniform scaling, orientation, and affine distortions to some extent – the real-world landmark variations are far more complex than that. In general, frames in which the landmark occupies the majority of the scene can provide more distinguishable local features, so they are easier to be recognized by the classifier. Furthermore, the similarities and differences between the training and test images also affect the classifier's decision. The training set is expected to encompass the landmark's visual diversity in order to well represent all possible situations that may appear in the test images, but as is shown in our experiments, the Flickr images cannot always represent the video frames well. For example, Figure 6.3 shows two images of the Singapore Flyer. Figure 6.3(a)

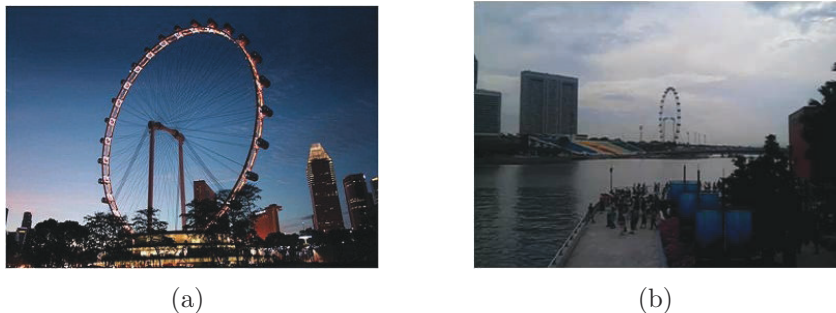


Figure 6.3: Two typical images of the Singapore Flyer landmark in the experimental dataset: (a) a representative of the Flickr training images (Copyright © 2013 Flickr [33]) and (b) a representative of the frames from the video dataset.

is a representative of the Flickr images and Figure 6.3(b) is a representative of the video frames. Since they were taken at different places, the appearance of the Singapore Flyer differs markedly. We may speculate that there are fewer locations where users take photos of landmarks compared to places where videos are taken, because videos may include actions such as pans and zooms and a landmark may only be part of a lengthy shot. Therefore, in videos recorded at places other than the favorite spots for taking Flickr photos, the landmark becomes more difficult to be recognized by the classifier.

When analyzing the execution time, one key difference between content-based and geo-based methods is that the former requires an extra preprocessing step before retrieval can be performed. Visual features need to be extracted and coded in the preprocessing phase for selected frames. Although in video retrieval applications the frame contents are usually analyzed only once and the features saved for indexing and searching, it can usually not be ignored because it is very time-consuming compared with the actual retrieval. In small to medium sized video sharing systems, the time spent on video preprocessing may be acceptable, but as it becomes significant in large video sharing systems

such as YouTube, issues arise such as determining the best time to preprocess a video. Though generally only a small portion of videos gain great popularity while the others receive little attention, it still remains a very difficult task to predict the popularity of a video beforehand. It may be wasteful to preprocess a video immediately after it is uploaded because it could turn out to be an unpopular clip that users rarely search for. On the other hand, if preprocessing is performed on demand of a retrieval request, it will cause major delays and possibly make the system impractical. The geo-based method, on the other hand, does not encounter this kind of concern.

Content- versus Geo-based: Strengths and Weaknesses.

As discussed above, the content-based method is susceptible to variations in illumination and landmark appearance. It is challenging to form a training set that can well represent all the possible conditions of video frames. Additionally content-based methods require a time-consuming preprocessing step before retrieval. The strength of content-based methods is that an *SVM* classifier outputs a probability. Thus by choosing different thresholds, one can find the best trade-off between precision and recall. The strength of the geo-based method is that it is more stable under various video conditions and landmark appearances. Since videos do not need to be preprocessed, they can be retrieved immediately after being uploaded to a server. Note that although the performance of the geo-based method depends on the accuracy of the sensor data, we found from our experiments that the performance is acceptable, since *GeoLVD* achieves an average recall of greater than 80% (see Table 6.1). For comparison purposes we also computed *ScSPM*'s average recall when it achieves an equal precision to *GeoLVD* and it is only less than 40% among day-time videos. However, the

geo-based method has its own drawback. It relies on geographic information services such as OpenStreetMap which may have uneven building and detail coverage, leading to missed occlusions and obstacles. However, if past experience is any guide then these data sources are rapidly improving in both coverage and details. Though this still leaves the problem of dynamic occlusions such as a bus passing by or newly planted trees.

6.3.3 Content-based Method Robustness Analysis

From our experiments we have an approximate idea of the performance that can be achieved by *ScSPM* and *LLC* in frame retrievals for certain landmarks. Though these methods have been reported to achieve promising classification results on some standard datasets (*e.g.*, Caltech 101, Caltech 256, and PASCAL VOC 2007) they face challenges when applied to large and multi-domain real world images and video frames. Here we mainly consider and discuss the following three factors that affect the retrieval effectiveness of content-based methods:

1. The complexity of visual features.
2. The representativeness of the training set.
3. The diversity of the test set.

In general, better performance is more likely achieved by adopting a more extensive spatial model and a larger visual vocabulary, but this will also cost more memory and computation time. The visual feature vector we used for each image has a length of $(1 + 4 + 16) \times 1,024 = 21,504$ floating point values which is already highly complex for video collections. Since the videos are

geo-referenced, we expect that their performance can be further improved with the help of the geographic metadata. The integration of content and context information will be discussed in Section 6.3.5.

Next, we investigated alternative image sources that can make the training set more representative. Google Maps StreetView would seem a good choice, since it is a very comprehensive dataset which consists of 360° panoramic views of almost all main streets and roads in a number of countries, with a distance of about 12 m between recording locations. To supplement the current training image set, we collected Google StreetView images on five main streets around Marina Bay. In Figure 6.4, the red pins represent the locations of images we collected, and at each location four directional side views were retrieved, ensuring that one of them would point to the landmark of interest, which is the Singapore Flyer in the case of Figure 6.4. For each landmark, 25 images were manually selected as positive instances, and the other 75 images with the same location but different side views were automatically selected as negative instances. To evaluate the effectiveness of this new image source, a new training set was formed by randomly selecting 25 positive and 75 negative instances from the previous training set and substituting them with the Google StreetView images. In the following experiment, we will use $Training_{Flickr}$ and $Training_{Flickr+StV}$ to denote the previous and new training sets, respectively.

For the third factor in our list, we conjecture that the appearance of objects may be more diverse in videos than in images because videos frequently contain moving scenes which are different than the composition of single, static picture. Thus, the partial appearance of a landmark is a common situation in video frames. It occurs due to scene transitions, pans, zooms, or partial occlusions. To measure the fraction of a landmark’s partial appearances in videos, we

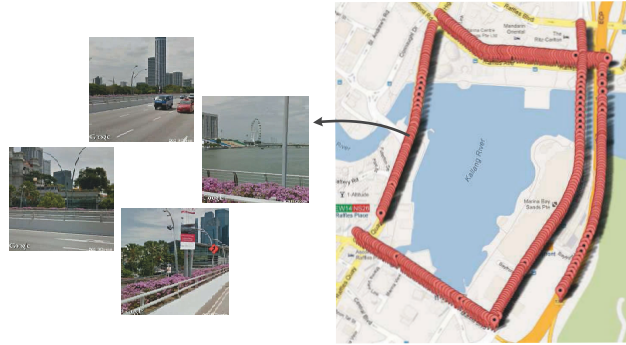
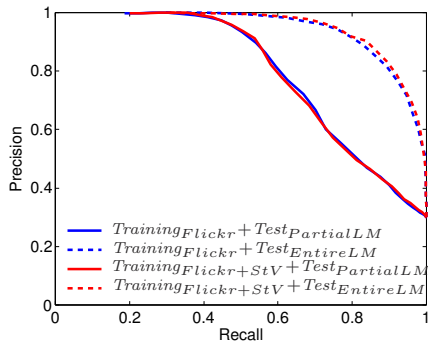


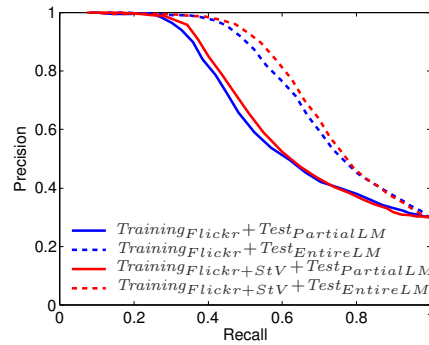
Figure 6.4: Details of the Google StreetView training images collected near Marina Bay. Right: image location distribution, left: an example of four side views per location. Copyright © 2013 Google [36].

filtered out all the frames in which the landmark is only partially visible and prepared a new test set considering only the landmark’s entire appearance as positive instances. We will use $Test_{PartialLM}$ and $Test_{EntireLM}$ to denote the previous and new test sets respectively in the following experiments.

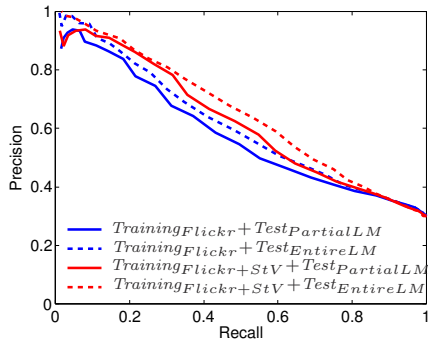
Experiments were carried out based on different combinations of training and test sets. For the content-based method we used $ScSPM$, and since the Google StreetView images we collected are all taken during the day, we performed the experiments among day-time videos only. As the frames were ranked based on the SVM probabilistic outputs, we computed the precision-recall curves by changing the threshold value. The results are shown in Figure 6.5. We observe that the StreetView images indeed enhance the representativeness of the training set and produce the greatest improvement for the Singapore Flyer landmark. Moreover, when the definition of positive instances is narrowed in the test set to the landmark’s entire appearance only, $ScSPM$ performs much better than before. This reveals not only $ScSPM$ ’s poor recognition rate with a landmark’s partial appearance, but also the widespread existence of such frames in videos. A landmark is more likely to show up partially



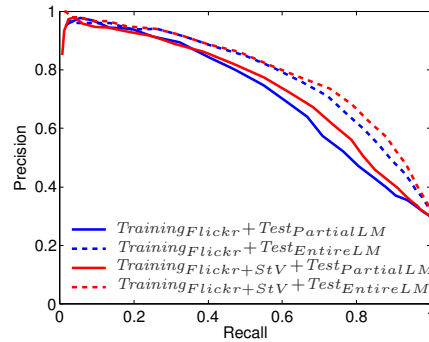
(a) Landmark: Marina Bay Sands hotel



(b) Landmark: Esplanade



(c) Landmark: Singapore Flyer



(d) Landmark: Art Science Museum

Figure 6.5: Evaluations with two training sets (Flickr only or Flickr with Google StreetView images, *Flickr+StV*) and two test sets (partial or entire view of landmarks) for the content-based *ScSPM* method.

if it is bigger in size, which is the reason why the performance on the Marina Bay Sands hotel improves the most.

6.3.4 Geo-based Method Robustness Analysis

The geo-based method geometrically computes a landmark’s visibility based on the geographic information of both the camera and the surrounding objects (*e.g.*, buildings). We summarise the factors that affect the retrieval effectiveness of geo-based methods into the following three aspects:

1. The accuracy of the smartphone sensor data.

Table 6.4: Details of landmark visibilities in Google StreetView images.

Condition	Visible	Invisible	Total
Marina Bay Sands	298	117	415
Esplanade	120	295	415
Singapore Flyer	233	182	415
Art Science Museum	90	325	415

2. The level of detail of the static geographic object models (*e.g.*, buildings).
3. The probability of the accidental appearance of real-world dynamic, moving objects such as people, vehicles, *etc.*

To evaluate the influence of each aspect, we used the Google StreetView images for the test set because they are associated with accurate location and orientation information. For each landmark, the test set is formed by all the images whose camera orientation is towards the landmark. We collected StreetView images at 415 locations, so there are overall 415 images in each test set. The details are shown in Table 6.4. We evaluated two variations of the geo-based method on StreetView images and video frames, respectively. The first approach is *GeoLVD*, which is termed the geo-advanced method as it checks if a landmark is hidden by obstacles. The other is termed the geo-basic method that does not perform the occlusion check, and hence functions for baseline comparisons.

For StreetView images, the geo-basic method always has a recall of 100% because all the test images point toward the landmark and will be surely retrieved without the occlusion check. Its precisions are 71.8%, 28.9%, 56.1%, and 21.7% respectively for the four landmarks. Comparatively, the geo-advanced method achieves higher precisions which are 85.2%, 42.0%, 66.5%, and 46.4%. The results are illustrated in Figure 6.6 and show that the geo-advanced method gains an average increase of 15.4% in precision over geo-basic. Since the

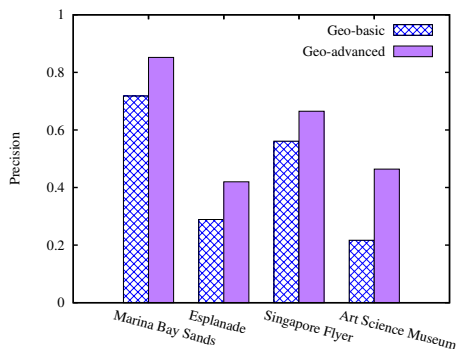


Figure 6.6: *GeoLVD* method results with retrieval queries based on Google StreetView images.

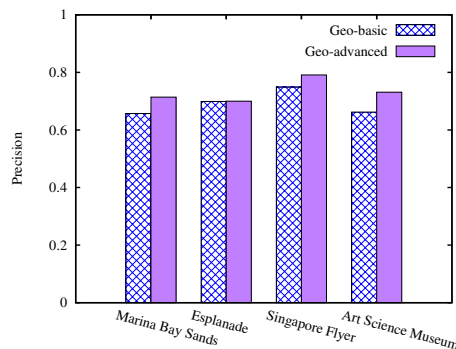


Figure 6.7: *GeoLVD* method results with retrieval queries based on video frames.

StreetView images are associated with accurate geographic information, any retrieval errors of *GeoLVD* are mainly attributed to the second and third factors listed earlier. Geographic information services such as OpenStreetMap only record data of major objects, hence it is not feasible to obtain information of unimportant buildings or trees which are potential obstacles that may hide a landmark. Moreover, even for the buildings recorded in OpenStreetMap, interestingly the height information is sometimes not available. For simplicity, we estimated the height according to a building’s name and type. For example, we used a height of 0 m for rivers and 30 m for ordinary buildings. Even this simple height estimation as the 3D occlusion check improves the precision by an average of 15.4%. However, it also worsens the recall slightly by an average of 4.3% resulting in a drop from 100% to 95.7%. We conclude that the geographic information collected from OpenStreetMap is currently not detailed enough for precise landmark visibility calculations. As a result the precisions of *GeoLVD* for the Esplanade and Singapore Flyer are still quite low (less than 50%). In general, smaller landmarks are affected more by the lack of information detail.

For video frames the retrieval precisions are visualized in Figure 6.7. We executed the experiments on 10 test sets for each landmark, and the resulting average precisions are 65.7%, 69.9%, 74.9%, 66.2% for the geo-basic method and 71.4%, 70.0%, 79.1%, 73.1% for the geo-advanced method, respectively. Compared with the results in Figure 6.6 we can make two major observations. First, both the geo-basic and the geo-advanced methods perform better on video frames, and second the performance gap between the geo-basic and the geo-advanced methods is significantly smaller on video frames. Given the assumption that the geographic sensor metadata is not as accurate as the data from StreetView, we would expect the retrieval performance to become worse. One explanation for the improvement might be that the locations for capturing landmarks have been selected by users and hence they have already been “filtered” to avoid occlusions. Hence, compared with the semi-robotically collected StreetView images, the probability that a camera view is pointing towards a famous landmark but is occluded by obstacles becomes smaller for video frames. Therefore, even without knowledge of minor inconsequential objects, the geo-based methods can still work well on video frames. The degree of sensor errors can be estimated by the recall values of the geo-basic method, which are 89.6%, 84.5%, 83.1%, and 82.1%, because the values are expected to be 100% if the sensor data is accurate. In terms of the geo-advanced method, only the recall for the Marina Bay Sands hotel decreases slightly to 89.1% while the other three values remain unchanged. This slight drop in recall, together with the other two observations above, indicate that the geo-based landmark retrieval is less susceptible to the lack of geographic information when queried with video frames.

6.3.5 Hybrid Integrated Content and Context Analysis

The encouraging results from Section 6.3.2 illustrate that contextual information, for example in the form of geographic data, is a powerful tool for the search of large-scale video archives. Since content- and context-based methods make use of complementary information, it seems natural to combine the two to further improve performance.

Here we propose a hybrid landmark retrieval method which is a late fusion approach that combines the scores of the content- and context-based methods in semantic space [106, 9]. As pointed out by Atrey *et al.* [9], the late fusion approach has the advantage of allowing us to use the most suitable methods for analyzing each single modality, *e.g.*, *SVMs* for visual content and *GeoLVD* for geographic context. In our study context refers to the location and orientation of the camera. However, the method may be extended to other contextual information in the future. The key idea of the hybrid approach is to first estimate the effectiveness of the content analysis based on the distance from the recorded frames to the landmark, and then use this measure as a weighting factor to combine the scores of the content- and context-based methods. The F1 score is a good indicator for the effectiveness of the content-based method as it considers both precision and recall (see Eqn. 6.3). Therefore, we grouped video frames based on their distance to the landmark and computed the F1 score for every group. We fit the F1 score to a Gaussian function of distance for each of the landmarks as shown in Figure 6.8. For an image that is taken far away from the landmark, the content-based method is not very reliable since the image contains limited details of the landmark but much irrelevant elements from its surroundings. As the camera location moves closer to a landmark, its structure

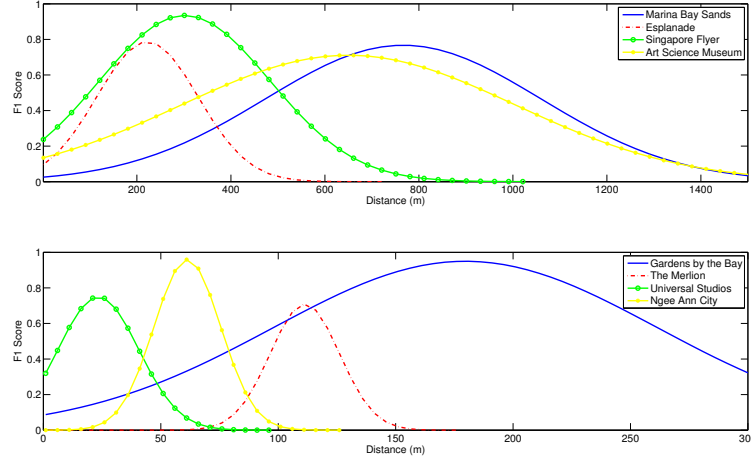


Figure 6.8: Estimated Gaussian Function of F1 Score over Distance for different landmarks.

is better visualized and the “noise” from the surroundings is reduced to some extent as well. However, if the distance is further reduced, the probability that an image focuses on only one part of the landmark instead of the whole structure becomes higher, and the resulting information loss also weakens the effectiveness of the content-based method to some extent.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (6.3)$$

We propose to combine the scores of content- and context-based methods, $Score_c$ and $Score_g$, using the formula below,

$$Score_h = \alpha \times Score_c + (1 - \alpha) \times Score_g$$

where α is the preference coefficient that controls the balance between the two scores. Based on the observations above, we define α as,

$$\alpha = a_i + b_i \times Gaussian_{F1}^i(Dist(landmark_i, frame))$$

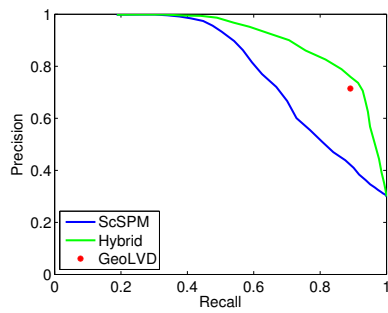
Table 6.5: F1 scores of methods based on content only, context only, and their hybrid integration.

Method	<i>ScSPM</i>	<i>GeoLVD</i>	<i>Hybrid</i>
Marina Bay Sands	0.6858	0.7930	0.8249
Esplanade	0.5736	0.7652	0.7878
Singapore Flyer	0.5628	0.8104	0.8104
Art Science Museum	0.6783	0.7737	0.8003
Gardens by the Bay	0.7347	0.8193	0.9376
The Merlion	0.7451	0.8557	0.8836
Universal Studios	0.6261	0.6900	0.8138
Ngee Ann City	0.6169	0.8574	0.8701

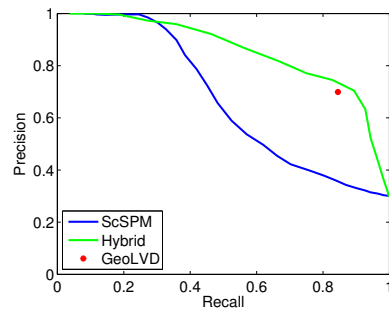
where $i = 1, 2, \dots, 8$ represents the index for the eight landmarks, $Gaussian_{F1}^i(x)$ denotes the estimated Gaussian function of the F1 score for landmark i , and $Dist(landmark_i, frame)$ computes the distance between landmark i and a given frame. Values a_i and b_i are constants such that $a_i + b_i \leq 1$. As indicated earlier, we use the F1 score as the measure of effectiveness for the content-based method. In practice, it should be avoided that α is close to zero because the location associated with a frame is acquired from GPS and thus contains some noise. Consequently, we use a_i to control the lower bound of α .

Figure 6.9 illustrates the precision-recall graphs of the proposed hybrid method as well as the results of *ScSPM* and *GeoLVD* for comparison. The experiments are performed on day-time videos. The positive instances of video frames are defined to include both the entire and partial appearances of a landmark. For each of the landmarks, a separate training set is selected where a_i and b_i ($i = 1, 2, \dots, 8$) are tuned. We also illustrate the highest F1 score that each of the methods achieves in Table 6.5.

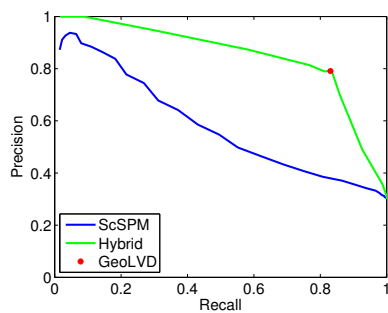
From the statistics we can see that the proposed hybrid method achieves the best result overall. It significantly improves the effectiveness of *ScSPM*. Moreover, it enhances *GeoLVD* not only by increasing the F1 score but also by



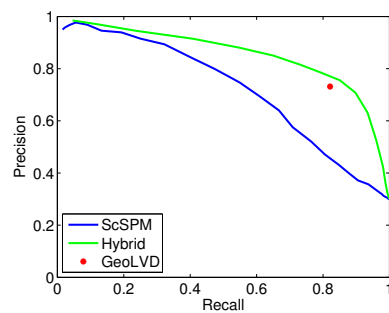
(a) Landmark: Marina Bay Sands



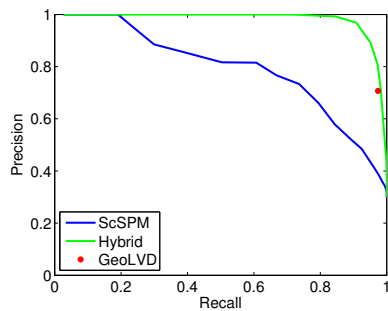
(b) Landmark: Esplanade



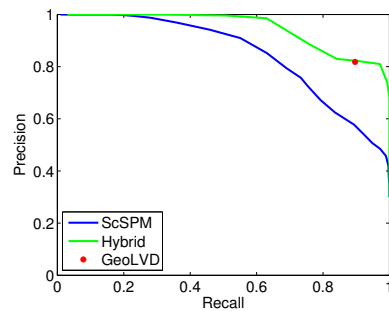
(c) Landmark: Singapore Flyer



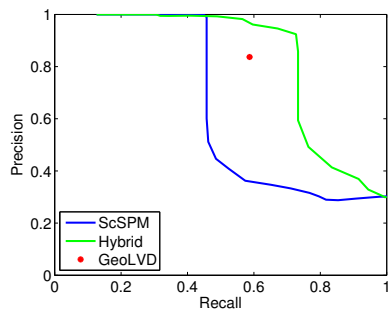
(d) Landmark: Art Science Museum



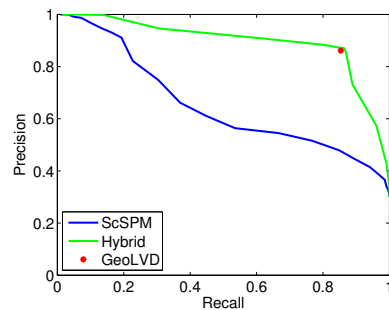
(e) Landmark: Gardens by the Bay



(f) Landmark: The Merlion



(g) Landmark: Universal Studios



(h) Landmark: Ngee Ann City

Figure 6.9: Precision-recall curves comparison of methods based on content only, context only, and their hybrid integration.

enriching the output from binary to a probability range. The best improvement is observed with the landmark Universal Studios Globe. We examined the test set and found that the frames are usually recorded very close to the globe because of its small size compared with other landmarks, and this therefore makes it the most sensitive to GPS errors. This is also the reason why *GeoLVD* only achieves an average recall of 58.7% with the globe. Fortunately, *ScSPM* compensates for such situations very well and consequently makes the hybrid method highly effective. On the other hand, most of the test frames from the Singapore Flyer are taken some distance away at Marina Bay which weakens the reliability of *ScSPM* in terms of compensation. The execution time of the hybrid method is approximately the sum of the content- and context-based methods we used, *i.e.*, 2.4 s for preprocessing and 0.47 ms for retrieval per frame, which is quite fast and acceptable. In summary, the selection of a suitable method depends on the application requirements of retrieval effectiveness which is closely related to the characteristics of the landmarks and the video collection.

6.4 Summary

In this study we have evaluated two state-of-the-art video landmark retrieval paradigms, namely media-content based and geo-context based retrievals. For the content-based retrieval, we selected two high performance methods, *ScSPM* and *LLC*, and for the geo-based retrieval, we introduced *GeoLVD*, which is inspired by the 3D visibility query processor proposed by Shen *et al.* [99].

From the comparison results we draw a number of interesting observations, chiefly among them is the importance of the illumination conditions for content-based methods, summarized as follows.

- When performing retrievals from our day-time video collection, it is always the case that content-based retrieval achieves a higher precision, while the geo-based retrieval achieves a higher recall.
- However, when carrying out retrievals from the night-time video collection, the geo-based method always outperforms the content-based method in terms of both precision and recall.
- The performance of the content-based method varies significantly when searching for different landmarks, while the geo-based method is relatively stable.

Therefore, we conclude that when the illumination or the appearance of a landmark is not favorable for content-based retrieval, the geo-based method is more suitable and should be chosen instead.

In terms of execution time, we observe the following.

- Both the content-based and the geo-based methods exhibit comparable retrieval speeds in the sub-milliseconds per frame, but the former involves an extra preprocessing step for visual feature extraction and coding which is usually time-consuming, taking on the order of 1 to 2 seconds per frame.
- *LLC*, which aims at speeding up the visual feature coding procedure, is much faster than *ScSPM* in the preprocessing step. However, it is also less accurate in the retrieval phase.

The time spent on the preprocessing step may not be a significant burden for small to medium video sharing systems. However for large video sharing platforms such as YouTube, it becomes a hassle to determine when is the best

time to preprocess a video. In such a scenario the geo-based retrieval method reveals its strength as no video preprocessing is needed beforehand. However, the standardized recording of geographic metadata is currently still in the exploratory stage among large online media sharing systems.

In the future, we plan to extend the evaluation from frame retrieval to segment retrieval by investigating video temporal continuity as well. Additionally, efforts have been made in the study and development of indoor positioning systems, *e.g.*, the Redpin project (<http://redpin.org>). We are interested in integrating such capabilities into our system to allow it to work both outdoors and indoors.

CHAPTER 7

Hybrid Video Similarity Search

7.1 Introduction

This work extends the keyword-based landmark retrieval studied in the previous chapter to a more general problem of precise example-based video similarity search. Query-by-example based video search refers to the automatic retrieval of video segments that are similar to a user-provided example from the video database. Considering the limited descriptiveness of textual annotations, example-based content-level processing of multimedia documents has recently been popular in image and video retrieval literature [50]. Unfortunately, content-based methods suffer from the semantic gap [51] that hinders an accurate discovery of video content of interest. To solve this issue, geographic contextual modeling has been investigated recently. Methods have been proposed to judge the relevance of documents based on the textual and

spatial similarity with a query [62]. In multimedia, most previous work fuses visual content and geo-context to facilitate image management, whereas little effort has been placed on video retrieval. For example, image location information is widely applied for geo-clustering in landmark mining [56, 26], or to create a conjunctive ranking in image annotation and retrieval [59, 54]. Such approaches cannot make full use of the geographic information since in most cases only the camera location is incorporated. For video, in most of the current geo-referenced retrieval systems [8, 7, 58], clips are ranked purely based on their spatial relevance to the geospatial queries. In this study, we focus on sensor-rich videos with fine granularity spatial data. Since such geographic properties are automatically recorded using a built-in GPS and compass, we use outdoor videos where the sensor readings are more accurate. We leverage the geographic metadata of videos to improve the performance of text-based and content-based video retrieval techniques. More robust and diverse semantic annotations and similarity search results can be obtained by applying multi-feature fusion.

One issue of the previous fusion approaches is that they utilize the camera location directly. However, such information only captures the camera properties (*e.g.*, photographer location in some street in Paris) rather than the video content (*e.g.*, the Eiffel Tower). This inconsistency motivated us to propose a new content-oriented geo-feature to facilitate video search. The key components of the approach are illustrated in Figure 7.1: the *Hybrid Model Generation* (see Section 7.2) and the *Geo-Codebook Generation* (see Section 7.3). In the *Hybrid Model Generation* module, a two-level hierarchical model is introduced where multiple cues collaboratively contribute to the video representation. At level one, we generate a geo-histogram which represents the regions that a video covers based on the camera’s field-of-view and a pre-defined geo-codebook.

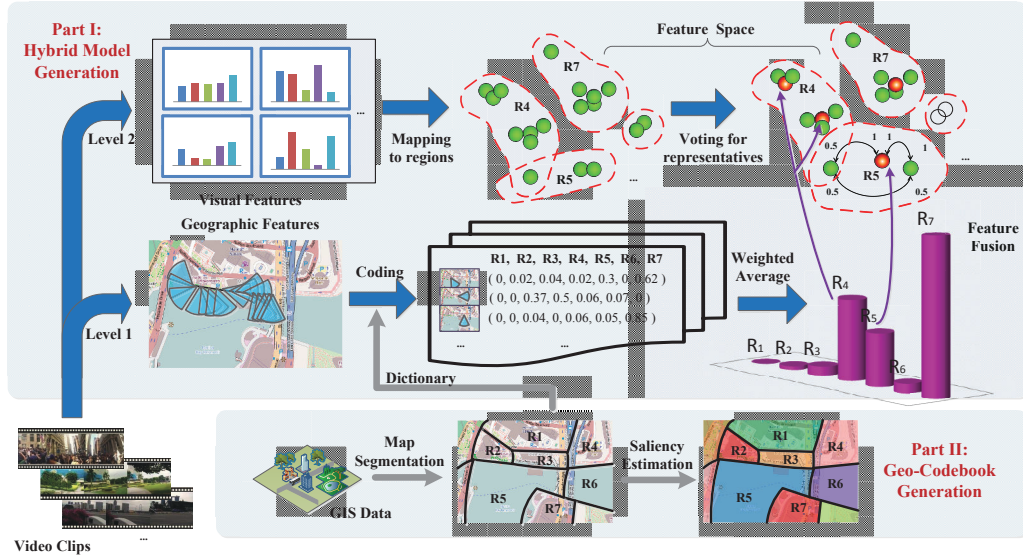


Figure 7.1: Illustration of key techniques for geographic and visual feature fusion in our proposed video retrieval system.

Different from the earlier viewable scene model [8] which focuses on individual frames, our proposed model describes the overall geographic coverage of a video. It enables the estimation of spatial relevance between videos through the cosine similarity between the two corresponding geo-histograms. At level two, we map frames to the regions they capture and select the visually representative ones. Note that in our model, frames are indexed by the regions they capture instead of the camera location. By doing so, geo and visual features are directly connected via regions. Thereafter, we propose a video similarity measure which sums up local similarity scores on a region-by-region basis. Next, toward a better encoding of the geographic coverage in the hybrid model, we present the *Geo-Codebook Generation* module. In this component, we propose an approach that can semantically segment a map into a collection of coherent regions as a geo-codebook. We further quantify the saliency of each region, as humans perceive geographic objects in different areas differently, *e.g.*, a building is more

likely to be of interest than a road. Finally, we built a video retrieval system based on the proposed model. Its effectiveness is shown through a performance comparison with existing methods.

7.2 Hybrid Model for Video Representation

While the viewable scene model [8] has been adopted for many geo-referenced video applications [7, 134, 99], one fundamental issue is it describes the camera properties rather than the video content. We argue that content-oriented geo features are highly desired because their consistency with visual clues can make the fusion more seamlessly. As illustrated in Figure 7.2, we propose a novel two-layer model in which frames are indexed by the regions they capture instead of the camera location. Therefore, geo and visual features are directly connected via regions.

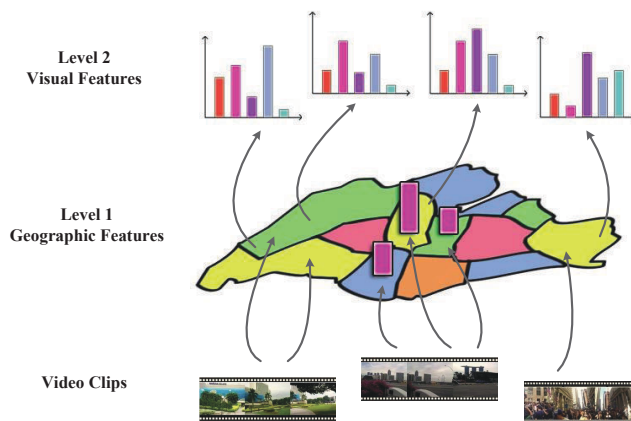


Figure 7.2: Illustration of the proposed hybrid model for video representation.

On the first level, this model computes the overall geographic coverage of a video instead of emphasizing individual frames for efficient spatial relevance measure. On the second level, it indexes frames by regions and selects a number

of representative ones based on the visual cues. In the rest of this section, we will first introduce the feature modeling of the proposed two-level video representation, then present a robust video similarity measure based on which more accurate search results can be retrieved.

7.2.1 L1: Geographic Coverage Calculation

As introduced earlier, level one aims to capture the overall geographic coverage of a video. To achieve this goal, we pre-segment a map into a set of regions with different saliency values. This is used as a geo-codebook to encode the geo-coverage of a video. The approaches for map segmentation and saliency estimation will be discussed in the next section. The geographic metadata is described by the viewable scene model proposed by Arslan Ay *et al.* [8] (referred to as *FOVScene*). Figure 7.3 illustrates the 2-dimensional $FOVScene(P, \vec{d}, \theta, R)$ model which is the 2D version of Figure 3.1. As we can see in this figure, the *FOVScene* overlaps with a geographic region, where *ol* represents the overlap, P^c denotes the centroid of overlap *ol*, and \vec{d}^c is the vector pointing from point P to P^c . These are the important concepts that will be used in the geo-coverage calculation.

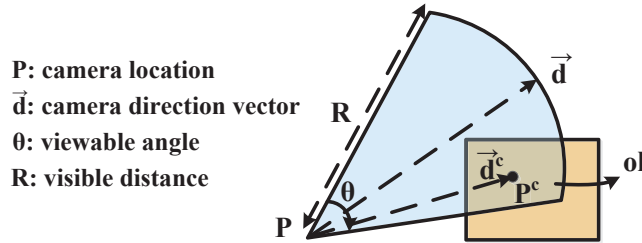


Figure 7.3: Illustration of *FOVScene* model in 2D.

To quantify what portion of a region is covered by a frame, we compute the overlap between the camera’s *FOVScene* and the regions in the geo-codebook

and use the overlap area to emphasize their spatial relevance [7]. As research indicates that people tend to focus on the center of an image [53], we prioritize regions that are close to the camera location and viewing direction [99, 134]. Let ol_{ij} denote the overlap between region r_i and frame f_j . We assign weights to the regions based on the following three criteria,

- *Normalized area of the overlap:* Considering the regions differ in size, we normalize the area of the overlap $A(ol_{ij})$ by the area of the region $A(r_i)$, that is $\hat{A}(ol_{ij}) = A(ol_{ij})/A(r_i)$.
- *Closeness to the camera location:* We compute the Euclidean distance $D(P_{ij}^c, P_j)$ between the overlap geometry center P_{ij}^c and the camera location P_j , and formulate this criterion as $\frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{D(P_{ij}^c, P_j)^2}{2\sigma^2})$.
- *Closeness to the viewing direction:* Let \vec{d}_{ij}^c denote the vector pointing from the camera location P_j to the overlap centroid P_{ij}^c . We compute the angular distance $D_\theta(\vec{d}_{ij}^c, \vec{d}_j)$ between vector \vec{d}_{ij}^c and the camera direction \vec{d}_j , and formulate this criterion as $\frac{1}{\sqrt{2\pi}\sigma_\theta} \exp(-\frac{D_\theta(\vec{d}_{ij}^c, \vec{d}_j)^2}{2\sigma_\theta^2})$.

Consequently, we compute the weight for region r_i captured in frame f_j using Eq. 7.1 given below.

$$hist_i^{geo}(f_j) = \frac{K_{\sigma, \sigma_\theta}(D(P_{ij}^c, P_j), D_\theta(\vec{d}_{ij}^c, \vec{d}_j)) \hat{A}(ol_{ij})}{\sum_k K_{\sigma, \sigma_\theta}(D(P_{kj}^c, P_j), D_\theta(\vec{d}_{kj}^c, \vec{d}_j)) \hat{A}(ol_{kj})} \quad (7.1)$$

where $K_{\sigma, \sigma_\theta}(d, d_\theta) = \frac{1}{2\pi\sigma\sigma_\theta} \exp\left(-\frac{1}{2}\left(\frac{d^2}{\sigma^2} + \frac{d_\theta^2}{\sigma_\theta^2}\right)\right)$. As a frame can cover multiple regions, the denominator is a factor that normalizes the sum of the region weights to one.

Subsequently, the histogram for video geo-coverage is calculated as the sum of $hist_i^{geo}(f_j)$ using Eq. 7.2. Since the video segments showing regions with a higher saliency value are more likely to be perceived by humans, we weight the histogram entries by the corresponding region saliency values $saliency(r_i)$, that is:

$$hist_i^{geo}(v) = saliency(r_i) \sum_{f_j \in v} hist_i^{geo}(f_j) \quad (7.2)$$

Finally, we normalize $hist^{geo}(v)$ by its Euclidean norm:

$$\hat{hist}_i^{geo}(v) = \frac{hist_i^{geo}(v)}{\|hist^{geo}(v)\|_2} \quad (7.3)$$

Now the geospatial relevance between videos can be efficiently measured as the cosine similarity between the generated geo-histograms, which quantifies the common areas covered by both of the videos:

$$S_g(v_i, v_j) = \sum_k \hat{hist}_k^{geo}(v_i) \hat{hist}_k^{geo}(v_j) \quad (7.4)$$

Note that for videos where only the GPS is available in the geo-metadata, it is possible to relax the direction criterion when generating the geo-histograms. We define the geographic area covered by such a frame to be a circle region centered at it with a radius of r . Therefore, Eq. 7.1 can be reduced to:

$$hist_i^{geo}(f_j) = \frac{K_\sigma(D(P_{ij}^c, P_j)) \hat{A}(ol_{ij})}{\sum_k K_\sigma(D(P_{kj}^c, P_j)) \hat{A}(ol_{kj})} \quad (7.5)$$

The regions are weighted based on the first two criteria which are $\hat{A}(ol_i)$ and $K_\sigma(D(P_{ij}^c, P_j)) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{D(P_{ij}^c, P_j)^2}{2\sigma^2})$.

7.2.2 L2: Representative Visual Features Selection

On the second level, visual features are extracted as the complementary information. In general, it is insufficient to measure video similarity purely based on the common geo-areas covered by both videos because (1) occlusions can occur due to moving objects such as people and vehicles, and (2) the geo-histogram generated on the first level is susceptible to sensor accuracy. Therefore, it is highly desired to compare visual features for a more robust similarity measure. The traditional content-based video similarity measure is mostly based on pairwise keyframe distances. Comparatively, with the prior knowledge of camera location and viewing direction, we can geographically index the frames of a geo-referenced video based on the regions they capture, and compute the local visual similarities in each region.

Since a large number of video frames are near-duplicate, it is necessary to cluster the frames and select the representative ones in each region. We build upon an effective lightweight clustering technique called the *reciprocal election approach* proposed by van Leuken *et al.* [115]. The key idea is to let every frame vote for the others. We make adaptations to the voting function to incorporate the frame geo-features. In a video v , let $F = \{f_1, f_2, \dots, f_n\}$ denote the set of frames of v that capture a same region r_i . For each frame f_j in F , we rank the others based on their visual similarities to f_j . Particularly, the visual similarity between frames is computed using Eq. 7.6.

$$W(f_i, f_j) = \exp\left(-\frac{\|f_i - f_j\|_2^2}{\sigma^2}\right) \quad (7.6)$$

Let f denote the k -th nearest neighbor of f_j . The vote f receives from f_j is defined to be $vote(f_j) = hist_i^{geo}(f_j)/k$. A smaller k indicates that the

two frames are highly similar and f is a good representative for f_j . A larger $hist_i^{geo}(f_j)$ indicates that f_j is highly relevant to region r_i and it is a salient frame in set F . Subsequently, the total votes f receives from the others is $\sum_j vote(f_j)$ where f_j is a frame in set F other than f .

After all the frames have cast their votes, the frame with the highest number of votes is selected as the first representative. The cluster around it is formed by those frames whose visual similarity to it exceeds a pre-defined threshold. Next, we exclude the first representative and its cluster members, and select the frame with the highest number of votes in the remaining set as the second representative. This process repeats until the percentage of the remaining frames is less than a threshold (0.05).

As the appearance of a region can change among videos, the visual similarity of a region's appearances can be measured based on its representative sets in different videos. To promote visually similar ones in ranking, we present an approach to fuse video spatial relevance with region visual similarity in the following section.

7.2.3 Video Similarity Measure

As introduced earlier, the proposed video representation transforms the original per-frame features into per-region features (spatial weight and visual representatives). Subsequently, video similarity can be decomposed as the sum of region feature similarities. As an example, Figure 7.4 shows two video clips A and B where a same region, the Marina Bay Sands hotel circled in red, is captured. Recall that the spatial relevance between two videos can be measured as the cosine similarity between the geo-histograms: $\sum_k \hat{hist}_k^{geo}(v_i) \hat{hist}_k^{geo}(v_j)$, that is

$0.88 \times 0.6 = 0.528$ between A and B. One issue arises in this case if we measure their similarity purely according to the spatial relevance. That is, the Marina Bay Sands hotel is occluded by trees in B, resulting in a low visual similarity score of 0.43. Furthermore, though the frames circled in blue and yellow show different regions, interestingly they happen to be visually similar.

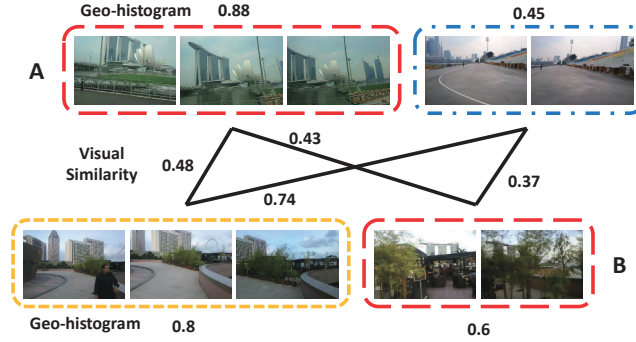


Figure 7.4: An example of similarity calculation between two videos.

Without loss of generality, let $w_k^{vis}(v_i, v_j)$ represent the local visual similarity between videos v_i and v_j in terms of region r_k . A small $w_k^{vis}(v_i, v_j)$ indicates that the region's appearances in the two videos are dissimilar, which is possibly caused by unpredictable occlusions, geo-metadata errors, or changes in illumination and viewpoints. Based on this observation, we penalize such situations by Eq. 7.7:

$$Sim(v_i, v_j) = \sum_k w_k^{vis}(v_i, v_j) \hat{hist}_k^{geo}(v_i) \hat{hist}_k^{geo}(v_j) \quad (7.7)$$

Note that $w_k^{vis}(v_i, v_j)$ can be computed by any existing visual similarity measure [23, 24, 98]. The proposed mechanism conjunctively leverages the geographic coverage similarity and the visual content similarity. $w_k^{vis}(v_i, v_j)$ controls the impact of visual features on the similarity calculation. If $w_k^{vis}(v_i, v_j)$ is set to one under all circumstances, Eq. 7.7 would become a histogram-based

approach which is similar to the one proposed by Arslan Ay *et al.* [7]. The difference is that their approach measures the spatial relevance between a video and a region query, whereas ours focuses on measuring the similarity between two videos. Such methods have the advantages of being highly efficient as the computation is only based on the geographic metadata, but without visual features its performance can degrade due to obstacles and occlusions.

On the other hand, the average size of the regions in the geo-codebook controls the impact of geographic features on the similarity calculation. Assume that there is only one region in the geo-codebook which is the entire globe, Eq. 7.7 would reduce to one of the existing visual-based similarity measures. In general, better precision can be achieved by using a geo-codebook with a finer granularity, as it arranges frames in smaller groups where the visual semantics are more explicit. But considering the errors in GPS and compass readings, a geo-codebook whose granularity is compatible with the size of the *FOVScene* model should be used.

In summary, the proposed model enables efficient spatial relevance calculations between videos as a dot-product of the geo-histograms on the first level and fuses visual clues to promote visually similar ones on the second level. By applying the geographic indexing of frames, our model not only reduces the computational costs, but also excludes noise that exists due to the mismatch between frames from different regions.

7.3 Geo-Codebook Generation

The geo-codebook is a key component in the hybrid model generation. Perhaps the simplest way to generate a geo-codebook is to use a grid-based map. How-

ever, a grid-based codebook suffers from two drawbacks as shown in Figure 7.5. First, geographic objects (*e.g.*, A, B and C) naturally differ in granularity while grid cells are equal-sized. Second, an object (*e.g.*, C) can be separated into multiple cells even if it is smaller than the cell size.



Figure 7.5: Limitations of a grid-based codebook that cannot satisfactorily capture the diverse granularity of geographic objects.

To solve the above two problems, we propose to construct a geo-codebook by a set of coherent regions that cover the map with no gaps or overlaps. There are several approaches that can discover the geographic coherent regions by investigating large image collections [110, 49]. However, such techniques cannot be applied for the geo-codebook generation because: (1) The regions discovered are usually not a full coverage of the map, and (2) the granularity of the generated regions is usually too coarse. Alternatively, geo-information services, *e.g.*, OpenStreetMap (OSM), provide information of the geographic objects all over the world. Compared with social image collections, this data source is more detailed and precise based on which a reliable geo-codebook can be generated.

7.3.1 Problem Formulation

For a geographic area, we first partition it into a set of square grid cells. Let $G = \{g_i | i = 1, 2, \dots, m \times n\}$ denote the set of cells, where m and n represent the number of rows and columns, respectively. Next we retrieve the information of geographic objects in each cell from OSM. Let $O^i = \{o_1^i, o_2^i, \dots, o_k^i\}$ represent the object set of grid cell g_i , where k is the total number of objects in it. Each object is represented by a quintuple, $o = \{id, name, tags, footprint, height\}$. A graph $G = (V, E)$ is constructed where the nodes V are grid cells and the edges E are weighted by node similarities. Thereby, the geo-codebook generation can be modeled as a graph clustering problem where each cluster represents a coherent region.

7.3.2 Clustering Cells into Coherent Regions

Based on the observation that adjacent similar cells should be merged into the same coherent region, we model the edges in graph G according to the following two criteria, the distance and the similarity between cells, in Eq. 7.8.

$$e_{ij} = K_\sigma(D(g_i, g_j)) \cdot S(g_i, g_j) \quad (7.8)$$

where $K_\sigma(d) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{d^2}{2\sigma^2})$; $D(g_i, g_j)$ and $S(g_i, g_j)$ denote the distance and the similarity between grid cells g_i and g_j , respectively.

Intuitively, cells should more likely be merged if they contain one or more common geographic objects. Therefore, we compute $S(g_i, g_j)$ based on the semantic similarity of the geographic objects in them. Recall that the geographic object set in cell g_i is $O^i = \{o_1^i, o_2^i, \dots, o_k^i\}$. Further, we assign a weight

to each object by measuring the percentage of area it occupies in cell g_i , *i.e.*, $P = \{p_1^i, p_2^i, \dots, p_k^i\}$. Thereafter, similarity $S(g_i, g_j)$ is computed as the weighted sum of the pairwise similarity of the geographic objects in grid cells g_i and g_j :

$$S(g_i, g_j) = \sum_{v,w} p_v^i p_w^j S(o_v^i, o_w^j) \quad (7.9)$$

Recently, Ballatore *et al.* proposed a mechanism to compute the semantic similarity of the OSM geographic classes [13]. They extracted a semantic network from the OSM Wiki website, and computed the tag-to-tag similarity score based on the network topology. As each geographic object can be assigned with multiple tags in OSM, we extend their approach to measure the object-to-object similarity by averaging the corresponding tag-to-tag similarities:

$$S(o_i, o_j) = \begin{cases} 1 & \text{if } o_i.id = o_j.id \\ \bar{S}(t^i, t^j) & \text{else} \end{cases} \quad (7.10)$$

where t^i and t^j are tags attached with objects o_i and o_j , and $\bar{S}(t^i, t^j)$ denotes the average value of the pairwise tag similarities.

After the graph is constructed, we adopt an effective clustering approach called *Newman and Girvan's Algorithm* [85]. This algorithm avoids the shortcomings of the traditional hierarchical clustering methods by detecting cluster peripheries instead of finding the strongly connected cores. Additionally, it provides a quality measurement called *modularity* which is more effective than empirically chosen thresholds. One issue is that finding a maximum-modularity clustering of a graph is computationally intractable. In our system, we utilized a Java implementation from the project *linloglayout*¹ which used an effective

¹<https://code.google.com/p/linloglayout/>

heuristic algorithm for modularity maximization.

Based on the above discussion, semantically coherent regions are obtained, resulting in a descriptive geo-codebook. Therefore, the features encoded in the hybrid model are more explicit and interpretable, leading to a better similarity estimation.

7.3.3 Region Saliency Estimation

As aforementioned, the importance of buildings and other geographic objects varies significantly in different areas. For example, landmarks are usually more attractive than ordinary buildings. Therefore, it is necessary to score the regions in the geo-codebook, based on which important objects appearing in a video can get emphasized in the video representation. *Visual saliency* and *social saliency* [102] complement with each other in attractiveness estimation. Here we estimate the region saliency according to these two criteria as follows.

Visual Saliency: Higher objects are more likely to draw the attention of the human eye, *e.g.*, a building is more likely to be of interest than a road. Based on this observation, we formulate this criterion as $VS(r) = \sum_i \{p_i \times o_i.height\}$, where o_i represents a geographic object in region r and p_i is the percentage of area covered by o_i in r .

Social Saliency: This criterion measures the impact of social factors on a region. We collect a set of geotagged images from Flickr, and compute the score for this criterion as $SS(r) = \sum_i K_\sigma(d_i)$, where d_i is the distance between the region center and the location of the i -th image. It can be viewed as the sum of image counts weighted by a Gaussian kernel based on distance.

To combine the above two criteria, the saliency of region r is calculated as

$saliency(r) = VS(r) + \lambda SS(r)$ where λ is a scaling factor. Recall that in the geo-coverage calculation, geo-histogram entries are weighted by region saliency scores. Therefore, our proposed hybrid model is able to promote important regions that are more likely to be of interest in the video representation.

7.4 Evaluation

We implemented a video search prototype and evaluated its effectiveness. We proceed in two steps. The first part shows two examples of the geo-codebook generation. The second part evaluates the performance of the proposed model in video retrieval.

7.4.1 Experimental Setup

We evaluated our proposed approach on the geo-referenced video dataset from the GeoVid website. Additionally, a supplementary dataset comprising 15,616 geotagged images was collected from Flickr by performing keyword-based search. Two types of tags were used as the query keywords: (1) the textual information of the geographic objects and (2) 25 popular concepts including airport, animal, birds, boat, bridge, buildings, cityscape, clouds, college, crowd, dancing, flowers, food, garden, grass, lake, person, plants, sky, street, sunset, temple, tree, vehicle, and water. This image dataset was used in the region saliency estimation.

For each of the frames and images, we extracted the following three low-level visual features in our experiments:

- *48-D Gabor Wavelet Texture*: Texture features extracted at four scales and

six orientations using a Gabor wavelet decomposition. [82]

- *225-D Block-Wise Color Moments*: The first (mean), the second (variance) and the third order (skewness) color moments in HSV space extracted over 5×5 fixed grid partitions. [108]
- *512-D Gist Descriptor*: The spatial structure of an image described by global features derived from the spatial envelope. [87]

These features are used for visual similarity measurement.

7.4.2 Geo-Codebook Generation

In our implementation, the geographic information of objects was collected from the OpenStreetMap. We recorded the name, the tags, and the footprint of each object. However, for buildings described in the OSM, interestingly the height attribute is sometimes not available. To solve this problem, we collected the building heights from EMPORIS², a real estate data mining company collecting and publishing data and photographs of buildings worldwide. In Singapore for example, it has records of 6,915 buildings, 321 of which have the height information. For the rest where the height of the building is not available, we estimate based on other clues, *e.g.*, the number of storeys.

Figure 7.6 presents examples of the generated geo-codebook in four different areas, namely Singapore, Chicago, Japan, and Hong Kong. The cell length of the grids was set to 50 m and the parameter σ in Eq. 7.8 was set to 65 m ($\sigma = 1.3 \times 50 = 65$ m). This parameter σ controls the connections between adjacent cells. If a large value is used, a cell will be bonded with its neighbors more tightly and therefore result in a coarse geo-codebook. Conversely,

²<http://www.emporis.com/>

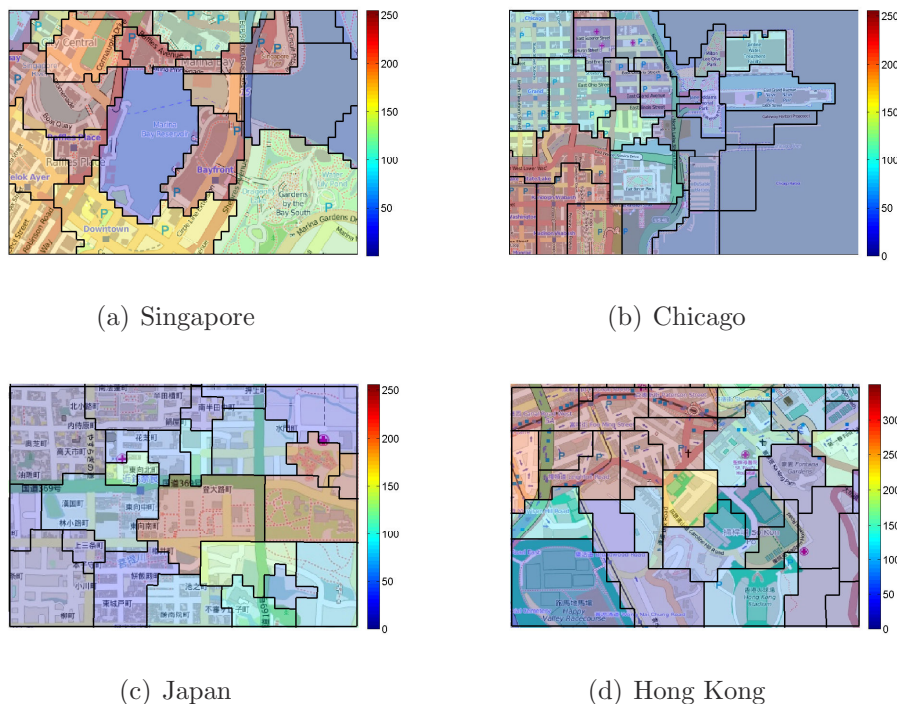


Figure 7.6: Examples of the generated geo-codebook in different areas around the world.

a small value of σ will result in a fine-grained geo-codebook. Additionally in Figure 7.6, the colors indicate the estimated saliency for each region and the scaling parameter λ was empirically set to 0.4. Compared with the grid-based codebook in Figure 7.5, we can see that this model successfully captures the diverse granularity of different geographic objects, and the estimated saliency is also consistent with human perception. Let us take Figure 7.6(a) as an example since it shows the same area as in Figure 7.5. In the center of the picture, we can see that the shape of Marina Bay (Object A in Figure 7.5) is well captured by the geo-codebook. The building on its right (Object B in Figure 7.5) is the most famous Marina Bay Sands hotel. Other salient regions marked in red are mainly the popular landmarks including the Singapore Flyer, the Esplanade, the Singapore River, and the financial district. In Figure 7.6(b), the salient

regions belong to the Loop which is the central business district of Chicago. In Figures 7.6(c) and 7.6(d), the salient regions are the Kofukuji Temple and the Time Square (Hong Kong), respectively.

Note that the current geo-codebook was generated within a city. For large-scale video datasets, our method can be easily scaled up by using a hierarchy: (1) segment the Earth surface into countries and cities, (2) generate geo-codebooks within cities, and (3) index videos using the generated geo-codebooks in various cities.

7.4.3 Evaluation on Video Retrieval

To evaluate the effectiveness of our proposed model in video retrieval, we collected a total of 423 videos throughout the world, ranging from 21 to 523 s in duration. The videos were further segmented into 1,656 shots, each of which are about 30 s in duration. Furthermore, we selected 30 video clips and 10 Flickr images (see Figure 7.7) as queries. The selection criterion is that they contain some recognizable places and landmarks which are more likely to be of interest.

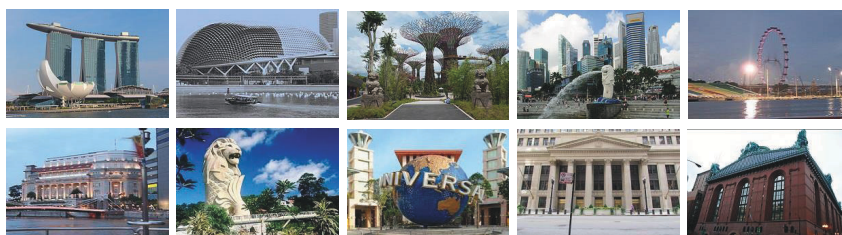


Figure 7.7: Ten geotagged Flickr images used as queries.

In our implementation, we adopt the method proposed by Cheung *et al.* [23] to measure the distance based on visual clues. Thereby, $w_k^{vis}(v_i, v_j)$ in Eq. 7.7

is computed as:

$$w_k^{vis}(v_i, v_j) = \exp\left(-\frac{D_k^{vis}(v_i, v_j)}{\sigma}\right) \quad (7.11)$$

$$D_k^{vis}(v_i, v_j) = \frac{\sum_{f_v \in \mathcal{R}_k(v_i)} \left(\min_{f_w \in \mathcal{R}_k(v_j)} \|f_v - f_w\|_2\right)}{|\mathcal{R}_k(v_i)| + |\mathcal{R}_k(v_j)|} + \frac{\sum_{f_w \in \mathcal{R}_k(v_j)} \left(\min_{f_v \in \mathcal{R}_k(v_i)} \|f_v - f_w\|_2\right)}{|\mathcal{R}_k(v_i)| + |\mathcal{R}_k(v_j)|} \quad (7.12)$$

where $\mathcal{R}_k(v)$ denotes the set of representative frames of region r_k in video v , $|\mathcal{R}_k(v)|$ represents its size, and $D_k^{vis}(v_i, v_j)$ is the visual distance between the two sets of frames, $\mathcal{R}_k(v_i)$ and $\mathcal{R}_k(v_j)$. As we can see, we first compute the local visual distance $D_k^{vis}(v_i, v_j)$ as the average distance between the closest matched frames using Cheung *et al.*'s method [23]. Then, we use a Gaussian kernel to acquire the local visual similarity score, which is $w_k^{vis}(v_i, v_j)$.

For the hybrid model generation, we empirically set $\sigma = \frac{R}{3}$ and $\sigma_\theta = \frac{\theta}{6}$ in Eq. 7.1, where R and θ denote the visible distance and the viewable angle of the *FOVScene* model illustrated in Figure 7.3.

Effectiveness comparison

To evaluate the effectiveness of our proposed region-aware video similarity measure, here we compared the following six methods and reported the results:

- *GEO*: It ranks videos based on the geospatial relevance using Eq. 7.4.
- *CRLF*: It filters the collection based on location, and then ranks the remaining based on visual similarity [86].
- *CRGV*: It ranks the collection based on a conjunctive function using both geographic distance and visual distance [54].

- *RASM*: It ranks videos based on the proposed region-aware similarity measure using Eq. 7.7.
- *OB*: A visual approach based on the state-of-the-art ObjectBank image descriptor. It represents an image based on its response to a large number of pre-trained object detectors [68].
- *BoS*: A visual approach based on the state-of-the-art Bag-of-Scene video representation. It generates a compact descriptor based on a dictionary of scenes, each of which represents a semantic concept [91].

As the existing work [86, 54] built their model using only GPS, to make it a fair comparison we generated the geo-histograms using Eq. 7.5 in this experiment. Later we will discuss how the performance can be further improved when camera direction is also available in the geo-metadata. For each of the queries, we examined the results and plotted the average precision at n ($P@n$) in Figure 7.8. We also compared the methods based on the Mean Average Precision (MAP) measure which is reported in Table 7.1.

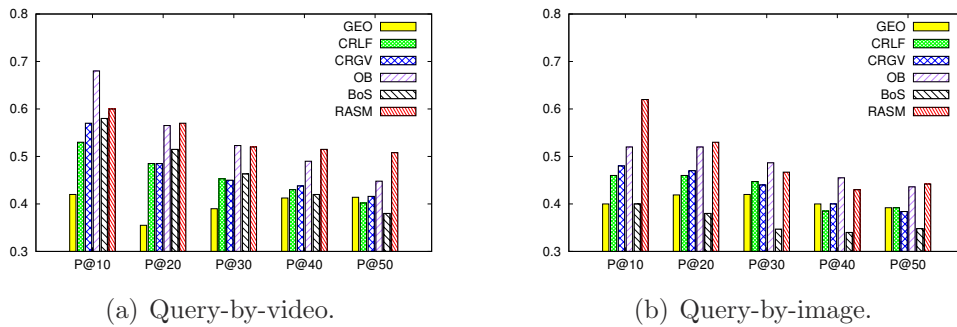


Figure 7.8: $P@n$ comparison of the proposed and the existing fusion methods.

GEO serves as a baseline method because it ranks videos based only on the geo-metadata. *CRLF* and *CRGV* outperformed the baseline method by integrating the visual clues. One issue is that these fusion approaches utilized the

Table 7.1: MAP comparison of the proposed and the existing fusion methods.

<i>Method</i>	<i>GEO</i>	<i>CRLF</i>	<i>CRGV</i>	<i>OB</i>	<i>BoS</i>	<i>RASM</i>
By-video	39.6%	40.6%	41.2%	44.6%	38.7%	49.2%
By-image	38.9%	39.7%	39.8%	41.7%	34.6%	44.2%

camera location directly. However, such information only captures the camera properties rather than the video content. This inconsistency between geo and visual features limited the effectiveness of such approaches. Additionally, we carried out experiments using the state-of-the-art visual features *OB* and *BoS* for comparison. *OB* is an object-level image descriptor which is generated based on pre-trained object detectors. It increased the MAP compared with methods *CRLF* and *CRGV* where the low-level visual features were adopted. However, due to the high dimensionality of the ObjectBank descriptor, it has the drawback of being time-consuming in feature extraction and similarity calculation. The time complexity of each method will be compared in Section 7.4.3. In contrast, *BoS* is a high-level compact video descriptor. In this experiment, we used a dictionary of 500 concept scenes and soft coding technique. The *BoS* descriptor represents a video segment using a single vector. Therefore, it is highly efficient in computing the similarity score between videos (see Table 7.6). However, it might be difficult to maintain a high MAP at the same time. As can be seen, our hybrid model *RASM* achieved the best results overall. It improved the MAP by 4.6% and 10.5% compared to *OB* and *BoS*. Our model generates the geo-coverage of a video which is a content-oriented geo-feature. Good performances can be achieved by fusing only with the low-level visual features. Moreover, our proposed model also works well with more advanced visual features such as *OB* and *BoS*. As reported in Table 7.2, our fusion technique can improve the MAP by as much as 7.7% compared with the original

content-based approaches.

Table 7.2: MAP comparison of fusion with OB and BoS.

<i>Method</i>	<i>OB</i>	<i>RASM_{OB}</i>	<i>BoS</i>	<i>RASM_{BoS}</i>
By-video	44.6%	51.9%	38.7%	46.4%
By-image	41.7%	48.4%	34.6%	42.1%

As a final point, our model can make use of multiple geo-features in the metadata, while how the camera direction can be utilized in other methods remains unknown.

Geo-metadata availability

Next, we studied how the retrieval performance varied when the geo-metadata was available at different levels. The comparison of average P@n is illustrated in Figure 7.9, and the MAP statistics are reported in Table 7.3. The subscript indicates which geo-metadata was used in the geo-coverage modeling.

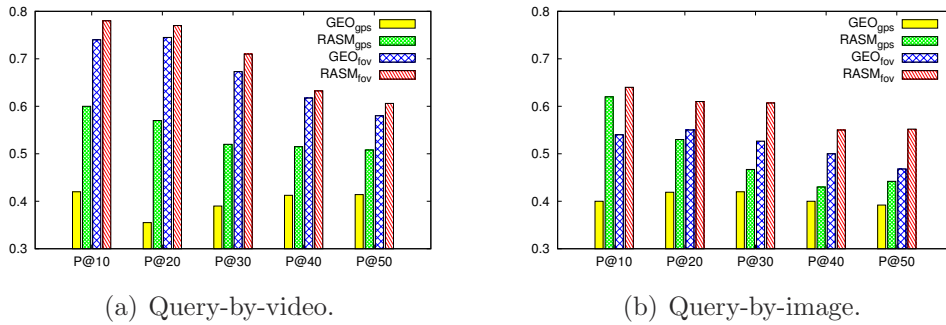


Figure 7.9: P@n comparison based on different availability of geo-metadata.

Table 7.3: MAP comparison based on different availability of geo-metadata.

<i>Method</i>	<i>GEO_{gps}</i>	<i>RASM_{gps}</i>	<i>GEO_{fov}</i>	<i>RASM_{fov}</i>
Query-by-video	39.6%	49.2%	66.9%	71.8%
Query-by-image	38.9%	44.2%	48.4%	53.2%

As can be seen, the effectiveness of both *GEO* and *RASM* was greatly improved by utilizing camera direction. It indicates the importance of camera orientation in video content analysis, but unfortunately compass record is still only available in the minority of multimedia documents. Such geo-restrictions can greatly help reduce the semantic gap between the low-level visual features and the high-level semantic concepts. For query-by-video, $RASM_{fov}$ improved the MAP by 22.6% compared to $RASM_{gps}$. For query-by-image, the increments were 9.0%. *RASM* is more robust than *GEO* because its similarity measure is more tolerant to dynamic obstacles and geo-metadata errors by analyzing the visual clues.

In terms of geo-metadata, social sharing platforms such as Flickr provide an accuracy level of geotags associated with photos. Therefore users can avoid using images with inaccurate geotags as queries. As pointed out by Hauff [41], the positional accuracy of the geotag information of Flickr images is highly dependent on the popularity of the venue. The average distance to the ground truth location is between 11 – 13 meters for images taken at popular venues, which is small compared to the size of the viewable scene model that we consider. Moreover, the good retrieval results shown in Figures 7.8 and 7.9 indicate that our method is robust within a certain range of geotag errors.

Step-By-Step Model Justification

The proposed video similarity measure includes two main components: geospatial relevance calculation and multi-feature fusion. To demonstrate the effectiveness of our proposed approach in each step, we replace our method by a functionally reduced counterpart and compare the corresponding retrieval performance.

- The geo-codebook generation is a key component in the first step. We use it to encode the geo-histograms, based on which the geospatial relevance between videos is computed. To illustrate its effectiveness, we replace it by a grid-based approach. Each region in the grid-based codebook is a square area that has a side length of 300 m.
- To justify the effectiveness of the region-aware fusion approach illustrated in Eq. 7.7, we compare it with the late fusion method [62]. The similarity is estimated as $S = \frac{1}{2}(S_g + S_v)$, where S_g and S_v denote the geospatial relevance and visual similarity, respectively. As shown in Figure 7.4, additional noise can be introduced by late fusion due to the mismatch between visual features from different regions.

Table 7.4: Mean average precision decrement.

<i>Query Type</i>	Query-by-video	Query-by-image
geo-codebook→grid map	-2.7%	-2.1%
region-aware→late fusion	-4.3%	-4.4%

As shown in Table 7.4, the MAP decreased when we replaced one component by an existing one. This demonstrates the effectiveness and the indispensability of our proposed approach.

System Efficiency

We performed the retrieval experiments on a desktop computer with a 3.20 GHz dual core CPU and 4 GB of main memory. The comparison of the execution time for feature extraction is reported in Table 7.5. For each query that we executed, we recorded the retrieval latency which includes the similarity calculation and the result ranking. The average value is reported in Table 7.6.

Table 7.5: The comparison of the execution time for feature extraction per image.

<i>Feature</i>	<i>GEO</i>	<i>Color</i>	<i>Texture</i>	<i>Gist</i>	<i>OB</i>	<i>BoS</i>
<i>Time</i>	0.01 ms	0.06 s	0.12 s	0.46 s	4.68 s	4.682 s

Table 7.6: The comparison of the average retrieval latency.

<i>Method</i>	<i>GEO</i>	<i>CRLF</i>	<i>CRGV</i>	<i>OB</i>	<i>BoS</i>	<i>RASM</i>
By-video	6 ms	512 ms	525 ms	927 ms	11 ms	295 ms
By-image	6 ms	83 ms	98 ms	185 ms	11 ms	64 ms

In comparison of the execution time for feature extraction, the encoding of the proposed geo-features is highly efficient as the calculation is only based on the camera location and orientation. In contrast, the time complexity for visual feature extraction is much higher. As can be seen, the low-level visual features such as color and texture would cost dozens of milliseconds for extraction, while the more descriptive ObjectBank representation would cost more than four seconds. *BoS* cost slightly more than *OB* as the former takes an extra step by soft encoding each frame to its nearest neighbors in the dictionary. Our proposed model can achieve high MAP while maintaining good efficiency. With the help of the proposed content-oriented geo-feature, effective retrieval performances can be achieved by using only the less descriptive low-level visual features, and thus the time complexity is greatly reduced.

As aforementioned, the videos in our system are indexed using inverted files based on the geographic regions. Therefore, only the geo-relevant videos are processed for similarity calculations. Method *GEO* is highly efficient because the high-dimensional visual features are not utilized in the similarity calculations. The visual approach *BoS* reduced the time complexity by generating a visual descriptor per video segment instead of per frame. Method *OB* is

the least efficient due to its high dimensionality compared with color, texture, and Gist used in other approaches. For hybrid approaches, the visual feature comparison is always the major cost for both storage and computation. Let \bar{n} denote the average number of keyframes in a video, then the complexity for the visual similarity calculation in *CRLF*, *CRGV*, and the late fusion approach will be $O(\bar{n}^2)$. Different from the above methods where a pairwise comparison between keyframes is required, our proposed approach *RASM* reduces the computational costs by geographic indexing where only the local visual similarities of each region are computed. If the keyframes of a video are divided into an average of \bar{k} region groups, the time complexity will be reduced to $O(\bar{n}^2/\bar{k})$. Comparatively, most of the previous work focused on the compact video representations that support efficient visual indexing [24, 98]. It is worth emphasizing that such techniques are parallel to our model, which can be integrated on the region-level after frames are geographically indexed. The geographic and the visual indexing complement with each other in a large video database. Considering the limitations on the availability of current geo-referenced videos, discussions of integrating efficient approximate visual similarity measure are left as part of the future work.

7.5 Summary

This work proposed the generation of content-oriented geo-features to facilitate video search. It does not focus on one specific visual similarity measure, rather it shows that the innovative fusion of visual and geo features provides improved performance over the existing approaches. A novel hybrid model is proposed as video representation, describing both the video geographic coverage

and the region-aware representative visual features. Additionally, we propose to construct a geo-codebook by utilizing the information available from the geo-information services to segment an area into a set of coherent regions. It overcomes the limitations of a grid-based codebook, based on which the geographic coverage of a video can be better encoded. Lastly, we developed a video retrieval prototype based on our proposed hybrid model. To evaluate its performance, we compared it to existing approaches. The results convincingly demonstrate the effectiveness of our proposed approaches.

Toward a more effective video retrieval system, we plan to improve several components of the proposed approach in the future. It will be interesting to study how to enrich the query types supported in the system (*e.g.*, search videos by a group of images). Additionally, more efforts will be made on the correction of geographic metadata and the acceleration of visual similarity calculations.

CHAPTER 8

Conclusions and Future Work

This dissertation studied the problem of geo-referenced video annotation and retrieval. The video dataset was recorded by smartphone applications developed by our group where the location and orientation of the camera are recorded along with the video streams. In order to benefit from the sensor metadata, we utilized the geographic information systems and services such as the OpenStreetMap (OSM). The OSM is a community based map application that can supply detailed information (*e.g.*, names, types, outlines) of numerous geographic objects. However, its completeness varies in different regions. Therefore, we alternatively leveraged social multimedia applications such as Flickr to enrich the vocabulary for video tagging. Moreover, hybrid methodologies by multi-feature fusion have been proposed to improve the effectiveness for video retrieval. Here we summarize our work as follows.

First, to reduce the noise and errors in the raw sensor data, we preprocessed

the geo-metadata by utilizing the geographic context derived from OSM. We built a comprehensive model to formulate the error terms, which incorporates smooth approximation, rotation estimation, pixel labeling and 3D projection. Earlier methods usually rely on the feature-based matching with 3D models reconstructed from large scale images. Comparatively, the feasibility of the proposed method has been greatly improved as we relaxed the preliminary data requisites that are required by Structure-from-Motion for robust feature matching.

Next, to enrich the vocabulary for the auto-annotation approach in our prior work, we showed how a positionable tag repository can be built based on social multimedia applications. To setup such a repository, we modeled the geographic distributions of tags by Gaussian mixture models. To judge whether a tag is positionable or not, we employed two features to build a classifier: (1) the number of peaks in the area-of-interest (AOI), and (2) the sum of the priors of the peaks in the AOI. Furthermore, we profiled their temporal distributions to determine their effective durations. Finally, we ranked the tag candidates based on their popularity and geographic bias.

Finally, we fused the geo and visual features to improve the retrieval effectiveness. For landmark search, we evaluated and compared two state-of-the-art video landmark retrieval paradigms, namely media-content based and geo-context based retrievals. From the comparison results we draw a number of interesting observations, based on which a hybrid integration of content and context was analyzed and shown to achieve significant improvements. For similarity search, we proposed a novel two-level video representation. At level one, we generated a content-oriented geo-histogram to describe the regions that a video captures. At level two, we generated region-aware visual features for

each entry of the geo-histogram. Based on this representation, a novel video similarity measure was proposed. Furthermore, toward a better encoding of the geographic coverage in the hybrid model, we presented an approach that segments a map into a collection of coherent regions which we refer to as a geo-codebook. Finally, we built a video retrieval prototype and demonstrated its effectiveness over existing approaches.

In the future, we plan to improve several components of the geo-tagged video retrieval prototype. First, we will enrich the query types supported in the system (*e.g.*, search videos by a group of images). Next, as currently the video frames are considered independently, we will investigate video temporal continuity for further improvements. Additionally, more efforts will be made on the correction of geographic metadata and the development of indoor positioning systems to allow it to work both outdoors and indoors.

Bibliography

- [1] *Electronic Statistics Textbook*. StatSoft, Inc, 2011. 65
- [2] YouTube Press, Statistics. <http://www.youtube.com/yt/press/statistics.html>, Apr. 2015. 1
- [3] G. Abdollahian and E. J. Delp. User Generated Video Annotation Using Geo-tagged Image Databases. In *IEEE ICME*, 2009. 17, 56
- [4] S. Ahern, M. Naaman, R. Nair, and J. Hui. World Explorer: Visualizing Aggregate Data from Unstructured Text in Geo-referenced Collections. In *ACM/IEEE-CS Joint Conference on Digital Libraries*, 2007. 19
- [5] G. Amato, F. Falchi, and F. Rabitti. Landmark Recognition in VISITO Tuscany. In *Multimedia for Cultural Heritage*, pages 1–13, 2012. 21
- [6] M. Ames and M. Naaman. Why We Tag: Motivations for Annotation in Mobile and Online Media. In *SIGCHI*, 2007. 55
- [7] S. Arslan Ay, R. Zimmermann, and S. Kim. Relevance Ranking in Georeferenced Video Search. *Multimedia Systems*, 16:105–125, 2010. vii, 4, 8, 20, 24, 88, 120, 122, 124, 129
- [8] S. Arslan Ay, R. Zimmermann, and S. H. Kim. Viewable Scene Modeling for Geospatial Video Search. In *ACM Multimedia*, pages 309–318, 2008. 2, 8, 14, 20, 24, 26, 28, 56, 89, 93, 120, 121, 122, 123

BIBLIOGRAPHY

- [9] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli. Multimodal Fusion for Multimedia Analysis: A Survey. *Multimedia Systems*, pages 345–379, 2010. [112](#)
- [10] Y. Avrithis, Y. Kalantidis, G. Toliás, and E. Spyrou. Retrieving Landmark and Non-landmark Images from Community Photo Collections. In *ACM Multimedia*, pages 153–162, 2010. [22](#), [88](#)
- [11] L. Ballan, M. Bertini, A. Del Bimbo, M. Meoni, and G. Serra. Tag Suggestion and Localization in User-generated Videos Based on Social Knowledge. In *Proceedings of Second ACM SIGMM Workshop on Social Media*, pages 3–8, 2010. [17](#), [56](#)
- [12] L. Ballan, M. Bertini, A. Del Bimbo, and G. Serra. Enriching and Localizing Semantic Tags in Internet Videos. In *ACM Multimedia*, pages 1541–1544, 2011. [56](#)
- [13] A. Ballatore, M. Bertolotto, and D. Wilson. Geographic Knowledge Extraction and Semantic Similarity in OpenStreetMap. *Knowledge and Information Systems*, pages 61–81, 2013. [132](#)
- [14] S. Bennett, J. Lasenby, A. Kokaram, S. Inguva, and N. Birkbeck. Reconstruction of the Pose of Uncalibrated Cameras via User-Generated Videos. In *International Conference on Distributed Smart Cameras*, pages 3:1–3:8, 2014. [39](#)
- [15] C. Brunson, A. Fotheringham, and M. Charlton. "geographically weighted summary statistics a framework for localised exploratory data analysis ". *Computers, Environment and Urban Systems*, pages 501 – 524, 2002. [18](#)
- [16] M. Campbell, A. Haubold, S. Ebadollahi, D. Joshi, M. R. Naphade, A. P. Natsev, J. Seidl, J. R. Smith, K. Scheinberg, J. Tešić, and L. Xie. IBM Research TRECVID-2006 Video Retrieval System. 2006. [23](#)
- [17] L. Cao and J. Krumm. From GPS Traces to a Routable Road Map. In *ACM SIGSPATIAL GIS*, pages 3–12, 2009. [5](#), [14](#), [34](#), [49](#)
- [18] L. Cao, Z. Li, Y. Mu, and S.-F. Chang. Submodular Video Hashing: A Unified Framework Towards Video Pooling and Indexing. In *ACM Multimedia*, pages 299–308, 2012. [25](#)
- [19] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2011. [75](#)

- [20] F. Chazal, D. Chen, L. Guibas, X. Jiang, and C. Sommer. Data-driven Trajectory Smoothing. In *ACM SIGSPATIAL GIS*, pages 251–260, 2011. [5](#), [14](#), [34](#)
- [21] D. Chen, G. Baatz, K. Koser, S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale Landmark Identification on Mobile Devices. In *IEEE CVPR*, pages 737–744, 2011. [22](#)
- [22] T. Chen, K.-H. Yap, and L.-P. Chau. Integrated Content and Context Analysis for Mobile Landmark Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1476–1486, 2011. [22](#)
- [23] S. Cheung and A. Zakhor. Efficient Video Similarity Measurement and Search. In *Image Processing.*, pages 85–88, 2000. [25](#), [128](#), [137](#), [138](#)
- [24] S. Cheung and A. Zakhor. Efficient Video Similarity Measurement with Video Signature. *Circuits and Systems for Video Technology, IEEE Transactions on*, pages 59–74, 2003. [128](#), [145](#)
- [25] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In *International Conference on Computer Vision*, pages 1–8, 2007. [21](#)
- [26] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the World’s Photos. In *ACM WWW*, pages 761–770, 2009. [24](#), [120](#)
- [27] N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, 2000. [67](#)
- [28] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual Categorization with Bags of Keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, pages 1–22, 2004. [7](#), [21](#), [87](#), [90](#)
- [29] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1977. [62](#), [65](#)
- [30] M. Ester, H.-p. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD*, 1996. [69](#)

BIBLIOGRAPHY

- [31] S. Feng and R. Manmatha. A Discrete Direct Retrieval Model for Image and Video Retrieval. In *International Conference on Content-based Image and Video Retrieval*, pages 427–436, 2008. [22](#)
- [32] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli Relevance Models for Image and Video Annotation. In *IEEE CVPR*, 2004. [16](#), [56](#)
- [33] Flickr. Flickr API. <https://www.flickr.com/services/api/>, 2013. [viii](#), [103](#)
- [34] Y. Gao, J. Tang, R. Hong, Q. Dai, T. S. Chua, and R. Jain. W2Go: A Travel Guidance System by Automatic Landmark Ranking. In *ACM Multimedia*, 2010. [19](#)
- [35] E. Gavves, C. G. Snoek, and A. W. Smeulders. Visual Synonyms for Landmark Image Retrieval. *Computer Vision and Image Understanding*, pages 238–249, 2012. [21](#)
- [36] Google. Google Maps. <https://maps.google.com/>, 2013. [viii](#), [ix](#), [2](#), [96](#), [107](#)
- [37] M. Haklay and P. Weber. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008. [30](#), [35](#)
- [38] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, pages 10–18, 2009. [75](#)
- [39] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, and F. Wu. 3D Visual Phrases for Landmark Recognition. In *IEEE CVPR*, pages 3594–3601, 2012. [21](#)
- [40] R. I. Hartley. Self-Calibration of Stationary Cameras. *International Journal of Computer Vision*, 22(1):5–23, 1997. [39](#)
- [41] C. Hauff. A Study on the Accuracy of Flickr’s Geotag Data. In *ACM SIGIR*, pages 1037–1040, 2013. [78](#), [142](#)
- [42] A. G. Hauptmann and M. G. Christel. Successful Approaches in the TREC Video Retrieval Evaluations. In *ACM Multimedia*, pages 668–675, 2004. [23](#)
- [43] E. Hecht. *Optics*. Addison-Wesley Publishing Company, 4 edition, 2001. [27](#), [94](#)
- [44] J. R. Hershey and P. A. Olsen. Approximating the Kullback Leibler Divergence between Gaussian Mixture Models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 317–320, 2007. [69](#)

- [45] N. Hoàng, V. Gouet-Brunet, M. Rukoz, and M. Manouvrier. "embedding spatial information into image content description for scene retrieval". *Pattern Recognition*, pages 3013 – 3024, 2010. [21](#)
- [46] R. Hong, J. Tang, H.-K. Tan, C.-W. Ngo, S. Yan, and T.-S. Chua. Beyond Search: Event-driven Summarization for Web Videos. *ACM Trans. Multimedia Comput. Commun. Appl.*, pages 35:1–35:18, 2011. [27](#)
- [47] B. K. P. Horn. Relative Orientation Revisited. *Journal of the Optical Society of America A*, 8:1630–1638, 1991. [15](#)
- [48] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video Search Reranking through Random Walk over Document-level Context Graph. In *ACM Multimedia*, pages 971–980, 2007. [23](#)
- [49] S. Intagorn and K. Lerman. Learning Boundaries of Vague Places from Noisy Annotations. In *ACM SIGSPATIAL GIS*, pages 425–428, 2011. [18](#), [130](#)
- [50] M. Jain, S. Vempati, C. Pulla, and C. V. Jawahar. Example Based Video Filters. In *ACM CIVR*, pages 24:1–24:8, 2009. [119](#)
- [51] R. Jain and P. Sinha. Content without Context is Meaningless. In *ACM Multimedia*, pages 1259–1268, 2010. [55](#), [88](#), [119](#)
- [52] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic Image Annotation and Retrieval Using Cross-media Relevance Models. In *ACM SIGIR*, pages 119–126, 2003. [17](#), [56](#)
- [53] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to Predict Where Humans Look. In *Computer Vision*, pages 2106–2113, 2009. [124](#)
- [54] J. Kamahara, T. Nagamatsu, and N. Tanaka. Conjunctive Ranking Function using Geographic Distance and Image Distance for Geotagged Image Retrieval. In *ACM GeoMM*, pages 9–14, 2012. [24](#), [120](#), [138](#), [139](#)
- [55] P. Kelm, S. Schmiedeke, and T. Sikora. A Hierarchical, Multi-modal Approach for Placing Videos on the Map Using Millions of Flickr Photographs. In *ACM workshop on Social and Behavioural Networked Media Access*, pages 15–20, 2011. [3](#), [20](#)

BIBLIOGRAPHY

- [56] L. S. Kennedy and M. Naaman. Generating Diverse and Representative Image Search Results for Landmarks. In *WWW*, pages 297–306, 2008. [24](#), [88](#), [120](#)
- [57] S. H. Kim, S. A. Ay, and R. Zimmermann. Design and Implementation of Geo-tagged Video Search Framework. *Visual Communication and Image Representation*, 21:773–786, 2010. [4](#)
- [58] Y. Kim, J. Kim, and H. Yu. GeoSearch: Georeferenced Video Retrieval System. In *ACM SIGKDD*, pages 1540–1543, 2012. [20](#), [24](#), [120](#)
- [59] J. Kleban, E. Moxley, J. Xu, and B. S. Manjunath. Global Annotation on Georeferenced Photographs. In *ACM Image and Video Retrieval*, pages 1–8, 2009. [120](#)
- [60] J. Kopf, M. F. Cohen, and R. Szeliski. First-person hyper-lapse videos. *ACM Trans. Graph.*, 33(4):78:1–78:10, 2014. [46](#)
- [61] M. Kroepfl, Y. Wexler, and E. Ofek. Efficiently Locating Photographs in Many Panoramas. In *ACM SIGSPATIAL GIS*, pages 119–128, 2010. [15](#), [16](#)
- [62] C. Kumar. Relevance and Ranking in Geographic Information Retrieval. In *BCS-IRSG FDIA*, pages 2–7, 2011. [120](#), [143](#)
- [63] Y.-H. Kuo, W.-H. Cheng, H.-T. Lin, and W. H. Hsu. Unsupervised Semantic Feature Discovery for Image Object Retrieval and Tag Refinement. *IEEE Transactions on Multimedia*, pages 1079–1090, 2012. [88](#)
- [64] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998. [47](#)
- [65] M. Larson, M. Eskevich, R. Ordelman, C. Kofler, S. Schmiedeke, and G. J. Jones. Overview of MediaEval 2011 Rich Speech Retrieval Task and Genre Tagging Task. In *Multimedia Benchmark Workshop*, 2011. [18](#)
- [66] M. Larson, M. Soleymani, P. Serdyukov, S. Rudinac, C. Wartena, V. Murdock, G. Friedland, R. Ordelman, and G. J. F. Jones. Automatic Tagging and Geotagging in Video Collections and Communities. In *ACM ICMR*, pages 51:1–51:8, 2011. [18](#)

- [67] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *IEEE CVPR*, pages 2169–2178, 2006. [21](#), [45](#), [88](#), [90](#)
- [68] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei. Object Bank: An Object-Level Image Representation for High-Level Visual Recognition. *International Journal of Computer Vision*, pages 20–39, 2014. [139](#)
- [69] X. Li, C. G. M. Snoek, M. Worring, and A. W. M. Smeulders. Fusing Concept Detection and Geo Context for Visual Search. In *ACM ICMR*, pages 4:1–4:8, 2012. [25](#)
- [70] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *ECCV*, pages 15–29, 2012. [15](#), [16](#), [34](#), [42](#)
- [71] Z. Li and K.-H. Yap. Content and Context Boosting for Mobile Landmark Recognition. *Signal Processing Letters*, pages 459–462, 2012. [22](#)
- [72] S. Liao, X. Li, X. Wang, and X. Du. Building Geo-aware Tag Features for Image Classification. In *IEEE ICME*, pages 1–6, 2014. [24](#)
- [73] H. Liu, T. Mei, J. Luo, H. Li, and S. Li. Finding Perfect Rendezvous on the Go: Accurate Mobile Visual Localization and Its Applications to Routing. In *ACM Multimedia*, pages 9–18, 2012. [15](#)
- [74] J. Liu, W. Lai, X.-S. Hua, Y. Huang, and S. Li. Video Search Re-ranking via Multi-graph Propagation. In *ACM Multimedia*, pages 208–217, 2007. [23](#)
- [75] J. Liu, B. Wang, M. Li, Z. Li, W. Ma, H. Lu, and S. Ma. Dual Cross-media Relevance Model for Image Annotation. In *ACM Multimedia*, pages 605–614, 2007. [17](#)
- [76] X. Liu, M. Corner, and P. Shenoy. SEVA: Sensor-enhanced Video Annotation. In *ACM Multimedia*, pages 618–627, 2005. [20](#)
- [77] D. Lowe. Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, pages 91–110, 2004. [21](#), [42](#)
- [78] X. Lu, C. Wang, J. M. Yang, Y. Pang, and L. Zhang. Photo2Trip: Generating Travel Routes from Geo-Tagged Photos for Trip Planning. In *ACM Multimedia*, 2010. [19](#), [34](#)

BIBLIOGRAPHY

- [79] J. Luo, D. Joshi, J. Yu, and A. Gallagher. Geotagging in Multimedia and Computer Vision—A Survey. *Multimedia Tools and Applications*, pages 187–211, 2011. [2](#)
- [80] Z. Luo, H. Li, J. Tang, R. Hong, and T.-S. Chua. ViewFocus: Explore Places of Interests on Google Maps Using Photos with View Direction Filtering. In *ACM Multimedia*, pages 963–964, 2009. [15](#), [16](#), [50](#)
- [81] Z. Luo, H. Li, J. Tang, R. Hong, and T.-S. Chua. Estimating Poses of World’s Photos with Geographic Metadata. In *Advances in Multimedia Modeling*, volume 5916, pages 695–700. 2010. [5](#), [15](#), [16](#), [34](#), [48](#), [50](#)
- [82] B. Manjunath and W. Ma. Texture Features for Browsing and Retrieval of Image Data. *IEEE Pattern Analysis and Machine Intelligence*, pages 837–842, 1996. [135](#)
- [83] F. Monay and D. G. Perez. On Image Auto-Annotation with Latent Space Models. In *ACM Multimedia*, 2003. [16](#), [17](#), [56](#)
- [84] E. Moxley, J. Kleban, and B. S. Manjunath. SpiritTagger: A Geo-Aware Tag Suggestion Tool Mined from Flickr. In *ACM MIR*, 2008. [17](#), [56](#)
- [85] M. E. Newman. Analysis of Weighted Networks. *Physical Review E*, page 056131, 2004. [132](#)
- [86] N. O’Hare, C. Gurrin, G. Jones, and A. Smeaton. Combination of Content Analysis and Context Features for Digital Photograph Retrieval. In *European Workshop on Integration of Knowledge, Semantics and Digital Media Technology.*, pages 323–328, 2005. [138](#), [139](#)
- [87] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *Int. J. Comput. Vision*, 42:145–175, 2001. [135](#)
- [88] C. Olsson and O. Enqvist. Stable Structure from Motion for Unordered Image Collections. In *Image Analysis*, volume 6688, pages 524–535. 2011. [35](#)
- [89] M. Park, J. Luo, R. T. Collins, and Y. Liu. Beyond GPS: Determining the Camera Viewing Direction of a Geotagged Image. In *ACM Multimedia*, pages 631–634, 2010. [5](#), [15](#), [16](#), [34](#), [48](#)

- [90] O. A. Penatti, F. B. Silva, E. Valle, V. Gouet-Brunet, and R. da S. Torres. "visual word spatial arrangement for image retrieval and classification". *Pattern Recognition*, pages 705 – 720, 2014. [21](#)
- [91] O. A. B. Penatti, L. T. Li, J. Almeida, and R. da S. Torres. A Visual Approach for Video Geocoding using Bag-of-Scenes. In *ACM ICMR*, pages 1–8, 2012. [22](#), [139](#)
- [92] D. Pollock. Smoothing with Cubic Splines. Department of Economics, Queen Mary and Westfield College, 1993. [40](#)
- [93] G. J. Qi, X. S. Hua, Y. Rui, J. Tang, T. Mei, and H. J. Zhang. Correlative Multi-label Video Annotation. In *ACM Multimedia*, 2007. [16](#)
- [94] A. Rae, V. Murdock, P. Serdyukov, and P. Kelm. Working Notes for the Placing Task at MediaEval 2011. 2011. [20](#)
- [95] T. Rattenbury, N. Good, and M. Naaman. Towards Automatic Extraction of Event and Place Semantics from Flickr Tags. In *ACM SIGIR*, 2007. [19](#)
- [96] T. Sattler, B. Leibe, and L. Kobbelt. Fast Image-based Localization using Direct 2D-to-3D Matching. In *ICCV*, pages 667–674, 2011. [15](#), [16](#), [34](#), [42](#)
- [97] P. Serdyukov, V. Murdock, and R. van Zwol. Placing Flickr Photos on a Map. In *ACM SIGIR*, pages 484–491, 2009. [3](#)
- [98] H. T. Shen, B. C. Ooi, and X. Zhou. Towards Effective Indexing for Very Large Video Sequence Database. In *ACM SIGMOD*, pages 730–741, 2005. [25](#), [128](#), [145](#)
- [99] Z. Shen, S. Arslan Ay, S. H. Kim, and R. Zimmermann. Automatic Tag Generation and Ranking for Sensor-rich Outdoor Videos. In *ACM Multimedia*, pages 93–102, 2011. [8](#), [20](#), [24](#), [28](#), [34](#), [56](#), [88](#), [94](#), [116](#), [122](#), [124](#)
- [100] S. Siersdorfer, J. San Pedro, and M. Sanderson. Automatic Video Tagging using Content Redundancy. In *ACM SIGIR*, 2009. [17](#), [56](#)
- [101] B. Sigurbjörnsson and R. Van Zwol. Flickr Tag Recommendation based on Collective Knowledge. In *ACM WWW*, 2008. [17](#), [56](#)

BIBLIOGRAPHY

- [102] T. H. Silva, P. O. S. Vaz de Melo, J. M. Almeida, J. Salles, and A. A. F. Loureiro. A Comparison of Foursquare and Instagram to the Study of City Dynamics and Urban Social Behavior. In *ACM SIGKDD UrbComp*, pages 1–8, 2013. [133](#)
- [103] R. Simon and P. Fröhlich. A Mobile Application Framework for the Geospatial Web. In *ACM WWW*, pages 381–390, 2007. [20](#)
- [104] S. Sizov. GeoFolk: Latent Spatial Semantics in Web 2.0 Social Media. In *ACM WSDM*, pages 281–290, 2010. [18](#)
- [105] N. Snavely, S. M. Seitz, and R. Szeliski. Photo Tourism: Exploring Photo Collections in 3D. In *ACM SIGGRAPH*, pages 835–846, 2006. [14](#), [15](#), [16](#)
- [106] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders. Early Versus Late Fusion in Semantic Video Analysis. In *ACM Multimedia*, pages 399–402, 2005. [112](#)
- [107] F. Souvannavong, B. Merialdo, and B. Huet. Region-based Video Content Indexing and Retrieval. In *International Workshop on Content-Based Multimedia Indexing*, pages 21–23, 2005. [22](#)
- [108] M. Stricker and M. Orengo. Similarity of Color Images. In *SPIE Storage and Retrieval for Image and Video Databases III*, pages 381–392, 1995. [135](#)
- [109] F. M. Suchanek, M. Vojnovic, and D. Gunawardena. Social Tags: Meaning and Suggestions. In *ACM CIKM*, 2008. [55](#)
- [110] B. Thomee and A. Rae. Uncovering Locally Characterizing Regions Within Geotagged Data. In *ACM WWW*, pages 1285–1296, 2013. [18](#), [130](#)
- [111] X. Tian, D. Tao, and Y. Rui. Sparse Transfer Learning for Interactive Video Search Reranking. *ACM Trans. Multimedia Comput. Commun. Appl.*, pages 26:1–26:19, 2012. [27](#)
- [112] X. Tian, L. Yang, J. Wang, Y. Yang, X. Wu, and X.-S. Hua. Bayesian Video Search Reranking. In *ACM Multimedia*, pages 131–140, 2008. [24](#)
- [113] J. Tighe and S. Lazebnik. Superparsing: Scalable Nonparametric Image Parsing with Superpixels. In *ECCV*, pages 352–365, 2010. [43](#)

- [114] K. Toyama, R. Logan, and A. Roseway. Geographic Location Tags on Digital Images. In *ACM Multimedia*, 2003. [19](#)
- [115] R. H. Van Leuken, L. Garcia, X. Olivares, and R. van Zwol. Visual Diversification of Image Search Results. In *ACM WWW*, pages 341–350, 2009. [126](#)
- [116] V. Viitaniemi and J. Laaksonen. Experiments on Selection of Codebooks for Local Image Feature Histograms. In *Visual Information Systems. Web-Based Visual Information Search and Management*, pages 126–137. 2008. [93](#)
- [117] G. Wang, B. Seo, and R. Zimmermann. Automatic Positioning Data Correction for Sensor-annotated Mobile Videos. In *ACM SIGSPATIAL GIS*, pages 470–473, 2012. [48](#)
- [118] G. Wang, Y. Yin, B. Seo, R. Zimmermann, and Z. Shen. Orientation Data Correction with Georeferenced Mobile Videos. In *ACM SIGSPATIAL GIS*, pages 400–403, 2013. [15](#)
- [119] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained Linear Coding for Image Classification. In *IEEE CVPR*, pages 3360–3367, 2010. [7](#), [21](#), [88](#), [89](#), [92](#)
- [120] S. Wang, S. Fidler, and R. Urtasun. Holistic 3D Scene Understanding From a Single Geo-Tagged Image. In *CVPR*, 2015. [30](#), [35](#), [43](#)
- [121] X.-Y. Wei and C.-W. Ngo. Ontology-enriched Semantic Space for Video Search. In *ACM Multimedia*, pages 981–990, 2007. [25](#)
- [122] L. Wu, L. Yang, N. Yu, and X. S. Hua. Learning to Tag. In *ACM WWW*, 2009. [17](#), [56](#)
- [123] X. Xu, T. Mei, W. Zeng, N. Yu, and J. Luo. AMIGO: Accurate Mobile Image Geotagging. In *ICIMCS*, pages 11–14, 2012. [15](#), [34](#)
- [124] R. Yan, A. Natsev, and M. Campbell. A Learning-based Hybrid Tagging and Browsing Approach for Efficient Manual Image Annotation. In *IEEE CVPR*, 2008. [55](#)
- [125] K. Yanai, H. Kawakubo, and B. Qiu. A Visual Analysis of the Relationship between Word Concepts and Geographical Locations. In *ACM International Conference on Image and Video Retrieval*, 2009. [18](#)

BIBLIOGRAPHY

- [126] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear Spatial Pyramid Matching using Sparse Coding for Image Classification. In *IEEE CVPR*, pages 1794–1801, 2009. [7](#), [21](#), [88](#), [89](#), [91](#)
- [127] K.-H. Yap, T. Chen, Z. Li, and K. Wu. A Comparative Study of Mobile-based Landmark Recognition Techniques. *Intelligent Systems*, pages 48–57, 2010. [21](#)
- [128] Y. Yin, B. Seo, and R. Zimmermann. Content vs. Context: Visual and Geographic Information Use in Video Landmark Retrieval. *ACM Trans. Multimedia Comput. Commun. Appl.*, 11(3):39:1–39:21, 2015. [12](#)
- [129] Y. Yin, Z. Shen, L. Zhang, and R. Zimmermann. Spatial-Temporal Tag Mining for Automatic Geospatial Video Annotation. *ACM Trans. Multimedia Comput. Commun. Appl.*, 11(2):29:1–29:21, 2015. [11](#)
- [130] Y. Yin, Y. Yu, and R. Zimmermann. On Generating Content-Oriented Geo Features for Sensor-Rich Outdoor Video Search. *IEEE Transactions on Multimedia*, 17(10):1760–1772, 2015. [12](#)
- [131] Z. Yin, L. Cao, J. Han, J. Luo, and T. S. Huang. Diversified Trajectory Pattern Ranking in Geo-tagged Social Media. In *Proceedings of SDM*, pages 980–991, 2011. [19](#)
- [132] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang. Geographical Topic Discovery and Comparison. In *ACM WWW*, pages 247–256, 2011. [19](#)
- [133] J. Yuan, Z.-J. Zha, Y.-T. Zheng, M. Wang, X. Zhou, and T.-S. Chua. Learning Concept Bundles for Video Search with Complex Queries. In *ACM Multimedia*, pages 453–462, 2011. [25](#)
- [134] B. Zhang, Q. Li, H. Chao, B. Chen, E. Ofek, and Y.-Q. Xu. Annotating and Navigating Tourist Videos. In *ACM SIGSPATIAL GIS*, pages 260–269, 2010. [vii](#), [3](#), [8](#), [20](#), [24](#), [33](#), [122](#), [124](#)
- [135] H. Zhang, M. Korayem, E. You, and D. J. Crandall. Beyond Co-occurrence: Discovering and Visualizing Tag Relationships from Geo-spatial and Temporal Similarities. In *ACM WSDM*, 2012. [19](#), [68](#)

- [136] Y.-T. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. Tour the World: Building a Web-scale Landmark Recognition Engine. In *IEEE CVPR*, pages 1085–1092, 2009. [22](#)