Jaeyoung Choi · Gerald Friedland
Editors

Multimodal Location Estimation of Videos and Images

Springer

Multimodal Location Estimation of Videos and Images

Jaeyoung Choi · Gerald Friedland
Editors

Multimodal Location Estimation of Videos and Images

Springer

Editors
Jaeyoung Choi
Gerald Friedland
International Computer Science Institute
Berkeley, CA
USA

ISBN 978-3-319-09860-9 ISBN 978-3-319-09861-6 (eBook)
DOI 10.1007/978-3-319-09861-6

Library of Congress Control Number: 2014951051

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

With the widespread use of GPS-equipped handheld devices, location metadata (a.k.a geotags) has rapidly become an integral part of photos and videos shared over the Web. This trend enabled location-based multimedia organization, search, and retrieval on many Internet services such as Google, Facebook, and Flickr. The main driving force behind these services is the creation of highly personalized user experiences, allowing for better recommendations and targeted advertisements. Even with this trend, it has been estimated that only about 5 % of the existing multimedia content on the Internet is actually geotagged. A significant amount of consumer-produced media content is still obtained using devices that do not have GPS functionality. Privacy concerns have motivated users to disable automatic geotagging of media. Furthermore, even GPS-enabled devices cannot provide accurate location information when the photo or video was captured in an indoor environment.

Nevertheless, the volume of high-quality geotagged videos and photos on the Web represents a quantity of training data for machine learning on an unprecedented scale, giving rise to the idea of creating an automated task that would try to locate non-geotagged media from the Web using models obtained through the geotagged subset. Put simply: Given a video and its associated textual metadata, can we infer the location where it was taken? This idea of "multimodal video location estimation" was proposed several years ago by the authors of the book.

Since then, the "Placing Task" of the MediaEval evaluation evaluated the task on a global scale and the United States Government has sponsored the research in their Finder program with a separate set of evaluations conducted by National Institute of Standards and Technology (NIST). As a result, the problem has been approached with diverse methods and ideas in the research community and significant improvements have been made. Multimodal Location Estimation has become a powerful tool and accuracies now come close to human capabilities.

The goal of the book is to present an overview of this field to software developers, engineers, and researchers and to bring together the different communities working in the area. Apart from research interest, forensics experts, developers, and engineers for targeted advertising tools, as well as many people working in social media retrieval have become interested in the subject.

Berkeley, CA, USA, June 2014

Jaeyoung Choi
Gerald Friedland

Acknowledgments

First and foremost, we would like to thank our families. Jaeyoung is very grateful to his parents and his sister for their tireless support and encouragement. They have always supported him in everything he did, and for this he wants to thank them most of all. Gerald would like to thank his wife Yvonne and his daughter Mona for supporting him through this project and so many others over the years.

Jaeyoung is indebted to Jiwon Kim, Choongbum Lee, Jennifer Jinju Lee, Eugene Suh, Michelle Jeung-Eun Lee, and Jun Woo Lim, who were there for him when he needed them the most while going through crisis during his years in the U.S. To my good friends/housemates, Kang Wook Lee, Chan Kim, and Eunkwang Joo; and Seunghoon Lee, former captain of KGSA soccer team, for laying all the groundwork for shaping ourselves into a better team, Euiyoung Kim, Kunwoo Lee, and Grace Jeon; It would have been a real boring life here if it weren't for dear friends that I've met at Berkeley: Sunghwan Kim, Sun Choi, Kevin Ahn, Ilhyung Lee, Insoon Yang, Shinhye Choi, and Hyojung Shin. All the good and bad moments we've shared and went through together won't be forgotten. Korean folks at CS department have given me precious advices in many aspects. I'm grateful for the time I got to spend with Gunho Lee, Minkyung Kang, Chang-seo Park, Wontae Choi, Woojong Ko, Sangjin Han, and Yunsup Lee.

We are both grateful to our friends and colleagues at ICSI who directly or indirectly helped and supported the research work put into this book, in particular; Nelson Morgan, who provided us a research home at ICSI, Adam Janin, Arlo Faria, Benjamin Elizalde, Luke Gottlieb, T.J. Tsai, Howard Lei, Oriol Vinyals, Trevor Darrell, Julia Bernd, Bryan Morgan, and Michael Ellsworth.

We were privileged to be collaborating with gifted minds of Berkeley BASICS Group, Venky Ekambaram, Giulia Fanti, and Kannan Ramchandran.

This work would not have happened without the steady encouragement of Martha Larson from TU Delft, who always kept a firm grip on making MediaEval a fun, organically organized event. The resulting highly constructive, noncompetitive climate of the evaluation resulted in some of the most insightful discussions with colleagues we ever had.

Throughout the process of writing this book, many individuals from the community have taken time out to help us out. We'd like to give special thanks to the MediaEval and GeoMM community for actively participating in the feedback and contributions for this book. The book is a collection of quality research works from many institutions and individuals. Special thanks to all the authors and contributors of the guest chapters for their hard work.

Last but not least, we would like to acknowledge our sponsors. Initial experiments of this work were supported by an NGA NURI grant #HM11582-10-1-0008 and Korean Foundation for Advanced Studies. Later work was supported by National Science Foundation Award No. 1065240 and also National Science Foundation Award No. 1251276.

Contents

Contributors

T. Chen Department of Information and Communication Engineering, The University of Tokyo, Bunkyo-ku, Tokyo, Japan

Jaeyoung Choi International Computer Science Institute, Berkeley, CA, USA; California State University, East Bay, CA, USA

Nicholas Corso Department of EECS, UC Berkeley, Berkeley, CA, USA

Alexei A. Efros University of California, Berkeley, CA, USA

Venkatesan Ekambaram University of California, Berkeley, CA, USA

Gerald Friedland International Computer Science Institute, Berkeley, CA, USA; California State University, East Bay, CA, USA

A. Gallagher School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA

Luke Gottlieb International Computer Science Institute, Berkeley, CA, USA

Claudia Hauff Delft University of Technology, Delft, The Netherlands

James Hays Brown University, Providence, RI, USA

Gareth J.F. Jones Dublin City University, Dublin, Ireland

Pascal Kelm Technische Universität, Berlin, Germany

Martha Larson Delft University of Technology, Delft, The Netherlands

Howard Lei International Computer Science Institute, Berkeley, CA, USA; California State University, East Bay, CA, USA

Houqiang Li Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China

Jason Zhi Liang Department of EECS, UC Berkeley, Berkeley, CA, USA

Heng Liu Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China

Jiebo Luo Department of Computer Science, University of Rochester, Rochester, NY, USA

Tao Mei Microsoft Research, Beijing, China

Vanessa Murdock Microsoft, Bellevue, WA, USA

Adam Rae Future Cities Catapult, London, UK

Kannan Ramchandran University of California, Berkeley, CA, USA

Sebastian Schmiedeke Technische Universität, Berlin, Germany

Steven Schockaert Cardiff University, Cardiff, UK

Pavel Serdyukov Yandex, Moscow, Russia

Thomas Sikora Technische Universität, Berlin, Germany

Bart Thomee Yahoo Labs, San Francisco, CA, USA

Michele Trevisiol Yahoo Labs, Pompeu Fabra University, Barcelona, Spain

Eric Turner Department of EECS, UC Berkeley, Berkeley, CA, USA

Olivier Van Laere Yahoo Labs, Barcelona, Spain

T. Yamasaki Department of Information and Communication Engineering, The University of Tokyo, Bunkyo-ku, Tokyo, Japan

Avideh Zakhor Department of EECS, UC Berkeley, Berkeley, CA, USA

Chapter 1
Introduction

Jaeyoung Choi and Gerald Friedland

In driving school, we learned that "traffic is defined by the fact that no two cars can be at the same point at the same time." While quantum physicists might argue differently, this statement can be generalized to all objects in the universe, making time and space coordinates a physically guaranteed unique database key. This is, of course, especially interesting for indexing and retrieval of objects that are dependent on space and time, such as most photographs and videos. It is therefore no wonder that the inclusion of GPS coordinates into photos and videos has become commonplace. A photo or video taken at the same time and at the same place has an overwhelming chance of showing the same event and/or objects, regardless of any other descriptive element that may be provided along with the media. Furthermore, location-based services are rapidly gaining traction in the online world. Besides major players like Google and Yahoo!, there are many smaller startups in the space as well. The main driving force behind these services is the enabling of a very personalized experience. In a parallel development, a growing number of sites now provide public APIs for structured access to their content, and many of these already come with geolocation functionality. Flickr, YouTube, and Twitter all allow queries for results originating at a certain location. After an incident, law enforcement agencies spend many person-months to find images and videos, including tourist recordings, that show a specific address, to find a suspect or other evidence. Also, intercepted audio, terrorist videos, and evidence of kidnappings is often most useful to law enforcement when the location can be inferred from the recording. To date, however, human expert analysts have to spend many hours watching for clues of the location of a target video.

In the end, however, still less than 10 % of videos and images are geotagged. Likewise, the belief is that retrofitting archives with location information will be attractive to many businesses and enable new usage scenarios. Therefore, and as part of the

J. Choi (✉) · G. Friedland
International Computer Science Institute, Berkeley, CA, USA
e-mail: jaeyoung@icsi.berkeley.edu

G. Friedland
e-mail: fractor@icsi.berkeley.edu

J. Choi and G. Friedland (eds.), *Multimodal Location Estimation of Videos and Images*,
DOI 10.1007/978-3-319-09861-6_1

Big Data movement, we introduced [3] a task called *Multimodal location estimation*. The task denotes the utilization of one or more cues potentially derivable from different media, e.g., audio, video, and textual metadata to estimate the geocoordinates of content recorded in digital media. Evaluation of algorithms works by utilizing images and videos that have geotags and comparing the results against them.

Pioneering early work, mostly focusing on a single modality, in the area of location estimation already existed before 2010. For example, in early articles [9, 12], the location estimation task was reduced to an image retrieval problem on small self-produced, location-tagged photo databases. Krotkov's approach [1] for robot applications extracts sun altitudes from images while Jacobs' system [5] relies on matching images with satellite data. In both these settings, single images have been used or images have been acquired from stationary webcams. In the work of [7], geolocation is also determined based on the estimate of the position of the sun. They provide a model of photometric effects of the sun on the scene, which does not require the sun to be visible in the image. The assumption, however, is that the camera is stationary, and hence only the changes due to illumination are modeled. This information in combination with time stamps is sufficient for the recovery of the geolocation of the sequence. A similar path is taken in [6]. Previous work that has been carried out in the area of automatic geotagging of multimedia content has been based on textual tags of Flickr images. User-contributed tags have a strong location component, as brought out by [11], who reported that over 13 % of Flickr image tags could be classified as locations using Wordnet. Rattenbury et al. [8] and Serdyukov et al. [10] estimate the posterior distribution of the geolocations given the tags or vice versa from the training database and use this to estimate the geolocation of a query video. We should not forget to mention the pioneering work of [4], which is still often used as a basis for multimodal location estimation systems.

Since the only-recent introduction and start of the field, work in the field of location estimation is currently creating progress in many areas of multimedia research. As discussed in [3], cues used to estimate location can be extracted using methods derived from current research areas. Since data found from the Internet is used, multimodal location estimation work is performed using much larger test and training sets than traditional multimedia content analysis tasks and the data are more diverse as the recording sources and locations differ greatly. This offers the chance to create machine learning algorithms of potentially higher generality. Overall, multimodal location estimation has the potential to advance many fields, some of which we don't even know of, as they are created based on user demand for new applications.

This book aims at presenting an overview of the field, summarizing research done in Europe, Asia, and the Americas. As ironic as it may seem, these political boundaries do matter, as even a task as globally inspiring as geolocation is mostly funded and therefore defined and evaluated through geographically limited research projects.

Our book is structured based on approaches. We start with a deeper overview of the field and benchmarking methodology in Chap. 2. Chapters 3, 4, and 5 then present three different visual approaches for location estimation before Chap. 6 presents an acoustic approach. Chapter 7 follows with a graphical framework approach to model locations with feature probabilities. Chapter 8 then presents a multimodal approach. We conclude the book with a study on human performance in Chap. 9 and an interesting application described in Chap. 10.

Another way to view the structure of this research field is based on data resources and modalities. We therefore also provide an overview along this structure.

As explained above, social network photo sharing services provide billions of useful resources for location estimation. However, these data samples are contributed by lay people and are thus very diverse and potentially noisy. Many services provide ways to geotag the posts, photos, or videos, and some of the services allow the usage of GPS information recorded by the camera module in the EXIF metadata. Chapters 2, 3, 5, 6, 7, 8, and 9, and 10 present work on this type of data.

Many videos and images are annotated with descriptive text, which forms a valuable source of information for geotagging as well. This is explained in Chaps. 2, 7, and 8. Social media videos also contain audio, thereby allowing the inference of location over environmental noise, which is the focus of Chap. 6. In fact, social media images and videos have become so important for research that close to the finish of this book, Yahoo! announced the release of a dataset containing 100 million images and videos, many of them geotagged, for research purposes.

An alternative source for images used in the community are Satellite and Aerial image databases, such as Google Earth and Microsoft Bird's Eye View, which virtually cover the entire globe. Images from these databases provide information about the environment of different areas and can provide coverage for regions with sparse photos and videos. This is explored in Chap. 4.

A Gazetteer is a geographical dictionary that translates location names into geocoordinates and vice versa. A popular gazetteer among location estimation developers is called GeoNames.org. GeoNames covers all countries and contains 8 million entries. It provides a web-based search engine which returns a list of matching entries ordered by their relevance to the query. The GeoNames database also contains semantic information about the point of interest. The popular WordNet[1] is a freely available online lexical database of English which contains a network of semantic relationships between words; Augmented-WordNet[2] is an extended version of WordNet with more data. Augmented-WordNet can be used for filtering out these common nouns by looking at their part-of-speech tag. It also has limited geographical annotation. Chapter 2 discusses the use of dictionaries. There has been a constant debate in the community on whether it is better to rely on statistical methods only or to rely more on so-called semantic methods using dictionaries. We intentionally left the discussion out of this book but refer the reader to [2].

We hope that this book can contribute to a unification of the different approaches. The authors firmly believe that multimodality is the key to approach location estimation. We hope this book will do its part to bring the field closer together. Most importantly, we hope the reader will find the book educational and entertaining.

[1] http://wordnet.princeton.edu.

[2] http://ai.stanford.edu/~rion/swn.

References

1. F. Cozman, E. Krotkov, Robot localization using a computer vision sextant, in *IEEE International Conference on Robotics and Automation*, pp. 106–106 (1995)
2. G. Friedland, J. Choi, H. Lei, A. Janin, Multimodal location estimation on Flickr videos, in *Proceedings of the 2011 ACM Workshop on Social Media*, ACM, Scottsdale, Arizona, USA, pp. 23–28 (2011)
3. G. Friedland, O. Vinyals, T. Darrell, Multimodal location estimation, in *Proceedings of ACM Multimedia*, pp. 1245–1251 (2010)
4. J. Hays, A. Efros, IM2GPS: estimating geographic information from a single image, in *IEEE Conference on Computer Vision and Pattern Recognition, 2008*, CVPR 2008, pp. 1–8 (2008)
5. N. Jacobs, S. Satkin, N. Roman, R. Speyer, R. Pless, Geolocating static cameras, in *IEEE International Conference on Computer Vision*, pp. 1–6 (2007)
6. I. Junejo, H. Foroosh, Estimating geo-temporal location of stationary cameras using shadow trajectories. Comput. Vision-ECCV **2008**, 318–331 (2008)
7. J. Lalonde, S. Narasimhan, A. Efros, What does the sky tell us about the camera? Comput. Vision-ECCV **2008**, 354–367 (2008)
8. T. Rattenbury, M. Naaman, Methods for extracting place semantics from Flickr tags. ACM Trans. Web (TWEB) **3**(1), 1 (2009)
9. G. Schindler, M. Brown, R. Szeliski, City-scale location recognition, in *IEEE Conference on Computer Vision and Pattern Recognition, 2007*, CVPR'07, pp. 1–7 (2007)
10. P. Serdyukov, V. Murdock, R. van Zwol, Placing Flickr photos on a map, in *ACM SIGIR*, pp. 484–491 (2009)
11. B. Sigurbjoernsson, R. V. Zwol, Flickr tag recommendation based on collective knowledge, in *ACM WWW*, pp. 327–336 (2008)
12. W. Zhang, J. Kosecka, Image based localization in urban environments, in *3D Data Processing, Visualization, and Transmission, 3rd International Symposium on*, pp. 33–40 (2006)

Chapter 2
The Benchmark as a Research Catalyst: Charting the Progress of Geo-prediction for Social Multimedia

Martha Larson, Pascal Kelm, Adam Rae, Claudia Hauff, Bart Thomee, Michele Trevisiol, Jaeyoung Choi, Olivier Van Laere, Steven Schockaert, Gareth J.F. Jones, Pavel Serdyukov, Vanessa Murdock and Gerald Friedland

Abstract Benchmarks have the power to bring research communities together to focus on specific research challenges. They drive research forward by making it easier to systematically compare and contrast new solutions, and evaluate their

M. Larson (✉) · C. Hauff
Delft University of Technology, Delft, The Netherlands
e-mail: m.a.larson@tudelft.nl

C. Hauff
e-mail: c.hauff@tudelft.nl

P. Kelm
Technische Universität, Berlin, Germany
e-mail: kelm@nue.tu-berlin.de

A. Rae
Future Cities Catapult, London, UK
e-mail: arae@futurecities.catapult.org.uk

B. Thomee
Yahoo Labs, San Francisco, CA, USA
e-mail: bthomee@yahoo-inc.com

M. Trevisiol
Pompeu Fabra University, Barcelona, Spain
e-mail: trevisiol@acm.org

O. Van Laere
Yahoo Labs, Barcelona, Spain
e-mail: vanlaere@yahoo-inc.com

J. Choi · G. Friedland
ICSI, Berkeley, CA, USA
e-mail: jaeyoung@icsi.berkeley.edu

G. Friedland
e-mail: fractor@icsi.berkeley.edu

S. Schockaert
Cardiff University, Cardiff, UK
e-mail: s.schockaert@cs.cardiff.ac.uk

J. Choi and G. Friedland (eds.), *Multimodal Location Estimation of Videos and Images*,
DOI 10.1007/978-3-319-09861-6_2

performance with respect to the existing state of the art. In this chapter, we present a retrospective on the Placing Task, a yearly challenge offered by the MediaEval Multimedia Benchmark. The Placing Task, launched in 2010, is a benchmarking task that requires participants to develop algorithms that automatically predict the geolocation of social multimedia (videos and images). This chapter covers the editions of the Placing Task offered in 2010–2013, and also presents an outlook onto 2014. We present the formulation of the task and the task dataset for each year, tracing the design decisions that were made by the organizers, and how each year built on the previous year. Finally, we provide a summary of future directions and challenges for multimodal geolocation, and concluding remarks on how benchmarking has catalyzed research progress in the research area of geolocation prediction for social multimedia.

2.1 Introduction

A benchmark is a standardized task that is carried out in order to evaluate alternative approaches to addressing the task and to facilitate a fair comparison between multiple strategies for tackling this task. Benchmarks bring research communities together to focus on a specific research challenge. This coming together of researchers with common interests to work on a specific task can drive research forward by enabling them to comparatively evaluate their work. In this chapter, we present a retrospective of the Placing Task, a challenge offered within the MediaEval Benchmarking Initiative for Multimedia Evaluation.[1] We track the development of the Placing Task over four editions from 2010 to 2013, and present an outlook to 2014. The evolution of the Placing Task in MediaEval illustrates the power of a benchmark to establish a new research topic and a community of collaborating researchers working to address the challenges of this topic, together with persistent datasets that enable researchers to evaluate their results with respect to the state of the art, and explore the effectiveness of the new algorithms that they develop.

[1] http://multimediaeval.org.

G.J.F. Jones
Dublin City University, Dublin, Ireland
e-mail: Gareth.Jones@computing.dcu.ie

P. Serdyukov
Yandex, Moscow, Russia
e-mail: pavser@yandex-team.ru

V. Murdock
Microsoft, Bellevue, WA, USA
e-mail: vanmur@microsoft.com

Geoprediction for social multimedia, also known as *placing*, is the task of inferring geocoordinates for images or videos that users have uploaded to social sharing websites. The key application of geoprediction technology is indexing images and videos online, making them easier, to find, manage, and browse, and, in general, more useful for users. Location, and concepts related to location, have a close connection with how users interpret, organize, and use multimedia, and consequently applications that use geocoordinates are considered to be important in allowing users to get the most out of social multimedia. Many of today's cameras and phones can and do record geoinformation. Nonetheless, a large number of videos and images are uploaded without georeference. For this reason, high-performance placing algorithms are necessary in order to generate metadata that makes it easier for users to retrieve and browse social multimedia.

This chapter discusses Placing Task, a challenge offered by the MediaEval Benchmarking Initiative for Multimedia Evaluation[2] to the multimedia research community with the goal of fostering the development of new algorithms addressing the task of automatically predicting the geocoordinates of social multimedia. While this chapter focuses on the topic of geoprediction, we anticipate that is also relevant to multimedia benchmarking and general. We hope that it might ultimately serve to support the consolidation and catalyzation of results in other research areas, which may benefit from applying the strategies and techniques used by the MediaEval Placing Task.

2.1.1 The Placing Challenge for Social Multimedia

The nature of social multimedia means that placing is fundamentally a multimedia challenge, involving different modalities. Images and videos uploaded by users are associated with metadata, such as titles and descriptions. They are also associated with user-contributed tags and comments. Often the user-uploaded multimedia items are connected in a social network: here, information such as social connections and views may also be available. In the case of video, the multimedia signal involves temporal patterns which can be exploited. Video typically involves both a visual and an audio channel. Audio includes spoken content, but also music and environmental sounds contained in videos.

Interest in technologies that infer geocoordinates was established by work such as [22] and [16]. Research effort devoted specifically to placing multimedia that users share on the Internet gained momentum along with the rise of social media. Geoprediction for social multimedia took on an independent form as a task in its own right, with the publication of a paper entitled,"Placing Flickr Photos on a Map" [52], which followed on the heels of [10]. Due to the influence of [52], the word "Placing" was adopted as the name of the task in the MediaEval benchmark.

It is important to note that the task of placing social multimedia is different from predicting geolocation of multimedia data that was not captured by users for personal use or social sharing. The phenomenon of people taking and sharing pictures

[2] http://multimediaeval.org.

and videos ranges from people who point-and-shoot in order to capture a moment or memory, to people who document events and objects, and people who pursue photography as a hobby. For whatever reason that people produce and distribute multimedia, it is clear that taking a picture or capturing a video is not a random act. Rather it occurs with a reason. The result is that social multimedia is characterized by a particular distribution of subject matter and of photographic style. In short, multimedia shared on the Internet can be considered a *social signal*, i.e., an information stream whose characteristics are determined by the underlying behavior of the people who produce it.

The challenge of placing social multimedia has multiple dimensions: First, placing algorithms must be able to confront the noise and uncertainty associated with social multimedia. The relationship between the visual content of an image and the location at which the image was taken is often a weak one. For example, two images can both depict a Black Labrador in front of a red Volkswagen. The content of both images is visually distinctive, making the images potentially very similar to each other with respect to image processing algorithms. Yet, it is possible that that the two were taken many kilometers apart. Conversely, two images taken at the same location, for example, a panorama and a close-up shot, may have no visual content in common. Another source of uncertainty is user-contributed metadata: titles, descriptions, and tags often receive only little attention from users uploading photos and can be incomplete or completely misleading.

Second, algorithms must be capable of effectively combining multiple modalities. Noise and uncertainty can be addressed by simultaneously exploiting multiple information sources. However, in order to benefit from the availability of multiple modalities, multimedia placing algorithms must be able to effectively exploit the complementary information that they contain. The contribution of each modality should enhance the ability of the algorithm to distinguish location. In the extreme cases, such exploitation requires the ability to identify cases in which one modality should be trusted and the others ignored.

Third, placing algorithms must be able to exploit large quantities of data. Geoprediction for social multimedia requires building algorithms that can place a multimedia item at any point in the world. The ability of algorithms to predict place reliably rests on their capacity to process and exploit the large amounts of social multimedia data that are available online, if they are to maximize their ability to cover the world's surface.

Finally, placing algorithms must be able to deal with the uneven distribution of the geotagged data that is available for training. Many locations are associated with rich resources in the form of a large number of multimedia items that have been taken there, and are associated with geocoordinates and available online. Other locations are represented by little to no data, creating an overall data sparseness problem. Fully facing the challenge of "placing" requires developing algorithms that explore a range of different techniques so that it is possible to cover each of these dimensions.

In this chapter, we trace the rise of interest and attention in the multimedia community to the task of placing multimedia items on the world map. Placing has received attention from a broad spectrum of researchers. However, a unique community of researchers has emerged who have worked together to actively define the task of placing social multimedia and to guide and pursue the development of placing solutions. This community, of which key members are the authors of this chapter, have used benchmarking as a tool to encourage and guide the development of algorithms that address the task of multimodal geolocation estimation for social media. Specifically, they have launched and grown the benchmarking task "Placing: Geocoordinate Prediction for Social Multimedia" within the MediaEval Benchmarking Initiative for Multimedia Evaluation.[3] In this chapter, we discuss the major developments in algorithms to address the challenges of "placing," and how these developments have been guided by the Placing Task at the MediaEval benchmark.

This chapter charts the development of the Placing Task over the years 2010–2014. In Sect. 2.1.2, we briefly discuss the history of benchmarking, and its ability to promote progress in specific areas of research. Then, in Sect. 2.2, we discuss the design of the task in each year and mentions major results. Based on the experiences in the benchmark, Sect. 2.3 presents an overview for the future challenges that are faced by researchers in the area of Placing. Finally, Sect. 2.4 finishes with a conclusion and outlook.

2.1.2 The Benefits of Benchmarking

A benchmark is a standardized task that is carried out in order to make possible a fair comparison among multiple algorithms. A benchmark generally consists of a description of a task, resources needed to address that task, and a standard metric or evaluation procedure used to judge the quality of algorithms that address the task. Although other areas have a slightly different definition, this definition holds for the fields of information retrieval and multimedia, from which the Placing Task draws most of its participants.

The roots of the benchmark evaluation movement that gave rise to the MediaEval Multimedia Benchmark can be traced to TREC, the Text REtrieval Conference,[4] which was established in 1992 by the US National Institute of Standards and Technology (NIST). TREC focused initially on text retrieval tasks, but has progressed to topics such as web search, various social media search tasks, and speech and video search, the latter within the TRECVid benchmark. TREC was followed by the establishment in 1999 of NTCIR[5] in East Asia and in 2000 of CLEF—Cross Language Evaluation Forum in Europe.[6] MediaEval was founded in 2008 as a track named "VideoCLEF" within CLEF. In 2008–2009, VideoCLEF ran tasks examining automatic video tagging and cross-language video search. In 2010, VideoCLEF

[3] http://www.multimediaeval.org.

[4] http://trec.nist.gov.

[5] http://research.nii.ac.jp/ntcir/.

[6] http://www.clef-campaign.org.

expanded its task offering and became an independent benchmarking initiative named MediaEval. In this year, the Placing Task was established within MediaEval.

The most often cited benefits of benchmarks is that they concentrate research effort on specific problems or challenges, enable cross-site comparison of approaches to address these problems, and drive forward the state of the art with respect to understanding methods or development of new ones. These benefits have been discussed specifically in relation to MediaEval in [36] and [34]. Another important benefit of benchmarks it that they provide stakeholder, e.g., companies who are interested in developing products based on placing technology, with a way to overview and to compare different solutions. Involving stakeholders in benchmarking can couple technologies to real-world needs [45].

One of the less-discussed benefits of benchmarking is that it allows researchers to learn how to deal with a new type of problem. Participating in a benchmark enables researchers to expand the horizons or scope of their research. The opportunities presented by benchmarks are valuable for both students who are developing thesis topics, and also for experienced researchers in the field who are interested in broadening the scope of their investigations. As a multimedia benchmark, MediaEval is related to TRECVid [54]. TRECVid is a video retrieval benchmark was first established as TREC track 2001–2002 and became an independent benchmark in 2003. Traditionally, TRECVid has been more closely tied to the signal processing aspects of multimedia challenges. MediaEval distinguishes itself from TRECVid by focusing on the human and social aspects of multimedia.

MediaEval follows a yearly cycle, like many other benchmarks. However, it is distinct in important aspects, which we discuss briefly here.

First, the tasks that are offered by MediaEval are decided on the basis of a community survey. This survey gives details of the tasks potentially available for the next round of the benchmark, and is circulated widely at the beginning of the year. The survey gives details of the offered task, but frequently also different possibilities for test data might be used or the specific research questions to be addressed. The survey gathers information about which tasks researchers and stakeholders find most interesting, and which aspects of the tasks that would be interested in focusing on. In the case of the Placing Task, the survey has allowed us to determine that researchers appreciate extra resources, such as visual features, to be released along with the task data, and has also made it possible to ensure that the dataset is structured so that it is computationally feasible for researchers in a wide variety of research labs to tackle the task.

Second, MediaEval tasks are largely autonomous from the overall benchmark, whose role is limited to ensuring the smooth running of the overall activity by maintaining the schedule for the year, and organizing the yearly workshop. This autonomy means that the entire responsibility for developing the task description and dataset, organizing submission of the “runs” (i.e., the results of experimental conditions) submitted by the task participants and their evaluation lies in the hands of the task organizers. In this way, the task organizers have the freedom to develop the most productive and useful task possible, but they also bear the responsibility of ensuring that the dataset is released on time, and that the task is successful.

Third, the MediaEval workshop is seen as a gathering place at which teams that have developed solutions to the tasks come together to exchange ideas and experiences and to forge communities with shared research interests. MediaEval encourages task participants to make use of the resources developed for the task and participants; shared experiences in tackling the task to move beyond the results reported in the Working Notes proceedings published at the workshop. After the workshop task participants, either as individual teams or in combined groups publish papers at mainstream venues. Placing is one example of a new task which has developed an active ongoing research community of shared interested through its participation in the MediaEval benchmark. The communities are typically highly energetic and are comprised of both established researchers and those developing their careers, such as postdoctoral researchers and those beginning their research journey, such as PhD students. Publication of MediaEval results outside the MediaEval workshop promotes the work of MediaEval to wider research communities.

A benchmark is useful if it allows researchers to make progress on a specific problem. It is even more useful if it allows researchers to make gain insight into a whole new class of problems. We believe that multimedia placing represents a new class of multimedia problems. Here, we briefly mention the aspects of the Placing Task that have led us to the conclusion that developing approaches to address the Placing Task will contribute to moving the field of multimedia forward as a whole.

Placing a photo on the world map involves inferring meaning from huge quantities of multimedia data, qualifying it as the type of problem currently often referred to be the term *Big Data*. As mentioned above, it is a fundamentally multimodal problem: creating solutions to address the Placing Task requires developing effective approaches for combining modalities. The data has been created by humans, and, as such, is characterized by an underlying social signal that reflects human behavior, such as travel and photo taking habits. The data used in the Placing Task is "found data." In other words, it is not collected under controlled conditions, but rather gathered in the wild. Finally, Placing Task algorithms build successfully on "classic" machine learning and information retrieval algorithms. For this reason, results of the Placing Task are likely to find application in other areas of the fields of multimedia and information retrieval.

2.2 Charting the Progress

In this section, we discuss, in turn, each year that the Placing Task has been offered at the MediaEval benchmark. The discussion documents the process of building each new year of the benchmark on the following year. Each year chose a different direction in which to push beyond the challenge that was addressed by the previous year into potentially productive, however, yet uncharted territory.

2.2.1 Placing Task 2010: Inception

The inception of the MediaEval Placing Task dates to 2010, the year that MediaEval became an independent benchmark. The task was launched by Pavel Serdyukov and Vanessa Murdock in the wake of their 2009 SIGIR paper entitled "Placing Flickr Photos on a Map" [52]. The paper, and its follow-up [44], showed the strength of methods that exploit metadata assigned by users to images in order to predict geocoordinates.

Previously, Flickr had started to allow users to upload videos of up to 90 s in length in addition to photos. The tagline associated by Flickr with the introduction of video was, "It is like a photo, but it moves!" Both the size limit and the tagline suggest that it should not be assumed a priori that Flickr videos have the same characteristics as videos uploaded elsewhere on the Internet. Instead, Flickr videos could be expected to have much in common with photographs.

The original paper had investigated photos, and the task was designed to investigate the problem of placing Flickr videos on the map. Use of open resources such as gazetteers or Wikipedia were encouraged. This made it possible to compare approaches that try to extract place names from user-contributed metadata using gazetteers to approaches that build statistical models of text features. Also, use of images to help predict the geolocation of videos was encouraged. In this sense, the task could be considered to be related to computer vision work carried out on scene matching. The ICCV 2005 Computer Vision Contest was called "Where am I?"[7] and required participants to use visual features to match images with unknown locations to images whose locations were known. In contrast to the Placing Task, this contest tackled a small-scale problem (in the order of hundreds of images), and also did not use social images shared online.

2.2.1.1 Design and Data

The MediaEval 2010 Placing Task[8] was carried out using a dataset of Creative Commons licensed videos crawled from Flickr containing 5125 videos in the training set and 5091 videos in the test set. The metadata for each video included user-contributed title, tags, description comments, and also information about the user who uploaded the video (including favorites and contacts). The metadata for ∼3 Million Creative Commons licensed, geotagged Flickr images was also included. Participants wishing to use visual features could use this metadata to download the images from Flickr.

A set of basic visual features extracted for all images and for keyframes of the videos was provided. Such a resource makes it possible for teams to participate in the benchmark that do not have the infrastructure necessary for large-scale visual feature extraction. Additionally, it ensures that everyone has access to features that were extracted using the same implementation of the feature extractor. In this way,

[7] http://research.microsoft.com/en-us/um/people/szeliski/visioncontest05/default.htm.

[8] http://multimediaeval.org/mediaeval2010/placing.

it is possible to control for the impact of minor variations in feature extraction on the comparison of geolocation approaches.

There are 16 zoom levels with which users can place multimedia items on the map in Flickr, and this corresponds to 16 accuracy levels between 1 (world-level) and 16 (street level). The videos in the dataset were selected to provide a broad coverage of users, but also because they have a high accuracy at the street level. The location at which the uploading user had placed the videos on the map was used as the ground truth. Performance was measured in terms of the distance between the location hypothesized by the experimental system and the reference location, i.e., ground truth. The results were reported in terms of the number of photos correctly placed within a 1, 5, 10, 50 and 100 km radius.

2.2.1.2 Results and Insights

Five groups "crossed the finish line" for this task in MediaEval 2010, submitting runs and also a working notes paper. Their papers can be found in the MediaEval 2010 Working Notes Proceedings [33]. The best results for each team are plotted in Fig. 2.1. Note that for the maximum evaluation radius, an approach will achieve 100 % accuracy unless it fails to produce a prediction for all multimedia items.

Here we summarize the major results in 2010. The best performing run was submitted by Ghent University, which used a two-step approach that made use of the textual metadata associated with the images. In the first step, a language model identified the most likely area of the video. In the second step, the location of the video was pinpointed by identifying the closest resources (images and videos) from the training set. The results were reported in the working notes paper [58] and also in [59].

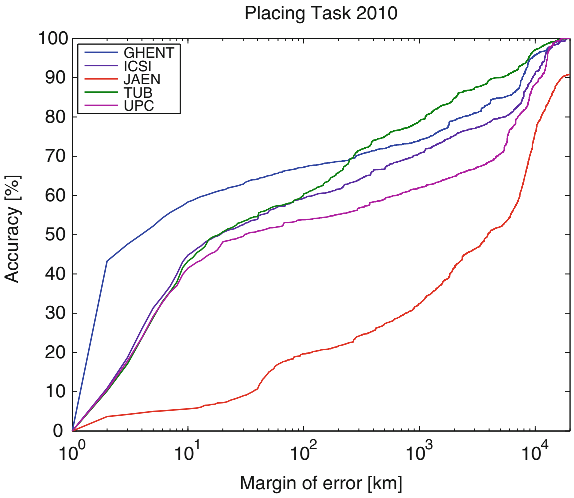


Fig. 2.1 Participants of the Placing Task 2010

International Computer Science Institute (ICSI) proposed two approaches [7]. The first exploited the prior distribution of tags, in particular choosing tag candidates extracted from video metadata on the basis of small spatial variance. The second approach also undertook supervised resolution of toponyms (using Geonames[9]).

The Universitat Politecnica de Catalunya (UPC) pursued an approach applying knowledge resources (a placenames database) combined with natural language processing used to detect and disambiguate geographical names in the metadata [12]. The SINAI group from the University of Jaen applied geographic-named entity recognizer [23].

The only group to make use of visual features in 2010 was the Technische Universität Berlin, who took a grid-based approach to the task, predicting the grid position of the video by combining a textual model based on metadata and a visual model based on low-level visual features [24, 26]. It is interesting to observe that this initial effort by a single group proved "contagious" and in future years more groups moved on to attempt to exploit not only visual but also audio features.

In the first year of the Placing Task, the importance of the user signal became clear. If a user has taken multiple images in the same place, and tagged them with the same tag set, the presence of one of these images in the training data will be enough to correctly identify the location of any of the other images in the test set. The challenge of placing reduces to a matching problem. Flickr allows users to carry out a so-called "bulk upload," i.e., uploading multiple items at the same time and assigning them the same tag set. On the one hand, "bulk uploads" can be seen as part of the overall problem of prediction the geolocation of multimedia items on Flickr. On the other hand, the fact that bulk uploads occur is strongly influenced by the technical possibilities offered by the platform. For this reason, the patterns in the data caused by bulk uploads cannot be directly attributed to the underlying social signal, i.e., human photo-taking behavior in the real world. Further, participants discovered that if the user specified a home location, this location provided a good fallback location. Systems placed a photo at the home location if there was no other way to predict location.

Already in the first year, the Placing Task embraced the model of publishing initial results in the MediaEval working note proceedings, but not encouraging the working notes paper to be a final product. Instead, researchers attended the benchmark, discussed the results, and wrote revised papers, which they submitted to other venues. The task contributed to a session called "Automatic Tagging and Geotagging in Video Collections and Communities" [34]. The conference included a paper from the Ghent University team concerning their two-step metadata-based approach [59]. It also included a paper from the Technische Universität Berlin team on their combined textual/visual approach [26]. Other publications were directed at ACM Multimedia 2011 and its satellite workshops [15, 25] and IEEE conferences [5, 37]. The presentation and discussion of the results at these conferences contributed to the growth of the task at future years of MediaEval.

[9] http://www.geonames.org.

An important insight of the first year was that participants were quickly tempted to download the dataset and start to implement approaches without reading the related work. The task organizers were concerned that participants' efforts would then result in them "re-inventing the wheel". In future years, the task description was accompanied by a list of papers entitled "Recommended Reading" and often even referred to as "Required Reading" by the task organizers. This practice was surprisingly helpful, and allowed teams in future years to boost the level of sophistication with which they approached the task.

2.2.2 Placing Task 2011: Consolidation

The main goal of MediaEval 2011 Placing Task[10] was to consolidate the task of geolocation prediction for social video by building on the 2010 edition. Pascal Kelm and Adam Rae joined Vanessa Murdock and Pavel Serdyukov as organizers of the task.

2.2.2.1 Design and Data

The videos in the test and development sets of 2010 were combined to create the 2011 development set (total of 10,216 videos). Again, metadata for ∼3 Million Creative Commons licensed Flickr images was also released for use in training, and task participants could use the metadata or also crawl images from this set from Flickr to develop their systems. Also, a basic set of visual features was again released. The test data consisted exclusively of video and contained 5,347 individual videos. As in 2010, the results were reported as the number of photos correctly placed within a 1, 5, 10, 50, and 100 km radius. Additional details can be found in [50].

2.2.2.2 Results and Insights

Six groups submitted results for the Placing Task 2011, and their papers can be found in the MediaEval 2011 Working Notes Proceedings [32]. ICSI, Ghent, and UPC returned to participate in the task again, and were joined by University of Campinas (UNICAMP), Delft University of Technology (WISTUD), and Chemnitz University of Technology (CUT). The best scoring runs of each of these groups is plotted in Fig. 2.2.

UPC [13] focused on textual features, an improvement of their 2010 approach [12], and also extended to explore and information retrieval approach. Their best scoring approach exploited a combination of the two. Chemnitz University of Technology [29] applied a text-based retrieval approach.

[10] http://multimediaeval.org/mediaeval2011/placing2011/.

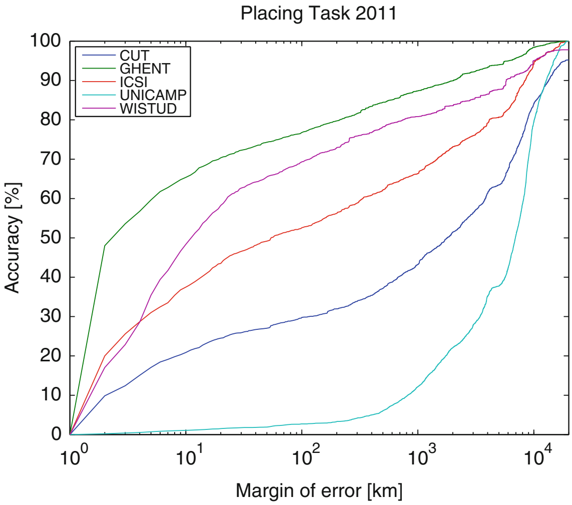


Fig. 2.2 Participants of the Placing Task 2011

ICSI proposed an approach integrating textual tags with visual and audio cues [9]. Their best scoring approach used tags refined with visual features.

WISTUD [18] used both textual and visual features, but achieved its best performance using filtered terms. Textual terms that were used by few users or that had a very high geographical spread were filtered out. Interestingly, at the 1 km scale, filtering was not useful.

The innovation of UNICAMP [39] was to exploit the visual content of the video. They used a Histogram of Motion approach [1]. Although the approach could not compete with text-only approaches, it was an important contribution because it represented an effort to use the temporal patterns in video in order to tackle the task. Other groups using visual features had confined themselves to static features derived from individual keyframes.

Ghent [60] extended their textual-metadata approach of 2010 [58]. They successfully exploited additional resources gathered from Flickr (mutually exclusive with the test set). Further, they built on an insight gained in 2010, namely that the granularity of the georegions used by the algorithm is critical. They use Dempster–Shafer theory to determine if there is sufficient evidence for an image to be placed using a given level of granularity [61]. The user's home location and visual similarity were used as fall back features.

As in the previous year, after the workshop, several of the participants improved and consolidated their approaches and published results in mainstream venue. Papers included the work of Claudia Hauff at TU Delft on exploiting information extracted from microblog posts (i.e., Twitter) and geographical priors [19, 20]. Also, ICSI developed a graph-based approach aimed at dealing with data sparsity [6], and UniCamp extended their approach [46]. Olivier Van Laere and colleagues from the

University of Ghent published a systematic overview of their approaches to the MediaEval 2010 and 2011 Placing Tasks as [62].

In 2011, for the first time, the task was described with an overview paper [50]. There were two reasons for the introduction of overview papers in MediaEval 2011. First, the overview paper describes the task, and communicates to the participants what the organizers are trying to achieve with the task. Second, participants were asked to focus their working notes paper on the unique contribution of their algorithm, and on analyzing their results. They can cite the overview paper for the formulation of the task and a description of the dataset.

In 2011, the Placing Task became interested in the impact of the identity of the uploader on the performance of algorithms. It was observed that the items uploaded by the same person are often near duplicates and/or have the same tag sets. For this reason, it is generally easier to predict the location of an item uploaded by a user, if other items uploaded by that user are included in the training set.

Note that in any given social multimedia collection, it can be expected that users upload multiple items and that some users upload substantially more than others. As a whole, user uploading patterns are part of the underlying characteristics of the social multimedia collection under study. However, the amount of user overlap between the test and the training set is not a fundamental property of the collection. Rather, it has strong dependencies on the data collection process. First, it is dependent on the amount of time that has elapsed between when the test set and when the training set is crawled. Second, it is dependent on the specific characteristics of the users that upload multimedia items in high volumes. Specific characteristics of one or two of these users (e.g., one of them happen to only upload multimedia captured in London), can have a disproportionally large impact on geolocation prediction results.

Although in 2011, the organizers became aware of the impact of exploiting "same-uploader" items for geolocation prediction, it was not until 2013 that the Placing Task issued a dataset in which the training and the test data were uploaded by mutually exclusive sets of users.

2.2.3 Placing Task 2012: Expansion

The MediaEval 2012 Placing Task[11] again addressed the challenge of generating geolocation predictions for video. It focused on extending the dataset from previous years. In order to encourage participants to develop methods exploiting video content, the task organizers required teams to submit one algorithm that used only video/audio features.

In 2012, Pascal Kelm and Adam Rae served as task organizers. Within the larger context of MediaEval, 2012 saw the handover of both the Placing Task and the Tagging Task (dedicated to automatically generating topical tags for video [51]) to organizer groups that did not include the original founders of the task, but rather were

[11] http://multimediaeval.org/mediaeval2012/placing2012.

comprised of a “second generation” of organizers, including people who had joined the task as participants. The ability of MediaEval to renew itself in this way has made an important contribution to the sustainability and the growth of the benchmark.

2.2.3.1 Design and Data

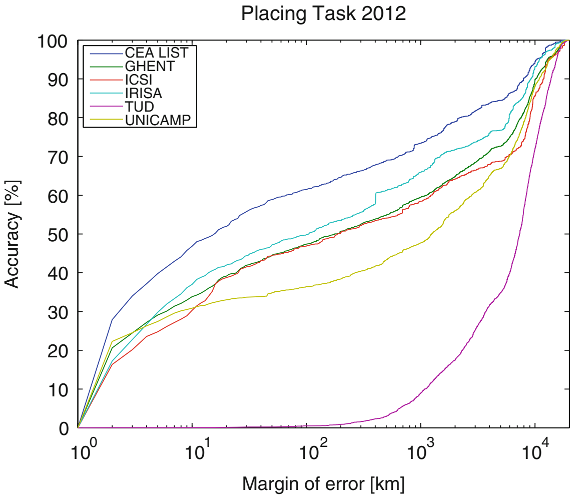
The test and development data of 2010 and 2011 was combined to create the 2012 development set (total of 15,563 videos). As in 2010 and 2011, metadata for ∼3 Million Creative Commons licensed Flickr images was also released. Again, a basic set of visual features were again released. The test set consisted of 4,182 videos. As in previous years, the results were reported as the number of photos correctly placed within a 1, 5, 10, 50, and 100 km radius. A basic set of visual features were again released. Additional details on the dataset can be found in [49].

2.2.3.2 Results and Insights

Seven groups submitted results for the Placing Task 2012, and their papers can be found in the MediaEval 2012 Working Notes Proceedings [35]. The best runs of each group are plotted in Fig. 2.3. New participants in 2012 were INRIA/IRISA [56] and CEA LIST [48].

The submission of CEA LIST [48] demonstrated the positive effect of user models on predicting geolocation. Previously, information on user tagging behavior had been exploited by [7] and information about the user home location had been used by [60].

Fig. 2.3 Participants of the Placing Task 2012



The graph-based approach of ICSI [4] carried out a joint estimation of all the locations of the test videos. Joint estimation was shown to lead to a performance improvement. Another notable contribution was the attempt of CEA LIST [48] to exploit motion features extracted from the videos.

Other approaches to the 2012 Placing Task combined textual and visual [40, 56, 63]. The Ghent University submission [63] built on its system from the previous year, exploring new feature selection and similarity search, but also experimenting with visual features. The UNICAMP approach is notable because it aggregated ranked-lists derived from various modalities. It was extended and subsequently published as [38]. The INRIA/IRISA approach was extended into an approach published as [57]. This approach exploits two-stage hypothesis refinement, which has proven effective in a number of different variants proposed to address the MediaEval Placing Task. It also exploits user's upload history, social network, and a visual-based matching technique, as well as visual similarity.

TUD proposed a visual-only approach that attempted, with unfortunately little success, to exploit geographical information [42]. Underlying geographical regions related to climate and anthropogenic biomes was used to create regions of the world, on the basis of the assumption that such regions would prove to be visually stable and lend themselves well to modeling.

TU Berlin [27] participated in the MediaEval 2013 Placing Task as an organizer team. As is customary practice within MediaEval, organizers do not report their results in the overall ranking.

Placing Task 2012 was the first time that participants shared code. Because this was also a first for MediaEval, Adam Rae was invited to give a talk at the MediaEval workshop on the importance of code sharing practices. He entitled his talk, "MediaEval Code of Conduct". The talk was aimed at reinforcing with the community the importance of releasing code.

2.2.4 Placing Task 2013: Volume

The organization of the MediaEval 2013 Placing Task[12] was taken over by a new team, Claudia Hauff, Bart Thomee, and Michele Trevisiol. The team made four crucial changes to the dataset, as well as the overall task compared to previous years [21]: (i) The task switched from predicting the geolocation of video to predicting the geolocation of images (a return to the original task tackled by [52]); (ii) in the 2013 edition, data collection was carried out in a way that ensured that the set of users contributing images to the training set was mutually exclusive with the set of users contributing images to the test set; (iii) the test set size was several orders of magnitude larger than in previous years to offer new *computational* as well as *algorithmic* challenges; and,

[12] http://multimediaeval.org/mediaeval2013/placing2013.

(iv) the new subtask of *Placeability* asks task participants to estimate the accuracy of the predicted locations, a necessary metric when employing item location prediction as a preprocessing step for an application.

2.2.4.1 Design and Data

Main task. Due to the changes described above, the existing datasets could not be reused. The 2013 PlacingTask dataset[13] contains nearly nine million Flickr images released by the owners with Creative Commons License. The sampling process ensured that regions which are popular by Flickr users are also represented as such in the dataset. The extracted metadata and extracted low-level image features follow the templates provided in previous editions, as does the evaluation.

The 2013 Placing Task pioneered the *Russian Dolls* approach to test set creation. Under this approach, the test data is available in five different sizes, in order to allow participation in the task even without very powerful computational resources—ranging from 5,300 images in the smallest test set and 262,000 images in the largest test set. The Russian Dolls approach ensures that the images of a smaller test set are also available in the larger test set as well.

Placeability subtask. The Placeability subtask was introduced to investigate whether it is possible to derive a measure of confidence for a predicted location. Past years' experiences had shown that many items can be placed with high accuracy; in 2013, the task made the next step and investigate whether we can also place items with high confidence. Based on the estimated confidence, in self-training, for instance, we may enlarge the training data only with those items that have been located accurately with high confidence, and in the location estimation task we can direct our computational resources to those items, whose confidence score is low.

For a first evaluation, task participants are asked to estimate the error for each prediction in kilometers. The linear and rank correlation coefficients are employed to compare the ability of the algorithms to estimate the error correctly: the true error distance in kilometers (as determined for the main task) is correlated with the predicted error distance. A high correlation coefficient indicates that the algorithm is able to infer the accuracy of the estimation.

As an example, a basic placeability approach can be the following: let us assume that the location estimation approach computes a ranked list of locations (and the top location is returned as estimated location). If the top n locations are distributed all over the globe, the method may have low confidence and thus the estimated error would be high. On the other hand, if the top n locations are spatially very close (e.g., having a standard deviation of a few kilometers), then the method's confidence in the location estimate would be high and the error thus low. To derive an error estimate in kilometers the mean distance among the top n ranked locations can be employed.

[13] Available for download: http://www.st.ewi.tudelft.nl/~hauff/placingTask2013Data.html.

2.2.4.2 Results and Insights

An overview of all Placing Task 2013 submissions, including details concerning the specific runs discussed in this section, can be found in the proceedings [31]. In total, 30 runs were submitted by seven different participants. Additionally, the organizers provided two baseline runs.[14] While four participants [11, 28, 43, 47] were able to process the largest test set, two participants [3, 41] opted for the middle ground (processing the third-largest test set with 53,000 test images) and one participant [55] processed the smallest test set. Of the submitted runs, 12 relied on textual metadata, 8 exploited a combination of textual and visual information, and 10 runs used visual information as sole features for location prediction.

As in previous years, relying solely on visual features was not competitive with text features. The best performing submission [43] in this category was able to correctly place less than 5 % of the test images (across all test set sizes) within 100 km of the true location, the maximum error distance we consider useful for the purpose of employing location prediction in practical settings. This approach employed a two-step process that first retrieved candidate images that were visually similar to the target image, and then refined the geoprediction hypothesis by considering additional visually similar images taken in the neighborhood of the candidate images.

In Fig. 2.4 we show the results of a number of submissions on the test set containing 53,000 images. Several points can be made about this figure. First, we note that `CEA_run5` [47] not only utilizes the provided dataset, but a large amount of additional training data was crawled (90 million items) as well. Most importantly, this includes images from users that are present in our test set, which means that this run exploits the same-user signal.

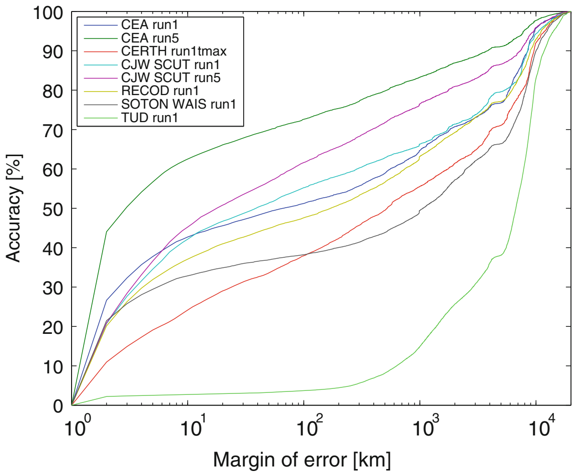


Fig. 2.4 Overview of a selected number of Placing Task 2013 submissions (evaluated on the test set containing 53,000 images)

[14] The baseline runs used out-of-the-box location prediction software: https://github.com/chauff/ImageLocationEstimation, with geographic filtering enabled.

Further, we point out that CJW_SCUT_run5 [3], although not relying on external training data, makes use of two aspects: the user's home location (if available) as well as the relationship between the test images themselves. A test user might contribute 100 images to the test set with only 20 having any tags associated with them—under this approach, the locations predicted for images with tags are also distributed to those images without tags if they have been taken in a short time frame. While the home location has been employed in previous years, the use of the relationship of images *within* the test set is relatively new, having been previously exploited by ICSI in 2012 [4].

Finally, run CJW_SCUT_run5 [3] shows that large gains are also possible when considering the relationship between test images themselves as well as additional user information. In contrast, most participants in 2013 who exploited textual metadata focused on the best way to model regions, although the change in effectiveness across the different modeling strategies was small.

The MediaEval 2013 Placing Task yielded several important overall insights. First, the organizers' baseline (based on an algorithm proposed in 2011) performed very well: across all runs and evaluation metrics relying solely on the provided data set, it always ranked second or third. This is a somewhat disappointing result, and indicates that forward progress in the task is quite slow.

Second, the test set, although randomly sampled, has a strong influence on the reported metrics, especially for the *ErrorMedian* metric, which is less stable than all metrics relying on a radius based error measure. As an example, while the median error of recod_run1 is 509 km for the smallest test set (5,300 items), it is 168 km for the middle test set (53,000 items). In contrast, *Error* 10 km is considerably more stable, changing from 32.5 to 37.6 % across the two test set sizes.

Third, we also note that despite the random sampling of test cases, some test sets are easier than others—in particular, test set three (53,000) shows the best performance for each run across all evaluated runs. This indicates, that results achieved on particular sub (or super) sets of images cannot be directly compared.

As mentioned above, the run CEA_run5 shows the influence the use of training data which *includes images from test users* has. *ErrorMedian* now ranges between 1.9 and 2.7 across test cases, while *Error* 10 km ranges from 58 to 63 %. This error magnitude is in line with previous years' results where training and test users were usually mixed. We stress that although increasing the training corpus also yields small performance gains, the largest gain is achieved by including images of test users in the training data—many users have a particular way of tagging, not only including geographical or in general dictionary terms but they also include long tags (compound phrases meshed together), nicknames or simply tags that only make sense to them. Those tags stand out, and make the location estimation task considerably easier. Thus, we conclude that by partitioning our data set according to training and test users, the task is considerably harder to solve. Exploratory experiments conducted post hoc suggest that after about 2 million sampled training images, the performance gain when adding more training data slowly levels off (assuming that the partitioning of training and test users is intact).

Finally, we turn to the Placeability subtask, which was attempted by one participating group [11]. Similarly to the intuitive example provided earlier, the error is estimated to be large for test images whose location, modeled as a Gaussian, has a high variance. The results of this task, with the linear correlation coefficient varying between 0.06 and 0.37, depending on the evaluated run, indicates that the task itself is feasible. At the same time, the moderate correlation achieved also points to the difficulty and the need for future work: in order to employ such error estimate for any practical means, such moderate correlation is not sufficient.

2.2.5 Placing Task 2014: Horizons

Currently, MediaEval is preparing to launch the 2014 edition of the Placing Task. Organization of the task was assumed by Bart Thomee, Jaeyoung Choi. Gerald Friedland, and Liangliang Cao. The MediaEval 2014 Placing Task[15] will make use of the Yahoo Flickr Creative Commons 100M dataset.[16] The dataset, also referred to as YLI, was created in collaboration with ICSI Berkeley[17] and the Lawrence Livermore National Laboratory[18] [53]. In this section, we provide a list of highlights that can be expected in the 2014 task.

Reintroduction of video: The 2010–2012 editions of the Placing Task focused on geolocation prediction for video, and included both videos and images as training material. The 2013 edition of the task, however, focused on data volume and to this end collected millions of photos for inclusion in the dataset, although at the expense of excluding videos. The exclusion was necessary due to the difficulty in downloading videos in large quantities and the required disk space to store them. In 2014, the task organizers worked to solve download and storage problem. They were prompted to do this by demand among task participants, especially from those that aim to exploit audio and motion features. The 2014 edition will therefore once again feature videos in addition to images, including them both in the training and test sets.

Evaluation granularity: In the previous editions of the task, the location predictions were evaluated to be correct within distances of 1, 10, 100, 1,000 and 5,000 km. However, since the accuracy of geotagging has been shown to be much more precise [17] and since higher granularity is a must for various purposes, the 2014 task additionally evaluates the predictions within distances of 10 m and 100 m.

Russian dolls: The 2014 test set will be again created according to the "Russian Dolls" approach that was introduced in the 2013 task. This approach divides the test set in multiple sets of varying size each, where each larger set is a superset of all

[15] http://multimediaeval.org/mediaeval2014/placing2014.

[16] http://webscope.sandbox.yahoo.com/catalog.php?datatype=i&did=67.

[17] http://www.icsi.berkeley.edu/icsi/.

[18] https://www.llnl.gov.

smaller sets. The Russian Dolls approach enables participants that do not have access to sufficient computational power and/or storage space to pick the largest subset they can handle, while at the same time allowing the evaluation results of all participants to at least be comparable on the smallest subsets they all have in common. This scheme proved successful in the 2013 edition, where one participant used the smallest test set of 5K photos and two participants used second smallest test set of 50K photos, while the remaining four participants used the largest test set of 250K photos.

Dataset contents: The dataset of the 2014 edition will be the largest and most complete in comparison with the previous tasks. The collection will contain thousands of videos, millions of photos, device metadata, textual metadata, visual features, and audio features. The dataset will be hosted in the cloud rather than on a single workstation or server to ensure high data availability and fast downloads for the participants.

2.3 Future Challenges for Geoprediction of Social Multimedia

The experiences that were gathered over the years of the Placing Task have allowed us to formulate a series of challenges that need to be addressed in order to further improve technology for multimodal geolocation. The ultimate aim of placing algorithms is to generate geolocation information that is truly useful for users within applications. For this reason, we view the challenges of placing as fundamentally determined by the needs of users. In order to ensure that new technologies are directly satisfying human needs, we allow the user perspective to drive our formulation of future challenges for placing.

In this section, we present a picture in which human understanding of the relationship between geolocation and social multimedia gives rise to technical challenges. Our discussion is organized into two parts. The first discusses the different types of relationships that exist between a multimedia item an a location. It uncovers new notions of 'place' that need to be taken into account if our systems are to predict geolocations that are maximally meaningful for users. The second discusses the connection between the needs of users and the definition of the Placing Task. Such considerations are necessary in order to ensure that we are addressing a Placing Task that is future proof, in the sense that it will continue to produce technologies that are relevant to user needs, as the use of location-related social multimedia continues to grow.

2.3.1 Further Development of Definitions of "Place"

Upon first consideration, defining "place" (i.e., location) seems nearly trivial: it is a set of geocoordinates that encode longitude and latitude. In the following discussion, we will see that there are actually many different notions of place. This variety exists because concepts of place arise from different sources. One source is geolocation as recorded by the devices that capture multimedia. Another source is human interpretations of geolocation. These sources do not always agree with each other. We discuss each in turn, and then explain why multiple notions of "place" are important for the geolocation prediction for social multimedia, and why it is also critical that we avoid conflating them, but rather that researchers maintain awareness that they are distinct from each other.

2.3.1.1 Automatically Captured Geocoordinates

From a technical point of view, the geocoordinates of the camera, or other capture device, used to record a multimedia item provide a straightforward definition of the location of that multimedia item. Information in the form of geocoordinates of the capture device makes an important contribution to the task of placing. It is an easy-to-understand source of location information. More importantly, it allows the problem of geolocation of social multimedia to be studied at large scale, since it is generated without human effort and is associated with a large number of images and videos that are shared by users on-line.

If the performance of geolocation predictions algorithms is to be judged against automatically captured geoinformation, the quality of geocoordinate information is critical so that geolocation prediction algorithms can be reliably assessed. However, limitations exist with respect to the ability of recording devices to capture location information, and media formats to encode location. These limitations can be organized along three dimensions that characterize location description: accuracy, precision, and granularity. Accuracy relates to systematic error between the stated location and the actual location. For automatic systems, accuracy describes how closely the coordinates given by the device match the actual location. The accuracy of the location captured by a recording device can be improved by collecting more readings over longer time periods, or by using more sophisticated signal processing techniques. In the US, the horizontal accuracy of civilian GPS service, Standard Positioning Service (SPS), is often within $\sim$1 m.[19] An augmentation system can provide even greater accuracies.[20]

Precision refers to how close individual location readings are to the mean of the reading. The precision of a capture device determines the degree to which a location can be defined (e.g., the number of decimal places in which the geocoordinates can be expressed). It quantifies the random error of geolocation capture. For satellite- and triangulation-based mechanisms, precision is determined by the capability of the system. For placing multimedia within the scope of the entire planet, the precision of geolocation devices is usually sufficient to encode a location such that it distinguishes it from other potential nearby locations. However, for placing with smaller scope,

[19] http://www.gps.gov/systems/gps/performance/accuracy.

[20] http://www.gps.gov/systems/augmentations.

such as being able to accurate distinguish locations inside a building (e.g., statues within a museum), current techniques are not sufficient.

Granularity refers to the geometric correctness-of-the-location description, be it a single point, polygon or polyhedron. The majority of approaches for capturing and encoding location information do so by storing a single geographic point representing the location of the camera. As a result, it is challenging to design placing tasks for social multimedia that move beyond the straightforward assumption that the position of the camera defines the "place" of a photo. Currently, no recording technologies are available that would make possible to create a large-scale social multimedia dataset annotated automatically with geometrically correct descriptions of the locations of objects. Theoretically, it may be desirable to describe an object, and especially a large object like a range of mountains, with bounding rectangle, a planar polygon or even a 3D polyhedron. Whether or not such geolocation descriptions are important for the Placing Task will ultimately depend on whether users find that they provide added value for social multimedia collections.

2.3.1.2 Beyond Automatically Captured Geocoordinates

In general, the geocoordinates associated with social multimedia on-line derive from two sources. First, automatically captured geocoordinates, just discussed, and, second, coordinates that are entered by the users at the moment that the images are uploaded. The possibilities for manually associating multimedia with geocoordinates are illustrated by Flickr, which provides users with a map interface. This interface allows them to position their images on a map, which results in an image being assigned geocoordinates, i.e., a geotag. Flickr records the location of photos at 16 levels of accuracy. In the Flickr API documentation, 1 is described as *World*, 3 as *Country*, 6 as *Region*, 11 as *City* and 16 as *Street*.

Location information added by users is dependent on a number of factors. First, it is important to note that the accuracy of manual annotation systems depends on how well the interface maps the provided coordinates to a location on the planet. In the case of manual annotation, the precision is limited only by the interface of the annotation system, be it either a case of adding numerical coordinates directly, or asking a user to select a point on a map. Second, users may be protecting their privacy by choosing to geotag their images at a relatively low level of precision. Third, users may have a limited amount of time and effort to devote to geotagging images, and decide to assign a batch of images to a single location though some of them might not have been taken at that location.

The overall combination of automatically captured geocoordinates and geocoordinates assigned by users results in a wide variation of the characteristics of the geolocation information associated with multimedia. More studies, such as the one on the accuracy of geocoordinate information on Flickr was carried out by [17], are necessary in order to gain a complete understanding of the underlying patterns. Minimally, it is important to be aware that when geotagged images are collected from

Flickr to use to evaluate placing algorithms, the underlying patterns of geotags will impact the measurement of the performance of the system.

Ultimately, the Placing Task will strive to go beyond the geocoordinates that are currently available associated with collections of social multimedia on-line. Specifically, the information should transcend the limitations imposed not only by capture devices, but also by metadata encoding standards. As an example of a common encoding standard, we mention Exchangeable Image File format (Exif), which was initially released in 1995. Exif can encode a wide range of information including technical status values of the capture device (such as aperture, shutter speed for photos) that can be added without human intervention, as well as location information. Currently, however, Exif can encode a single creation location using latitude, longitude and altitude coordinates, as well as a single "destination" location, commonly interpreted as a location pertinent to the content of the image.

Here, we would like to point out several directions that this expansion could take. First, we point out that GPS systems are capable of generating more information than a mere set of geocoordinates. If the error of the capture device is known, it would be helpful to keep information about the error associated with the captured multimedia. When automatically recorded geocoordinates are used as ground truth for evaluating geolocation prediction algorithms, the error limits the resolution with which it is possible to use the ground truth to measure performance.

Second, elevation has an important role to play in defining place. For example, multiple floors in the same building with have the same geolocation, but different elevation. Currently, there is no simply, widely-used manner to add elevation information to social multimedia, either automatically with consumer capture devices, or manually via input interfaces.

Third, it is possible that multiple locations are important for an image. Returning to the example of the mountain range mentioned above, this could be the locations of the individual mountain peaks. Multiple locations are certainly important for video, since the position of the camera may move as the video is recorded. A single set of geocoordinates is not capable of characterizing the capture location of a video, but rather, an entire geopath is necessary.

The catalog of possible descriptions of locations is a long one. Researchers developing placing algorithms need to be able to prioritize the descriptions that they focus on in their research. A great deal of the time, researchers are well served by focusing on using the descriptions of locations that are most readily available at large scale. This point explains the focus in the literature on using the geotags of photos on Flickr as ground truth. Despite the shortcomings of the geoinformation associated with online photos mentioned above, there is no other source of ground truth that would allow research datasets of comparable size to be created. However, to the extent to which resources for datasets are available, it is possible to take other priorities into account. Specifically, we focus on descriptions of locations that will best capture the way in which users understand the concept of location as it is related to social images.

2.3.1.3 Human Judgements and Interpretations of "Place"

The move towards human views on location starts with the question, "What do we consider to be the *location* of a multimedia item?" Asking this question allows us to uncover *georelevance*, which we define as the connection made by a user between an image or a video and a place. Here, we examine some ways in which people perceive how location is related to multimedia, and reflect on how these correspond to different types of georelevance.

Perhaps the simplest contrast between two types of georelevance is the difference between the location of the camera and the location of the subject material depicted by the multimedia content. This distinction was already mentioned above, in reference to the example of the image of a mountain. We turn to discuss that example again now. The camera that is used to take a picture of the mountain could be positioned kilometers from the mountain itself. It should be noted that the difference in location between the photographer and the subject of the photo is regarded as an inherent fact of photography, and not an exception. It is a common procedure when taking a photo to move away from the subject in order to capture it better. We should be careful in assuming that by default the photograph is at approximately the same location as the subject.

It is easy to ignore the difference between the position of the photographer and the position of the subject. Humans generalize so quickly from different views of objects to the identity of the object themselves. For example, we can recognize the same house taken from different points of view without effort. It is easy to imagine that this interpretation takes place with no conscious awareness of the position of the photographer. This effect can explain why people looking at photos and videos are not specifically aware of the difference in position of the cameraperson and the subject material. That difference is certainly not relevant for interpretation.

More sophisticated notions of relevance require more detailed consideration of the connection between social multimedia and users. Researchers who study social multimedia often study so-called "found" content: content collected from the Internet. It is important to keep in mind that such content is not "found" in a vacuum. Rather, social multimedia comes into existence because it was captured by users. As such, it is natural that a given multimedia item (i.e., an image or a video) will be interpreted differently by different people, and that this different interpretation will also be relevant to place.

In order to understand how the relationship between a person and a multimedia item impacts how that person perceives the connection of that item to location, it is helpful to consider the case of people judging the location of a particular photo. We consider the example of the image in Fig. 2.5, taken in Pisa, Italy.

The relationships that users have with an image determine the evidence and techniques that they have at their disposal for making the decision of the location of that image. Relationships of people to images can be considered to fall into different classes. The classes should not be considered absolute, or necessarily mutually exclusive. However, the existence of these classes shows that different perspectives on location exist among social multimedia users. The following is a list of ten

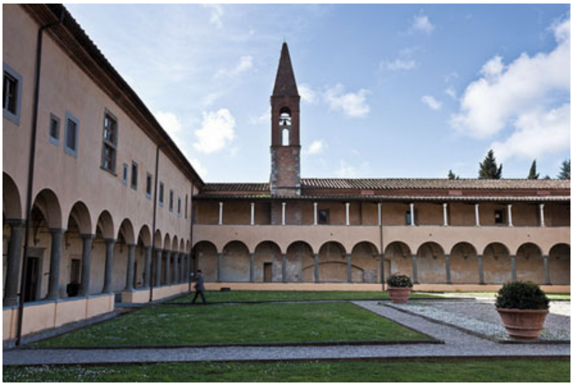


Fig. 2.5 Image taken in Pisa, Italy (Flickr: PGBrown1987)

different statements, which could conceivably be made by ten different people (P1–P10), regarding the ways in which they could judge that the location of the image is Pisa.

- P1: I took the picture and when I see it, I remember it.
- P2: I was there when the picture was taken and when I see it, I remember this moment.
- P3: I have been there, and I remember what the place looks like.
- P4: Someone told me about a picture that was taken in Pisa and there is something that I see in this picture that tells me that this must be the picture that the person was talking about.
- P5: I know of another picture that looks just like this one and it was labeled “Pisa”.
- P6: I’ve seen other pictures like this an recognize it (the specific buildings that appear).
- P7: Someone who has been there described this place to me, and on the basis of what we discussed about the place, I recognize it in the picture.
- P8: I’ve been there and recognize characteristics of the place (the type of architecture).
- P9: I am an Internet user, and I can formulate either keyword queries, or use content-based image retrieval to find information that will allow me to identify the picture as Pisa.
- P10: I am a multimedia forensic expert and have established a chain of logic that identifies the place as Pisa.

From these statements, it is clear that a person in class P1, who knows the location of the picture because they took it themselves, will give a different type of answer than a person who is relying on descriptions or other external information. Further, a person who is relying on information will judge the image differently depending on what type of information is available: is it a near duplicate of the same photo, which can be compared side by side, or is it a set of clues concerning details in the pictures that would elude a human judge who was not inclined to detective work.

In the literature, it is conventional to assume a ground truth that is generated by a certain class of users. For example, in formulations of the task of placing that use uploader contributed geocoordinates, the assumption is that users are in class P1. The work of Choi et al. [8], comparing human and machine abilities to generate geopredictions assumed people belonged to class P9. Specifically, in order to collect judgements on human geoprediction performance, crowdsourcing workers were asked to judge images using resources from the Internet, including Google Maps and Streetview.

If different groups of users have different relationships to images, it is not surprising that they also have different notions of the definition of place that is relevant to an image. Another example is helpful in order to communicate this point.

Consider an image of a house. A user could describe this house as “my grandmother’s house”. This definition of place is relative to a particular user. It would be most likely for someone who had taken the picture themselves, or had spent time in the house (approximately corresponding to P1–P4 above). Such a person would judge a geolocation prediction system to have failed if the prediction was off by a few meters (e.g., if the house were assigned the geocoordinates of the neighbor’s house). Another user could describe this house as a “southern bungalow”, referring to a particular architectural style (approximately corresponding to P5–P10 above). This person would judge a geolocation prediction system to have succeeded if it located the neighbors’ house, and quite possibly would be quite satisfied if it correctly placed the image in Florida, USA.

Currently, work in the area of geoprediction conflates these two notions of relevance. It is an open question whether work being done to optimize systems for one type of georelevance directly helps to advance systems optimized for another.

In order to develop detailed and well-supported notions of georelevance, it will ultimately be necessary to carry out careful studies of user interpretations of multimedia location and their location-related multimedia needs. However, we point out already that some findings of such studies may be surprising. For example, it is not unlikely that a large number of users consider an image of a sand dune to be a georelevant depiction of the whole Sahara desert, and that the exact location of the sand dune is relevant only to a minority of users. In this way, geolocation prediction for social multimedia stands in clear contrast with geolocation prediction for military or scientific purposes, in which the exact location may be critical.

Such examples should not automatically be assumed to be exceptions. Social multimedia contains many images which were taken for the purpose of artistic value or emotion impact. Think of a beautiful image of a starry night. An astronomer would be bothered by a small deviation in the accuracy of a geoprediction of the camera position, but a user who is looking for an inspiring or relaxing image would not feel the image to be relevant to any particular geocoordinates. Such considerations would also apply to images of fantastic landscapes, or photographs made as art.

Moving forward, how users interpret place will be strongly impacted by the types of systems that are available for them to interact with geotagged multimedia. As an example of this impact, we consider the case of video. A multimodal location prediction system may be capable of generating a geopath that traces the movement of the camera during the video. However, unless there is some way to visualize this path for the user, the existence of the path is not useful. A dot tracing the position of a video on the map as the video plays is an interesting solution. However, many questions are left open, such as how to display multiple videos at once.

While allowing more comprehensive geometries to be used to describe a location does not solve this problem, it does give those annotating media better tools for describe location better suited to their media. In addition, explicit annotation for media that have no logical location associated with them would also be valuable. This would help distinguish between media that have no sense of location and those that have not had any location annotated yet.

In summary, users' interpretations of the place that are most relevant for a photo will depend on to which other photos it is being compared, and also on the way in which the user is looking at the photo, or the purpose for which the photo should be used. In the context of annotating multimedia, the existing state of correctness-of-location could possibly be considered sufficient for current needs. However, as technology develops and users begin to demand more from their media collections and systems that handle them, more comprehensive location metadata schema may be required.

Systems will need to be able to serve users who do not present geocoordinates as queries, but rather designate locations by other means. For example, the location of a photo of a person's house may be described by a set of unambiguous coordinates, but also by a postal address, or even a description tied to a specific user, such as "Grandma's House". Multimedia systems need to be able to handle the translation between objective, computer friendly descriptors and subjective, human friendly location descriptors. This would allow multimedia information retrieval systems to take in location queries in forms that are intuitive to humans and be able to search an index of media that have formal location descriptions. The context of the query as well as the personal profile of the user would need to be taken into consideration to deliver effective results.

If the user does not have access to a map that can be referenced for the given lat/lon pair, it is more useful to provide the answer in more practical, abstract or conceptual way such as the postal address or a "tall red building".

2.3.2 Definitions of the Task of Placing Social Multimedia

An important part of allowing the user perspective to drive our formulation of future challenges for "placing" is understanding the underlying patterns of social multimedia. Social multimedia collections arise due to human behavior, rather than being constructed according to a overall plan or premeditated purpose. For this reason they may be influenced by a range of factors that may not be immediately obvious upon first consideration. Here, we point out that definitions of the Placing Task that do not take these factors into consideration, risk promoting the development of algorithms that address and artificial problem, whose solutions may not transfer to real-world use scenarios.

A key observation is that different multimedia sharing platforms cater to different communities with different values and, as such, can be considered to have specific cultures. The culture of the platform, for example, includes the extent to which it fosters photography as a hobby, in contrast to other reasons for capturing and sharing images. Also, the technical possibilities of the platform, for example, how easy it is to upload videos, and associate them with tags and geotags, have a large impact on the characteristics of multimedia collections, including patterns of user contributed metadata. Finally, governments impose restrictions around the world on the collection of geotagged multimedia. The definition of what constitutes security risk varies from region to region, and can be expected to have an impact on social multimedia data collections.

The usual reflex of a researcher would be to understand the impact of these factors by making a systematic comparison between multiple datasets and data sources. However, this method of understanding variation between datasets is limited when studying geolocation for social images. As a multimedia sharing platform, Flickr large and so widely used. Other large social multimedia sharing services are able to develop alongside Flickr only to the extent that they serve a different purpose, or supported a different group of users or type of sharing culture. In short, researchers must study social multimedia sharing without having a large number of examples of social multimedia collections of the same type, which would make it possible to formulate a set of properties characteristic of a specific social multimedia collection. The study of social multimedia is the study of specific collections, and not the study of an ideal.

For this reason, it is important when creating a dataset that is to be used to develop geolocation prediction algorithms, that the underlying characteristics of the social multimedia collection from which it is drawn are preserved as much as possible. In other words, dataset development should leave "found data" as much as possible in the context in which it was found. The more that we change the data from its "in the wild" state, the less certain we can be that we are developing algorithms, or applications, that are relevant to multimedia that users actually produce and are appropriate for the patterns with which they produce it.

Unfortunately, it is easy to get frustrated with "found data" because its properties are a priori unknown and out of control of the researcher. The natural temptation is to seek to somehow "improve" it. This temptation arises from our drive, as researchers, to have the best possible dataset on which to carry out our research. It is critical, that we define "best possible" dataset by considering the ultimate goal of our research, and not focusing on what characteristics that our algorithms require in a dataset in order to perform well. However, before we begin to devote effort to address a particular "drawback" or "shortcoming" of our dataset, it is important to give careful consideration to whether or not that shortcoming is a reflex of the underlying problem to be solved.

An example serves to illustrate issues that can arise when we are overhasty to identify an aspect of a dataset as a weakness. If we are tackling the goal of creating a better search and browsing functionalities for users of social multimedia websites, we may feel that one "shortcoming" of the dataset is the fact that people take pictures inside buildings, which possibly contain very little information distinctive for location. If we "improve" our dataset by discarding all images taken inside buildings, we actually have changed that underlying problem. We will no longer be working toward algorithms that have a chance of improving search and browsing functionalities overall, but rather only for a portion of all user-uploaded images. Changing the dataset changes important aspects such as the prior probability of certain locations, the composition of the negative class, and the relative proximity of images to each other in feature spaces, such as the visual feature space. In practice, difficult challenges may be tackled by isolating subchallenges and addressing them individually. However, the overall implications of simplifying the dataset must be carefully considered.

We can formulate best practice with the following statement: Researchers should decide on the ultimate goal of the research independently of the collection of the dataset, and where every possible datasets should be collected "in the wild" from the communities that are ultimately meant to benefit from the algorithms that researchers develop. Decisions to change the distribution of a dataset should not be viewed as "improvements," but rather should clearly be given the status of informed modifications of the original data, carried out with the purpose of accomplishing a subgoal. Note that because the multimodal geolocation requires such large amounts of data, the decisions to "change" datasets from their naturally occurring state does not take place after the data has been collected, but rather during the collection process. It is important to reflect thoroughly on whether the methodology used for crawling or sampling pushes the dataset away from the "use case" corresponding to the ultimate goal of the research.

2.4 Conclusion and Outlook

This article has presented a retrospective on the Placing Task offered by the MediaEval Multimedia Benchmark. We have traced the first four years of the benchmark 2010–2013, and provided an outlook on 2014. In this final section, we summarize essential points concerning the ground covered by the task thus far, and anticipate the new challenges that it will tackle in the immediate future.

2.4.1 Where Placing has Been

The Placing Task has successfully provided large-scale datasets to the multimedia community that have supported cross-site comparison of multimodal location prediction algorithms.

Several important principles have emerged that transcend specific algorithms. One is that two-stage approaches are highly effective. The first stage creates a set of candidate hypotheses and the second stage seeks to refine these hypotheses. The first two-state approach was introduced by Olivier Van Laere in 2010 [58] and has also been exploited by Michele Trevisiol and colleagues [56]. Another principle is that the test data can be jointly exploited to address issues of data sparsity. Both Jaeyoung Choi and colleagues in 2012 and Jiewei Cao in 2013 exploited this principle within two different algorithms. Finally, the importance of user modeling has been made clear, both by approaches that build models of past tagging behavior and approaches that default to users' home locations. Here, notable contributions include [7] in 2010, [60] in 2011, and [48]. In 2013, the task was designed to encourage participants to emphasize aspects other than the same-uploader signal.

The MediaEval Placing Task has successfully guided researchers to explore new areas. In 2010, only a single group (TU Berlin [24]) made use of visual features alongside image metadata. By 2013, over half of the submitted runs used a combination of textual and visual features, or visual features alone. Likewise, the task has successfully guided researchers away from less promising areas. Already in 2010, [7] and the follow-up work [14] reveals that the use of knowledge resources such as gazetteers that contain lists of geographically related words, are not the silver bullet for addressing the task of geolocation social multimedia. Instead, if enough data is available, data-driven approaches will outperform approaches that use knowledge resources. As a result, MediaEval participants used gazetteers judiciously, and did not automatically assume that knowledge resources were necessary to implement effective text-based geolocation prediction approaches.

Other results achieved by MediaEval involve the research community as a whole, rather than individual algorithms. The fact that researchers worked together within the framework of a benchmark rather than as individuals has strengthened the research in this area. The Placing Task 2010–2013 demonstrates that researchers can successfully improve their approaches via discussion with other teams at the yearly workshop, and results on extended algorithms have gone on to be published in mainstream venues.

The benchmarking framework is also important in driving forward the state of the art. Specifically, the Placing Task experience has shown that encouraging state-of-the-art baseline works: When provided with specific papers and code, participants did better at starting where others had left off, rather than re-inventing the wheel. Lowering the threshold works: Sites could participate that would not otherwise have been able to take part, had they not had access to data, features, and also had the "Russian Doll" approach allowed them to work on a smaller dataset. Students benefit from the support of the community as they carry out their thesis research.

Finally the Placing Task has brought benefits to the research community by opening the door to exploration of new topics. It is a laudable goal to make social multimedia more easily findable and browsable by annotating it with location information. However, it is also critical to remember that geoinformation does not always make a positive contribution. Many Internet users avoid geotagging their videos and photos, or turn the GPS functionality of their phones off, since they do not want the location of their media items to be known. An important development was that ICSI was able to use the MediaEval 2010 dataset in order to move forward with its work in the area of privacy [14].

2.4.2 Where Placing Is Going

As discussed in Sect. 2.3, researchers developing automatic geoprediction algorithms for social multimedia stand before a large number of challenges. We have argued that the most worthwhile future challenges are those driven by user needs or user behavior.

We have discussed a variety of notions of georelevance that arise from how people interpret images. These new notions of georelevance will also require new evaluation metrics. Currently, the Placing Task evaluates the quality of a prediction by calculating the Haversine distance between the predicted geocoordinates and the ground truth. Distance does not necessarily correspond to human perceptions of relevance, however. We have already mentioned the importance of the relationship of the person to the multimedia item in determining their perception of georelevance. However, the problem cannot be solved by merely enlarging the evaluation radius, but rather researchers must dig deeper.

One effect that they will uncover is that human interpretations of georelevance can be assumed to be spatially discontinuous. Users may consider an image to be accurately geolocated if its location is predicted to be anywhere along the shore of a lake. However, if the location is predicted to be in the middle of the lake, users would feel that the geolocation is inaccurate. Such judgements are clearly related to the semantic content of the example.

In the future the Placing Task will need to decide the extent to which it is addressing a challenge of image understanding. When humans interpret images, it is possible for them to find image content to be relevant to a geolocation without the image content being physically located at the geolocation. Clearly, if a geolocation prediction system identifies a picture of a cowboy in New York City as “Texas” would be less disturbing for a user than geotagging the Statue of Liberty “Texas”. User studies are necessary to understand the extent to which users expect and are able to use systems that can predict geoconnections going beyond physical location.

In any case, future work will necessarily involve a better understanding of the properties of social multimedia data that allow geocoordinate prediction systems to work. For example, in the case of visual features, it remains unclear if systems should be optimized to identify near duplications, to match scenes, to match specific landmarks, or to match types of objects. This choice will determine the types of visual features that are best suited to the task.

Overall, geoprediction stands to benefit from a better understanding of when particular approaches are most likely to work well and when they should be avoided. For example, currently, many systems apply the same approach to predicting the geolocation of multimedia items in regions that are represented with a wealth of multimedia content to multimedia items where data is sparse. Instead, a system could first predict whether it is confronting a “easy” or “hard” prediction, and then react accordingly. The initial groundwork has been laid for such approaches by the Placing Task 2013 Placeability subtask. Further, the use of audio for predicting the geolocation of social video has delivered some initially promising results. Here, it is still necessary to understand the types of situations in which audio can be used to the best advantage.

In all cases, algorithms must anticipate the challenges they face as the amount of available social multimedia grows larger and perhaps even changes in composition. In this chapter, we have focused heavily on Flickr. This focus is primarily dictated by the availability of the data for research purposes. However, many other sources of online multimedia exist, including: YouTube,[21] Facebook,[22] Vimeo,[23] blip.tv,[24] Instagram,[25] and Vine.[26] New patterns in multimedia collections can be expected as new capture devices are introduced (e.g., sports cameras, Google Glass) and users change their capture and sharing habits.

In the face of increasing amounts of multimedia data, it is important to remember that ultimately we are best served by not only reflecting on where we can obtain more data, but also, how we can be sure that we are using the right data. Ultimately, the Placing Task must be steered by the geoprediction needs of users for multimedia-associated geoinformation. We should be alert to cases in which users needs can be addressed by a smaller number of specific multimedia items. Such cases could be expected to arise, for example, in geolocation systems for constrained location, i.e., geolocating images taken within a museum.

Moving forward, it will be important to pursue new avenues of research that are opened by the MediaEval Placing Task. As already mentioned above, developing algorithms that are capable of automatically prediction the geolocation of social multimedia has important implications for user privacy. In view of these implications, it is important to study not only the geolocation problem, but also the inverse challenge. Addressing this challenge involves determining if it possible to maintain those properties that make multimedia items understandable and interesting to users, while at the same time hiding the information necessary to produce geopredictions. We refer to this challenge as “Geo-Cloaking”. A geocloaking task could involve merging the Placing Task with another task offered at MediaEval, namely Visual Privacy [2]. The Visual Privacy task seeks to develop methods of obscuring video that are acceptable to human viewers, while at the same time also obscuring person information. In order to test the effectiveness of such systems, state-of-the-art geolocation prediction algorithms are necessary.

Another interesting avenue is to turn the Placing Task on its head. Instead of inferring information about an multimedia item in the form of a location, it is possible

[21] https://www.youtube.com.

[22] https://www.facebook.com.

[23] https://vimeo.com.

[24] http://blip.tv.

[25] http://instagram.com.

[26] https://vine.co.

to conceptualize the task as inferring information about a location by gathering information about multimedia items. Ghent University has taken the lead in work dedicated to learning geographically relevant information by using georeferenced social media [30]. This perspective is an interesting one for future pursuit.

We close this chapter with a few words on the larger implications of the MediaEval Placing Task. We have seen in this article that the Placing Task represents a multi-year, international cooperative effort to solve the overall problem of predicting the geocoordinates of a social multimedia item, wherever in the world that item might be located. It distinguishes itself from other initiatives in the area of geocoordinate prediction by its focus on large collections of social multimedia—it seeks to solve the problem using "found" datasets gathered from online multimedia sharing platforms and transferred as directly as possible into a task. The most advanced resource imaginable to tackle this problem would be a multimedia collection containing every item available in the world. If individual research sites were to develop such a collection, few would be able to afford to carry out Placing research. The Placing Task can, for this reason, be considered to share similarities with future-oriented initiatives that are formulated on a grand scale, such as the International Space Station.[27] Such projects require an international coalition which comes together to build and maintain a resource that allow scientific research to be carried out that would not be possible any other way. The Placing Task has made a strong start, and its organizers look forward to further growth in the future. As they push forward the state of the art in geolocation of social multimedia, the sky is the limit.

References

1. J. Almeida, N. Leite, R. Torres, Comparison of video sequences with histograms of motion patterns, in *18th IEEE International Conference on Image Processing (ICIP)*, September 2011, pp. 3673–3676
2. A. Badii, M. Einig, T. Piatrik, Overview of the MediaEval 2013 Visual Privacy Task, in Larson et al. [31]
3. J. Cao, Photo set refinement and tag segmentation in georeferencing Flickr photos, in Larson et al. [31]
4. J. Choi, V. Ekambaram, G. Friedland, K. Ramchandran, The 2012 ICSI/Berkeley video location estimation system, in Larson et al. [35]
5. J. Choi, G. Friedland, Data-driven vs. semantic-technology-driven tag-based video location estimation, in *Proceedings of the 2011 IEEE Fifth International Conference on Semantic Computing, ICSC '11*. IEEE Computer Society, Washington, DC, pp. 243–246 (2011)
6. J. Choi, G. Friedland, V. Ekambaram, K. Ramchandran, Multimodal location estimation of consumer media: dealing with sparse training data, in *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo, ICME '12*. IEEE Computer Society, Washington, DC, pp. 43–48 (2012)
7. J. Choi, A. Janin, G. Friedland, The 2010 ICSI video location estimation system, in Larson et al. [33]
8. J. Choi, H. Lei, V. Ekambaram, P. Kelm, L. Gottlieb, T. Sikora, K. Ramchandran, G. Friedland, Human versus machine: establishing a human baseline for multimodal location estimation, in

[27] http://www.nasa.gov/mission_pages/station.

Proceedings of the 21st ACM International Conference on Multimedia, MM '13, ACM, New York, pp. 867–876 (2013)

9. J. Choi, H. Lei, G. Friedland, The 2011 ICSI video location estimation system, in Larson et al. [32]
10. D.J. Crandall, L. Backstrom, D. Huttenlocher, J. Kleinberg, Mapping the world's photos, in *Proceedings of the 18th International Conference on World Wide Web, WWW '09*, ACM, 2009, pp. 761–770
11. J. Davies, J. Hare, S. Samangooei, J. Preston, N. Jain, D. Dupplaw, P. Lewis, Identifying the geographic location of an image with a multimodal probability density function, in Larson et al. [31]
12. D. Ferrès, H. Rodríguez, TALP at MediaEval 2010 Placing Task: geographical focus detection of Flickr textual annotations, in Larson et al. [33]
13. D. Ferres, H. Rodriguez, TALP at MediaEval 2011 Placing Task: georeferencing Flickr videos with geographical knowledge and information retrieval, in Larson et al. [32]
14. G. Friedland, J. Choi, Semantic computing and privacy: a case study using inferred geo-location. Int. J. Semant. Comput. **5**(1), 79–93 (2011)
15. G. Friedland, J. Choi, A. Janin, VIDEO2GPS: a demo of multimodal location estimation on Flickr videos, in *Proceedings of the 19th ACM International Conference on Multimedia, MM '11*, ACM, New York, pp. 833–834 (2011)
16. A. Gallagher, D. Joshi, J. Yu, J. Luo, Geo-location inference from image content and user tags, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2009, CVPR Workshops 2009,* June 2009, pp. 55–62
17. C. Hauff, A study on the accuracy of Flickr's geotag data, in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13*, ACM, New York, pp. 1037–1040 (2013)
18. C. Hauff, G.-J. Houben, WISTUD at MediaEval 2011: placing task, in Larson et al. [32]
19. C. Hauff, G.-J. Houben, Geo-location estimation of Flickr images: social web based enrichment, in *Proceedings of the 34th European Conference on Advances in Information Retrieval, ECIR'12*. Springer, Berlin, pp. 85–96 (2012)
20. C. Hauff, G.-J. Houben, Placing images on the world map: a microblog-based enrichment approach, in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12*, ACM, New York, pp. 691–700 (2012)
21. C. Hauff, B. Thomee, M. Trevisiol, Working notes for the placing task at MediaEval 2013, in Larson et al. [31]
22. J. Hays, A.A. Efros, Im2gps: estimating geographic information from a single image, in *CVPR*. IEEE Computer Society (2008)
23. J.M. Perea-Ortega, M.Á. García-Cumbreras, L. Alfonso Ureña-López, M. García-Vega, SINAI at Placing Task of MediaEval 2010, in Larson et al. [33]
24. P. Kelm, S. Schmiedeke, T. Sikora, VIDEO2GPS: geotagging using collaborative systems, textual and visual features: MediaEval 2010 Placing Task, in Larson et al. [33]
25. P. Kelm, S. Schmiedeke, T. Sikora, A hierarchical, multi-modal approach for placing videos on the map using millions of Flickr photographs, in *ACM Multimedia 2011 (Workshop on Social and Behavioral Networked Media Access—SBNMA)*, ACM, November 2011
26. P. Kelm, S. Schmiedeke, T. Sikora, Multi-modal, multi-resource methods for placing Flickr videos on the map, in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ICMR '11*, ACM, New York, pp. 52:1–52:8 (2011)
27. P. Kelm, S. Schmiedeke, T. Sikora, How spatial segmentation improves the multimodal geotagging, in Larson et al. [35]
28. G. Kordopatis-Zilos, S. Papadopoulos, E. Spyromitros-Xioufis, A.L. Symeonidis, Y. Kompatsiaris, CERTH at MediaEval Placing Task 2013, in Larson et al. [31]
29. F. Krippner, G. Meier, J. Hartmann, R. Knauf, Placing media items using the XTrieval framework, in Larson et al. [32]
30. O.V. Laere, S. Schockaert, V. Tanasescu, B. Dhoedt, C. Jones, Georeferencing Wikipedia documents using data from social media. ACM Trans. Inf. Syst. **32**(3), (2014)

31. M. Larson, X. Anguera, T. Reuter, G.J.F. Jones, B. Ionescu, M. Schedl, T. Piatrik, C. Hauff, M. Soleymani (eds.), in *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop*, Barcelona, Spain, October 2013, CEUR-WS.org, online http://ceur-ws.org/Vol-1043 (2013)
32. M. Larson, A. Rae, C.-H. Demarty, C. Kofler, F. Metze, R. Troncy, V. Mezaris, G.J.F. Jones (eds.), in *Working Notes Proceedings of the MediaEval 2011 Workshop*, Pisa, Italy, September 2011, CEUR-WS.org, online http://ceur-ws.org/Vol-807 (2011)
33. M. Larson, M. Soleymani, P. Serdyukov, V. Murdock, G.J.F. Jones (eds.), in *Working Notes Proceedings of the MediaEval 2010 Workshop*, Pisa, Italy, October 2010, online http://multimediaeval.org/mediaeval2010/2010worknotes (2010)
34. M. Larson, M. Soleymani, P. Serdyukov, S. Rudinac, C. Wartena, V. Murdock, G. Friedland, R. Ordelman, G.J.F. Jones, Automatic tagging and geotagging in video collections and communities, in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ICMR '11*, ACM, New York, pp. 51:1–51:8 (2011)
35. M. Larson, S. Schmiedeke, P. Kelm, A. Rae, V. Mezaris, T. Piatrik, M. Soleymani, F. Metze, G.J.F. Jones (eds.), in *Working Notes Proceedings of the MediaEval 2012 Workshop*, Pisa, Italy, October 2012, CEUR-WS.org, online http://ceur-ws.org/Vol-927 (2012)
36. M. Larson, M. Soleymani, M. Eskevich, P. Serdyukov, R. Ordelman, G. Jones, The Community and the Crowd: Multimedia Benchmark Dataset Development. MultiMedia, IEEE. **19**(3), 15–23 (2012)
37. H. Lei, J. Choi, G. Friedland, Multimodal city-verification on Flickr videos using acoustic and textual features, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 2273–2276
38. L. Li, D. Pedronette, J. Almeida, O. Penatti, R. Calumby, R. Torres, A rank aggregation framework for video multimodal geocoding, pp. 1–37 (2013)
39. L.T. Li, J. Almeida, R.D.S. Torres, RECOD working notes for placing task MediaEval 2011, in Larson et al. [32]
40. L.T. Li, J. Almeida, D.C.G Pedronette, O. Penatti, R.D.S. Torres, A multimodal approach for video geocoding, in Larson et al. [35]
41. L.T. Li, J. Almeida, O. Penatti, R. Calumby, D.C.G. Pedronette, M.A. Gonçalves, R.D.S. Torres, Multimodal image geocoding: the 2013 RECOD's approach, in Larson et al. [31]
42. X. Li, C. Hauff, M.A. Larson, A. Hanjalic, Preliminary exploration of the use of geographical information for content-based geo-tagging of social video, in Larson et al. [35]
43. X. Li, M. Riegler, M. Larson, A. Hanjalic, Exploration of feature combination in geo-visual ranking for visual content-based location prediction, in Larson et al. [31]
44. N. O'Hare, V. Murdock, Modeling locations with social media. Inf. Retr. **16**(1), 30–62 (2013)
45. J. Oomen, P. Over, W. Kraaij, A. Smeaton, Symbiosis between the TrecVid benchmark and video libraries at the Netherlands Institute for Sound and Vision. Int. J. Digit. Libr. **13**(2), 91–104 (2013)
46. O.A.B. Penatti, L.T. Li, J. Almeida, R.D.S. Torres, A visual approach for video geocoding using bag-of-scenes, in *Proceedings of the 2Nd ACM International Conference on Multimedia Retrieval, ICMR '12*, ACM, New York, pp. 53:1–53:8 (2012)
47. A. Popescu, CEA List's participation at MediaEval 2013 Placing Task, in Larson et al. [31]
48. A. Popescu, N. Ballas, CEA List's participation at MediaEval 2012 Placing Task, in Larson et al. [35]
49. A. Rae, P. Kelm, Working notes for the Placing Task at MediaEval 2012, in Larson et al. [35]
50. A. Rae, V. Murdock, P. Serdyukov, P. Kelm, Working notes for the Placing Task at MediaEval 2011, in Larson et al. [32]
51. S. Schmiedeke, C. Kofler, I. Ferrané, Overview of the MediaEval 2012 Tagging Task, Working Notes Proceedings of the MediaEval 2012 Workshop, Santa Croce in Fossabanda, Pisa, Italy, October 4–5, CEUR-WS.org, ISSN 1613–0073 (2012)
52. P. Serdyukov, V. Murdock, R. van Zwol, Placing Flickr photos on a map, in *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09*, ACM, New York, pp. 484–491 (2009)

53. D.A. Shamma, One hundred million creative commons Flickr images for research. http://yahoolabs.tumblr.com/post/89783581601/one-hundred-million-creative-commons-flickr-images-for, month = June, note = Accessed: 30 June 2014 (2014)
54. A.F. Smeaton, P. Over, W. Kraaij, Evaluation campaigns and TrecVid, in *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, MIR '06*, ACM, New York, pp. 321–330 (2006)
55. S. Subramanian, V. Vidyasagaran, K. Chandramouli, VIT@MediaEval 2013 Placing Task: location specific tag weighting for language model based placing of images, in Larson et al. [31]
56. M. Trevisiol, J. Delhumeau, H. Jégou, G. Gravier, How INRIA/IRISA identifies geographic location of a video, in Larson et al. [35]
57. M. Trevisiol, H. Jégou, J. Delhumeau, G. Gravier, Retrieving geo-location of videos with a divide & conquer hierarchical multimodal approach, in *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval, ICMR '13*, ACM, New York, pp. 1–8 (2013)
58. O. Van Laere, S. Schockaert, B. Dhoedt, Ghent University at the 2010 Placing Task, in Larson et al. [33]
59. O. Van Laere, S. Schockaert, B. Dhoedt, Finding locations of Flickr resources using language models and similarity search, in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ICMR '11*, ACM, New York, pp. 48:1–48:8 (2011)
60. O. Van Laere, S. Schockaert, B. Dhoedt, Ghent University at the 2011 Placing Task, in Larson et al. [32]
61. O. Van Laere, S. Schockaert, B. Dhoedt, Georeferencing Flickr photos using language models at different levels of granularity: an evidence based approach. J. Web Semant. **16**, 17–31 (2012)
62. O. Van Laere, S. Schockaert, B. Dhoedt, Georeferencing Flickr resources based on textual meta-data. Inf. Sci. **238**, 52–74 (2013)
63. O. Van Laere, S. Schockaert, J. Quinn, F. Langbein, B. Dhoedt, Ghent and CARDIFF University at the 2012 Placing Task, in Larson et al. [35]

Chapter 3
Large-Scale Image Geolocalization

James Hays and Alexei A. Efros

Abstract In this chapter, we explore the task of global image geolocalization—estimating where on the Earth a photograph was captured. We examine variants of the "im2gps" algorithm using millions of "geotagged" Internet photographs as training data. We first discuss a simple to understand nearest-neighbor baseline. Next, we introduce a lazy-learning approach with more sophisticated features that doubles the performance of the original "im2gps" algorithm. Beyond quantifying geolocalization accuracy, we also analyze (a) how the nonuniform distribution of training data impacts the algorithm (b) how performance compares to baselines such as random guessing and land-cover recognition and (c) whether geolocalization is simply landmark or "instance level" recognition at a large scale. We also show that geolocation estimates can provide the basis for image understanding tasks such as population density estimation or land cover estimation. This work was originally described, in part, in "im2gps" [9] which was the first attempt at global geolocalization using Internet-derived training data.

3.1 Introduction

Is it feasible to estimate the location of generic scenes? One of the main questions addressed by this study is as much about the Earth itself as it is about computer vision. Humans and computers can recognize specific, physical scenes that they've seen before, but what about more generic scenes that may be impossible to specifically localize? We know that our world is self-similar not just locally but across the globe.

J. Hays (✉)
Brown University, Providence, RI, USA
e-mail: hays@cs.brown.edu

A.A. Efros
University of California, Berkeley, CA, USA
e-mail: efros@eecs.berkeley.edu

J. Choi and G. Friedland (eds.), *Multimodal Location Estimation of Videos and Images*,
DOI 10.1007/978-3-319-09861-6_3

Film creators have long taken advantage of this (e.g., “Spaghetti Westerns” films that were ostensibly set in the American Southwest but filmed in Almería, Spain.) Nonetheless, it must be the case that certain visual features in images correlate strongly with geography even if the relationship is not strong enough to specifically pinpoint a location. Beach images must be near bodies of water, jungles must be near the equator, and glaciated mountains cover a relatively small fraction of the Earth’s surface.

Consider the photographs in Fig. 3.1. What can you say about where they were taken? The first one is easy—it’s an iconic image of the Notre Dame cathedral in Paris. The middle photo looks vaguely Mediterranean, perhaps a small town in Italy, or France, or Spain. The rightmost photograph is the most ambiguous. Probably all that could be said is that it’s a picture of a seaside in some tropical location. But even this vague description allows us to disregard all noncoastal, nontropical areas—more than 99.9 % of the Earth’s surface! Evidently, we humans have learned a reasonably strong model for inferring location distribution from photographs.

What explains this impressive human ability? Semantic reasoning, for one, is likely to play a big role. People’s faces and clothes, the language of the street signs, the types of trees and plants, the topographical features of the terrain—all can serve as semantic clues to the geographic location of a particular shot. Yet, there is mounting evidence in cognitive science that *data association* (ask not “What is it?” but rather “What is it *like*?”) may play a significant role as well [2]. In the example above, this would mean that instead of reasoning about a beach scene in terms of the tropical sea, sand and palm trees, we would simply remember: “I have seen something similar on a trip to Hawaii!”. Note that although the original picture may not actually be from Hawaii, this association is still extremely valuable in helping to implicitly define the *type* of place that the photo belongs to.

Of course, computationally we are quite far from being able to semantically reason about a photograph (although encouraging progress is being made). On the other hand, the recent availability of truly gigantic image collections has made data association, such as brute-force scene matching, quite feasible [8, 33].

In this chapter, we examine algorithms for estimating a distribution over geographic locations from an image using a data-driven scene matching approach. For this task, we leverage a dataset of over 6 million GPS-tagged images from Flickr.com.

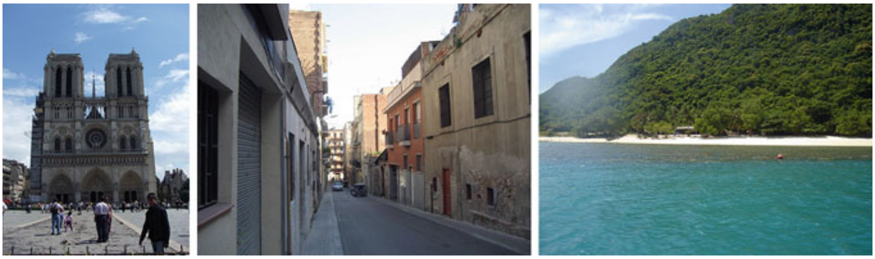


Fig. 3.1 What can you say about where these photos were taken?

We measure how often a geolocation strategy can correctly locate a query photo, where "correct" is defined as "within 200 km of the actual location." with meta-tasks such as land cover estimation and urban/rural classification.

A key idea of the im2gps work is that humans or algorithms can estimate the location of a photograph *without* having to perform "instance level" or "landmark" recognition. While instance-level recognition techniques are impressive, we show that such matches only account for about one half of successful geolocalizations.

3.1.1 Background

There exist a variety of geolocalization algorithms operating on different input modalities. While we will review a few techniques, see [19] for a broader survey. Im2gps assumes the input is an unlabeled photograph while other methods make use of sequences of photos [12] or try to relate ground level views to aerial imagery [17]. Jacobs et al. [11] propose a clever method to geolocalize a webcam by correlating its video-stream with satellite weather maps over the same time period. Visual localization on a topographical map was one of the early problems in computer vision. It turns out to be challenging for both computers and humans [32], but recent methods [1] based on terrain "curvelet" features work surprisingly well.

The availability of GPS-tagged images of urban environments coupled with advances in multiview geometry and efficient feature matching led to a number of groups developing place recognition algorithms, some of which competed in the "Where am I?" Contest [31] at ICCV'05 (winning entry described in [37]). Similar local feature geometric matching approaches have also been successfully applied to co-registering online photographs of famous landmarks for browsing [30] and summarization [28], as well as image retrieval in location-labeled collections, e.g. [4]. Landmark photos are linked to Wikipedia photos and articles within a specified city in [26]. Since the publication of im2gps, [6] and [38] have attacked the global landmark recognition problem, in the latter case scaling up to thousands of landmarks with high accuracy.

But can these geometric local feature matching approaches scale up to all photos of world? This is unlikely in the near future, not just because of computational cost, but simply because the set of all existing photographs is still not large enough to exhaustively sample the entire world. Yes, there are tens of thousands of photos of a many landmarks, but some ordinary streets or even whole cities might be entirely missing. Even with a dense visual sample, much of the world is too self-similar (e.g., the 300,000 square kilometers of corn fields in the USA). Clearly, a generalization of some sort is required.

On the other side of the spectrum from instance-level recognition is the task of scene categorization which tries to group forests with forests, kitchens with kitchens, deserts with deserts, etc. A large body of work exists on scene recognition [16, 22, 27, 34, 35] which involves defining a fixed taxonomy of scene categories and using various features to classify a novel image into one of these categories.

We use a combination of features from both the local feature matching literature (best suited for instance-level recognition) as well as features more commonly seen in category recognition (best suited for recognizing broader geographic concepts, e.g., "Mediterranean"). If the query image is a famous landmark, there will likely be many similar images of the same exact place in the database, and our approach is likely to return a precise GPS location. If the query is more generic, like a desert scene, many different deserts could match, producing a location probability that is high over the dry, sandy parts of the world.

3.1.2 Chapter Outline

In Sect. 6.4 we create training and testing databases from geotagged Internet images. In Sect. 3.3 we discuss the original, relatively simple "im2gps" geolocalization algorithm [9]. In Sect. 3.4 we add new features and utilize a lazy learning technique to nearly double the original "im2gps" performance. In Sect. 3.5, we analyze factors affecting geolocalization accuracy such as geographic bias in the influence of instance-level landmark matching.

3.2 Building a Geo-tagged Image Dataset

In order to reason about the global location of an arbitrary scene we first need a large number of images that are labeled with geographic information. This information could be in the form of text keywords or it could be in the form of GPS coordinates. Fortunately there is a huge (and rapidly growing) amount of online images with both types of labels. For instance, Flickr.com has hundreds of millions of pictures with either geographic text or GPS coordinates.

But it is still difficult to create a useful, high-quality database based on user collected and labeled content. We are interested in collecting images that depict some amount of geographic uniqueness. For instance, pictures taken by tourists are ideal because they often focus on the *unique* and *interesting* qualities of a place. Many of these images can be found because they often have geographic keywords associated with them (i.e., city or country names). But using geographic text labels is problematic because many of them are ambiguous (e.g., Washington city/state, Georgia state/country, Mississippi river/state, and LA city/state) or spatially broad (e.g., Asia or Canada).

Images annotated only with GPS coordinates are geographically unambiguous and accurate, but are more likely to be visually irrelevant. Users tend to geo-tag all of their pictures, whether they are pet dog pictures (less useful) or hiking photos (more useful). In fact, the vast majority of online images tagged with GPS coordinates and to a lesser extent those with geographic text labels are not useful for image-based geolocation. Many of the images are poor quality (low resolution, noisy, black and white) or depict scenes, which are only marginally useful for geolocation (most portraits, wedding pictures, abstracts, and macro photography). While these types of photos can sometimes reveal geographic information (western-style weddings are popular in Europe and Japan but not in India; pet dogs are popular in the USA but not in Syria) the customs are so broadly distributed that it is not very useful for geolocation.

However, we find that by taking the intersection of these groups, images with both GPS coordinates and geographic keywords, we greatly increased the likelihood of finding accurately geolocated *and* visually relevant training data. People may geotag images of their cats, but they're less likely to label that image with "New York City" at the same time. Our list of geographic keywords includes every country and territory, every continent, the top 200 most populated cities in the world, every US state, and popular tourist sites (e.g., "Pisa," "Nikko," "Orlando").

This results in a pool of approximately 20 million geotagged and geographic text-labeled images from which we excluded all photos which were also tagged with keywords such as "birthday," "concert," "abstract," and "cameraphone." In the end, we arrived at a database of 6,472,304 images. All images were downsized to max dimension 1024 and JPEG compressed for a total of 1 terabyte of data.

While this is a tremendous amount of data it cannot be considered an exhaustive visual sampling of Earth. Our database averages only 0.0435 pictures per square kilometer of Earth's land area. But as Fig. 3.2 shows the data is very nonuniformly distributed towards places where people live or travel. We will revisit this nonuniform distribution in Sect. 3.5.1. It can be seen as a desirable property in that this is the same distribution from which people would generate query images or undesirable since it leaves huge portions of the world under-sampled.

3.2.1 Evaluation Test Set

To evaluate geolocalization performance we use a separate, held-out test set of geolocated images. We built the test set by drawing 400 random images from the original dataset. From this set, we manually remove the types of undesirable photos that we

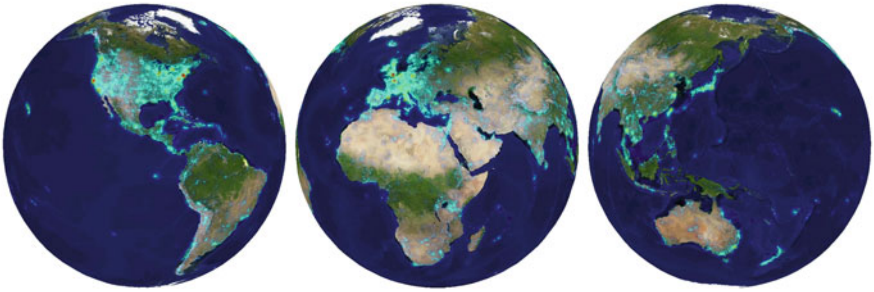


Fig. 3.2 The distribution of photos in our database. Photo locations are cyan. Density is overlaid with the "jet" color map (log scale)

Fig. 3.3 A sample of the 237 image im2gps test set. Note how difficult it is to specifically geolocalize most of the images

tried to excluded during database construction—abstract photos, overly processed or artistic photos, and black and white photos. We also exclude photos with significant artifacts such as motion blur or extreme noise. Finally we remove pictures with easily recognizable people or other situations that might violate someone's privacy. To ensure that our test set and database are independent we exclude from the database not just test images, but all other images from the same photographers.

Of the 237 resulting images, about 5 % are recognizable as specific tourist sites around the globe but the great majority are only recognizable in a generic sense (See Fig. 3.3). Some of the images contain very little geographic information, even for an astute human examiner. We think this test set is extremely challenging but representative of the types of photos people take.

3.3 Simple, Baseline Geolocalization Method

This section briefly describes the original "im2gps" method [9]. We treat this as a baseline for later studies in Sects. 3.4 and 3.5. In this section, we first look at a handful of relatively simple "baseline" global image features. We hope that some of these image properties correlate with geographic location.

Tiny Images. The most trivial way to match scenes is to compare them directly in color image space. Reducing the image dimensions drastically makes this approach more computationally feasible and less sensitive to exact alignment. This method of image matching has been examined thoroughly by Torralba et al. [33]. Inspired by this work we will use 16 by 16 color images as one of our base features.

Color histograms. We build joint histograms of color in CIE L*a*b* color space for each image. Our histograms have 4, 14, and 14 bins in L, a, and b, respectively for a total of 784 dimensions. We have fewer bins in the intensity dimension because other descriptors will measure the intensity distribution of each image. We compute distance between these histograms using χ^2 distance.

Texton Histograms. Texture features might help distinguish between geographically correlated properties such ornamentation styles or building materials in cities or vegetation and terrain types in landscapes. We build a 512 entry universal texton dictionary [20] by clustering our dataset's responses to a bank of filters with eight orientations, two scales, and two elongations. For each image, we then build a 512 dimensional histogram by assigning each pixel's set of filter responses to the nearest texton dictionary entry. Again, we use χ^2 distances between texton histograms. This representation is quite similar to dense "visual words" of local features.

Line Features. We have found that the statistics of straight lines in images are useful for distinguishing between natural and man-made scenes and for finding scenes with similar vanishing points. We find straight lines from Canny edges using the method described in Video Compass [13]. For each image, we build two histograms based on the statistics of detected lines- one with bins corresponding to line angles and one with bins corresponding to line lengths. We use L1 distance to compare these histograms.

Gist Descriptor + Color. The gist descriptor [23] has been shown to work well for scene categorization [22] and for retrieving semantically and structurally similar scenes [8]. We create a gist descriptor for each image with 5 by 5 spatial resolution where each bin contains that image region's average response to steerable filters at 6 orientations and 4 scales. We also create a tiny L*a*b image, also at 5 by 5 spatial resolution.

Geometric Context. Finally, we compute the geometric class probabilities for image regions using the method of Hoiem et al. [10]. We use only the primary classes- ground, sky, and vertical since they are more reliably classified. We reduce the probability maps for each class to 8×8 and use L2 distance to compare them.

We precompute all features for the 6.5 million images. At 15 s per image this requires a total of 3.08 CPU years, but is trivially parallelized.

Our baseline geolocation algorithm is quite simple—for each query we find the nearest neighbor scene in our database according to these features. We then take the GPS coordinate of that nearest neighbor match as our geolocation estimate.

3.3.1 Is the Data Helping?

A key question for us is *how strongly does image similarity correlate with geographic proximity*? To geolocalize a query we don't just want to find images that are similarly structured or of the same semantic class (e.g., "forest" or "indoors"). We want image matches that are specific enough to be geographically distinct from otherwise similar scenes. How much data is needed start to capture this geography-specific information? In Fig. 3.4 we plot how frequently the 1-NN increase the size of the database. With a tiny database of 90 images, the 1-NN scene match is as likely to be near the query as a random image from the database. With the full database we perform 16 times better than chance.

Given a photo, how often can we pin-point the right city? Country? Continent? With our simple baseline geolocalization algorithm, the first nearest neighbor is within 64 km of the true location 12 % of the time, within 200 km 16 % of the time, within 750 km 25 % of the time, and within 2,500 km 50 % of the time.

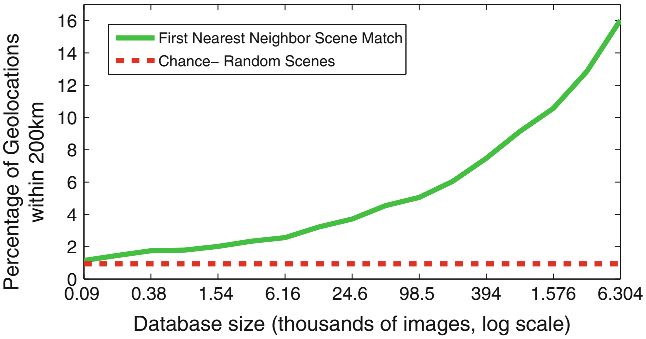


Fig. 3.4 *Accuracy of simple geolocalization baseline across database sizes*. Percentage of test set images that were correctly localized within 200 km of ground truth as function of dataset size using 1-NN. As the database shrinks the performance converges to chance

3.3.2 Grouping Geolocation Estimates

1-NN approaches are sensitive to noise. Alternatively, we also consider a larger set of kNN ($k = 120$ in our experiments). This set of nearest neighbors together forms an implicit estimate of geographic location—a probability map over the entire globe. The hope is that the location of peak density in this probability map corresponds to the true location of the query image. One way to operationalize this is to consider the modes of the distribution by performing mean-shift [5] clustering on the geolocations of the matches. We represent the geolocations as 3d points and re-project the mean-shift clusters to the Earth's surface after the clustering procedure. We use a mean-shift bandwidth of 200 km (although performance is not especially sensitive to this parameter). The clustering serves as a kind of geographic outlier rejection to clean up spurious matches, but can be unfavorable to locations with few data-points. To compute a geolocation estimate, one approach is to pick the cluster with the highest cardinality and report the GPS coordinate of its mode. In practice, this works no better than 1-NN, but we will use these mean shift clusters as the basis for our learning algorithm in Sect. 3.4.4. For some applications, it might be acceptable to return a list of possible location estimates, in which case the modes of the clusters can be reported in order of decreasing cardinality. We show qualitative results for several images in Fig. 3.5. Cluster membership is indicated with a colored border around the matching scenes and with colored markers on the map.

3.4 Improving Geolocalization with More Features and Lazy Learning

The features (global histograms, gist, bag of textons, etc) and prediction method (1 nearest neighbor) in the previous section represent the capabilities of the original im2gps system [9]. However, we can dramatically improve global geolocalization accuracy with more advanced features and more sophisticated learning.

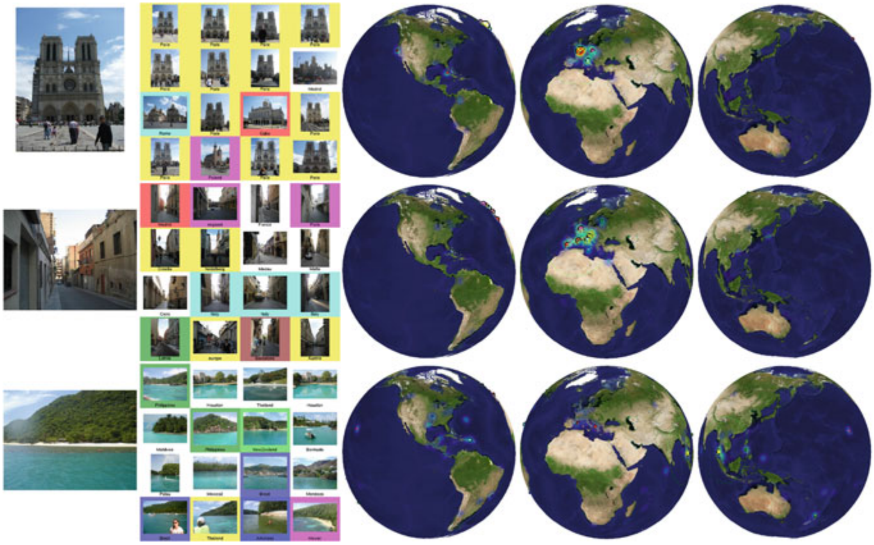


Fig. 3.5 *Results of simple geolocalization baseline*. From left to right: query images, nearest neighbors, and three visualizations of the estimated geolocation probability map. The probability map is shown as a jet-colorspace overlay on the world map. Cluster modes are marked with circumscribed "X"'s whose sizes are proportional to cluster cardinality. If a scene match is contained in a cluster it is highlighted with the corresponding color. The ground truth location is a cyan asterisk surrounding by green contours at radii of 200, 750, and 2,500 km. From top to bottom, these photos were taken in Paris, Barcelona, and Thailand

First, we describe additional scene matching features which are intended to be more robust than those used in the previous section. Two shortcomings of the baseline features are (1) sensitivity to scene layout and (2) poor performance at instance-level recognition. To address these problems we describe additional geometry derived and SIFT-derived features.

Second, we use "lazy learning" with these additional features. We train a multiclass, kernel SVM to decide which mean shift cluster of scene matches a query belongs to. Together, the new features and lazy learning double the baseline im2gps performance.

3.4.1 Geometry Specific Color and Texton Histograms

The baseline scene descriptors are all "global"—encompassing statistics of the entire image or built on fixed image grid regardless of scene layout. This means that irrelevant scene transformations (e.g., cropping the image, shifting the horizon) produce huge changes in the global descriptors and thus huge distances according to our distance metrics. This lack of invariance means that inconsequential image differences

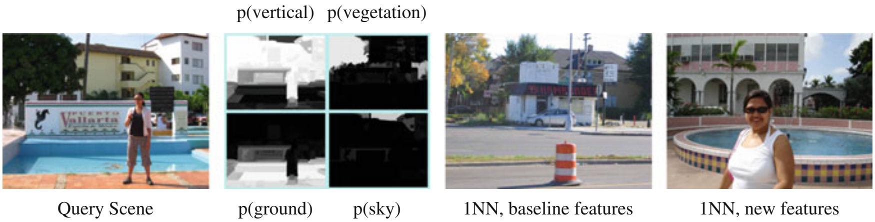


Fig. 3.6 For each geometric class we build separate color and texton histograms. Scene matching is improved by restricting the histogram comparisons to corresponding geometric regions

will prevent otherwise good scene matches from being retrieved. To address this and make histogram comparisons more meaningful we build color and texton histograms for *each geometric class* in an image. For example, texture histograms for vertical surfaces in an image. By restricting texture and color comparisons to geometrically like regions of images, we expect their distances to be more reliable (Fig. 3.6).

We use geometric context [10] to estimate the probability of each image region being "ground," "vertical," "sky," or "porous" (i.e., vegetation). For any pixel, the probability of "ground," "sky," and "vertical" sums to one, while "porous" is a subset of "vertical." We build color and texture histograms for each geometric class by weighting each pixel's contribution to each histogram according to the geometric class probabilities. We also build global texture and color histograms in which the "vertical" pixels get much higher contribution (the intuition being that the appearance of vertical image content is more likely to be correlated with geolocation than the sky or ground). Our approach is similar to the "illumination context" proposed in Photo Clip Art [14] in which scenes are matched with color histograms built from ground, sky, and vertical image regions.

The geometric context classification is not entirely reliable, especially for unusual scenes, but the mistakes tend to be fairly *consistent* which is arguably more important than accuracy in this task (e.g., if clouds were 100 % classified as "vertical," our feature distances would still be reasonable because the scenes would be decomposed into consistent, although mixed, semantic groups). The geometric context probability maps are themselves resized to 8×8 image features.

3.4.2 Bags of SIFT Features

SIFT [18] derived features have been used for scene representations with spatial pyramids composed of densely sampled local features still near the state of the art for scene recognition [15]. In these cases, the quantization of visual words is typically rather coarse (e.g., 500 visual words). Quantized SIFT features have also been shown to work well for instance-level recognition in large datasets [29]. Larger vocabularies (1 million visual words) and geometric verification of candidate matches improve performance further [24]. Landmark geolocation methods [6, 38] have relied entirely on these types of features.

Inspired by these successes, we compute SIFT features at interest points detected by Hessian-affine and MSER [21] detectors. For each interest point type, we build vocabularies of 1,000 and 50,000 visual words based on a random subset of the database. The intuition is that a vocabulary of 1,000 captures texture qualities of the scene while a vocabulary of 50,000 captures instance specific (landmark) image elements. To build the visual vocabularies, we use 20 million SIFTS sampled from roughly 1 million images. To build the 50,000 entry vocabularies a two level hierarchy is used, as k-means would otherwise be prohibitively slow. The hierarchy is only used to construct the vocabulary after which the leaf nodes are treated as a flat vocabulary. We use "soft assignment" as described in [25], assigning each SIFT descriptor to its nearest 5 vocabulary centers, inversely weighted by distance. Because we use soft assignment, the 50,000 entry histograms are not sparse enough to merit an inverted file system search.

3.4.3 Geolocalization with Additional Features

While these features perform especially well when coupled with a more sophisticated machine learning method (Sect. 3.4.4), as a baseline we again use the first nearest neighbor method. We use L1 distance for all image features (gist, geometric context maps) and χ^2 (chi squared) measure for all histograms (texture, color, lines, SIFT). The scene matching process is implemented hierarchically—first, 2,000 nearest neighbors are found with the baseline features and then distances are computed for the new geometry derived and SIFT features and the matches are reranked.

Compared to the baseline features, the new features perform significantly better at instance-level recognition, as would be expected from the new large-vocabulary SIFT histograms. Scene matches for more "generic" scenes are also improved. Figure 3.7 shows cases where the first nearest neighbor with the new features is dramatically improved from the baseline features. For common scene types under canonical viewpoints, the difference is less noticeable.

Recall that with the 237 image im2gps test set and base im2gps features, the first nearest neighbor is within 200 km of a query 16 of the time. Using the four SIFT histograms by themselves (after the initial hierarchical search) gives an accuracy 18.6 %. Using all features improves accuracy to 21.1 %.

3.4.4 Lazy Learning for Large-Scale Scene Geolocalization

Nearest neighbor methods are attractive because they require no training, they are trivially parallelizeable, they perform well in practice, and their query complexity scales linearly with the size of the dataset. In fact, it is often possible to perform nearest neighbor search in less than linear time, especially if one is willing to adopt approximate methods. However, nearest neighbor methods lack one of the fundamental advantages of supervised learning methods—the ability to learn which dimensions are relevant for a particular task.

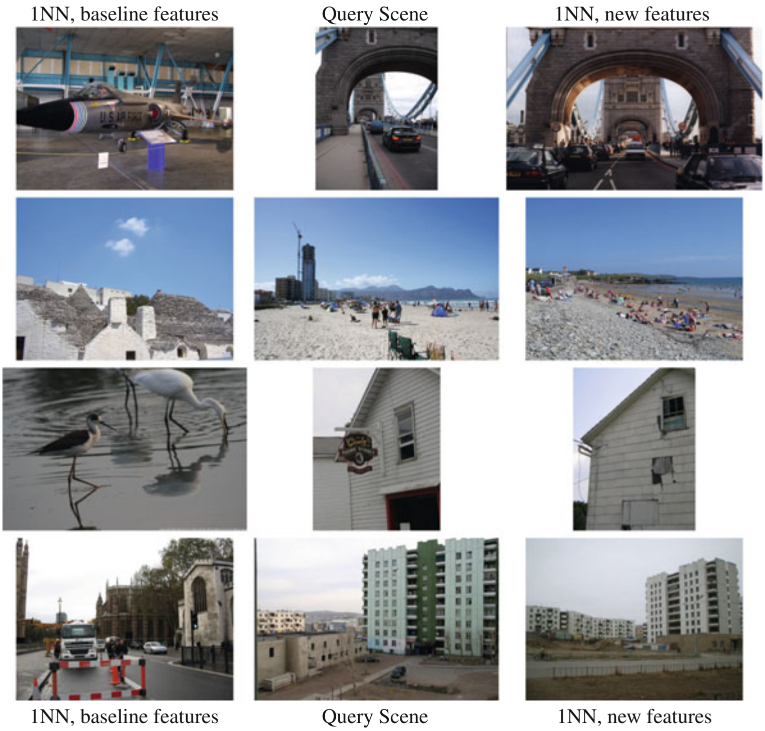


Fig. 3.7 *Nearest Neighbors with New Features*. The features introduced in this section are dramatically better at landmark recognition, especially when the viewpoints do not match, as in the *top row*. This is to be expected from the SIFT features. The remaining figures show nonlandmark scenes for which the matches are much better. The *last row* is an ideal case—even though an exact, instance-level match can not be found, the new features have found a scene that is architecturally very similar. Even more impressive, both photos are in Mongolia where there are few photos to match to

This is critical because our feature representation is quite high-dimensional. In total, including the features from the baseline method, we have an over-complete set of 22 elementary features. The baseline features total 2,201 dimensions, while the features proposed in this section total 109,436 dimensions dominated by the two 50,000 entry SIFT histograms. Such high feature dimensionality is problematic for nearest-neighbor methods. Unfortunately, more sophisticated learning approaches are difficult to apply when the number of training samples is large (over 6 million in our case) and the feature dimensionality is high (over 100,000 in our case).

We adopt a "lazy learning" approach inspired by SVM-KNN [36] and prior supervised, KNN enhancements (See [3] for an overview of "local" learning methods). Lazy learning methods are hybrids of nonparametric, KNN techniques and parametric, supervised learning techniques. Our supervised lazy learning can be seen as a post-process to refine the nearest-neighbor search we use as a baseline.

The philosophy driving these works is that learning becomes *easier* when examining the local space around a query instead of the entire problem domain.

Consider the image geolocation problem. The boundary between geographic classes (e.g., Tokyo and London) is extraordinarily complex because it must divide a wide spectrum of scenes types (indoor, urban, landscape, etc...) that occur in both locations. There is no simple parametric boundary between these geographic classes. However, within a space of similar scenes (e.g., subway carriage photos) it may be trivially easy to divide the classes and this allows one to employ simpler, faster, and easier to interpret learning methods. Thus lazy learning is promoted not as an approximation method, but as a learning enhancement. But it is the scalability to very large datasets that makes lazy learning attractive to us.

For a novel query, our algorithm is:

1. Find $K_{sl} = 2{,}000$ nearest neighbors using the “baseline” features defined in Sect. 3.3.
2. Reduce the K_{sl} nearest neighbors to K using both “baseline” features and the additional features introduced in this section.
3. Cluster the K nearest neighbors according to their geographic locations using mean shift. We use a bandwidth of 200 km. Each of the $\mathscr{C}$ clusters is now considered a distinct class for the sake of learning. Typical values of $\mathscr{C}$ are 30–60, depending on the minimum allowed cluster size.
4. Compute the all-pairs distances between all K nearest neighbors using both the “base” and additional features with L1 and χ^2 (chi squared) distances.
5. Convert the all-pairs distances into a positive semi-definite kernel matrix (i.e., the “Kernel Trick”) and use it to train $\mathscr{C}$ 1-vs-all nonlinear SVMs.
6. For each of the $\mathscr{C}$ classifiers, compute how far the query point is from the decision boundary. The class for which this distance is most positive is the “winner,” and the query is assigned to that mean shift cluster.
7. The final geolocation estimate for the query is then the average GPS coordinate of all members of the winning cluster.

As K becomes small, the technique reduces to 1NN. As K becomes large, the technique reduces to a standard kernel SVM (which is intractable with our scale of data).

Our approach depends on the nearest-neighbor search in steps 1 and 2 retrieving enough geographically relevant scenes to train the SVM. If a query photo is from Pittsburgh and none of the retrieved scenes are nearby, the learning can not hope to recover. However, for 75 % of queries, the $K_{sl} = 120$ nearest neighbors according to the baseline features have at least one image within 200 km of the ground truth location. Thus we can have some confidence that geographically nearby scenes are being included among our nearest neighbors and taking part in the learning process.

A point of interest about our approach is that our *classes* for the supervised learning emerge in a lazy manner (after a nearest neighbor search) rather than being pre-defined as in SVM-KNN [36]. Because the output of a geolocation estimation system is a real-valued GPS coordinate, it might seem like it is more naturally a regression problem. But for any query scene, the geolocation problem ends up being a decision between several discrete, disconnected possibilities (e.g., Alps vs. Cascades vs. Rockies vs. Andes). Therefore we think it is natural to treat it as a classification problem.

3.4.4.1 Complexity and Running Time

As with KNN-SVM, our complexity is linear with respect to N, the number of "base" distances we compute to find K nearest neighbors, and quadratic with respect to K. In our case, $N = 6471706$ and $K = \sim 200$ and our running time is still dominated by the initial search which takes ~2.5 min (amortized over many queries). We have made little effort to optimize the initial search although "tiny images" [33] reports good results from a very low dimensional initial search of PCA bases. Step 1 is amenable to approximation because it does not need to have high precision, only high recall, assuming that step 2 will filter out spurious matches.

3.4.5 Geolocalization Results with New Features and Lazy Learning

With a one nearest neighbor algorithm, our accuracy is 16 % with baseline features and 21 % with more advanced, higher dimensional features. Replacing the one nearest neighbor prediction with the lazy learning method raises our accuracy to 31 %, nearly doubling the performance of the original im2gps publication [9]. We show four geolocalization results in Figs. 3.8 and 3.9.

3.5 Why Does it Work? Deeper Performance Analysis

3.5.1 Measuring Performance Without Geographic Bias.

Since the geographic distribution of data appears to be peaked in relatively few places (Fig. 3.10), one concern is that our performance could be a result of random guessing. In fact, the chance that two photos are within 200 km in the im2gps database is about 1.2 %. For our test set of 237 images sampled from the database chance is 0.97 %. For individual test cases, chance ranges from less than 0.01 % in Libya, Colombia, and Greenland to 4.9 % near London. That is to say, 4.9 % of the im2gps database images (and probably 4.9 % of Internet images) are within 200 km of London. For other cities the values are: New York City 4.3 %, San Francisco 3.1 %, Paris 2.8 %, Chicago 1.9 %, Tokyo 1.8 %, and Barcelona 1.5 %.

How would our simple baseline geolocalization algorithm perform if the test set distribution was not geographically peaked? To quantitatively evaluate this issue we

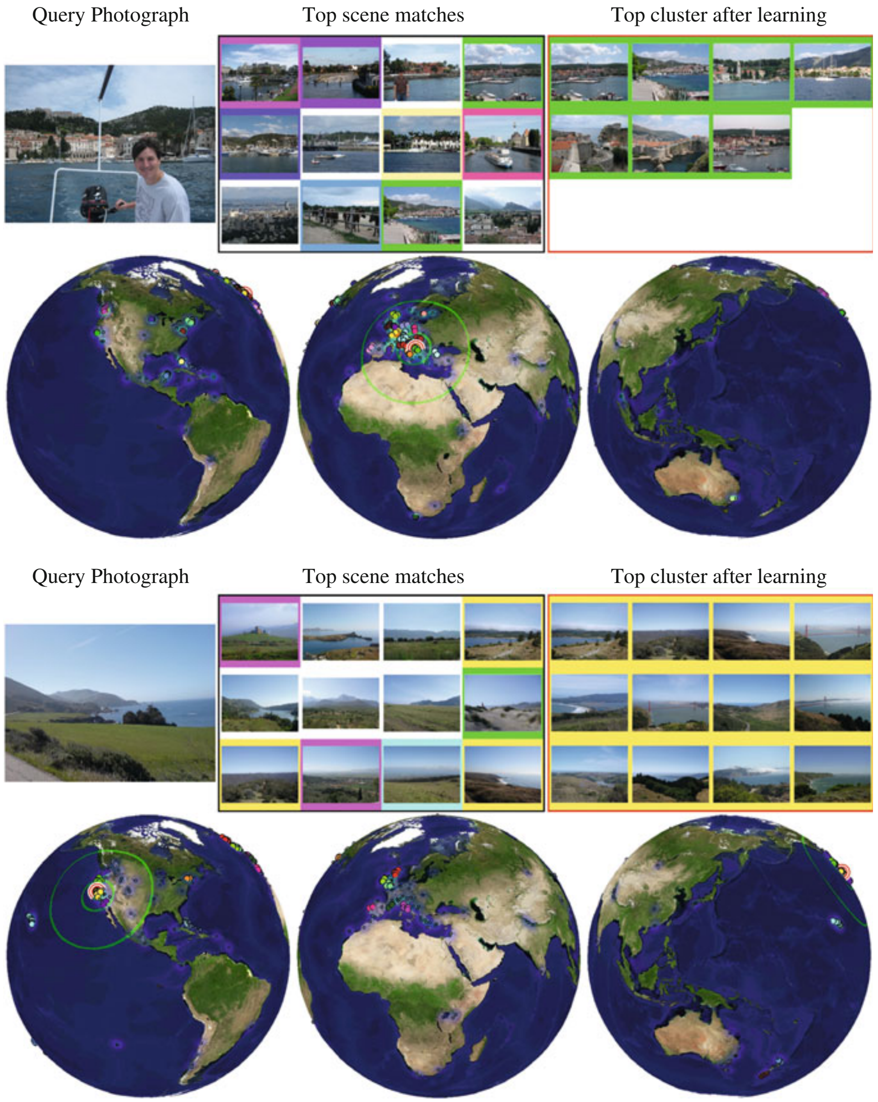


Fig. 3.8 *Geolocalization Results with Lazy Learning*. Results are generated from $K = 200$ nearest neighbors clustered with a mean shift bandwidth of 200 km and a minimum cluster size of 3. The scene match montages are scanline ordered according to scene match distances. The colors of scene match borders and globe markers indicate cluster membership. The coloring of the clusters indicates their ordering by cardinality—*yellow* is largest, then *cyan*, *magenta*, *red*, *green*, and *blue*. The geolocation estimate from learning is indicated by the *red* and *white* concentric rings. The ground truth location is marked by concentric green rings of radius 200, 750, and 3,000 km. The density of scene matches on the globe is indicated by a jet colormap overlay. Scene matches without a cluster are plotted as *black rings*

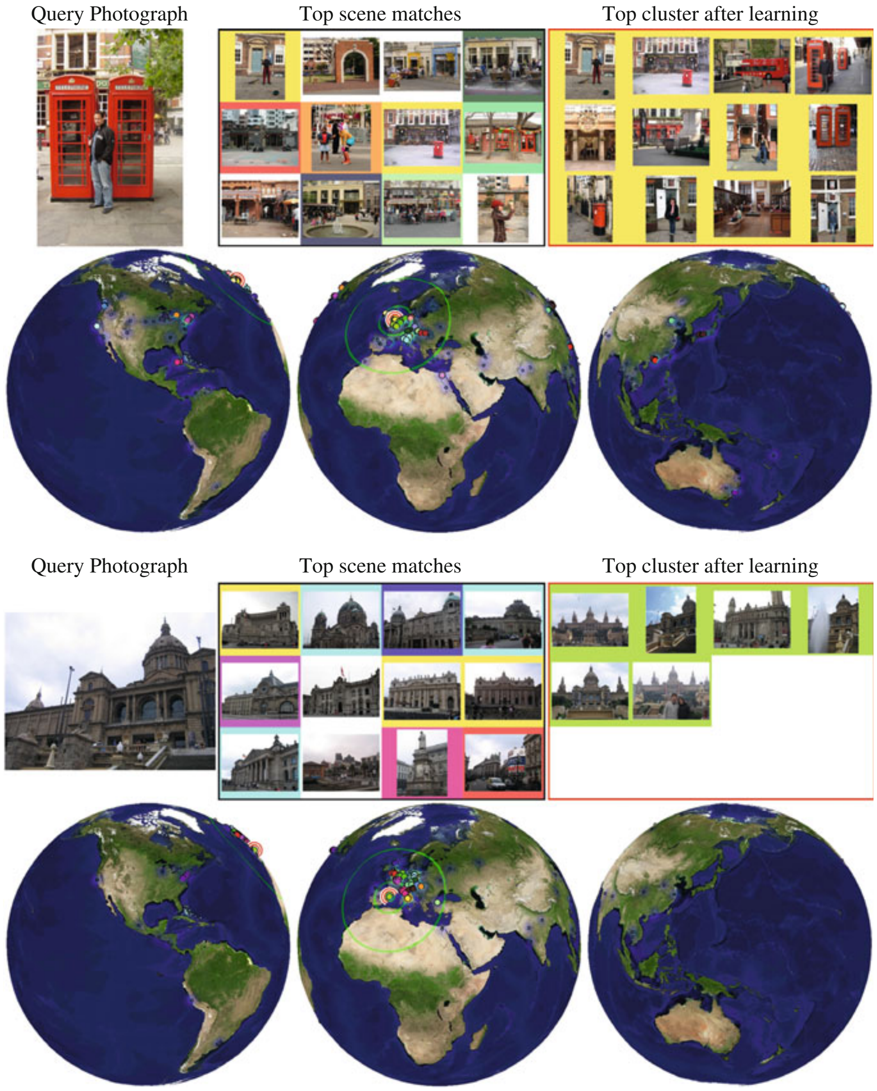


Fig. 3.9 Additional Geolocalization Results with Lazy Learning

define a new *geographically uniform* test set. We tessellate the globe into quadrilateral regions roughly 400 km on edge (Fig. 3.10). We take one query from each of the 955 regions in that have at least ten photographs. Chance is an order of magnitude lower

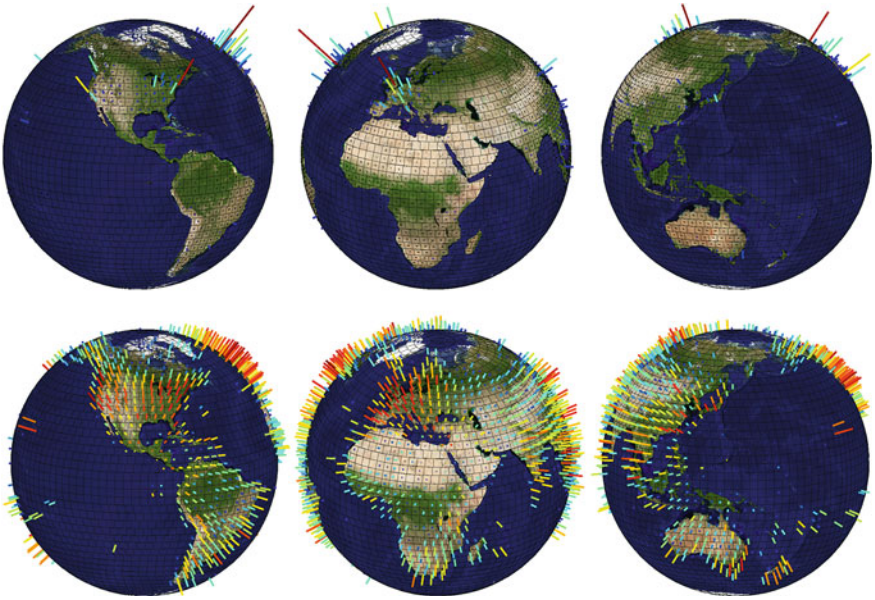


Fig. 3.10 *Photo density in im2gps database*, linear scale (*top*) and natural log scale (*bottom*). The height of each bar is proportional to the density of geotagged photos in each equal area region. The bars are colored according to the Matlab "jet" color scheme. Regions with zero photos have no bar

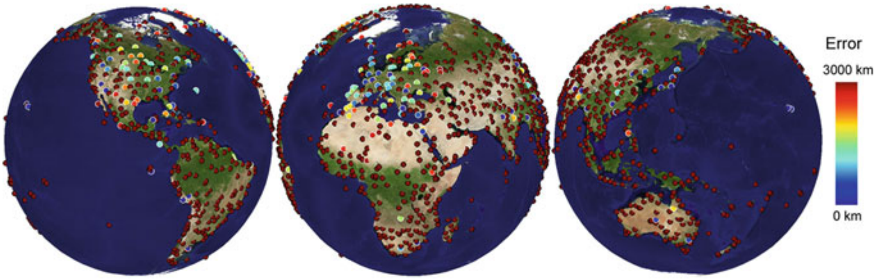


Fig. 3.11 *Accuracy on Geographically Uniform Test Set*. For each photo in the test set, the marker color indicates how accurately the photo was geolocated

for this database—only 0.13 %.[1] Figure 3.11 shows the geographic distribution of the test set, as well as the geolocation accuracy for each photo. We are unable to correctly localize any queries in large regions of South America, Africa, and Central Asia. Overall, for only 2.3 % of the test set is the first nearest neighbor is within

[1] This value was calculated by counting the number of database photos close enough to each query in the test set. Alternatively, each geolocation guess has an area of 126,663 km^2 and the land area of the Earth is 148,940,000 km^2, suggesting that a truly uniform test set would have a chance guessing accuracy of 0.084 %. Chance is higher for our test set because our database (and thus test set) contain no photographs in some regions of Siberia, Sahara, and Antarctica.

200 km of the query photo's location. Interestingly, relative to chance, this is just as much of an improvement as on the im2gps test set (∼16 times better).

The fundamental, unavoidable issue is that we do not have enough data for many locations around the world. A generic photo of Brazilian rain forest will find many more matches in Hawaii, Thailand, or more temperate locations than in the correct location. It is not a matter of database peakedness drowning out the correct matches—if a scene is visually distinct it will often be geolocated even if it is rarely photographed. But for generic scenes, where the visual features distinguishing one location from another are extremely subtle, a large amount of reference data is needed. So it is certainly the case that im2gps performance is inextricably tied to the geographic distribution of our test set and database. A biased sampling strategy at database creation time could help smooth out these differences, but there is not enough geotagged data on Flickr to completely remove the geographic bias of photo taking.

3.5.2 Measuring Category Level Geolocation Performance.

While we have demonstrated that our geolocation accuracy is far better than chance, random guessing is arguably not a realistic baseline comparison. Just by retrieving scenes of the same broad semantic category as a query (for instance "beach," "mountain," "city," "forest," "indoors," etc...) chance must rise considerably. Does category level guessing account for im2gps performance, or is it picking up on more subtle geographic variations?

As we increase the size of the im2gps database we see a slow but steady increase in performance (Fig. 3.4). If random matching within the same scene broad scene category could account for im2gps performance, it is likely that performance would saturate with a dramatically smaller database. Previous work has shown 90 % accuracy in 4-way categorization using a couple thousand training examples and nearest neighbor classification with the gist descriptor [22]. Why does our performance double as our database increases from 600,000 to 6 million geolocated examples? Part of the gain is likely because the scene matches become more discriminative (not just forest but rain forest, not just cities but European cities).

Figure 3.12 shows three queries that would fit into a broadly defined "city" category. Notice how different the geographic distribution of scene matches is for each query. The German city geolocation estimate is correctly peaked in central Europe. The Hong Kong skyline is confused with other skylines (New York, Shanghai, Tokyo, and Chicago). Hong Kong is the 5th largest cluster. The Alabama warehouse matches many paved areas or streets in the USA, although none near the correct location. The im2gps scene matches can definitely be more specific than typically defined scene categories.

We can quantify how accurately im2gps would perform with perfect category level scene recognition and random guessing within that category for our test sets. We use land cover maps to assign a ground truth geographic scene category to each image in a test set. The categories are "city," "forest," "water," "shrubland," "rain forest," "barren," "snow and ice," "crops and grassland," and "savanna."

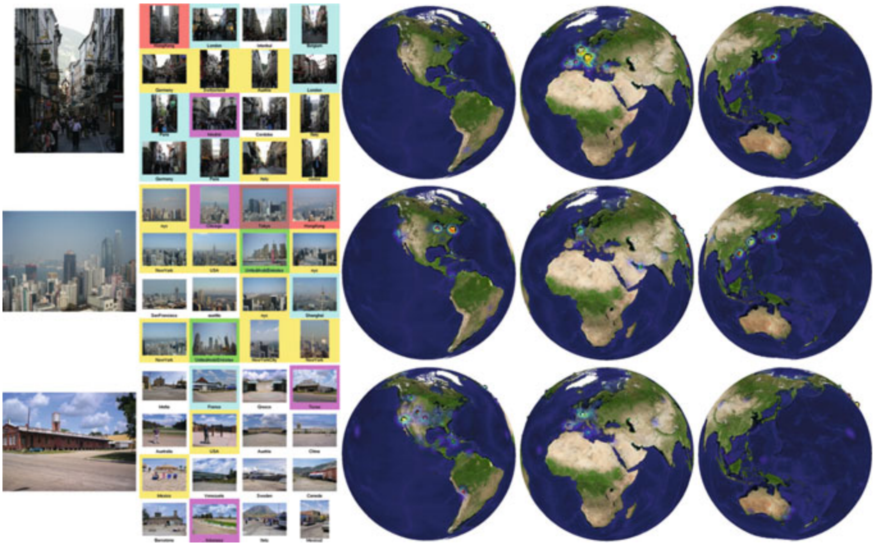


Fig. 3.12 *im2gps results for different cities*. These city queries from Germany, Hong Kong, and Alabama produce very different geolocation estimates

We classify the entire im2gps database into the same 9 categories. Then for each photo in a test set, we calculate the probability that randomly matching to scenes of the same category will produce a geolocation guess within 200 km. This is something of an ideal case because it assumes that any two categories, e.g., "shrubland" and "savannah," can always be distinguished. Chance under this perfect categorical matching is still quite low—2.09 % for the im2gps test set (up from 0.97 %) and 0.36 % for the geographically uniform test set (up from 0.13 %). We can safely say that our geolocation method is discriminating far more than just scene categories.

3.5.3 Measuring Landmark Geolocation Performance

Perhaps 5 to 7 % of photos in the im2gps test set are readily recognizable landmarks such as Sagrada Familia or the Sydney Opera House. A very geographically knowledgeable person might even recognize the exact physical scene for 10 % of the test cases. Landmarks are visually distinctive and often photographed so it makes sense that they contribute a large amount to im2gps performance. For our baseline algorithm, of the 16 % of queries whose first nearest neighbor is within 200 km, 58 % of the 1 NN matches depict the same *physical scene*. Many of these would not be considered "landmarks" by a layperson—an aircraft in the Smithsonian, an Apple store in New York City, or a bridge in Portugal. At the same time certain possible landmarks, such as the Millennium Wheel in London, are missed by the first nearest neighbor.

We also evaluate the contribution of instance-level matching when using the higher dimensional features and lazy learning introduced in Sect. 3.4. With the improved features, the first nearest neighbor is the same scene for 40 % of successfully localized queries. The cluster chosen by the learning contains an instance-level match 58 % of the time. In some of these cases, the geolocation probably would have been correct even without the instance matches.

Thus, instance-level recognition does account for a slim majority of successful geolocalizations for both the simpler and more complex geolocalization strategies. But we are also able to localize a significant number of photos that are not landmarks and would presumably fall through the cracks of methods such as [6, 38].

3.6 Discussion

Not only is photo geolocalization an important problem in itself, but it could also be tremendously useful to many other vision tasks. Knowing the distribution of likely locations for an image provides huge amounts of additional meta-data for climate, average temperature for any day, vegetation index, elevation, population density, per capita income, average rainfall, etc. Even a coarse geo-location can provide a useful object prior for recognition. For example, knowing that a picture is somewhere in Japan would allow one to prime object detection for the appropriate type of taxi cabs, lane markings, average pedestrian height, etc.

Im2gps [9] was the first study of global image geolocation, a task that only became possible because of the emergence of large-scale geotagged Internet imagery. While the baseline im2gps approach was relatively simple, with the additional features and learning discussed in Sect. 3.4, our results are qualitatively and quantitatively greatly improved. In fact, our geolocalization accuracy exceeds that of nonexpert humans [7]. Typically, humans are implicitly treated as an upper bound for performance in vision tasks (e.g., object detection). Have we saturated performance for automatic image geolocalization? Definitely not. There is still a great deal of room for improvement. As Fig. 3.11 shows, the algorithm has trouble localizing photographs from sparsely sampled regions of the world unless they contain distinct landmarks. While it was hoped that our scene matching might be able to pick up on subtle landscape, vegetation, or architecture cues to geolocalize images this is rarely observed. Our algorithm's advantage over humans is its large visual memory, not its ability to relate scene statistics to geographic locations. Geolocalization performance should increase as algorithms include more high-level reasoning about architecture, writing, clothing, lighting direction, geology, and plant and animal species.

Acknowledgments We thank Steve Schlosser, Julio Lopez, and Intel Research Pittsburgh for helping us overcome the logistical and computational challenges of this project. All visualizations and geographic data sources are derived from NASA data. Funding for this work was provided by an NSF fellowship to James Hays and NSF grants CAREER 1149853, CAREER 0546547, and CCF-0541230.

References

1. G. Baatz, O. Saurer, K.Köser, M. Pollefeys, Large scale visual geo-localization of images in mountainous terrain, In *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, (2012), pp. 517–530
2. M. Bar, The proactive brain: using analogies and associations to generate predictions. Trends Cogn. Sci. **11**(7), 280–289 (2007)
3. S.S. Chris Atkeson, Andrew Moore, Locally weighted learning. AI. Review **11**, 11–73 (1997)
4. O. Chum, J. Philbin, J. Sivic, M. Isard, A. Zisserman, Total recall: Automatic query expansion with a generative feature model for object retrieval, in *Proceedings of ICCV*, 2007
5. D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
6. D.J. Crandall, L. Backstrom, D. Huttenlocher, J. Kleinberg. Mapping the world's photos, in *WWW '09: Proceedings of the 18th international conference on World wide web 2009*, pp. 761–770, 2009
7. J. Hays, A. Efros. Where in the world? human and computer geolocation of images, in *Vision sciences society meeting*, 2009
8. J. Hays, A.A. Efros. Scene completion using millions of photographs, in *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007
9. J. Hays, A.A. Efros. im2gps: estimating geographic information from a single image, in *CVPR*, 2008
10. D. Hoiem, A. Efros, M. Hebert, Recovering surface layout from an image. Int. J. Comput. Vision. **75**(1), 151–172 (2007)
11. N. Jacobs, S. Satkin, N. Roman, R. Speyer, R. Pless, *Geolocating static cameras*, in *Proceedings, ICCV*, 2007
12. E. Kalogerakis, O. Vesselova, J. Hays, A.A. Efros, A. Hertzmann. Image sequence geolocation with human travel priors, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '09)* (2009)
13. J. Kosecka, W. Zhang. Video compass, in *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, 2002, pp. 476–490
14. J.-F. Lalonde, D. Hoiem, A.A. Efros, C. Rother, J. Winn, A. Criminisi. Photo clip art. *ACM Transactions on Graphics (SIGGRAPH 2007)*, vol. 26(3) (August 2007)
15. S. Lazebnik, C. Schmid, J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in *CVPR* (2006)
16. L.-J. Li, L.F. Fei, *What, where and who? classifying events by scene and object recognition*, in *Proceedings, ICCV*, (2007)
17. T.-Y. Lin, S. Belongie, J. Hays. Cross-view image geolocalization, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Portland, OR, June 2013)
18. D. Lowe, Object recognition from local scale-invariant features. ICCV **2**, 1150–1157 (1999)
19. J. Luo, D. Joshi, J. Yu, A. Gallagher, Geotagging in multimedia and computer visiona survey. Multime'd Tools Appl. **51**, 187–211 (2011)
20. D. Martin, C. Fowlkes, D. Tal, J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in *Proceedings ICCV* (July 2001)
21. J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. **22**(10), 761–767 (2004)
22. A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vision **42**(3), 145–175 (2001)
23. A. Oliva, A. Torralba. Building the gist of a scene: The role of global image features in recognition, in *Visual Perception, Progress in Brain Research*, 2006, vol. 155
24. J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman. Object retrieval with large vocabularies and fast spatial matching, in *CVPR* (2007)

25. J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2008)
26. T. Quack, B. Leibe, L. Van Gool. World-scale mining of objects and events from community photo collections, in *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval* (2008)
27. L.W. Renninger, J. Malik, When is scene recognition just texture recognition? Vis. Res. **44**, 2301–2311 (2004)
28. I. Simon, N. Snavely, S.M. Seitz. Scene summarization for online image collections, in *Proceedings, ICCV* (2007)
29. J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in videos. ICCV **2**, 1470–1477 (2003)
30. N. Snavely, S.M. Seitz, R. Szeliski, Photo tourism: exploring photo collections in 3d. ACM Trans. Graph. **25**(3), 835–846 (2006)
31. R. Szeliski. "Where am I?": ICCV 2005 Computer Vision Contest. http://research.microsoft.com/iccv2005/Contest/
32. W. Thompson, C. Valiquette, B. Bennett, K. Sutherland, Geometric reasoning for map-based localization. Spatial Cogn. Comput **1**(3), 291–321 (1999)
33. A. Torralba, R. Fergus, W.T. Freeman, 80 million tiny images: a large dataset for non-parametric object and scene recognition. IEEE PAMI **30**(11), 1958–1970 (2008)
34. J. Vogel, B. Schiele, Semantic modeling of natural scenes for content-based image retrieval. Int. J. Comput. Vis. **72**(2), 133–157 (2007)
35. J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo, in *CVPR* (2010)
36. H. Zhang, A.C. Berg, M. Maire, J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition, in *CVPR '06* (2006)
37. W. Zhang, J. Kosecka. Image based localization in urban environments, in *3DPVT '06* (2006)
38. Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, H. Neven. Tour the world: building a web-scale landmark recognition engine, in *CVPR* (2009)

Chapter 4
Vision-Based Fine-Grained Location Estimation

Heng Liu, Tao Mei, Houqiang Li and Jiebo Luo

Abstract In this chapter, we explore a variety of vision-based location estimation techniques, in which the goal is to determine the location of an image at a fine-grained level. First, we introduce the concept about *image-based location and landmark recognition* (Sect. 4.1), which determines the location of a given image by leveraging collections of geo-located images. Early techniques usually treat this as a similar image matching problem and use the geo-tags transferred from the matched database images. Some recent works have examined how to estimate more fine-grained and comprehensive geo-context information, such as the *viewing direction estimation* (Sect. 4.3) of photos. Next we will review the techniques for *city-scale location recognition*, *informative codebook generation*, and *geo-visual clustering* (Sect. 4.4). Moreover, we will introduce the structure-from-motion technique, which is closely related to estimating the camera geo-location by generating 3D models. With the 3D scenes reconstructed from the image collections, images are localized by *2D–3D alignment* (Sect. 4.5). The camera location, viewing direction, and scene location are estimated simultaneously, which are essential to various applications. Moreover, another class of vision-based location estimation technique using *satellite-imagery* database is also described (Sect. 4.6).

H. Liu (✉) · H. Li
Department of Electronic Engineering and Information Science, University of Science
and Technology of China, Hefei, China
e-mail: sorcerer@mail.ustc.edu.cn

H. Li
e-mail: lihq@ustc.edu.cn

T. Mei
Microsoft Research, Beijing, China
e-mail: tmei@microsoft.com

J. Luo
Department of Computer Science, 611 Computer Studies Building, University of Rochester,
Rochester, NY, USA
e-mail: jluo@cs.rochester.edu

J. Choi and G. Friedland (eds.), *Multimodal Location Estimation of Videos and Images*,
DOI 10.1007/978-3-319-09861-6_4

4.1 Landmark and Location Recognition

The problem of location recognition has been studied in the past few decades, and a variety of approaches have been proposed [1, 2]. The basic idea is to calculate the position of a query image with respect to a database of geo-referenced images [2]. Most of these methods consider this as an image matching problem, that is, accurate matching to images of the same scene (Sect. 4.2).

However, estimating the geo-information of an image is not only finding similar images. More comprehensive parameters, such as viewing direction can also be estimated by considering geometric relationships between ground-level image pairs (Sect. 4.2) or between the ground-level image and satellite-imagery (Sect. 4.6) or aligning image to 3D scene models (Sect. 4.5). There is a trend of using massive amount data for large-scale vision-based location estimation, related techniques such as large-scale image database indexing, discriminative code-book generation, geo-visual clustering are described (Sect. 4.4). The comprehensive and fine-grained geo-information is useful for a variety of applications (Sect. 4.6).

4.2 Image-Based Location Recognition

Given a user provided photo as the query image, the image-based location recognition is to determine the geo-location of the query image. Typically, it requires to leverage collections of geo-located images for training or matching. Furthermore, location recognition methods generally attempt to directly match feature points or structures from an unknown image to the images with known locations. In [3], a method is proposed to find a probabilistic distribution of the location of an unknown image by searching similar images. The idea is to use a data-driven scene matching approach. An image is summarized using typical scene descriptors including color histograms, GIST [4], and texton histograms and then compared against a dataset of over 6-million GPS-tagged images [3]. This approach is able to estimate the locations of images by summarizing the likelihood that such an image is captured in a particular part of the world. In essence, the training dataset is used to represent general appearance (e.g. color, the amount of vegetation, and the type of architecture) of different locations. The estimated location is correct to within 200 km about 16 % of the time.

Instead of using global features that summarize the scene of an image, the methods using interest point detections show better performance for accurate location recognition [1, 5, 6]. The interest points (detected using local feature detectors such as SIFT [7], SURF [8], or MSER [9]) from a query image are matched to the interest points in one or more training image sets to find the images which contained at least some overlap with the query image. In [2], the location estimation is further refined using a triangulation procedure based on planar homography transformations to the top image matches. The landmark recognition is performed by first finding a set of candidate matches [10]. Each candidate receives a score measuring the similarity between the query and the candidate. Then the top scored image is used to determine the location.

4.3 Estimating the Camera Viewing Direction

In [11], the problem of image geo-tagging is extended to estimate the viewing direction given the camera location of the image. An approximation method that uses the homography constraint only is proposed, which is less dependent on the accuracy of camera parameters of the database images. Instead of estimating homography transformation and the camera parameter directly, it is assumed that any 3D plane that induces homography is mostly vertical on street-level images. Thus a simplified method is proposed for viewing direction estimation as follows.

Given a query image, U, and the relevant images (S_i for $i = 1 \sim \mathrm{N}$) collected from the SIFT matching process, there are N viewing directions associated with S_i and N sets of matching correspondences, denoted by M_{us} between U and S_i. To estimate an initial rough viewing direction, each field of view is examined at the center of matched street view images. The overlapping regions between all street view images are found and given relevance weights proportional to the number of inliers, as shown in Fig. 4.1a.

Then a Parzen window estimation is used to find the highest mode of the 2D distribution of the location of interesting region and to obtain an initial estimation of the viewing direction as a ray coming from the center of user location to the highest mode, as shown in Fig. 4.1b. When estimating the 2D viewing direction, (i.e., the yaw angle), the problem can be relaxed by considering only x-axis information from M_{us}. Through this relaxation, the optimization that would be required by triangulation of rays with some assumptions is avoided. It is assumed that the principle point of each

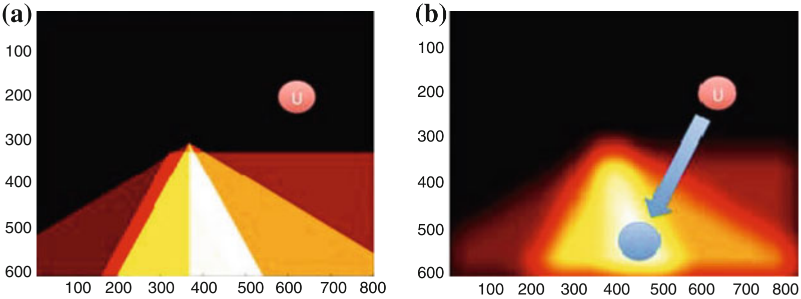


Fig. 4.1 The initial estimate of yaw angle. The *red circles* in (**a**) and (**b**) indicate the user the *blue circle* in (**b**) indicates 2D location of interesting object. **a** Field of view (FOV) at every Street View center. Each region covered by FOV is given a relevance weight proportional to the number of inliers found in matching. **b** Parzen window estimation of interesting area seen by Street View images

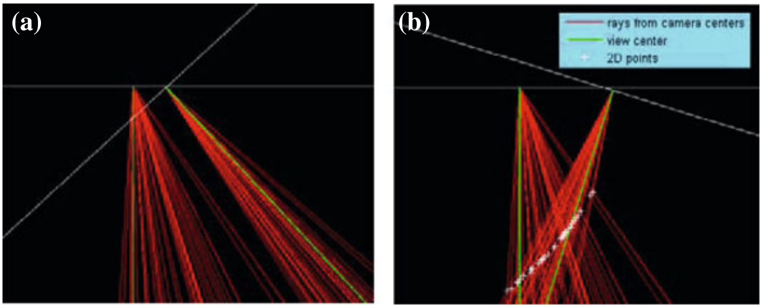


Fig. 4.2 The *red lines* are rays from each camera center. The *green lines* are rays passing through the principal points. The *white crosses* are the triangulated 2D points. **a** Triangulation of rays cannot find any crossing points visible by both cameras showing the given viewing direction is not possible. **b** 2D points that are visible by both cameras are computed by triangulation of rays

S_i and U is the center of each image, and the FOV of U is extracted from the camera metadata. Then the 3D rays are projected onto the $y = 0$ plane to form 2D rays. Planes intersect with the $y = 0$ plane results in 2D lines on the $y = 0$ plane. Therefore, the proposed method tries to find collinear 2D points on the $y = 0$ plane, since the projection of a 3D plane onto the $y = 0$ plane remains to be a line approximately. By varying the FOV horizontally for each S_i and the viewing direction of the user photo, a hypothetical 2D point for each of M_{ui} is computed by triangulation, as shown in Fig. 4.2b. Since these 2D points, (x_k, y_k) for $k = 1 \sim m$ should be approximately collinear. A scatter matrix of the 2D points is built by:

$$S = \frac{1}{m}\begin{bmatrix} \sum_{k=1} m(x_k - \bar{x})^2 & \sum_{k=1} m(x_k - \bar{x})^2 \\ \sum_{k=1} m(x_k - \bar{x})^2 & \sum_{k=1} m(y_k - \bar{y})^2 \end{bmatrix} \tag{4.1}$$

where $\bar{x}$ and $\bar{y}$ are the mean of x_k and y_k, respectively. Then the eigenvalues of S is computed to measure the collinearity of the point (x_k, y_k). The viewing direction and horizontal FOV of S_i that minimizes a ration r is selected by:

$$r = \frac{\lambda_{\min}}{\lambda_{\max}}, \tag{4.2}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalues of the scatter matrix S.

4.4 City-Scale Location Recognition

When a database contains more images, e.g., the street view images for a city (up to million images), the computational cost of forementioned approaches which use direct feature matching is unaffordable. There are usually hundreds to thousands features extracted from an ordinary street view image. So for a city scale street view image database with million images, there will be up to billion features in total.

On the other hand, several techniques that have emerged to make location recognition approaches scale up for such huge collections. First we will review large-scale image database indexing (Sect. 4.4.1). Next, several approaches for selecting discriminative features or generating informative visual codebooks are described (Sect. 4.4.2). Clustering images that are geographically proximate and visually similar into subsets is also an efficient way to boost the location estimation performance in large datasets (Sect. 4.4.3).

4.4.1 Large-Scale Image Database Indexing

To address the problem for fast and accurate image similarity calculation with high dimensional feature, several techniques are widely used. One is based on nearest neighbor search, which uses methods such as kd-tree for approximate and fast feature matching. For example, FLANN [12] is used to index all the SIFT features of a database image into a kd-tree to estimate the location of a query [6]. However, these methods need to store all the image features thus the memory cost is large and will not be suitable for million scale database. Another way is using bag-of-words (BoW) model inherited from document retrieval. Sivic and Zisserman first proposed to use BoW in image retrieval by quantizing the local descriptors [13]. Each descriptor is mapped to an integer index representing the visual word ID. An image is represented by a visual word histogram instead of its feature descriptors. The similarity between two images is measured by the distance between their BoW histograms. Typically, the visual words are trained from a sampled SIFT descriptor set by clustering, while the resulting descriptors are defined as visual words. For efficient clustering and quantization, a hierarchical tree-based structure is adopted and the resulting leaf nodes are considered as visual words [14]. Compared with nearest neighbor descriptor matching, BoW-based methods are better in terms of compactness and memory requirements thus have better scalabilities. In the city-scale location recognition, Schindler et al. extend feature point matching methods to investigate the use of large vocabulary trees for finding the location of query image by matching it to large geotagged image database [1]. Extensions of interest point-based approaches are widely seen in large-scale location recognition works such as [5, 15–18]. The hierarchical tree has two basic settings: branching factor B and hierarchical layer H, where B controls the cluster numbers at each iteration and H determines the depth of tree. In general, the visual word number is approximate B^H and the descriptor comparisons which quantize a descriptor is equivalent to $B \times H$. For example, in the implementation of the proposed system "accurate mobile visual localization" (AMVL) [16] the parameters are set to $B = 10$, $H = 6$, yielding a codebook with 1 million visual words and the inverted file system is exploited to index the images with the trained vocabulary tree.

4.4.2 Informative Codebook Generation

One of the main challenges using interest point-based matching methods for location recognition is the mismatches in image retrieval caused by noisy image features. Especially in urban areas, features from common objects such as vegetation, automobiles and pedestrians will lead to incorrect retrieval results. Since these objects are widely distributed and the features have similar appearances that quantized to the same visual word may actually come from faraway objects. Some researchers try to avoid the mismatches brought by these confusing features. One method is tracking features in a video sequence to remove moving objects such as cars or pedestrians in the query [17]. However, the static confusing objects such as trees or road marks cannot be handled by this method. Turcot and Lowe take features that are robust to view changes [19].

A similar scheme is proposed by Knopp et al. to detect confusing objects by using the available geo-tags as image labels to train feature classifiers [20]. The images from far away locations are considered negative examples, since they depict different places, so image patches in which detected features match these negative examples are detected as confusing layers. These features are suppressed in image representation to improve retrieval accuracy. The afore mentioned two methods need to match features between each image pair and build an image adjacency graph in the training stage, making the process computationally very energy-consuming. In another paper [21], Doersch et al. tried to find representative image patches of a certain city. They propose using support vector machines (SVM) for iteratively learning the patches that frequently appear in a city and distinguish this city from other cities. More commonly, a term frequency-inverse document frequency (tf-idf) model is used to weigh the visual words [13], but this only considers information from descriptors. Similarly, Ji et al. proposed to create a ranking sensitive projection matrix for each region to adjust the codebook weighing [22].

In order to avoid noisy and select informative visual words for more accurate image retrieval results in location recognition. A straightforward yet effective codebook weighting scheme that takes the geo-tags of the database image as prior knowledge [16]. The proposed Location-based Codebook Weighting (LCW) embeds the geographic distribution of each visual words into the codebook. Similar to the tf-idf model, the proposed LCW prone to give lower weights to visual words from common objects, which will lead to confusing matches while giving higher weights to distinctive visual words. The motivation of LCW is that, if a visual word is related to features which are patches from common objects, such as trees, cars, or road signs, the locations of the images contained in this visual word are prone to be uniformly distributed since these objects can be seen everywhere. On the other hand, if the visual words are generated from feature patches that are in some unique buildings or scenes (such as the Arch), most of the images will be located near this building and form a peak in this area. The 2D geographic distribution of each visual word is calculated using the location of indexed images in the database. Examples of visual word distribution can be seen in Fig. 4.3, the Kurtosis of the 2D distribution is calculated as the node weight of the visual word (WoV), given by:

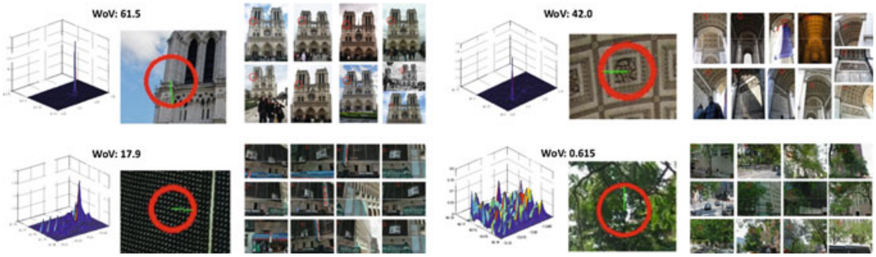


Fig. 4.3 Examples of visual word patches and their geographic distributions. The visual words that related to unique patches (*top left* Norte Dame, *top right* inner The Arch, *bottom left* a pattern in New York street view images) are in distributions with more concentrated peaks. While for nondistinctive visual words (*bottom right* patches from trees in New York street view images), the images have this visual words seem to be uniformly distributed

$$WoV(v_k) = -3 + \frac{1}{\sigma^4} \sum_{Lat} \sum_{Lon} (Lat - u_{Lat})^2 (Lon - u_{Lon})^2 \phi_{v_k}(Lat, Lon) \quad (4.3)$$

where $\phi_{v_k}(Lat, Lon)$ is the distribution function of v_k with (Lat, Lon) as the variables, u_{Lat} and u_{Lon} are expected values of (Lat, Lon) respectively. We can see from Fig. 4.3 that with higher Kurtosis value, the related images are more concentrated. Otherwise the image location distribution is diverse. Therefore, the computed WoV is used to weight visual words in the codebook for better location recognition performance.

4.4.3 Geo-Visual Clutering

As we know, there are numerous images with GPS tags and other contextual information in geo-referenced image datasets. There are several types of geo-referenced datasets, which are collections of images with GPS information. One way for collecting images is harnessing street-view images with vehicles equipped with cameras and other sensors (e.g., GPS) [5]. Images collected in this way are consistent and have high level of accuracy (e.g., Google Street View and Bing Streetside). Another commonly seen type of geo-referenced image datasets are images collected from increasingly popular and accessible online photo-sharing services. More and more people are sharing their photos on these websites such as Flickr and Panoramio. A portion of these social images are captured using devices with a GPS module or manually labeled with geographic information. While these community resources are plentiful, their contents are often diverse and prone to noisiness. These abundant resources of geo-referenced images have proven very useful in location recognition.

As can be seen from previous section Sect. 4.4.1, indexing images with local feature descriptors is an efficient way for location recognition. To better utilize these geo-referenced datasets for location recognition, a number of strategies similar to query expansion are used. Clustering visually consistent images into groups is an effective approach. Avrithis et al. [15] proposed to cluster images to compress a large corpus of images to form “scene maps” which depict different views of the same scene. All views are mapped to one reference image to construct the 2D scene map by preserving details from all images while discarding repeating visual features. The indexing, retrieval and spatial matching scheme then operates directly on scene maps. This can boost the retrieval recall performance thus lead to more precise location recognition. Visual clustering is very helpful for the speeding up of 3D reconstruction in a hierarchical way [10], which is related to more accurate location estimation with 2D–3D matching that will be introduced in Sect. 4.4. Similar with recent approaches, a two-layer clustering scheme that considers both geographic and visual similarity is proposed in [23]. First the location information (latitude, longitude) is used. Only geographically close images are grouped in the same cluster, assumed that faraway images are impossible depicting the same scene. This significantly reduce the computational costs of the subsequent visual-clustering. Images far away from each other are disconnected by leveraging an open-ball operation for simplicity, with the radius r_g. Thus the geographical similarity matrix is computed as follows:

$$\mathrm{Sim}_g(i, j) = \begin{cases} g_{i,j}, & d_{\mathrm{geo}}(i, j) < r_g \\ 0, & d_{\mathrm{geo}}(i, j) \geq r_g \end{cases}, \tag{4.4}$$

where $g_{i,j} = \exp\{\frac{-d_{\mathrm{geo}}(i,j)}{\sigma_g}\}$ denotes the geo-proximity of two images, $d_{\mathrm{geo}}(i, j)$ is the distance from the location of image i to image j, and σ_g is the scale factor. Then, the affinity propagation (AP) algorithm is adopted to get the geo-clusters [24]. The total number of groups is automatically inferred from the data.

When the geo-clusters are determined, clustering based on visual similarity of images is further conducted. Same as geo-clustering, it requires to compute a matrix Sim_v for each geo-cluster that represents the pairwise visual similarity between images in this cluster. Unlike geo-clustering, to compute the pairwise similarity of images is the most computational procedure. Typically this is done by matching local visual features according to a variety of different methods like mutual nearest neighbors or distance ratio, followed by checking consistency in geometry by means of RANSAC and other forms of spatial matching [25]. The inverted-indexing technique is employed here to speed up this procedure. For each image in the cluster, its features are quantized to visual words as the same as described in Sect. 4.4.1, then each image is represented using the histogram of visual words. Matching between this image and other images in the cluster is conducted using the inverted index file and followed by a fast geometric verification. This matching is conducted in a small subset (within a geo-cluster) instead of the whole database thus the computational cost is greatly reduced. Furthermore, we restrict that only image pairs that have an inlier match feature number more than r_v are connected in the visual graph. So the visual similarity matrix is:

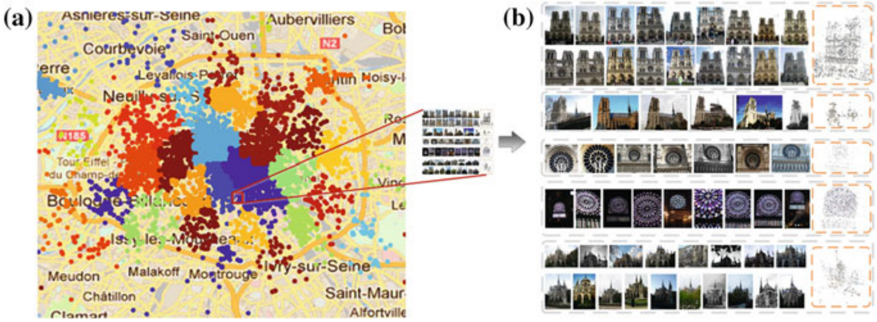


Fig. 4.4 An example of geo-visual clustering in the city of Paris. **a** The geographical regions are generated by employing affinity propagation clustering. Different clusters are denoted by different colors. **b** Part of the visual clusters in one geo-cluster near Norte Dame de Paris. Each row represents images in the same visual cluster along with the reconstructed 3D scene model

$$\mathrm{Sim}_v(i, j) = \begin{cases} v_{i,j}, & v_{i,j} \geq r_v \\ 0, & v_{i,j} < r_v \end{cases}, \tag{4.5}$$

where $v_{i,j}$ is the number of inlier matches between image i and j, representing their visual similarity. The clustering results of geo-clustering of community photo collections in a city and an example of visual clusters of a geo-cluster are shown in Fig. 4.4, and each row in Fig. 4.4b depicts the images in the same visual cluster. Reconstructed 3D model is shown at end of each row in Fig. 4.4b.

4.5 Location Estimation by 2D–3D Alignment

2.4 Another topic that is closely related to location estimation is 3D scene model reconstruction from image collections. In this section, we will briefly review the 3D reconstruction technique first Sect. 4.5.1 and then describe how the comprehensive geo-information including camera location, viewing direction and scene location is estimated using 2D–3D alignment Sect. 4.5.2.

4.5.1 3D Model Reconstruction

The 3D information related to the location itself can also be incorporated to location recognition. Estimating image camera geo-location is a topic that closely related to the fields of camera calibration. In [26], image features that are visible in two or more images of a scene are used to estimate the geometric relations between image pairs. Then both the coordinates of the image features and the camera parameters (including the location of each cameras in the 3D coordinate frame, rotation matrix, and other intrinsic parameters) are estimated in an external 3D coordinate frame. In [19], the structure-from-motion (SfM) technique is explored to reconstruct 3D models of Internet photo collections of landmarks. The estimation of image location is accomplished by placing the reconstructed 3D model within the context of the coordinate system of the Earth. In [10, 27–30], images are localized by matching the 2D image feature points to the 3D point cloud model.

In [16, 23], the location estimation technique based on 2D–3D feature matching is extended to city-scale cooperating with the forementioned geo-visual clustering process. Through the fore two-stage clustering, images are grouped into clusters. Images in the same cluster are visually similar and geographically close. Let $\{\mathscr{C}_i\}_{i=1}^{N}$ depicts the N image clusters . Each cluster $\mathscr{C}_i = \{I_i^m\}_{m=1}^{M_i}$ consist of images depicting the same scene. Instead of representing the scenes with sets of images, a better choice is using 3D models which contains stronger geometric constraints. These geometric constraints deliver the pose of images directly compared to the 2D image matches.

3D models are created using multiple-view vision methods from image clusters $\mathscr{C} = \{I^m\}_{m=1}^{M}$. In [16, 23] the reconstruction of 3D scenes from images in the same cluster is mainly based on *SfM* algorithm [31]. The relevant steps of 3D scene reconstruction can be briefly summarize here. First, for each pair of image, the key point descriptors is matched using the approximate nearest neighbors (ANN) kd-tree algorithm [32]. Then, the set of cameras and 3D points can be reconstructed by bundle adjustment. The reconstruction starts from a selected initial image pair that has a large number of matches while with certain viewpoint changes. The 5-point algorithm is used to estimate camera parameters for this initial pair [33] and matched image points are triangulated to estimate their 3D coordinates. Then, the scene model is incrementally built by adding a few images to the model at a time.

The reconstructed model $\mathscr{S}(\mathscr{C}) \triangleq (\mathscr{X}, \mathscr{A}, \mathscr{W})$ of a cluster $\mathscr{C}$ is represented with following structures: (1) A set of 3D points $\mathscr{X} = \{X^n\}_{n=1}^{N}$ in 3D Euclid coordinate (U, V, Z) , (2) A set of camera $\mathscr{A} = \{A^m\}_{m=1}^{M}$, where each camera A^m consists of an image I^m, a rotation matrix $\mathbf{R}^m$, a translation $\mathbf{t}^m$, and a focal length l^m, and (3) a set of Binary mappings $\mathscr{W} = \{w_{n,m}\}_{n,m=1}^{n=N,m=M}$ between the points and cameras indicates whether X^n could be observed from camera A^m. If $X^n = (X_U^n, X_V^n, X_Z^n)$ is observable in A^m, then there exists an 2D feature point $x^{n,m} = (x_u^{n,m}, x_v^{n,m})$ represented in 2D image coordinate (u, v). At each iteration, the reconstructed model is further refined using bundle adjustment. It is equivalent to the following optimization problem that minimize the reprojection error:

$$\arg \min_{X,\mathbf{R},t,l} \sum_{X^n,\mathbf{R}^m,t^m,l^m} w_{n,m} \parallel x^{n,m} - \widehat{x}^{n,m} \parallel^2, \tag{4.6}$$

where $\widehat{x}^{n,m} = (\widehat{x}_u^{n,m}, \widehat{x}_v^{n,m})$ is the reprojection of X^n into image I^m, the reprojection is computed as:

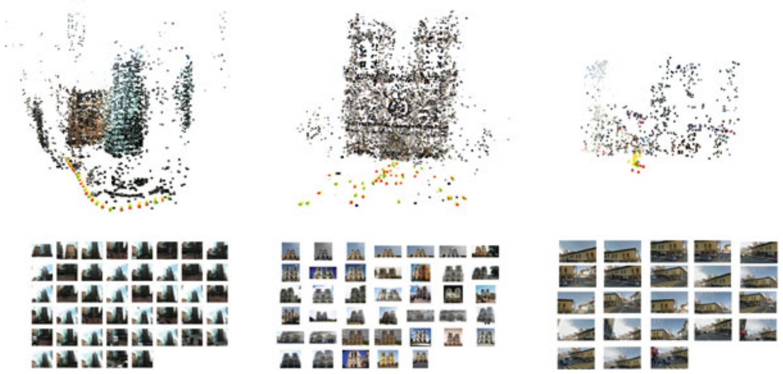


Fig. 4.5 Visual clusters and reconstructed 3D scene models. The examples are from three different datasets. *Left* Model in "SF street view" [5]. *Middle* Model in "Flikcr five cities" [23]. *Right* Model in "Google street view" [16]

$$\lambda \begin{pmatrix} \widehat{x}_u^{n,m} \\ \widehat{x}_v^{n,m} \\ 1 \end{pmatrix} = l^m(\mathbf{R}^m \mathbf{X}^n + \mathbf{t}^m), \tag{4.7}$$

where λ is an arbitrary homogeneous scaling factor. Thus bundle adjustment jointly refines the estimated camera parameters and 3D points. Finally, when the reconstruction has been completed and the coordinate is converted to the real world coordinate. With labeled GPS information of the images, a similar transformation is estimated to register the 3D model into real world coordinate. Then, the query image is localized by searching and registering to the 3D scenes in this superior geo-referenced database Fig. 4.5 shows some 3D model reconstructed from image visual clusters.

4.5.2 Image Localization by View Registration

The reconstructed 3D models and the related images are indexed using forementioned BoW model with inverted files. When estimating the location, first similar images are retrieved for the query image. The retrieval system then returns a short list of candidate images, along with filtered feature matches between the query image and these candidates. At the same time, all the 3D scene models $\{\mathscr{S}(\mathscr{C}_i)|q \cap \mathscr{C}_i \neq \oslash\}$ in the database that has some candidate images related to are scored using these retrieved images and feature matching results. We use a majority voting scheme to select the 3D scene model that the query image q has the most overlap with. For each 3D scene model $\mathscr{S}(\mathscr{C}_i)$, a voting score $V(\mathscr{S}(\mathscr{C}_i), q)$ is accumulated as follows:

$$V(\mathscr{S}(\mathscr{C}_i), q) = \sum\nolimits_{f^q \in q} \delta(f^q, , \mathscr{S}(\mathscr{C}_i)), \tag{4.8}$$

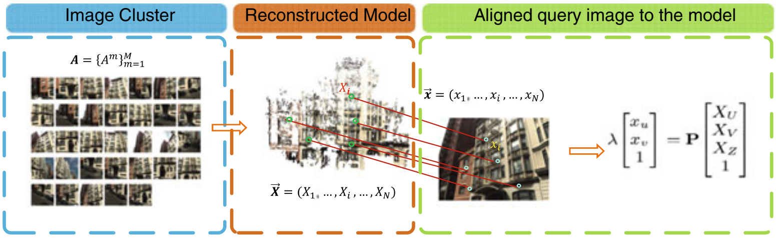


Image Cluster
Reconstructed Model
Aligned query image to the model
$A = \{A^m\}_{m=1}^M$
X_i
$\vec{X} = (X_1, \dots, X_i, \dots, X_N)$
$\vec{x} = (x_1, \dots, x_i, \dots, x_N)$
x_i
$\lambda \begin{bmatrix} x_u \\ x_v \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_U \\ X_V \\ X_Z \\ 1 \end{bmatrix}$

Fig. 4.6 3D models are reconstructed from image clusters. For localization, the query image is aligned to the scene model which shares most image feature overlap with it

where $\delta(f^q, , \mathscr{S}(\mathscr{C}_i)) \rightarrow \{0, 1\}$ indicates whether an feature point f^q in query image can find match in the candidate images that related to the scene $\mathscr{C}_i$. Thus $V(\mathscr{S}(\mathscr{C}_i), q)$ denotes the strength of the link between query image and $\mathscr{S}(\mathscr{C}_i)$ approximately by measuring how many of the feature points in the query image are covered by the scene model. The scene model with highest hits is chosen to further align the query image for accurate localization.

From the previous voting procedure, the most related scene model is determined. Figure 4.6 illustrates the registration procedure. The camera pose is then estimated by registering the query image to 3D points. For descriptors of each 3D model, a KD-tree is established. Nearest neighbors are searched in this tree to find matches for the features extracted from the query image. When there is sufficient correspondences between image features x and 3D points X, the 3×4 projection matrix $\mathbf{P}$ is estimated.

$$\lambda \begin{bmatrix} x_u \\ x_v \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_U \\ X_V \\ X_Z \\ 1 \end{bmatrix}, \tag{4.9}$$

where λ is an arbitrary homogeneous scaling factor same as in Sect. 4.5.1. There are 11 degrees of freedom (DOF) in $\mathbf{P}$ and usually the 6-point DLT algorithm is adopted [26]. $\mathbf{P}$ can be decomposed into a 3×3 intrinsic matrix $\mathbf{K}$, a 3×3 rotation matrix $\mathbf{R}$, and a translation vector $\mathbf{t}$. where

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}], \tag{4.10}$$

$$\mathbf{K} = \begin{bmatrix} l & \gamma & u_0 \\ 0 & l & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.11}$$

where l is the focal length of the camera, γ is the skew of the camera, which is often 0, and u_0 and v_0 define the principal point of the image, usually in the center. With these assumptions, only l need to be considered in $\mathbf{K}$. Notice that $\mathbf{R}$ is actually determined by three constituent Euler angles and t is a 3-vectors, and the total DOF is reduced to 7 and only 4-points are needed.

There are methods that can estimate the focal length and other parameters like 4-point-solver recently proposed in [34]. When focal length is given, e.g., in EXIF, the 3-point perspective pose estimation method could be used. According to the report in [10, 30], the camera position of the image is given by $\mathbf{R}$ and $\mathbf{t}$, the camera position of the image is given by $-\mathbf{R}'\mathbf{t}$ and view direction is $-\mathbf{R}'\begin{bmatrix}0 & 0 & 1\end{bmatrix}'$. The scene position is estimated using the 3D points that could be observed from the query image and we use the mean of them as the scene location of the image. Finally, by estimating a similar transform from 3D model's horizontal axis to the GPS coordinate, the image is registered to the real world map. Then $(\mathbf{l}_u, \theta_u, \mathbf{l}_s)$ denoting camera location, camera view direction, and scene location is calculated respectively by this similar transform. Figure 4.6 shows an example of an image aligned to the retrieved 3D scene model.

4.6 Accurate Mobile Visual Localization and Its Applications

Location recognition has many potential applications, such as automatically image annotation, geo-trajectory mining [35], visual summarization [10, 31] and virtual navigation [16, 23].

In [16], a visual-based localization approach that take advantage fo the forementioned large-scale landmark image recognition technique and image localization by 2D–3D alignment is proposed. The proposed approach is able to estimate accurate and comprehensive location information (including the camera location, viewing direction and distance to the captured scene) for the user according to the phone captured photo (typically associated with a rough GPS). The framework of the proposed system is shown in Fig. 4.7.

The comprehensive set of geo-context attributes can lead to a wide variety of LBS applications; in the following there are three applications:

Application I: On spot tour guide As its core functionality, the framework [16] can supply accurate localization and mapping service for a mobile user. Augmented reality is a promising application on mobile devices. This relies on accurate and

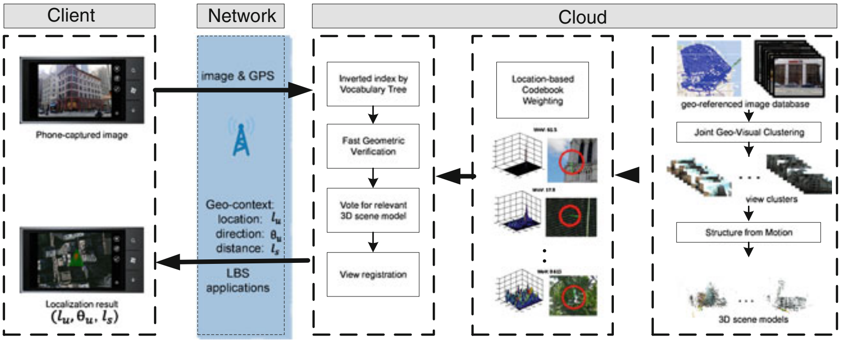
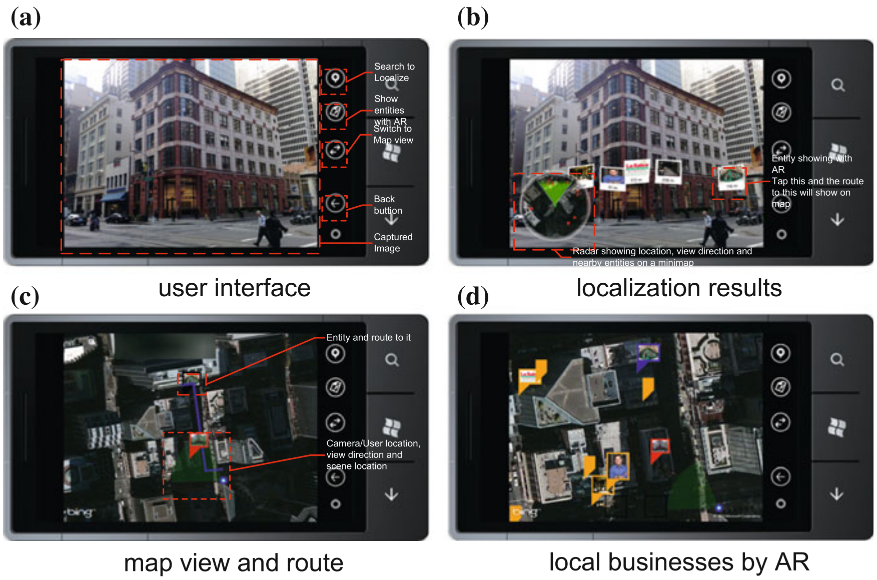


Fig. 4.7 The proposed framework for accurate mobile visual localization in [16]

comprehensive position information obtained by the camera. For example, a tourist wants to have a clearer view of the surrounding or to find his hotel around him. He needs his accurate location and direction in such applications. The user interface for this application is shown in Fig. 4.8. The user interface for mobile visual localization: (1) the user captures the scene with his phone camera, (2) the location and viewing direction are estimated by our proposed mobile visual localization approach, and the nearby local businesses are also displayed on the screen by augment reality, (3) the map view of the localization results and route to the selected destination, and (4) the map view overlaid with recommended nearby local businesses. First the user uses the camera to capture a photo as a fingerprint of the scene. Then by pressing the "localization" button the image and rough GPS is transmitted to the cloud. The service returns ($\mathbf{l_u}, \theta_\mathbf{u}, \mathbf{l_s}$), which defines the location, view direction, and location of captured scene. With the rich and accurate information, better mapping services are supplied to help the user better explore the surrounding. In addition, we can recommend the nearby local business to user and show related entities along the view direction θ_u using augment reality techniques. For example in Fig. 4.8, the radar shows the surrounding area of the user within a given radius.

Application II: Collaborative routing Another scenario is that friends in a crowded area may get separated from each other. It is difficult for them to find others, especially in an unfamiliar scene. To locate themselves and find their way back together quickly, each of them can take a photo of a landmark at his/her location and upload the photos


(a)
Search to Localize
Show entities with AR
Switch to Map view
Back button
Captured Image
(b)
Entity showing with AR
Tap this and the route to this will show on map
Radar showing location, view direction and nearby entities on a minimap
(c)
Entity and route to it
Camera/User location, view direction and scene location
bing
(d)
bing

user interface

localization results

map view and route

local businesses by AR

Fig. 4.8 The user interface for mobile visual localization. **a** shows that the user captures the scene with the phone camera. **b** Then the location and view direction are estimated by the proposed framework. The nearby local businesses are shown to the screen by augment reality. **c** The map view of the localization result and route to the selected destination. **d** The map view with nearby local businesses

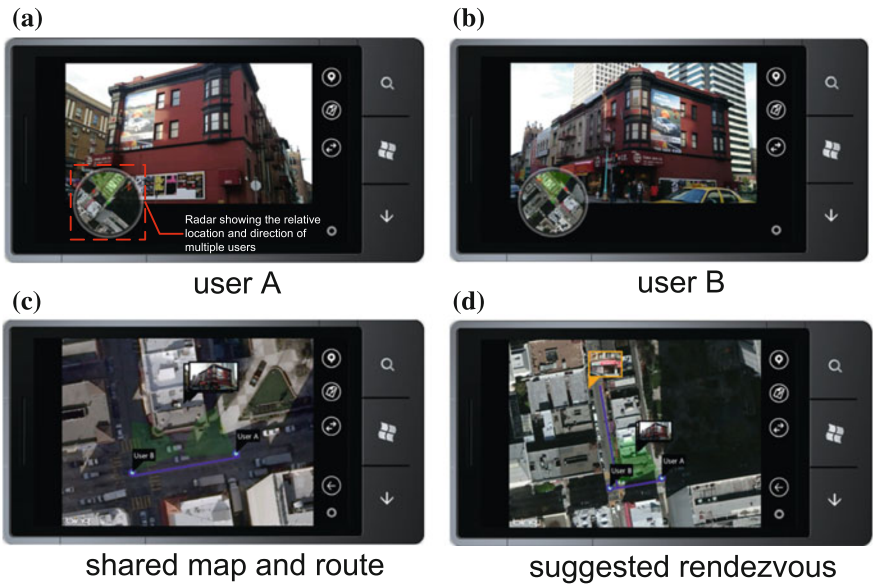


Fig. 4.9 Multiple user system interface. **a**, **b** Two users are located in collaboration. **c** Their relative locations, view directions and suggested rendezvous for them shared on the same map. **d** The suggested rendezvous and route

to the cloud. With the multiple image localization service, they will receive the accurate geographic information of themselves and their friends' . Let $(\mathbf{l}_{u1}, \theta_{u1}, \mathbf{l}_{s1})$ and $(\mathbf{l}_{u2}, \theta_{u2}, \mathbf{l}_{s2})$ denote their localized context respectively. These information are shown with a radar on the screen as a clear guideline for them: whether they have been the same scene ($\mathbf{l}_{u1}$ close to $\mathbf{l}_{u2}$), which direction should they turn to $(\theta_{u1} \leftrightarrow \theta_{u2})$. Moreover, we will recommend the rendezvous for them. According to the users' purpose (e.g., dine together) and the geo-context, and suggest the route to there (Fig. 4.9).

Application III: Sightseeing guide There is another application that especially useful for tourists. When browsing on the Internet, people may be attracted to some pictures. They will think about where these pictures were taken and decide to visit the same scenery spot. Furthermore, they probably want to take a similar photo of the scene. But what they have is only the picture with no other information. The system could help the user to find and reach the specific "rendezvous" point with the scenery. When the user feel attracted by an online photo, he could use our plugins on mobile phone to search and locate the photo. As described in Sect. 4.5, the accurate localization system returns $\mathbf{R}, \mathbf{t}, \mathbf{K}$ of the photo to user, which could be further translated to $(\mathbf{l}_p, \theta_p, \mathbf{l}_s)$. And other parameters could be suggested to the user for photographing too. Through this application, the user can have suggestions: where to go ($\mathbf{l}_p$), what to photograph ($\mathbf{l}_s$), with what camera pose (θ_p could further be decomposed to heading, tilt and roll of the camera) and other parameters (e.g., focal length) to generate a satisfied photo [36] (Fig. 4.10).

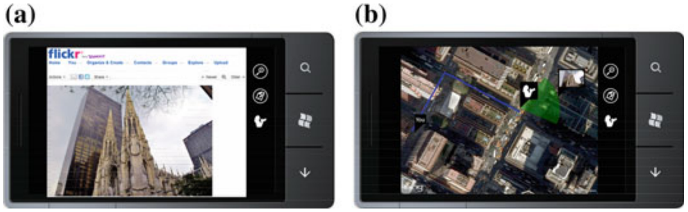


Fig. 4.10 User interface of sight seeing guide. When a user is browsing a photo online, we could calculate the location and view direction for the user to take a similar photo and suggest route from user's current location to the rendezvous for photographing

4.6.1 Aerial-Imagery Matching

In previous section, the approaches for location recognition when ground-level images are validated when the database is available. Unfortunately, the ground-level images such as Google Street View images are not always available. To further estimate the geo-location of the user photo, other modalities such as satellite-imagery and bird-view imagery are considered.

Aerial image databases such as Google Earch and Micorsoft Bird's Eye View capture the globe at vary levels of detail and can be used for acquiring geographical knowledge. In this section, we describe algorithms that use aerial-imagery matching techniques. Ground-level user photos are matched to near orthogonal satellite imagery or 45° bird-view imagery.

The satellite view image is top-down from above, while computing the match between the satellite view image and ground-level image is extremely challenging using local features such as SIFT because the appearance of common objects can vary significantly due to the wide view-point changes and different imaging conditions. The proposed method in [11] is on the basis that the ground plane and fixture objects on the ground are visible from both the satellite view and ground view (Fig. 4.11).

Provided that there are structures visible from both views, we need to align the two planes in a way that minimize the alignment error. First, the field of view (FOV) is extracted from the user image to simulate a ground-level view in a certain viewing direction by rotating the FOV on the co-located satellite image. Image patch covered by the FOV is extracted and warped to the ground-level view (Fig. 4.11). Then the horizon of the user image is detected and ground plane region is picked for the matching.

4.6.1.1 Horizon Detection

First, a user image is segmented into several regions. Then the boundary of each segmentation is taken as an edge. And the edge responses along the x axis is summed to yield a vector with length equal to the height of the image. Then a box filter is used to convolve the computed vector since the image may not be perfectly normal

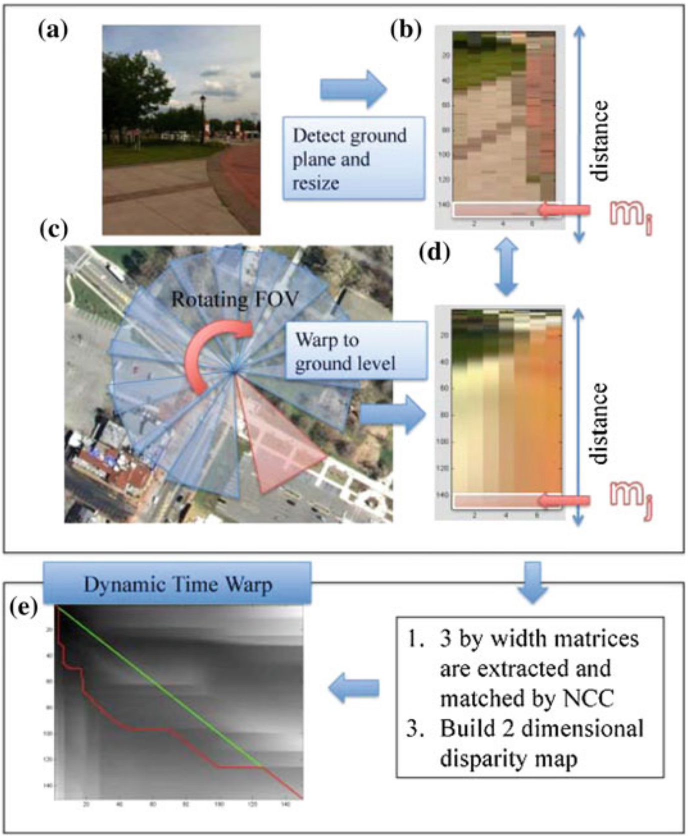

(a)
(b)
Detect ground plane and resize
distance
m_i
(c)
(d)
Rotating FOV
Warp to ground level
distance
m_j
Dynamic Time Warp
(e)
1. 3 by width matrices are extracted and matched by NCC
3. Build 2 dimensional disparity map

Fig. 4.11 Matching the ground-level user photo to a satellite view: **a** user photo, **b** detected ground plane from the user photo using horizon detection, **c** extraction of the ground plane at a specific user photo location, viewing direction, and FOV, **d** simulated ground-level view using the result of (**c**), **e** dynamic time warping and disparity score for (**b**) and (**d**)

to the ground plane. To detect a rotated horizon with possible large tilt change, the size of the box filter can be increased. Formally, the solution is given as follows:

$$I_r(y) = \sum_{x=1}^{\text{width}} I_m(y, x) \tag{4.12}$$

where $I_m(y, x)$ is the edge magnitude at pixel (x, y) on a segmented image and $I_r(y)$ is an edge response at vertical axis y.

$$I_r(y) = \frac{I_r(y)}{\sum_{y=1}^{\text{height}} I_r(y)} \tag{4.13}$$

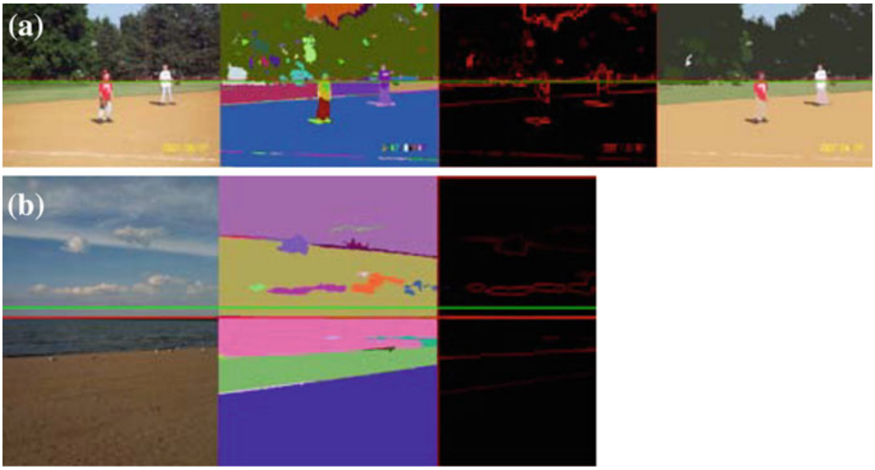


Fig. 4.12 Horizon detection: *red horizontal line* is a maximum likelihood solution and green is a minimum mean squared solution

$$y_{ML} = \arg\max_{y} (I_r * BOX)(y) \tag{4.14}$$

$$y_{\text{MMSE}} = \sum_{y=1}^{\text{height}} y \times (I_r * BOX)(y) \tag{4.15}$$

where *BOX* is a box filter and $*$ is a convolution operator. Some examples of detected horizon in the images can be seen in Fig. 4.12.

4.6.1.2 Alignment of Ground-Level Image and Satellite View

Then both images are resized into small patches, which are called codes, to normalize the horizontal axis and vertical axis (Fig. 4.11b, d). Since the same FOV is use when simulating a ground-level view from the satellite image, this normalization makes the horizontal axes of the two codes approximately correspond to each other. However, sometimes the tilt angle of the camera is unknown so the y-axes that relate to distances from a camera center may not correspond to each other. Discussion about how to take care of this is given in Sect. 4.6.2.

4.6.2 Intensity-Based Matching Through Dynamic Time Warping

Regarding the vertical axis as a time axis, the matching problem has a conceptual similarity with time series analysis problem where two signals have different speed and acceleration (e.g., speech). The similarity between two time series is evaluated by the similarity score of the two $3 \times w$ matrices (m_i, m_j), where w is the width of the code extracted from both codes at distance (i, j) at a given time (i, j) (see Fig. 4.11b, d). Having converted this matching problem to time-series analysis, then normalized cross correlation (NCC) can be used to generate a 2D disparity map (Fig. 4.11) between the codes and dynamic programming can be used to find the minimum shortest path (Fig. 4.11). Any types of appearance similarity scores and features such as the earth mover's distance, color histogram, NCC and texture can be used to help overcome the differences in terms of optics, weather, lighting and other factors which originated from two extremely different imaging conditions (one image captured by a camera on a satellite vs. one captured by a consumer-level camera on the ground). Finally,the viewing direction that generates the minimum shortest path is chosen as the solution. When estimating viewing direction a dataset of 55 images captured with smart phone with ground truth viewing directions in Washington D.C., New York City, Rochester, NY, and State College, PA areas for experiments, they show an average mean error of 11.1° and standard deviation of 9.5°.

4.6.3 Conclusions

In this chapter, we have covered a broad range of vision-based fine-grained location estimation. Starting with landmark recognition, we have seen how images can be used for geo-location estimation. Moreover, instead of studying location estimation only, we have also explored to techniques that can calculate more fine-grained and comprehensive geo-information parameters such as viewing direction. Techniques that closely related to vision-based location estimation such as 3D reconstruction, large-scale image database indexing, and image clustering are also described. There are multimodal geo-image database that are useful for location estimation, we have also described location estimation method using satellite imagery. A variety of applications have shown that vision-based location estimation is essential and very helpful for users. Having the broad material covered in this chapter, one of the recent trends in vision-based fine-grained location estimation is to use massive amount of images on the Internet together with data-driven approaches and advanced computer vision techniques that can provide more detailed information such as building structures, to get more accurate understanding of the geo-context.

References

1. G. Schindler, M. Brown, R. Szeliski, City-scale location recognition. in *Proceedings of the Computer Vision and Pattern Recognition* (CVPR) 2007. IEEE Conference on, pp. 1–7. IEEE (2007)
2. W. Zhang, J. Kosecka, Image based localization in urban environments. in 3D Data Processing, Visualization, and Transmission, Third International Symposium on, pp. 33–40. IEEE (2006)

3. J. Hays, A. Efros, Im2gps: estimating geographic information from a single image. In: *Proceedings of the Computer Vision and Pattern Recognition* (CVPR) 2008. IEEE Conference on, pp. 1–8. IEEE (2008)
4. A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vis. **42**(3), 145–175 (2001)
5. D. Chen, G. Baatz, K. Koser, S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al., City-scale landmark identification on mobile devices. in *Proceedings of the Computer Vision and Pattern Recognition* (CVPR) 2011, IEEE Conference on, pp. 737–744. IEEE (2011)
6. A. Zamir, M. Shah, Accurate image localization based on google maps street view. Comput. Vis.-ECCV **2010**, 255–268 (2010)
7. D. Lowe, Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
8. H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features. In: Computer Vision-ECCV 2006. (Springer, Berlin, 2006), pp. 404–417
9. J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. **22**(10), 761–767 (2004)
10. X. Li, C. Wu, C. Zach, S. Lazebnik, J. Frahm, Modeling and recognition of landmark image collections using iconic scene graphs. Comput. Vis.-ECCV, 427–440 (2008)
11. M. Park, J. Luo, R. Collins, Y. Liu, Beyond gps: determining the camera viewing direction of a geotagged image. in *Proceedings of the international conference on Multimedia*, pp. 631–634. ACM (2010)
12. M. Muja, D.G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration. in International Conference on Computer Vision Theory and Application VISSAPP'09). INSTICC Press (2009). pp. 331–340
13. J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos. Computer Vision, 2003. in *Proceedings of the Ninth IEEE International Conference on*, pp. 1470–1477. IEEE (2003)
14. D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree. in *Proceedings of the Computer Vision and Pattern Recognition*, 2006 IEEE Computer Society Conference on, vol. 2. IEEE (2006), pp. 2161–2168
15. Y. Avrithis, Y. Kalantidis, G. Tolias, E. Spyrou, Retrieving landmark and non-landmark images from community photo collections. in *Proceedings of the international conference on Multimedia*. ACM (2010), pp. 153–162
16. H. Liu, T. Mei, H. Li, J. Luo, S. Li, Robust and accurate mobile visual localization and its applications. ACM Trans. Multimedia Comput. Commun. Appl. **9**(1s), 51:1–51:22 (2013). doi:10.1145/2491735. http://doi.acm.org/10.1145/2491735
17. G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, E. Steinbach, Mobile visual location recognition. Signal Proc. Mag. IEEE **28**(4), 77–89 (2011)
18. F. Yu, R. Ji, S. Chang, Active query sensing for mobile location search. in *Proceedings of the 19th ACM international conference on Multimedia*. ACM (2011), pp. 3–12
19. P. Turcot, D. Lowe, Better matching with fewer features: The selection of useful features in large database recognition problems. in Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, pp. 2109–2116. IEEE (2009)
20. J. Knopp, J. Sivic, T. Pajdla, Avoiding confusing features in place recognition. Comput. Vis.-ECCV 2010 **6311**, 748–761 (2010)
21. C. Doersch, S. Singh, A. Gupta, J. Sivic, A.A. Efros, What makes paris look like paris? ACM Trans. Graph. **31**(4), 101:1–101:9 (2012)
22. R. Ji, L. Duan, J. Chen, H. Yao, J. Yuan, Y. Rui, W. Gao, Location discriminative vocabulary coding for mobile landmark search. Int. J. Comput. Vis. **96**(3), 290–314 (2012)
23. H. Liu, T. Mei, J. Luo, H. Li, S. Li, Finding perfect rendezvous on the go: accurate mobile visual localization and its applications to routing. in *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pp. 9–18. ACM, New York, NY, USA (2012). doi:10.1145/2393347.2393357. http://doi.acm.org/10.1145/2393347.2393357

24. B. Frey, D. Dueck, Clustering by passing messages between data points. Science **315**(5814), 972–976 (2007)
25. J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching. in Computer Vision and Pattern Recognition (CVPR) 2007. IEEE Conference on, pp. 1–8. IEEE (2007)
26. R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd edn. (Cambridge University Press, Cambridge, 2004) ISBN: 0521540518
27. A. Irschara, C. Zach, J. Frahm, H. Bischof, From structure-from-motion point clouds to fast location recognition. in *Proceedings of the Computer Vision and Pattern Recognition*, (CVPR) 2009. IEEE Conference on, pp. 2599–2606. IEEE (2009)
28. Y. Li, N. Snavely, D. Huttenlocher, Location recognition using prioritized feature matching. Comput. Vis.-ECCV 2010 **88**, 791–804 (2010)
29. Y. Li, N. Snavely, D. Huttenlocher, P. Fua, Worldwide pose estimation using 3d point clouds. in *Proceedings of the Computer Vision*-ECCV 2012. Springer (2012), pp. 15–29
30. T. Sattler, B. Leibe, L. Kobbelt, Fast image-based localization using direct 2d–3d matching. in Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 667–674. IEEE (2011)
31. N. Snavely, S. Seitz, R. Szeliski, Photo tourism: exploring photo collections in 3d. In: ACM Transactions on Graphics (TOG), vol. 25, pp. 835–846. ACM (2006)
32. S. Arya, D. Mount, N. Netanyahu, R. Silverman, A. Wu, An optimal algorithm for approximate nearest neighbor searching fixed dimensions. J. ACM **45**(6), 891–923 (1998)
33. D. Nistér, An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**(6), 756–770 (2004)
34. K. Josephson, M. Byrod, Pose estimation with radial distortion and unknown focal length. in *Proceedings of the Computer Vision and Pattern Recognition* (CVPR) 2009. IEEE Conference on, pp. 2419–2426. IEEE (2009)
35. C. Chen, K. Grauman, Clues from the beaten path: Location estimation with bursty sequences of tourist photos. in *Proceedings of the Computer Vision and Pattern Recognition* (CVPR) 2011, IEEE Conference on, pp. 1569–1576. IEEE (2011)
36. S. Bourke, K. McCarthy, B. Smyth, The social camera: a case-study in contextual image recommendation. in *Proceedings of the 16th international conference on Intelligent user interfaces*. ACM (2011), pp. 13–22

Chapter 5
Image-Based Positioning of Mobile Devices in Indoor Environments

Jason Zhi Liang, Nicholas Corso, Eric Turner and Avideh Zakhor

5.1 Introduction

Indoor positioning allows for many commercially viable applications, such as navigation, behavior and movement tracking, and augmented reality (AR). These applications all require the user's location and orientation to be reliably estimated. Positioning is noticeably more challenging indoors than outdoors since GPS is typically unavailable in interior environments due to the shielding effect of structures. As a result, much research has been focused on relying on other types of signals, or in our case, images as a basis for positioning.

A variety of sensors are capable of performing indoor positioning, including image [1], optical [2], radio [3–7], magnetic [8], RFID [9], and acoustic [10]. WiFi-based indoor positioning takes advantage of the proliferation of wireless access points (AP) and WiFi capable smartphones and uses the signal strength of nearby WiFi beacons to estimate the user's location. A few drawbacks are that APs cannot be moved or modified after the initial calibration, and that a large of number of APs are required to achieve reasonable accuracy. For instance, 10 or more wireless hotspots are typically required to achieve submeter accuracy [4]. The most debilitating drawback of WiFi positioning is its inability to estimate the user's orientation, which is necessary for AR applications. Other forms of indoor positioning that rely on measuring radio signal strength such as bluetooth, GSM, and RFID, also share the same strengths and weaknesses of WiFi-based indoor positioning.

There have also been previous attempts at indoor image-based positioning [1]. An image-based positioning system involves retrieving the best image from a database

J.Z. Liang (✉) · N. Corso, E. Turner · A. Zakhor
Department of EECS, UC Berkeley, Berkeley, CA, USA
e-mail: jasonzliang@eecs.berkeley.edu

N. Corso
e-mail: ncorso@eecs.berkeley.edu

E. Turner
e-mail: elturner@eecs.berkeley.edu

A. Zakhor
e-mail: avz@eecs.berkeley.edu

J. Choi and G. Friedland (eds.), *Multimodal Location Estimation of Videos and Images*,
DOI 10.1007/978-3-319-09861-6_5

that matches to the user's query image, then performing pose estimation on the query/database image pair in order to estimate the location and orientation of the query image. The authors in [1] take advantage of off the shelf image matching algorithms, namely color histograms, wavelet decomposition, and shape matching and achieve room level accuracy with more than 90 % success probability, and meter-level accuracy with more than 80 % success probability for one floor of the computer science building at Rutgers University. This approach however, cannot be used to determine the absolute metric position of the camera, nor its orientation. Thus, it cannot be used in augmented reality applications where precise position and orientation is needed.

In this chapter, we demonstrate an image-based positioning system for mobile devices capable of achieving submeter position accuracy as well as orientation recovery. The three stages of that pipeline are: (1) preparing a 2.5D locally referenced image database, (2) image retrieval, and (3) pose recovery from the retrieved database image. We also present a method to estimate confidence values for both image retrieval and pose estimation of our proposed image-based positioning system. These two confidence values can be combined to form an overall confidence indicator. Furthermore, the confidence values for our pipeline can be combined with that of other sensors such as WiFi in order to yield a more accurate result than each method by itself.

Our pipeline can be summarized as follows:

1. Database Preparation, shown in Fig. 5.1a: We use a human operated ambulatory backpack outfitted with laser scanners, cameras, and an orientation sensor (OS), as seen in Fig. 5.2, to map the interior of a building in order to generate a locally referenced 2.5D image database complete with SIFT features [11–13]. By locally referenced image database, we mean that the absolute six degrees of freedom pose of all images, i.e., x, y, z, yaw, pitch, and row, are known with respect to a given coordinate system. By 2.5D, we mean that for each database image, there is a sparse depthmap that associates depth values with image SIFT keypoints only.
2. Image Retrieval, shown in Fig. 5.1b: We load all of the image database SIFT features into a k-d tree and perform fast approximate nearest neighbor search to find a database image with most number of matching features to the query image [14–16].
3. Pose Estimation, shown in Fig. 5.1c: We use the depth of SIFT feature matches along with cell phone pitch and roll to recover the relative pose between the retrieved database image in step (2) and the query image. This results in complete six degree of freedom pose for the query image in the given coordinate system [17].

In Sect. 5.2, we describe our approach for generating sparse depthmaps during database preparation. In Sect. 5.3, we will go over image retrieval and pose estimation. Section 5.4 includes estimation of confidence values for both image retrieval and pose estimation are estimated. In Sect. 5.5, we show experimental results, characterizing the accuracy of our pipeline. Section 5.6 includes conclusions and future work.

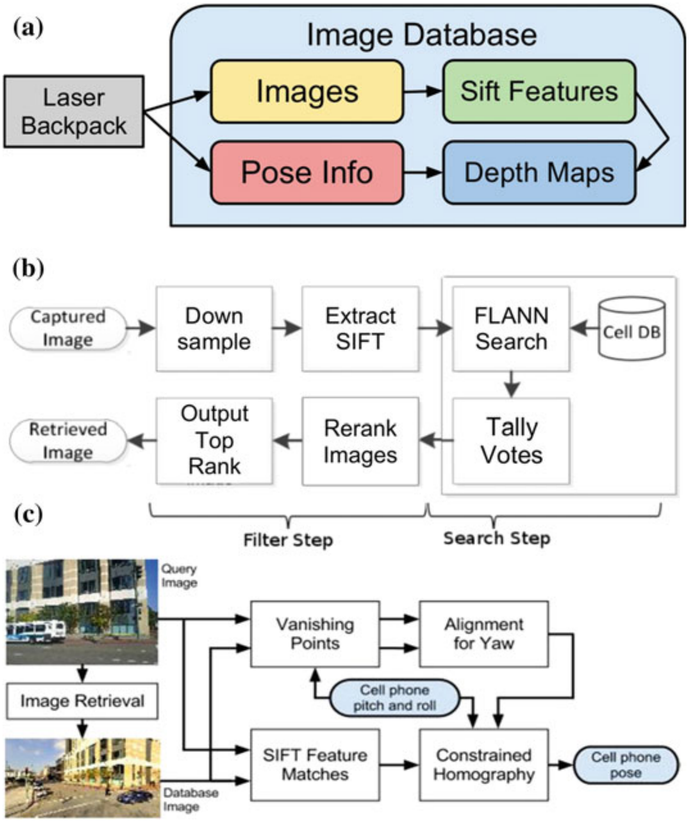


Fig. 5.1 Overview of our indoor positioning pipeline. The pipeline is composed of **a** database preparation, **b** image retrieval, and **c** pose estimation steps

5.2 Database Preparation

In order to prepare the image database, an ambulatory human operator first scans the interior of the building of interest using a backpack fitted with two 2D laser scanners, two fish-eye cameras, and one OS as shown in Fig. 5.2. The database acquisition system requires two laser scanners, namely the pitch and yaw scanners in Fig. 5.2. Measurements from the backpack's yaw and pitch laser range scanners are processed by a scan matching algorithm to localize the backpack at each time step and recover its six degrees of freedom pose [18]. Specifically, the yaw scanner is used in conjunction with a 2D positioning algorithm in [11–13] to recover x, y and yaw, the OS is used to recover pitch and roll, and the pitch scanner is used to recover z [11]. Since the cameras are rigidly mounted on the backpack, recovering the backpack pose essentially implies recovering camera pose. Figure 5.3a shows the recovered path of the backpack within a shopping center, while Fig. 5.3b shows

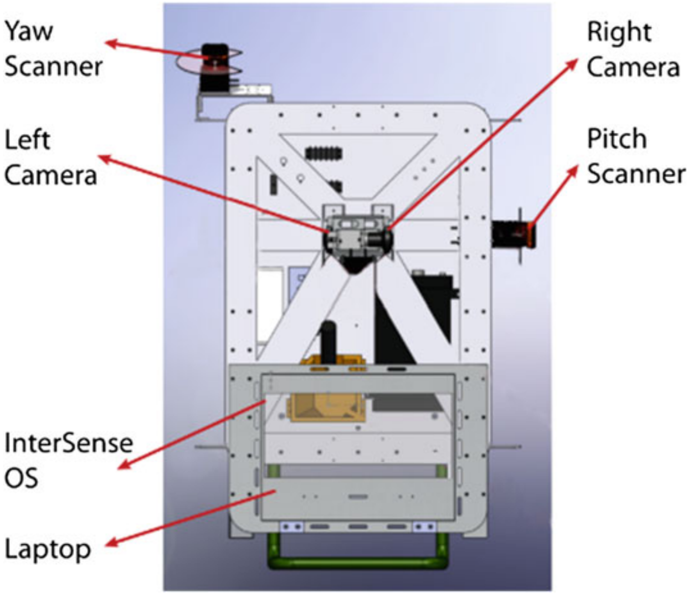


Fig. 5.2 Diagram of the data acquisition backpack

the surrounding wall points recovered by the backpack by projecting the yaw scans onto the ground plane [19]. These wall points can be connected interactively via commercially available CAD software to produce an approximate 2D floorplan of the mall as seen in Fig. 5.3c. The recovered pose of the rigidly mounted cameras on the backpack are then used to generate a locally referenced image database in which the location, i.e., x, y, and z, as well as orientation, i.e., yaw, pitch, and roll, of each image is known within one coordinate system.

To create a sparse depthmap for the database images, we first temporally subsample successive captured images on the backpack while maintaining their overlap. We then extract SIFT features from each pair of images and determine matching feature correspondence pairs through nearest neighbor search. In order to ensure the geometric consistency of the SIFT features, we compute the fundamental matrix that relates the two sets of SIFT features and removes any feature pairs which do not satisfy epipolar constraints.

We then triangulate matching SIFT keypoint pairs in 3D space. As seen in Fig. 5.4, for each pair of SIFT correspondences, we calculate the 3D vectors that connects the camera centers of the images to the respective pixel locations of their SIFT features. In doing so, we make use of the database images' poses and intrinsic parameters to ensure both vectors are correctly positioned within the same world coordinate frame. Next, we determine the depth of the SIFT features by finding the intersection of these rays and computing the distance from camera center to the intersection point. We

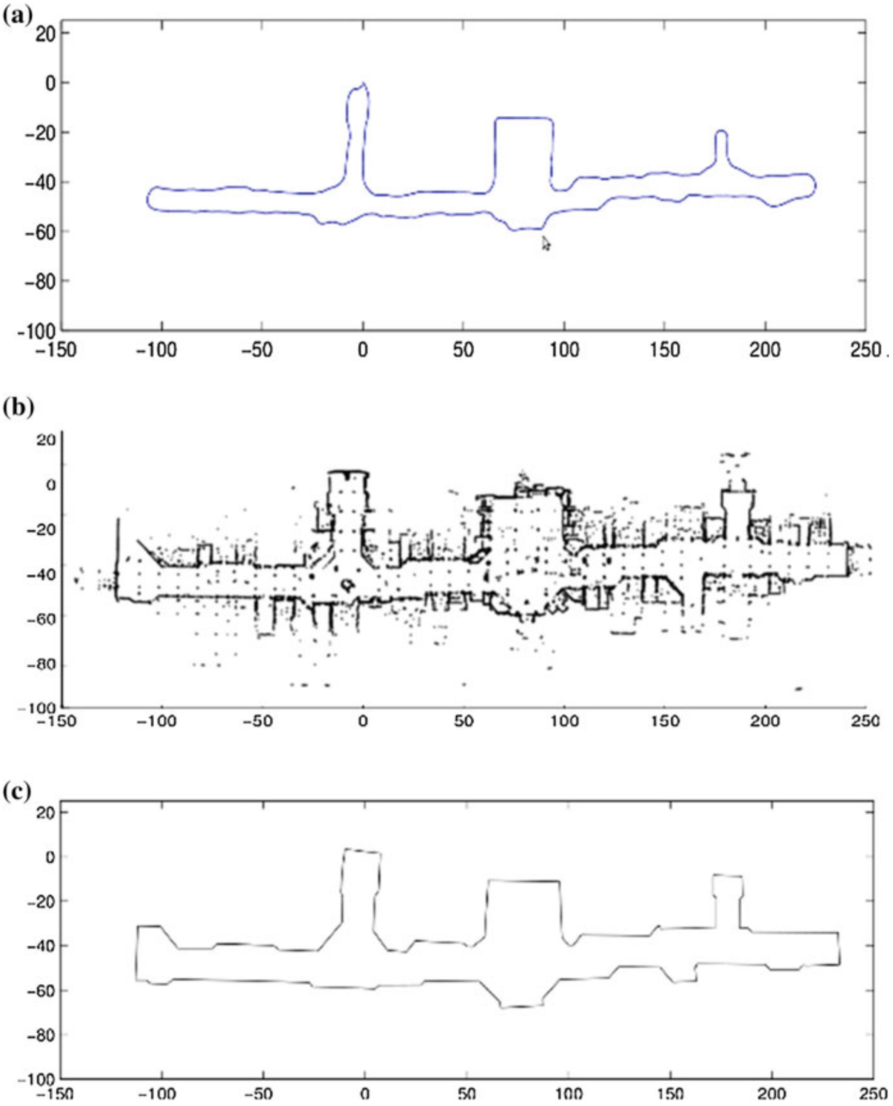


Fig. 5.3 **a** Recovered path of backpack traversal. **b** Wall points generated by backpack. **c** 2D floorplan recovered from wall points

use the following to determine the intersection point or the point mutually closest to the two vectors:

$$x = \left(\sum_{i=1}^{2} I - v_i v_i^T \right)^{-1} \left(\sum_{i=1}^{2} \left(I - v_i v_i^T \right) p_i \right) \tag{5.1}$$

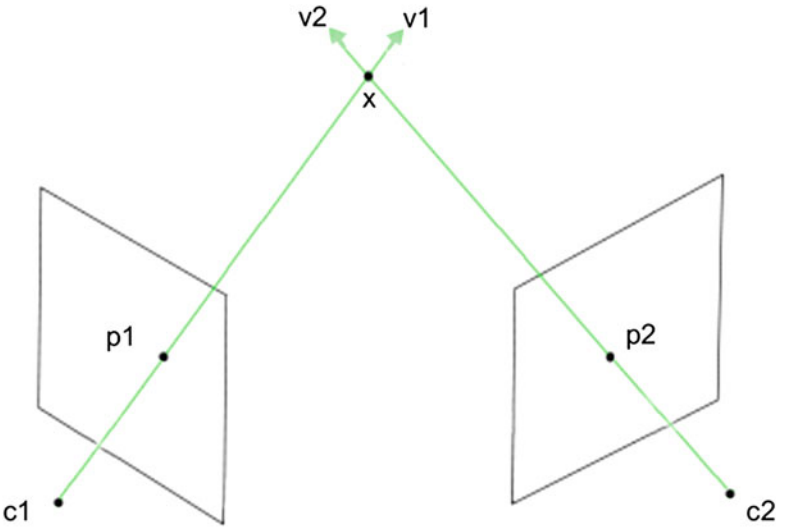


Fig. 5.4 Triangulation of two matching SIFT features. v1 and v2 are the resulting vectors when the camera centers c1 and c2 are connected to the SIFT features p1 and p2 on the image planes. The two vectors intersect at x

where x is the intersection point, v_i is the normalized direction of the ith vector, and p_i is a point located on the ith vector. The availability of highly optimized library functions for determining fundamental matrices and performing linear algebra operations means that sparse depthmap generation can be done in a matter of seconds per image. For debugging and visualization purposes, we combine the intersection points of SIFT features from every database image into a single sparse 3D point cloud, shown in Fig. 5.5a, b.

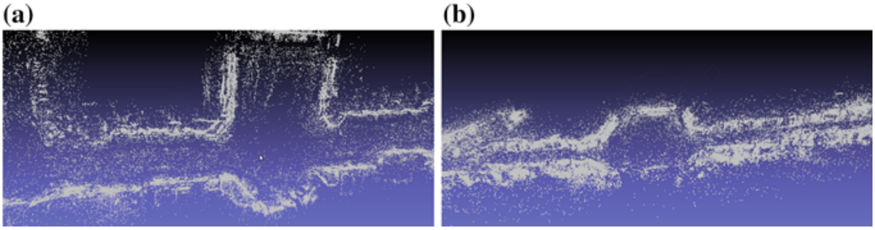


Fig. 5.5 **a** *Top-down* and **b** *side views* of sparse 3D point cloud generated from triangulation of SIFT feature correspondence of the database images

5.3 Image Retrieval and Pose Estimation

The next step of our image-based positioning pipeline shown in Fig. 5.1c is image retrieval, which involves selecting the best matching image from the image database for a particular query image. Our indoor image retrieval system loads the SIFT features of every database image into a single k-d tree [16]. Next, we extract SIFT features from the query image and for each SIFT vector extracted, we lookup its top N neighbors in the kd-tree. For each closest neighbor found, we assign a vote to the database image that the closest neighbor feature vector belongs to. Having repeated this for all the SIFT features in the query image, the database images are ranked by the number of matching SIFT features they share with the query image.

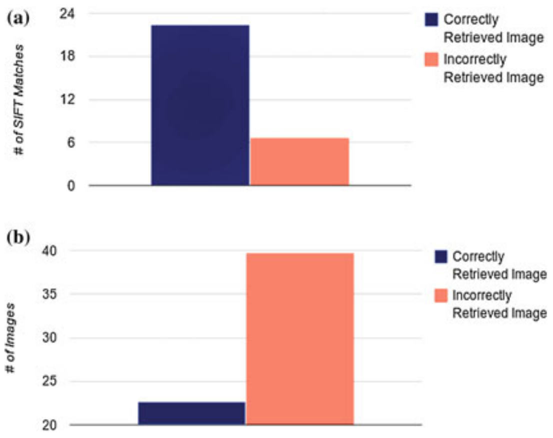
After tallying the votes, we check geometric consistency and rerank the scores to filter out mismatched SIFT features. We then solve for the fundamental matrix between the database and query images and eliminate feature matches that do not satisfy epipolar constraints [14]. We also remove SIFT feature matches where the angle of SIFT features differ by more than 0.2 rad. Since these geometric consistency checks only eliminate feature matches and decrease the scores of database images, we only need to partially rerank the database images. The database image with the highest score after reranking is exported as the best match to the query image. The image retrieval step takes roughly 2–4 s depending on the processing power of the processor used.

As shown in Fig. 5.1c, the last step of our indoor positioning pipeline is pose recovery of the query image. Pitch and roll estimates from cell phone sensors are used in vanishing point analysis to compute yaw of the query image [17]. Once we estimate orientation, SIFT matches are used to solve a constrained homography problem within Random Sample Consensus (RANSAC) to recover translation between query and database images. The method for scale recovery of the translation vector only requires depth values at the SIFT features which are considered inliers from the RANSAC homography. These depth values are present in the sparse depthmaps generated during the database preparation step of Sect. 5.2. We have also found that reducing the size of the query images significantly reduces the number of iterations required for RANSAC homography. This is because the resolution of our database images is significantly lower than that of the query image camera. If the RANSAC homography fails to find inliers, we use the pose of the matched database image as the solution. Depending on the image and speed of the processor, pose estimation requires 2–10 s.

5.4 Confidence Estimation

Our confidence estimation system consists of several classifiers that output confidence values for both the image retrieval and pose recovery steps in our pipeline. These classifiers are trained using positive and negative examples from both image

Fig. 5.6 Comparison of **a** number of SIFT matches after geometric consistency check and **b** vote ranking distribution before geometric consistency check for correctly and incorrectly retrieved images



retrieval and pose recovery stages of our proposed pipeline in Sect. 5.3. We have empirically found a logistic regression classifier to perform reasonably well even though other classifiers can also be used for confidence estimation. In order to evaluate the performance of our confidence estimation system, we create a dataset of over 270 groundtruth images where roughly 25 % are used for validation and the rest for training. To boost classifier performance, 50 out of the 270 images in the validation set are chosen to be "negative" images that do not match to any image database.

To generate confidence values for image retrieval, we train a logistic regression classifier based on features obtained during the image retrieval process. We assign groundtruth binary labels to the images in the training set that indicate whether the retrieved images matches the query images. For a given query image, the retrieval classifier generates both a predicted binary label and a retrieval confidence value between 0 and 1. We have found the following features to be well correlated with image retrieval accuracy [14]: (a) number of SIFT feature matches between query and database image before geometric consistency checks; (b) number of SIFT matches after geometric consistency checks; (c) the distribution of the vote ranking before geometric consistency checks; (d) the distribution of the vote ranking after geometric consistency checks; (e) physical proximity of the top ranking database images in the vote ranking. For example, as shown in Fig. 5.6a, the average number of SIFT matches after geometric consistency checks for correctly matched query images is over three times that of incorrectly matched query images. Likewise, as shown in Fig. 5.6b, the number of database images with at least half the number of votes of the top ranked image before the geometric consistency check is much lower for correctly retrieved images than the incorrectly retrieved ones.

Similarly for pose estimation, we train a separate logistic regression classifier on another set of features that correlate well with the pose recovery accuracy. We assign a groundtruth "True" label if the location error of a training image is below a pre-specified threshold of 4 m, and a "False" label otherwise. As with the image retrieval classifier, our pose estimation classifier generates a predicted binary label

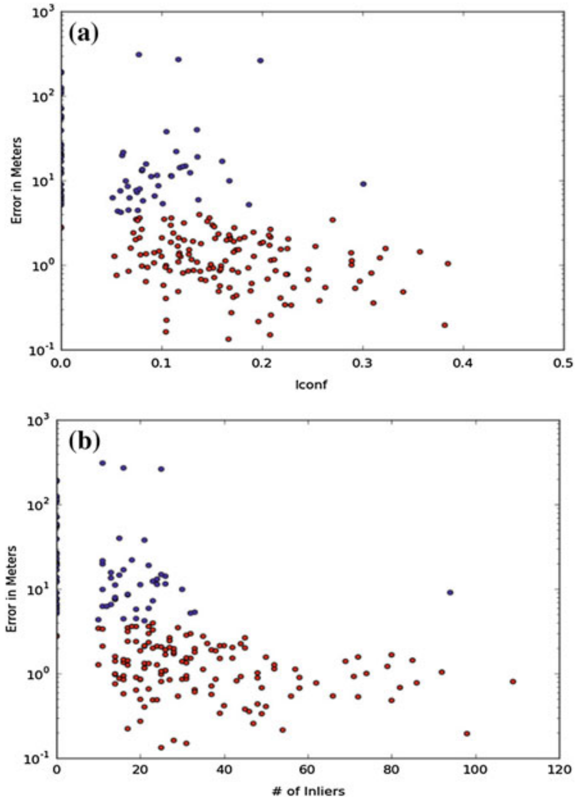


Fig. 5.7 Scatterplot of a confidence metric used in [17] and **b** number of inliers after RANSAC homography versus location error. *Red* (*blue*) *dots* correspond to images with less (more) than 4 m of location error

and a confidence value between 0 and 1. The features use to train the classifier are: (a) number of inliers after RANSAC homography; (b) reprojection error; (c) number of SIFT feature matches before RANSAC homography; (d) number of RANSAC iterations; (e) a confidence metric in [17] that is used to choose the optimal inlier set. In Fig. 5.7, we use scatterplots to visualize the correlation between some of these features and pose recovery accuracy. Specifically, Fig. 5.7a plots the relationship between the confidence metric used to choose the optimal inlier set and location error of the pose estimation while Fig. 5.7b does the same for the number of inliers remaining after RANSAC homography and location error. The red (blue) dots in the scatterplots correspond to images with less (more) than 4 m of location error. As seen, query images with larger location error tend to have less inliers and a smaller inlier set confidence metric.

We also perform support vector regression (SVR) on the training set and use the resulting regression model to predict location error of the testing set for our proposed

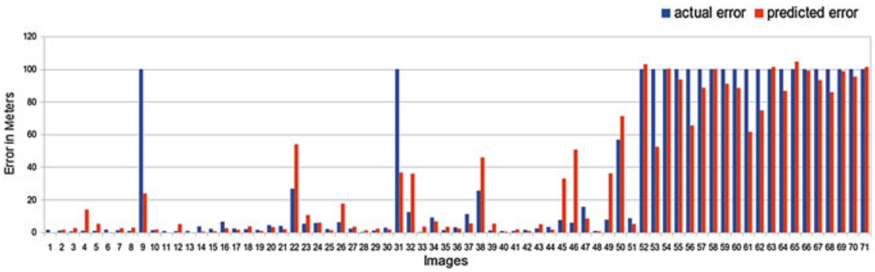


Fig. 5.8 Plot of actual (*blue*) versus predicted (*red*) location error for images in validation set using SVR regression. For the negative examples in the validation set, we set the actual error to be an arbitrary high value of 100 m

pose recovery method. In doing so, we assign an arbitrarily large location error of 100 m to the negative examples in the validation set. As seen in Fig. 5.8, there is a reasonable correlation between our predicted and actual location error.

We find the predicted binary label of the image retrieval and pose estimation confidence system to be in agreement with the actual groundtruth label 86 and 89 % of the query images in the validation set respectively. Figure 5.9a, b show the distribution of confidence values for image retrieval and pose estimation respectively on the validation set. Green (red) bars represent the confidence distribution of images whose predicted label (do not) match the groundtruth label. To create an overall system confidence score between 0 and 1 and prediction label, we use the following algorithm below:

```
overall confidence = 0.5 * (retrieval confidence + pose confidence);
if overall confidence >0.5 then
    prediction label = true;
else
    prediction label = false;
end
```

By comparing the groundtruth and the overall confidence prediction labels for the query images in the validation set, the accuracy of the overall confidence estimation is determined to be 86 %. Figure 5.9c shows the distribution of overall confidence scores for the validation set.

To determine the optimal location error threshold, we set it to values ranging from 1 to 12 m and test the accuracy of the pose estimation confidence system. As shown in Fig. 5.10, the optimal value for the threshold is around 3–5 m.

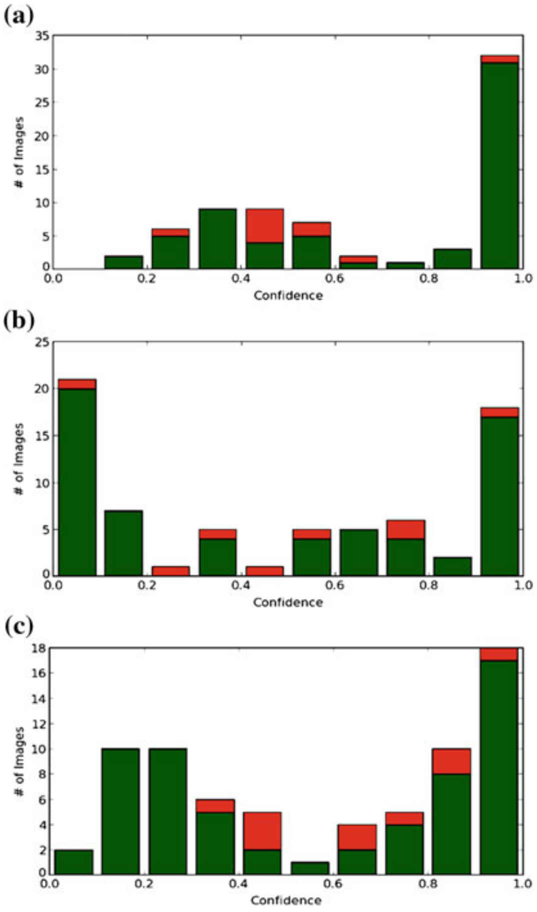


Fig. 5.9 Confidence distribution for **a** image retrieval, **b** pose recovery, **c** overall system on the validation set; *Red* (*green*) *bars* correspond to incorrectly (correctly) predicted images

5.5 Experimental Results

For our experimental setup, we use the ambulatory human operated backpack of Fig. 5.2 to scan the interior of a two story shopping center located in Fremont, California. To generate the image database, we collect thousands of images with two 5 megapixel fish-eye cameras mounted on the backpack. These heavily distorted fish-eye images are then rectified into 20,000 lower resolution rectilinear images. Since the images overlap heavily with each other, it is sufficient to include every sixth image for use in the database. By reducing the number of images, we are able to speed up image retrieval by several factors with virtually no loss in accuracy.

Our query image data set consists of 83 images taken with a Samsung Galaxy S3 smartphone. The images are approximately 5 megapixels in size and are taken using the default settings of the Android camera application. Furthermore, the images

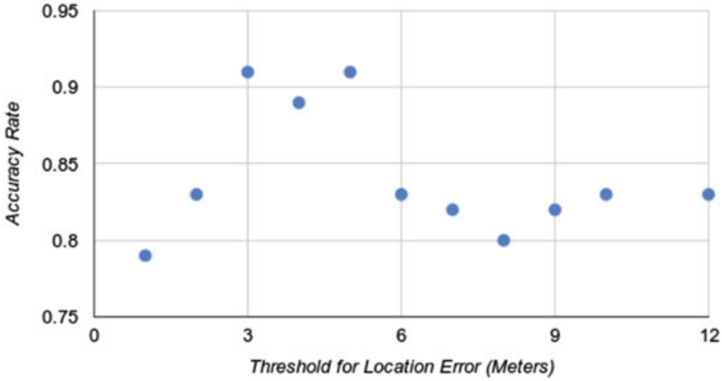


Fig. 5.10 Scatterplot showing relationship between pose recovery confidence estimation accuracy and the threshold t_s used for location error

consist of landscape photos either taken head-on in front of a store or at a slanted angle of approximately 30°. After downsampling the query images to the same resolution as the database images, i.e., 1.25 megapixels, we successfully match 78 out of 83 images to achieve a retrieval rate of 94 %. Detailed analysis of the failure cases reveal that two of the incorrectly matched query images correspond to a store that does not exist in the image database. Therefore, the effective failure rate of our image retrieval system is 3 out of 80 or less than 4 %. As shown in Fig. 5.11a, successful retrieval usually involves matching of store signs present in both the query and database images. In cases such as Fig. 5.11b where retrieval fails, there are few matched features on the query image's store sign.

Next, we run the remaining query images with successful retrieved database images through the pose estimation part of the pipeline. In order to characterize pose estimation accuracy, we first manually groundtruth the pose of each query image taken. Groundtruth is estimated by using the 3D model representation of the mall, and distance and yaw measurements recorded during the query dataset collection. We first locate store signs and other distinctive scenery of the query image within the

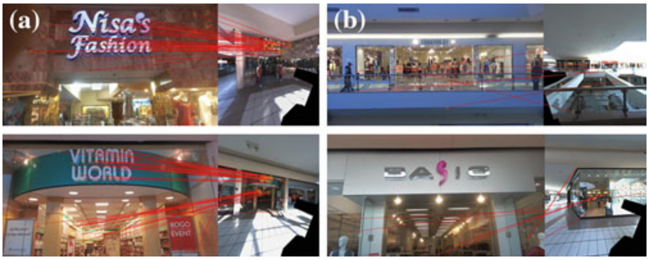


Fig. 5.11 **a** Successful and **b** unsuccessful examples of image retrieval. *Red lines* show SIFT feature matches

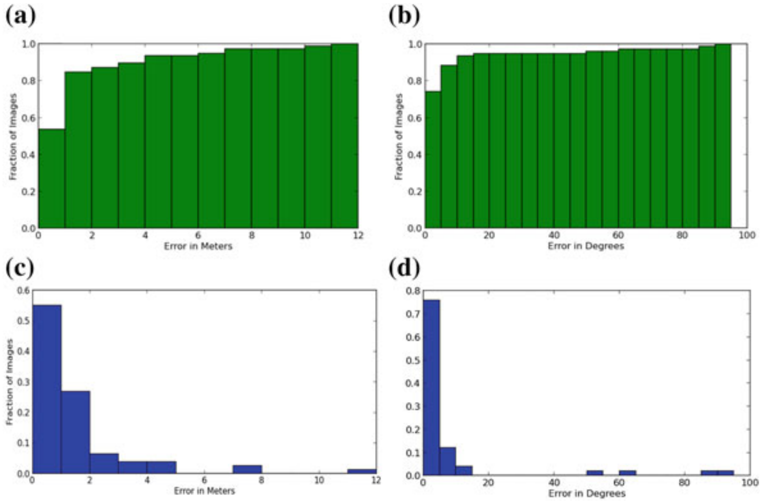


Fig. 5.12 Cumulative density function of **a** location, **b** yaw error of and probability density function of **c** location, **d** yaw error of our indoor positioning pipeline

3D model to obtain a rough estimate of the query image pose, which is then refined using the measurements. The resulting groundtruth values are in the same coordinate frame as the output of the pose recovery step.

Figure 5.12 summarizes the performance of the pose estimation stage of our pipeline. Figure 5.12a, b show the cumulative distribution functions of location and yaw error respectively while Fig. 5.12c, d show the probability distribution functions of location and yaw error. As we can see, over 80 % of the images have their yaw correctly estimated to within 10° of the groundtruth values. Furthermore, over 55 % of all the images have a location error of less than 1 m. As seen in the example in Fig. 5.13a, when the location error is less than 1 m, the SIFT features of corresponding store signs present in both query and database images are matched by the RANSAC homography [17]. Conversely, in less accurate cases of pose estimation where the location error exceeds 4 m, the RANSAC homography finds “false matches” between unrelated elements of the query and database images. For instance in Fig. 5.13b, different letters in the signs of the two images are matched. In general, we find that images with visually unique signs perform better during pose estimation than those lacking such features.

On a 2.3 GHz i5 laptop, our complete pipeline from image retrieval to pose recovery takes on average 10–12 s to run. On an Amazon EC2 extra-large computing instance, the runtime is reduced further to an average of 4.5 s per image. The individual runtimes for each image is highly variable, with some images taking twice as long as the average time.

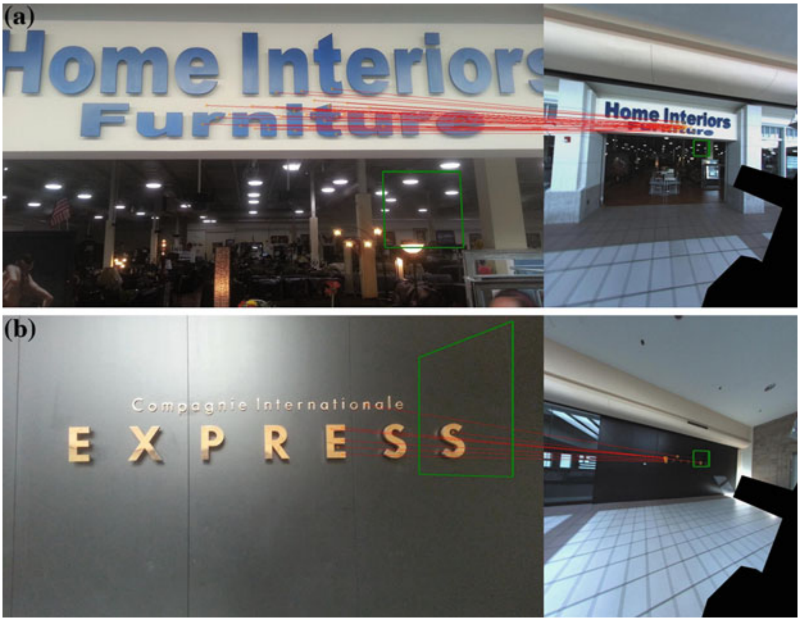


Fig. 5.13 **a** Example of accurate pose estimation on query image. **b** Example of inaccurate pose estimation. Notice how different letters in the same sign are matched

5.6 Conclusion

In this chapter, we have presented a data acquisition system and processing pipeline for image-based positioning in indoor environments. Several possible improvements to our image-based positioning pipeline include tracking the position of the user and reducing the amount of noise in the depthmaps by utilizing more images for the sparse depthmap generation process. For future research, we are planning to examine ways to further increase the accuracy of indoor positioning. One method we are exploring is to combine our image-based indoor positioning pipeline with a WiFi-based indoor positioning system. The final position is determined by a particle filter that receives measurement updates from both positioning systems.

References

1. N. Ravi, P. Shankar, A. Frankel, A. Elgammal, L. Iftode, Indoor localization using camera phones, in *Mobile Computing Systems and Applications* (2006)
2. L.I.U. Xiaohan, H. Makino, M.A.S.E. Kenichi, Improved indoor location estimation using fluorescent light communication system with a nine-channel receiver. IEICE Trans. Commun. **93**(11), 2936–2944 (2010)

3. Y. Chen, H. Kobayashi, Signal strength based indoor geolocation, in *International Conference on Communications* (2002)
4. J. Biswas, M. Veloso, WiFi localization and navigation for autonomous indoor mobile robots, in *International Conference on Robotics and Automation* (2010)
5. G. Fischer, B. Dietrich, F. Winkler, Bluetooth indoor localization system, in *Proceedings of the 1st Workshop on Positioning, Navigation and Communication* (2004)
6. S.S. Chawathe, Low-latency indoor localization using bluetooth beacons, in *12th International IEEE Conference on Intelligent Transportation Systems* (2009)
7. A. Varshavsky, E. de Lara, J. Hightower, A. LaMarca, V. Otsason, GSM indoor localization. Perv. Mobile Comput. **3**(6), 698–720 (2007)
8. J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, M. Wiseman, Indoor location sensing using geo-magnetism, in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, pp. 141–154 (2011)
9. S. Schneegans, P. Vorst, A. Zell, Using RFID snapshots for mobile robot self-localization, in *European Conference on Mobile Robots* (2007)
10. H.-S. Kim, J.-S. Choi, Advanced indoor localization using ultrasonic sensor and digital compass, in *International Conference on Control, Automation and Systems* (2008)
11. G. Chen, J. Kua, S. Shum, N. Naikal, M. Carlberg, A. Zakhor, Indoor localization algorithms for a human-operated backpack system, in *3D Data Processing, Visualization, and Transmission*, May 2010
12. T. Liu, M. Carlberg, G. Chen, J. Chen, J. Kua, A. Zakhor, Indoor localization and visualization using a human-operated backpack system, in *International Conference on Indoor Positioning and Indoor Navigation* (2010)
13. J. Kua, N. Corso, A. Zakhor, Automatic loop closure detection using multiple cameras for 3D indoor localization, in *IS&T/SPIE Electronic Imaging* (2012)
14. J. Zhang, A. Hallquist, E. Liang, A. Zakhor, Location-based image retrieval for urban environments, in *International Conference on Image Processing* (2011)
15. D.G. Lowe, Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004)
16. M. Muja, D.G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, in *International Conference on Computer Vision Theory and Applications* (2009)
17. A. Hallquist, A. Zakhor, Single view pose estimation of mobile devices in urban environments, in *Workshop on the Applications of Computer Vision* (2013)
18. N. Corso, A. Zakhor, Indoor localization algorithms for an ambulatory human operated 3D mobile mapping system. Remote Sensing **2013**, 6611–6646 (2013)
19. E. Turner, A. Zakhor, Watertight as-built architectural floor plans generated from laser range data, in *Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission* (2012)
20. E. Turner, A. Zakhor, Watertight planar surface meshing of indoor point clouds with voxel carving, in *3D Vision*, Seattle, June 2013
21. R. Hartley, A. Zisserman, *Multiple View Geometry* (Cambridge University Press, Cambridge, 2004)

Chapter 6
Application of Large-Scale Classification Techniques for Simple Location Estimation Experiments

Howard Lei, Jaeyoung Choi and Gerald Friedland

Abstract This chapter describes an application using established classification techniques for performing simple multimodal location estimation experiments. It demonstrates the use of Gaussian Mixture Model (GMM)—and language model-based approaches for verifying the cities from which Flickr videos are taken based on the videos' audio and textual metadata. The methods used in most of the approaches are described in detail, allowing people with no background in location estimation to perform simple experiments. The city-verification results for the approaches are not eye-popping by any means, but are above-random and present opportunities for future work in the development of better approaches. The techniques may also be suitable for class projects, for students who wish to gain hands-on experience in performing location estimation.

6.1 Introduction

This chapter is based on the work of Lei et al. [1], and discusses the potential for the use of large-scale classification algorithms for multimodal location estimation at the city-scale. It uses consumer-produced videos from the Internet, and provides the reader a tutorial for the use of matured algorithms that have been successfully applied to other tasks. The use of large-scale algorithms is of interest to researchers, due to increasing amounts of multimedia data being uploaded to the web. It has become increasingly attractive for researchers to build massive corpora out of videos, images, and audio files. A corpora could potentially contain thousands

H. Lei (✉) · J. Choi · G. Friedland
International Computer Science Institute, Berkeley, CA, USA
e-mail: howard.lei@csueastbay.edu

J. Choi
e-mail: jaeyoung@icsi.berkeley.edu

G. Friedland
e-mail: fractor@icsi.berkeley.edu

H. Lei
California State University, East Bay, CA, USA

J. Choi and G. Friedland (eds.), *Multimodal Location Estimation of Videos and Images*,
DOI 10.1007/978-3-319-09861-6_6

of hours of audio/video, where large-scale classification algorithms would be needed for their location estimation. Furthermore, consumer produced content on the Internet is completely uncontrolled, and therefore imposes a massive challenge for current signal processing and machine learning algorithms. These challenges present many opportunities for the development of new approaches, as well as the application of approaches from existing big-data tasks, for multimodal location estimation.

In approaching the location estimation task, the goal is to discover an approach that's suitable for the location estimation dataset, and apply the approach to the dataset in hopes of obtaining satisfactory results. While it's inherently desirable to invent new specialized approaches specifically tuned to the dataset, the challenge lies in the fact that the datasets can be sufficiently "wild"—especially if consumer-uploaded videos from the Internet are being used for the task. Those without any location estimation and/or classification background may have trouble deciding where to start.

For those wishing to "jump in" to the fun of performing location estimation, we present the application of well-established Gaussian Mixture Model (GMM)-based approaches to a city-scale location verification task. The audio and textual metadata of the videos are used in the task. While much of the information in the videos are discarded by using only the audio and metadata, our approaches demonstrate the cross-domain adaptability of well-established techniques in acoustic and language modeling. The audio-based approach uses supervised GMM training, with audio from consumer-produced Flickr videos, while the textual metadata-based approach uses unigram, bigram, and trigram language models. The task proves extremely difficult because of the "wild" nature of the videos, which contain many different background and foreground noises, and textual metadata. However, the approaches do achieve results that are significantly above-random. They also leave room for the reader to make improvements that are appropriate for future work in the field.

6.2 Approaching the City-Verification Task

The biggest challenge to any large-scale classification task is determining models that are suitable for the entities that need to be modeled. For the task of audio-based city verification, the challenge lies in determining models suitable for modeling the audio from different noises and sounds of each city.

Our use of the GMM-based approach stems from the fact that GMMs are easy to statistically train, and is suitable for modeling data under a wide range of distributions. We borrowed the GMM-UBM [2] approach that has been widely used in speaker verification until around 2005, to demonstrate how a simple approach can be applied to the difficult city-scale location estimation task.

The approach trains a single GMM per city in the dataset using acoustic feature vectors of the dataset's audio. The GMMs are used to represent the distribution of acoustic feature vectors of each city, and to classify features with unknown city labels. The acoustic features are the Mel Frequency Cepstral Coefficients (MFCCs) [3], which are widely used in speech processing applications. While the features have been designed for speech, they are based primarily on the audio's frequency-domain representation, which is useful for audio signal processing and analysis. The MFCC features are hence applicable to any audio-based task requiring the parameterization of the audio into acoustic feature vectors, with subsequent statistical modeling of the features.

6.2.1 MFCC Acoustic Feature Extraction

Figure 6.1 illustrates the process of obtaining the MFCC features. A detailed feature extraction process can be found in [3, 4].

The process involves taking 25 ms windows (i.e., Hamming windows) of the audio signal. For each windowed signal, a Fourier Transform is applied to obtain the frequency spectrum. A filterbank is then applied to the frequency spectrum, where each triangular filter is centered at frequency locations that are linear in the Mel-Frequency scale. Integration of the frequency spectrum magnitude is performed

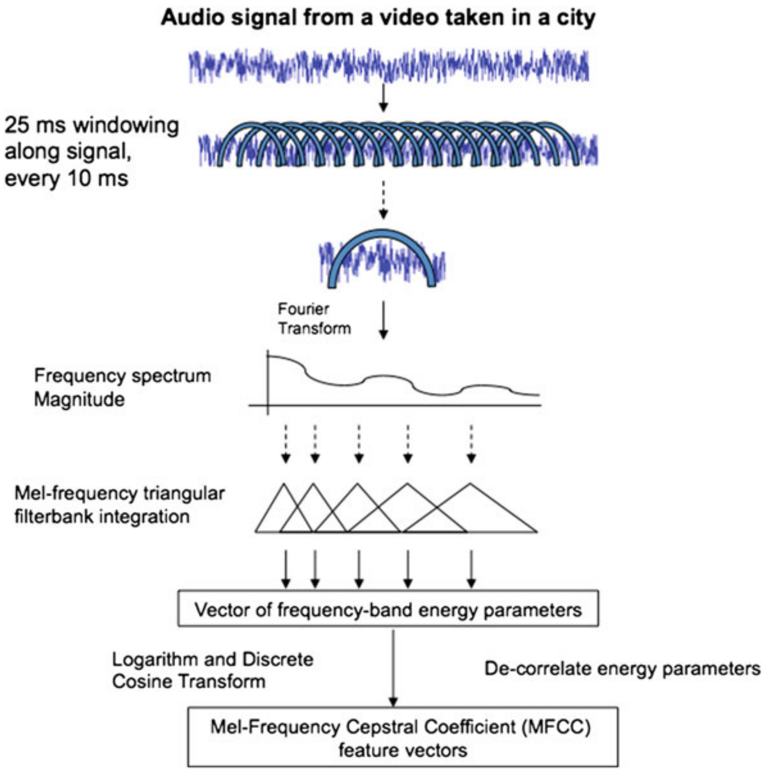

Audio signal from a video taken in a city
25 ms windowing along signal, every 10 ms
Fourier Transform
Frequency spectrum Magnitude
Mel-frequency triangular filterbank integration
Vector of frequency-band energy parameters
Logarithm and Discrete Cosine Transform
De-correlate energy parameters
Mel-Frequency Cepstral Coefficient (MFCC) feature vectors

Fig. 6.1 The Mel-Frequency Cepstral Coefficient (MFCC) extraction process

with each filter, and a logarithm is applied to each integration result to obtain a vector of log-energies. Finally, a Discrete Cosine Transform (DCT) is applied to the vector of log energies to obtain the final MFCC features for a given 25 ms window of the signal. The process is repeated for each 25 ms window, which are spaced 10 ms apart. The MFCC features of each audio file can be mean and variance-normalized for increased robustness, since each file may have different recording environments and noises. Mean and variance normalization of MFCC features by no means compensate for all variability from the noises, but is an easy enough procedure to implement. The related task of speaker verification typically uses MFCC mean subtraction (also known as Cepstral Mean Subtraction) [2], and Gaussian Feature Warping [5].

The MFCC features used in the experiments described in this chapter are the 0th through 19th (C0-C19) parameters after application of the DCT. C0 correlates with the energy of the frequency spectrum. Delta and acceleration coefficients are appended to the 20-dimensional MFCC parameters, for a total of 60 parameters per feature vector.

6.2.2 Gaussian Mixture Modeling

Once the MFCCs are obtained, the next step involves obtaining GMMs for each city in the training dataset, using the MFCCs of each city's audio. This is a two-step process, first involving the training of a city-independent model, or Universal Background Model (UBM) [2]. The UBM is trained using MFCCs from a large set of cities in the training data. Once the UBM is trained, the city-dependent GMM models are adapted (via the maximum a-posteriori method) from the UBM using city-specific MFCC feature vectors [2]. Figures 6.2 and 6.3 illustrates this process.

Once a GMM is trained for each city, a similarity score can be generated between MFCC feature vectors from a city in the test set, to one of the cities for which a GMM has been trained. Figure 6.4 illustrates this process.

Specifically, a probabilistic log-likelihood ratio is computed for the MFCC vectors using a city-dependent GMM (numerator of likelihood ratio) and the UBM (denominator of likelihood ratio). The UBM is used for score normalization purposes. This is the standard approach used in GMM-UBM speaker verification [2].

6.2.3 GMM-SVM Approach

A alternative, closely related approach to the GMM-UBM is the GMM-SVM approach, which was first used by Campbell et al. for speaker verification [6]. In this approach, a city-dependent GMM is trained for each city (regardless if the city is in the training or test dataset). Once the GMM is trained, the mean parameters of the GMM are extracted into a city-dependent "supervector." A Support Vector Machine (SVM) model is trained for each city in the training data using that city's supervector, and supervectors from a large collection of other cities. The city-dependent supervector serves as the positive SVM training example, while the rest serve as

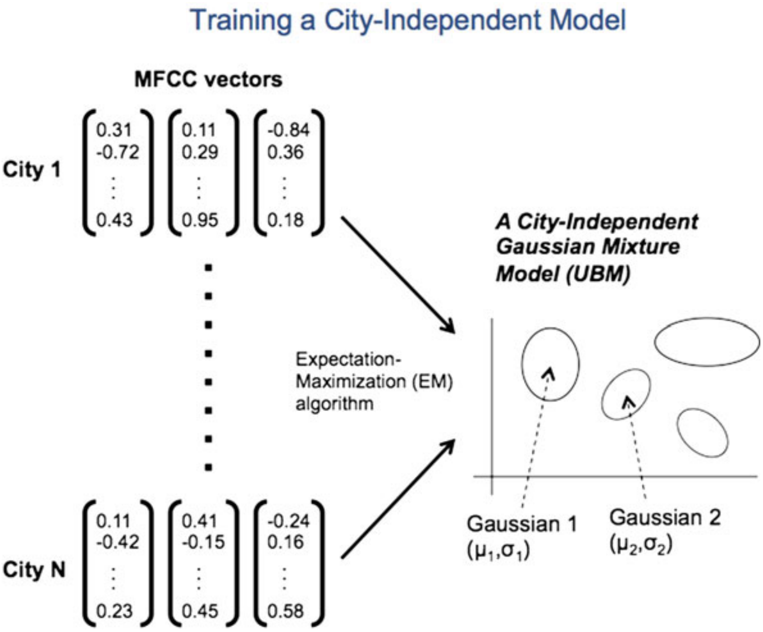

Training a City-Independent Model
MFCC vectors
City 1
0.31
-0.72
0.43
0.11
0.29
0.95
-0.84
0.36
0.18
City N
0.11
-0.42
0.23
0.41
-0.15
0.45
-0.24
0.16
0.58
Expectation-Maximization (EM) algorithm
A City-Independent Gaussian Mixture Model (UBM)
Gaussian 1 (μ_1,σ_1)
Gaussian 2 (μ_2,σ_2)

Fig. 6.2 City-independent model (i.e., UBM) training using MFCCs from N cities

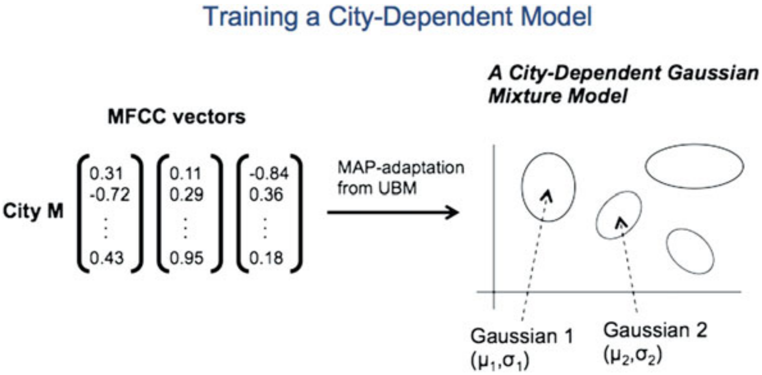

Training a City-Dependent Model
MFCC vectors
City M
0.31
-0.72
0.43
0.11
0.29
0.95
-0.84
0.36
0.18
MAP-adaptation from UBM
A City-Dependent Gaussian Mixture Model
Gaussian 1 (μ_1,σ_1)
Gaussian 2 (μ_2,σ_2)

Fig. 6.3 City-dependent model training using MFCCs from a single city, along with the UBM

negative training examples. Note that multiple positive examples can be used for a given city if sufficient data is available. The classification and scoring of a city in the test dataset is achieved by computing an SVM classification (or regression) score against each SVM model from a city in the training dataset. Figures 6.5 and 6.6 illustrate supervector extraction and SVM model training.

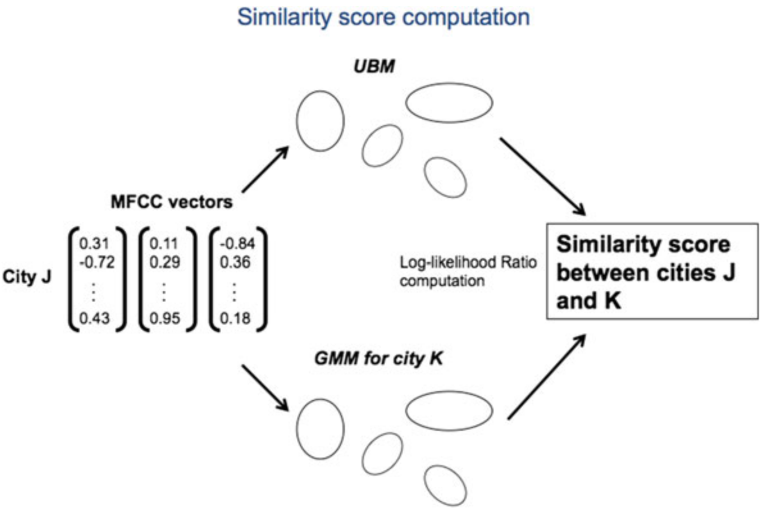

Similarity score computation
UBM
MFCC vectors
City J
0.31 -0.72 ⋮ 0.43
0.11 0.29 ⋮ 0.95
-0.84 0.36 ⋮ 0.18
Log-likelihood Ratio computation
Similarity score between cities J and K
GMM for city K

Fig. 6.4 Computing similarity scores between cities using city-dependent MFCC feature vectors and GMM models

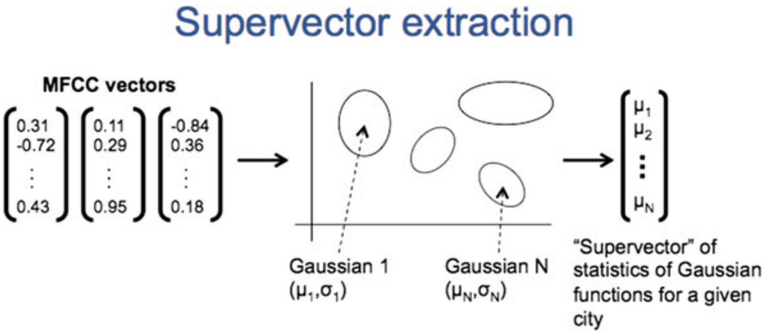
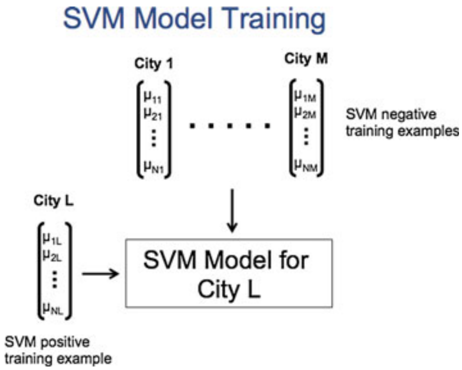

Supervector extraction
MFCC vectors
0.31 -0.72 ⋮ 0.43
0.11 0.29 ⋮ 0.95
-0.84 0.36 ⋮ 0.18
Gaussian 1 (μ_1,σ_1)
Gaussian N (μ_N,σ_N)
μ_1 μ_2 ⋮ μ_N
"Supervector" of statistics of Gaussian functions for a given city

Fig. 6.5 City-dependent supervector extraction from MFCC features and GMM models

Fig. 6.6 Support Vector Machine model training from positive and negative training examples


SVM Model Training
City 1
μ_{11} μ_{21} ⋮ μ_{N1}
City M
μ_{1M} μ_{2M} ⋮ μ_{NM}
SVM negative training examples
City L
μ_{1L} μ_{2L} ⋮ μ_{NL}
SVM Model for City L
SVM positive training example

We note that for both the GMM-UBM and GMM-SVM approaches, the GMM models are trained using the open-source ALIZE toolkit [7], and the MFCC features are obtained via HTK [8]. SVM models are implemented using the SVM$^{\text{light}}$ toolkit [9], with wrapper scripts provided by SRI International.

6.2.4 Language Modeling

The language-modeling based approach involves training back-off language models, implemented using the SRILM toolkit [10]. Uni-, bi-, and trigram word language models are trained for each city using the metadata (keywords and descriptions) of all videos for the city in the training data. The probabilistic likelihoods of the metadata of test videos are computed using each city's language model to determine similarity scores of each test video versus each city.

6.2.5 Performance Evaluation

For each of the above approaches, a similarity score is generated for a pair of cities—one in the training set, one in the test set. A scoring threshold is establish to separate the scores with matching-cities (matching scores), from scores with nonmatching cities (nonmatching scores). The Equal Error Rate (EER) occurs at a scoring threshold where the percentage of matching scores classified as nonmatching scores (misses) equals the percentage of nonmatching scores classified as matching scores (false alarms). The EER corresponds to either percentage when the percentages are equal. We note that in practice, it's rarely the case that the percentage of misses and false alarms can be made exactly equal. The EER can then be computed from an average of the percentages.

6.3 Related Work

There has been a considerable amount of work done on multimodal location estimation. Articles [11, 12] on location estimation of images have indicated that the task may be approached as a retrieval problem on a location-tagged image database. In the work of [13], the goal was to estimate a rough location of an image as opposed to its exact GPS location. For example, images of certain landscapes could occur only in certain places on Earth. Jacobs' system [14] relied on matching images with satellite data. The above work (along with other work) relied on the detection or matching of a set of explicit visual features (e.g., landmarks or sun altitudes) rather than performing an implicit matching of unknown cues as performed in this article. Works have also been performed in the 2011 MediaEval Placing task evaluation [15], where participants must obtain the geo-location of Flickr videos based on textual metadata, video, and audio. While the accuracies achieved there were better than city-scale, audio usage had been virtually ignored in all systems. Lei et al. in [1] performed work on city-scale location estimation, which this chapter is based on. More recently, Kelm et al. [16] has performed work on novel fusion techniques integrating multiple modalities of knowledge for location estimation, while Choi et al. [17] considered human versus machine performance for multimodal location estimation, using Amazon's Mechanical Turk [18].

6.4 Dataset

The audio tracks for the experiments are extracted from videos distributed as a training dataset for the Placing Task of MediaEval 2010 [19], a multimedia benchmark evaluation. The dataset consists of 5,125 Creative Commons licensed Flickr videos uploaded by Flickr users. Manual inspection of the dataset led us to conclude that most visual/audio contents of the videos lack reasonable information for estimation of their origin. For example, some videos have been recorded indoors or in private spaces such as the backyard of a house, which makes the Placing Task nearly impossible if only the visual and audio contents were examined. This indicates that the videos have not been pre-filtered or pre-selected in any way to make the dataset more relevant to the city-verification task.

From an examination of 84 videos from the dataset, we found that most of the videos' audio tracks are quite "wild." Only 2.4 % of them have been recorded in a controlled environment such as inside a studio at a radio station. The other 97.6 % are home-video style with ambient noise. 65.5 % of the videos have heavy ambient noises; 14.3 % of the videos contain music. About 50 % of the videos do not contain human speech, and even for the ones that contain human speech, almost half are from multiple subjects and crowds in the background speaking to one another. 5 % of the videos are edited to contain changed scenes, fast-forwarding, muted audio, or inserted background music. While there are some audio features that may hint at the city-scale location of the video—features such as the spoken language in cases where human speech exist, type and genre of music, etc.—such factors are not prevalent, and are often mixed with heavy amounts of background noise and music. The maximum length of Flickr videos is limited to 90 s. About 70 % of videos are less than 50 s. The relatively short lengths of each audio track should also noted as can be seen in Fig. 6.7.

A listening of the audio also gave us insight into the extent to which city-scale geo-locations of videos are correlated with their audio features. We rarely found city-specific sounds that would enable a human listener to accurately perform city verification of the videos based on their audio. In addition, while location and/or language-specific metadata tags could sometimes be found in the Flickr videos to aid in the city-verification task, the useful metadata was sparse. Hence, this work demonstrate the power of machine learning algorithms in performing a task that would likely be difficult for humans. Because the audio and metadata modalities are likely

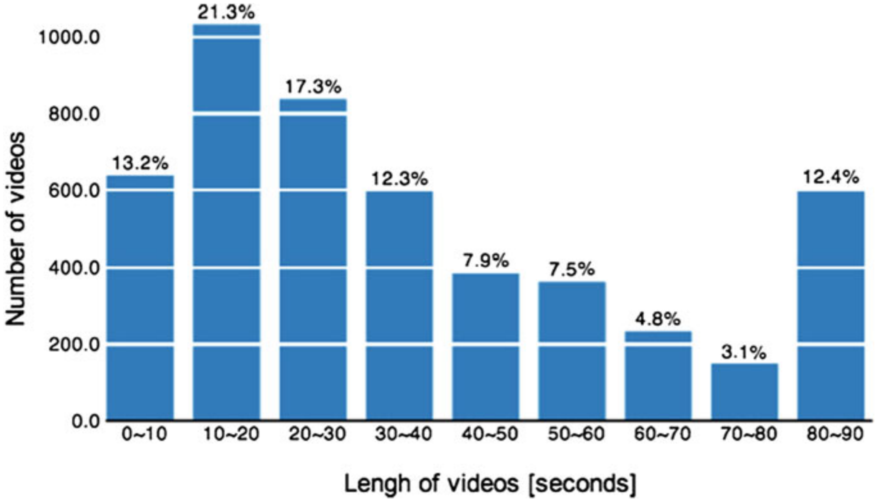


Fig. 6.7 A histogram visualizing the duration of the videos of the MediaEval 2010 dataset

complementary to other modalities (i.e., video), achieving success using only the audio and metadata modalities would suggest the potential for further improvements when additional modalities are incorporated.

For the task of city verification, a video is considered to be located within a city if its geo-coordinates are within 5 km of the city center. The following cities are considered for verification because of the predominance of videos belonging to these cities: *Bangkok*, *Barcelona*, *Beijing*, *Berlin*, *Chicago*, *Houston*, *London*, *LosAngeles*, *Moscow*, *NewYork*, *Paris*, *Praha*, *Rio*, *Rome*, *San Francisco*, *Seoul*, *Sydney*, and *Tokyo*.

6.5 Experiments and Results

Experiments are performed using the GMM-UBM and GMM-SVM approaches, along with unigram, bigram, and trigram language models to obtain city verification results. For audio-based experiments, the entire duration of each audio track is used, and MFCC features are mean- and variance-normalized prior to GMM and UBM training. Different combinations of data are used for training and testing. The main audio experiment uses a 117-video development set, a 1,080-video training set, and a 285-video test set with no common users (i.e., common Flickr user accounts hosting the videos) in the training set. The city-specific distribution of videos in the 1,080-video training set is such that 43 % of videos are from *San Francisco*, 17 % are from *London*, and each remaining city has 7 % or less of the total number of videos. The distribution in the 285-video test set is such that 25 % of videos are from *San Francisco*, 22 % are from *London*, and each remaining city has 7 % or less of the

Table 6.1 Results for the GMM-UBM and GMM-SVM audio-based approaches for city verification

Approach	EER (%)
GMM-UBM	32.3
GMM-SVM	32.3

total number of videos. The 285-video test set allows us to generate 5,130 similarity scores (with 285 matching scores). Table 6.1 shows the audio-based results.

According to the results in Table 6.1, both the GMM-UBM and GMM-SVM approaches give identical EER performance (32.2 %). While the result is not impressive by any means, it is significantly above random (50 % EER). Metadata-based approaches are also examined, primarily in combination with the audio-based approach. To combine the audio and metadata-based approaches at the score level, a Multi Layer Perceptron (MLP) with 2 hidden nodes and 1 hidden layer, implemented using Lnknet [20], is used. The EER results represent averaged EER values over 100 splits amongst the training cities and test videos, where each split contained training and test subsplits. For each of the 100 splits, MLP weights are trained using the training subsplit, and applied to the test subsplit.

The metadata-based experiments use a newly defined training dataset of 542 videos, and a test dataset of 541 videos (i.e., different training and test splits of the same overall dataset). A small fraction of the videos across the new training and test sets share common users. The EER averaging is done for all results using this training and test data combination. Table 6.2 shows the metadata-based results, along with its combination with the GMM-UBM approach. The Unigram LM, Bigram LM, and Trigram LM approaches use unigram, bigram, and trigram Language Models (LM). The GMM-UBM result is also shown for purposes of comparison.

The metadata experiments give surprisingly similar results compared to the audio experiments. The Unigram LM approach gives a 23.9 % EER, which is a 5.5 % relative EER improvement over the GMM-UBM approach (25.3 % EER). Combining the Unigram LM and GMM-UBM approaches results in a 21.8 % EER, an 8.8 % relative EER improvement over the Unigram LM standalone. Note that the Unigram LM approach (23.9 % EER) gives lower EER than the Bigram LM (29.4 % EER) and

Table 6.2 Results of metadata-based approaches for city verification, using a different set of training and test videos

Approach	EER (%)
Unigram LM	23.9
Bigram LM	29.4
Trigram LM	30.9
GMM-UBM	25.3
GMM-UBM + Unigram LM	*21.8*

Combining the Unigram LM approach with the GMM-UBM approach gives a minimum EER of 21.8 %

Trigram LM (30.9 % EER) approaches. This is likely because most metadata keywords have no lexical connections with other keywords, and the video descriptions are short. Overall, there is an average of 6.3 usable lexical tags per video in our dataset, such that higher-order language models are sparse and would likely result in overtraining. Given the sparsity of metadata information, it is surprising that the metadata-based approaches are comparable in terms of their performances to the audio-based approaches.

6.6 Discussion and Analysis

The audio-based results are interesting considering that after listening to a random sample of the videos across different cities, we did not get the sense that there were any clear, distinctive audio features for each city. For instance, there are no sounds that would clearly verify audio as belonging to the city of San Francisco. However, a close listening to the test videos with the high true trial scores indicates that speech may play a significant role in city verification. Test videos with the top three true trial scores are all from Rio and contain monologue speech from a family excursion, where the words "Rio De Janeiro" are spoken. One high-scoring test video from London contains speech with British accents, while one from Paris contains city-specific ambulance noise. Many high-scoring videos hence appear to contain some kind of city-specific audio feature (i.e., speech or language/dialect marker, or other city-specific noise). However, there are also high-scoring test videos without city-specific audio features—a video from Tokyo contains audio of a train arriving, one from San Francisco contains bagpipe music, and one from Paris contains loud engine noise.

An analysis of high-scoring test videos for the unigram language model experiments indicates that many high-scoring test videos contain location- or language-specific metadata keywords. Such videos include one from Barcelona, with Spanish metadata *bicicletas*, *policia*, *brigada*, along with the location-specific word *barcelona*. A video from London contains the metadata *london*, one from San Francisco contains *sanfrancisco*, and one from Beijing contains *asia*, *china*, *tibetan*, and *buddha*. However, some high-scoring test videos do not contain any location or language-specific metadata.

Because high-scoring test videos from the audio-based approaches differ from those for the metadata approaches, the two approaches are complementary, resulting in an EER improvement in their combination. Furthermore, potential improvements in city verification can be obtained by combining other modalities, such as video and keyframe image data, as well as making better use of the audio and metadata.

6.7 Conclusion and Future Work

This chapter illustrates applications of simple large-scale classification algorithms for the task of city-verification. The results are above random, but not admirable by any means. Nevertheless, the experiments described serve as a teaching tool, demonstrating the applicability of cross-domain algorithms and techniques for handling multimodal location estimation. Anyone new to location estimation can begin with such algorithms as a starting point for experiments. The experiments described also illustrate many challenges, and may suggest to the reader potential areas of improvement in future projects.

For instance, an investigation of how the classification approaches can better handle the audio and metadata modalities, and incorporating other modalities to enhance performance, can be performed. Other investigations on how to better handle the conglomeration of factors in the audio, such as differences in music, language, loudness, can also be performed. Overall, the experiments described do not serve to demonstrate large-scale multimodal location estimation being a solved problem, but as a task that should inspire future research.

Acknowledgments The experiments described in this work were supported by NGA NURI grant number HM11582-10-1-0008, NSF EAGER grant IIS-1138599, and NSF Award CNS-1065240. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors.

References

1. H. Lei, J. Choi, G. Friedland, Multimodal city-verification on Flickr videos using acoustic and textual features, in *Proceedings of ICASSP*, Kyoto, Japan, (2012)
2. D.A. Reynolds, T.F. Quatieri, R. Dunn, Speaker Verification using Adapted Gaussian Mixture Models. Digit. Signal Process. **10**, 19–41 (2000)
3. S. Davis, P. Mermelstein, Comparison of Parametric Representations of Monosyllabic Word Recognition in Continuously Spoken Sentences, in *Proceedings of ICASSP* (1980)
4. D.A. Reynolds, R.C. Rose, Robust text-independent speaker identification using Gaussian mixture speaker models, in *IEEE Transactions on Speech and Audio Process*, vol. 3, pp. 72–83 (1995)
5. J. Pelecanos, S. Sridharan, Feature Warping for Robust Speaker Verification, in *Speaker Odyssey: The Speaker Recognition Workshop*, Crete, Greece, (2001)
6. W. Campbell, D. Sturim, D. Reynolds, Support Vector Machines using GMM Supervectors for Speaker Verification. IEEE Signal Process. Lett. **13**, 308–311 (2006)
7. J.F. Bonastre, F. Wils, S. Meignier, ALIZE, a free Toolkit for Speaker Recognition, in *ICASSP*, vol. 1, pp. 737–740 (2005)
8. HMM Toolkit (HTK), http://htk.eng.cam.ac.uk
9. T. Joachims, Making Large Scale SVM Learning Practical, in *Advances in Kernel Methods—Support Vector Learning*, ed. by B. Schoelkopf, C. Burges, A. Smola (MIT-press, Cambridge, 1999)
10. A. Stolcke, SRILM—An Extensible Language Modeling Toolkit in *Proceedings of the International Conference Spoken Language Processing*, Denver, Colorado, (2002)

11. G. Schindler, M. Brown, R. Szeliski, City-scale Location Recognition, in *IEEE Conference on Computer Vision and Pattern Recognition* (2007)
12. W. Zhang, J. Kosecka, Image based Localization in Urban Environments in *3rd International Symposium on 3D Data Processing, Visualization, and Transmission* (2006)
13. J. Hays, A. Efros, IM2GPS: Estimating Geographic Information from a Single Image, in *IEEE Conference on Computer Vision and Pattern Recognition* (2008)
14. N. Jacobs, S. Satkin, N. Roman, R. Speyer, R. Pless, Geolocation Static Cameras, in *IEEE International Conference on Computer Vision* (2007)
15. A. Rae, V. Murdock, P. Serdyukov, P. Kelm, Working Notes for the Placing Task at MediaEval 2011, in *Proceedings of MediaEval* (2011)
16. P. Kelm, S. Schmiedeke, J. Choi, G. Friedland, V. Ekambaram, K. Ramchandran, T. Sikora, A Novel Fusion Method for Integrating Multiple Modalities and Knowledge for Multimodal Location Estimation, in *GeoMM'13*, Barcelona, Spain, (2013)
17. J. Choi, H. Lei, V. Ekambaram, P. Kelm, L. Gottlieb, T. Sikora, K. Ramchandran, G. Friedland, Human vs Machine: Establishing a Human Baseline for Multimodal Location Estimation, in *ACM SIGMM International Conference on Multimedia* (2013)
18. P. Ipeirotis, Analyzing the Amazon Mechanical Turk Marketplace, in *ACM XRDS (Crossroads)*, vol. 17, No. 2, (2010)
19. MediaEval Web Site, http://www.multimediaeval.org
20. R.P. Lippmann, L.C. Kukolich, E. Singer, LNKnet: Neural Network, Machine Learning, and Statistical Software for Pattern Classification. Linc. Lab. J. **6**, 249–268 (1993)

Chapter 7
Collaborative Multimodal Location Estimation of Consumer Media

Venkatesan Ekambaram, Kannan Ramchandran, Jaeyoung Choi and Gerald Friedland

7.1 Introduction

With the emergence of Web 2.0 and with GPS devices becoming ubiquitous and pervasive in our daily life, location-based services are rapidly gaining traction in the online world. The main driving force behind these services is the enabling of a very personalized experience. Social-media websites such as Flickr, YouTube, Twitter, etc., allow queries for results originating at a certain location. Likewise, the belief is that retro-fitting archives with location information will be attractive to many businesses, and will enable newer applications. The task of estimating the geo-coordinates of a media-recording goes by different names such as "geo-tagging", "location estimation" or "placing". Geo-tagging multimedia content has various applications. For example, geo-location services can be provided for media captured in environments without GPS, such as photos taken indoors on mobile phones. Vacation videos and photos can be better organized and presented to the user if they have geo-location information. With the explosive growth of available multimedia content on the Internet (200 million photos are uploaded to Facebook daily), there is a dire need for efficient organization and retrieval of multimedia content, which can be enabled by geo-tagging. Geo-location information further helps develop a better semantic

V. Ekambaram (✉) · K. Ramchandran
University of California, Berkeley, CA, USA
e-mail: venkyne@eecs.berkeley.edu

K. Ramchandran
e-mail: kannanr@eecs.berkeley.edu

J. Choi · G. Friedland
International Computer Science Institute, Berkeley, CA, USA
e-mail: jaeyoung@icsi.berkeley.edu

G. Friedland
e-mail: fractor@icsi.berkeley.edu

J. Choi and G. Friedland (eds.), *Multimodal Location Estimation of Videos and Images*,
DOI 10.1007/978-3-319-09861-6_7

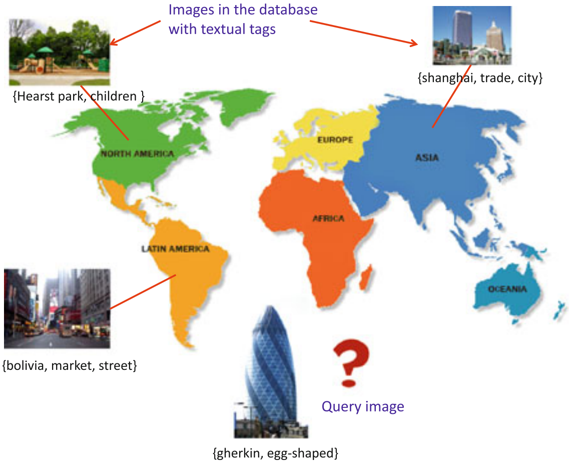


Fig. 7.1 *Geo-tagging:* given a database of training images/videos with their geo-coordinates and textual data, estimate the geo-location of a query video given its textual metadata, visual and audio features

understanding of multimedia content. These are some of the main motivations of the MediaEval Placing task [1, 2].

Even though many of the high-end cameras and video recorders are retrofitted with GPS chips, it has been estimated that only about 5 % of the existing multimedia content on the Internet is actually geo-tagged [3]. Most of the consumer-produced media content are obtained using low-end cameras that do not have GPS chips. Further, privacy concerns have motivated users to disable automatic geo-stamping of photos taken on their phones. However, users usually tag their uploaded videos with textual data that can have some geo-location information. Under this scenario, we ask the question, "Given a set of videos and their associated textual tags, how do we determine their geo-locations?" To aid this process, we are provided with a training database of videos with their geo-location estimates (Fig. 7.1).

The main contribution of our work is the development of an estimation theoretic collaborative framework based on graphical models for the purpose of determining the geo-coordinates of query videos. We pose the problem of geo-tagging as one of inference over an underlying graph constructed using the query video database and *jointly* estimate the geo-locations of all the query videos. The main advantage of this approach is that, even when the training data is sparse, the query videos whose geo-tags are estimated act as "virtual" training data for incoming query data, thereby effectively bootstrapping the geo-tagging process. Our results show that we obtain up to 10 % performance improvements over existing state-of-the-art algorithms. The generic nature of the framework facilitates fusing multimodal data sources such as textual tags, audio and visual features. Thus, any other additional information obtained can be easily incorporated into the framework.

Our paper is organized as follows. Section 7.2 provides a brief overview of the existing work in this field and the novelty of our work as compared to this literature. Section 7.3 describes in detail the problem of data sparsity and a simple example illustrating the intuition behind our algorithm. Section 7.4 describes in detail our graphical model framework for geo-tagging. Section 7.5 describes the datasets we use and our simulation results. Section 7.6 concludes with a summary of the paper and possible research directions.

7.2 Literature Review

The problem of geo-tagging is closely related to the problem of user location estimation using images captured from his/her camera. Some of the earlier work in this domain [4, 5] posed the problem as one of image retrieval from a database of self-produced, location-tagged images using specialized visual features. Similar approaches based on visual descriptors such as [6] have been used to geo-tag images. Larson et al. [7] provide a comprehensive overview of the ongoing research in this field. Most users tag the media they upload to sites like Flickr with text that can include information regarding the location or activity captured by the images. Rattenbury et al. [8] and Serdyukov et al. [9] estimate the posterior distribution of the geo-locations given the tags or vice-versa from the training database and use this to estimate the geo-location of a query video.

The 2010, 2011 and 2012 MediaEval Placing tasks [1] provided a common platform to evaluate different geo-tagging approaches on a corpus of randomly selected consumer-produced videos. One of the top performing systems proposed by Van Laere et al. [10] used a combination of language models and similarity search to geo-tag the videos purely based on their textual tags. Several other proposed approaches [11–14] relied on both textual and visual features. However, none of these systems utilized audio features. A subset of the authors of this paper proposed a hierarchical system [15] that uses the spatial variance of the tags' geo-location distribution to find an initial estimate of the query image location, which is used as an anchor point for a visual nearest neighbor search in the next stage. Their enhanced system [16] incorporates audio features as well, motivated by their previous work on location estimation of ambulance videos from different cities [3] using audio features.

All of the existing approaches described above have the common feature of processing each query video independently and estimating its geo-location based on textual, visual, and audio features using a geo-tagged training database. Clearly, the performance of these systems largely depends on the size and quality of the training database. However, data sparsity is one of the major issues that can adversely affect the performance of these systems. Our approach differs from the existing work in the literature in the aspect that we *jointly* estimate the geo-locations of all the input query images. Each query image added to the database enhances the quality of the database by acting as "virtual" training data and thereby boosts the performance of the algorithm. We elaborate on the data sparsity issue in the next section and provide an intuition for our algorithm.

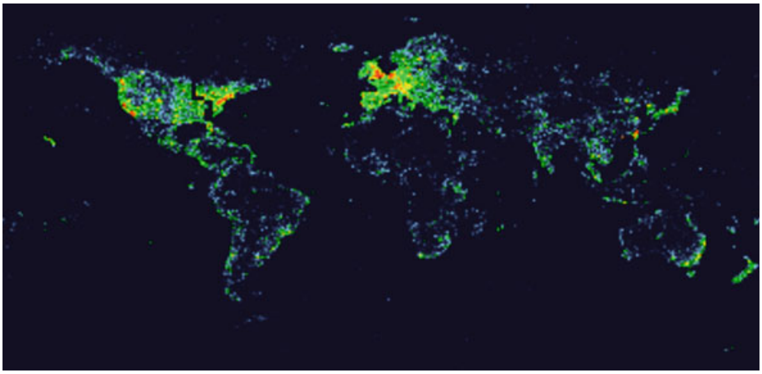
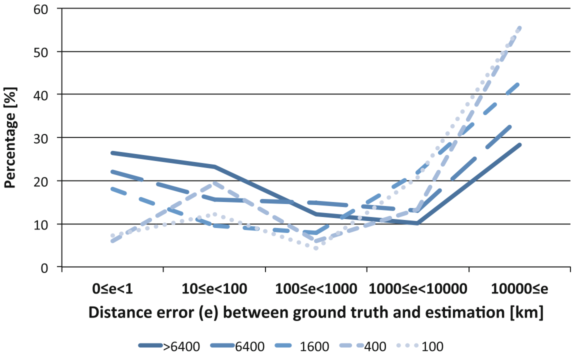


Fig. 7.2 Distribution of videos and images of the MediaEval 2010 Placing Task training set. Randomly sampling videos from Flickr results in a non-uniform geographical prior

Fig. 7.3 Comparison of the performance of a data-driven algorithm [15] on grids with different training data density. Query video from a denser area has higher chance of being estimated with lower error in distance



7.3 Data Sparsity

Traditional approaches such as [12, 15] use training sets that are several orders of magnitude larger than the test set. These approaches suffer from the drawback that their accuracies are significantly affected when the training data is sparse. There are two reasons for sparsity in training data. First, it is estimated that only 5 % of Internet videos are actually geo-tagged [3], and hence the training set is typically much smaller than the test set, contrary to what is assumed in the literature. Second, the training database is largely skewed toward certain geographical regions (see Fig. 7.2).

For the MediaEval dataset [1], we analyzed the performance of a data-driven algorithm from [15] in different regions with varying data densities. The world map was divided into 64,800 grids of one latitude by one longitude each and the number of training data in each grid was counted. Figure 7.3 shows the performance of the algorithm for different data densities. The different curves are for different values of the training data density, i.e., we look at the performance in grids with varying quantities of videos: over 6,400, 6,400, 1,600, 400, and 100. The x-axis corresponds to the different error ranges in km and the y-axis to the percentage of geo-tagged videos in these error ranges. Grids with a denser population of training data perform significantly better than those with lesser training data. Thus, estimation models must be developed to handle data sparsity.

One can exploit the fact that while the training data may be small, test data may be very large. Existing algorithms do not take this into account. Thus the question of interest is, "Can we intelligently process the test/query videos in such a way that each additional query video not only is placed but also improves the quality of the existing database?" To take a simple example demonstrating our idea, let us suppose that we have two query videos, Q1 and Q2, with associated textual tags {berkeley, sathergate, california} and {sathergate, california} respectively. Assume that the training set only contains the tags, {berkeley, california}. If Q1 and Q2 were to be processed independently, then Q1's location estimation would be good whereas the location ambiguity of Q2 would be much larger. However, if we jointly process Q1 and Q2, then given that Q1 and Q2 have the tag "sathergate" in common, it is very likely that their locations are also very close by and hence an intelligent algorithm would estimate their locations to be the same, which would improve the location accuracy of Q2. The proposed graphical model framework in the next section applies this principle to a database of query videos and appropriately weighs the edges based on the common tags between the videos.

7.4 Graphical Models for Geo-tagging

Graphical models provide an efficient representation of dependencies amongst different random variables and have been extensively studied in the statistical learning theory community [17]. The random variables in our setup are the geo-locations of the query videos that need to be estimated. We treat the textual tags and visual and audio features as observed random variables that are probabilistically related to the geo-location of that video. The goal is to obtain the best estimate of the unobserved random variables (locations of the query videos) given all the observed variables. We use graphical models to characterize the dependencies amongst the different random variables and use efficient message-passing algorithms to obtain the desired estimates. We give a brief introduction to graphical models and apply the framework to our setup.

An undirected graphical model or a Markov Random Field (MRF) $G(V, E)$ consists of a vertex set V and an edge set E. The vertices (nodes) of the graph represent random variables $\{x_v\}_{v \in V}$ and the edges capture the conditional independencies amongst the random variables through graph separation [17]. The joint probability distribution for a N-node pairwise MRF can be written as follows [17],

$$p(x_1, \ldots, x_N) = \prod_{i \in V} \psi(x_i) \prod_{(i,j) \in E} \psi(x_i, x_j). \tag{7.1}$$

$\psi(.)$'s are known as potential functions that depend on the probability distribution of the random variables. A typical problem of inference over a graphical model involves finding the marginal distribution of the random variables $p(x_i)$. Finding the exact marginals is in general an NP-hard problem [17] and approximation algorithms such as the sum-product algorithm are used in practice. In the sum-product algorithm, messages are passed between nodes that take the following form:

$$m_{j\to i}(x_i) \propto \int\limits_{x_j} \psi(x_i, x_j)\psi(x_j) \prod_{k\in N(j)/i} m_{k\to j}(x_j)dx_j, \tag{7.2}$$

where $m_{j\to i}(x_i)$ is the message passed from node j to node i and $N(j)$ is the set of neighbors of j. The messages are iteratively passed until convergence and the final estimate of $p(x_i)$ is obtained as follows,

$$\hat{p}(x_i) \propto \psi(x_i) \prod_{j\in N(i)} m_{j\to i}(x_i). \tag{7.3}$$

This algorithm is seen to work well in practice for many applications.

In order to obtain a graphical model representation for our problem setup, we need to model the joint distribution of the query video locations given the observed data. Since it is hard to obtain the exact probability distribution, we will use a simplistic conditional dependency model for the random variables as described below. Each node in our graphical model corresponds to a query video and the associated random variable is the geo-location of that query video (i.e., latitude and longitude). Intuitively, if two images are nearby, then they should be connected by an edge since their locations are highly correlated. The problem is that we do not know the geo-locations a priori. However, given that textual tags are strongly correlated to the geo-locations, a common textual tag between two images is a good indication of the proximity of geo-locations. Hence, we will build the graphical model by having an edge between two nodes if and only if the two query videos have at least one common textual tag. Note that this textual tag need not appear in the training set.

Figure 7.4 shows an example of one such graph. The edge potential functions appropriately weigh the edge based on the common textual tag. The potential

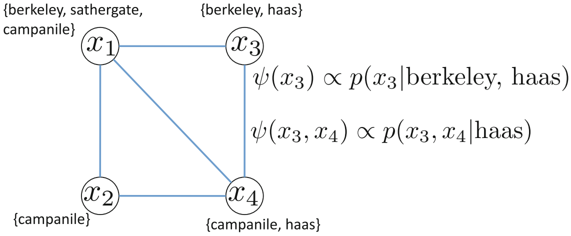

{berkeley, sathergate, campanile}
x_1
{berkeley, haas}
x_3
$\psi(x_3) \propto p(x_3|\text{berkeley, haas})$
$\psi(x_3, x_4) \propto p(x_3, x_4|\text{haas})$
x_2
{campanile}
x_4
{campanile, haas}

Fig. 7.4 An example graphical model for geo-tagging. Each node corresponds to an image/video and an edge is drawn between two nodes if they share a common textual tag. Potential functions are associated with each edge that reflect the strength of the connection

functions associated with the edge that reflects the strength of the edge, are a function of the common tag. For example, if the common tag is highly location specific (e.g. "sathergate"), then one would expect the potential function to have more weight than compared to a tag that is more generic (e.g. "school").The model could be further improved using the audio and visual features as well.

Let x_i be the geo-location of the ith video and $\{t_i^k\}_{k=1}^{n_i}$ be the set of n_i tags associated with this video. Based on our model, under a pairwise MRF assumption, the joint probability distribution factorizes as follows:

$$p(x_1, \ldots, x_N | \{t_1^k\}, \ldots, \{t_N^k\}) \propto \prod_{i \in V} \psi(x_i | \{t_i^k\}) \prod_{(i,j) \in E} \psi(x_i, x_j | \{t_i^k\}, \{t_j^k\}). \quad (7.4)$$

Given the potential functions, one could use the sum-product iterates (7.2), to estimate the marginal distribution, $p(x_i | \{t_1^k\}, \ldots, \{t_N^k\})$.

We now need to model the node and edge potential functions. The literature provides numerous techniques for modeling potential functions and adaptively learning them from the given data [18]. We use the following simple model for the potential functions. Given the training data, we fit a Gaussian Mixture Model (GMM) for the distribution of the location given a particular tag t, i.e., $p(x|t)$. The intuition is that tags usually correspond to one or more specific locations and the distribution is multi-modal (e.g., the tag "washington" can refer to two geographic places). To estimate the parameters of the GMM, we use an algorithm based on Expectation Maximization [19] that adaptively chooses the number of components for different tags using a likelihood criterion. Assuming conditional independence for different tags, we take the node potential as follows, $\psi(x_i) \propto \prod_{k=1}^{n_i} p(x_i | t_i^k)$. For the potential functions, $\psi(x_i, x_j | \{t_i^k\}, \{t_j^k\})$, we use a very simple model. Intuitively, if the common tag between two query videos i and j occurs too frequently either in the test set or the training set, that tag is most likely a common word like "video" or "photo" which does not really encode any information about the geographic closeness of the two videos. In this case, we assume that the edge potential is zero [drop edge (i, j)] whenever the number of occurrences of the tag is above a threshold. When the occurrence of the common tag is less frequent, then it is most likely that the geographic locations are very close to each other and we model the potential function as an indicator function,

$$\psi(x_i, x_j | \{t_i^k\}, \{t_j^k\}) = \begin{cases} 1 \text{ if } x_i = x_j, \\ 0 \text{ otherwise.} \end{cases} \quad (7.5)$$

This model is a hard-threshold model and we can clearly use a soft-version wherein the weights on the edges for the potential functions are appropriately chosen.

Further, we propose the following simplification, which leads to analytically tractable expressions for the potential functions and message updates. Given that for many of the tags, the GMM will have one strong mixture component, the distribution $\psi(x_i)$, can be approximated by a Gaussian distribution with the mean $(\tilde{\mu}_i)$ and variance $(\tilde{\sigma}_i^2)$ given by,

$$(\tilde{\mu}_i, \tilde{\sigma}_i^2) = \left(\frac{\sum_{k=1}^{n_i} \frac{1}{(\sigma_i^k)^2} \mu_i^k}{\sum_{k=1}^{n_i} \frac{1}{(\sigma_i^k)^2}}, \frac{1}{\sum_{k=1}^{n_i} \frac{1}{(\sigma_i^k)^2}} \right), \tag{7.6}$$

where μ_i^k and $(\sigma_i^k)^2$ are the mean and variance of the mixture component with the largest weight of the distribution $p(x_i|t_i^k)$. Under this assumption, the iterations of the sum-product algorithm take on the following simplistic form. Node i at iteration m, updates its location estimate $(\hat{\mu}_i(m))$ and variance $(\hat{\sigma}_i^2(m))$ as follows,

$$\hat{\mu}_i(m) = \frac{\frac{1}{\tilde{\sigma}_i^2}\tilde{\mu}_i + \sum_{j \in N(i)} \frac{1}{\hat{\sigma}_j^2(m-1)} \hat{\mu}_j(m-1)}{\frac{1}{\tilde{\sigma}_i^2} + \sum_{j \in N(i)} \frac{1}{\hat{\sigma}_j^2(m-1)}}, \tag{7.7}$$

$$\hat{\sigma}_i^2(m) = \frac{1}{\frac{1}{\tilde{\sigma}_i^2} + \sum_{j \in N(i)} \frac{1}{\hat{\sigma}_j^2(m-1)}}. \tag{7.8}$$

The location estimate for the ith query video $\hat{x}_i$ is taken to be $\hat{\mu}_i(m)$ at the end of m iterations, or when the algorithm has converged. The variance $\hat{\sigma}_i^2(m)$ provides a confidence metric on the location estimate. Figure 7.5 provides an illustration of the algorithm.

Given the graphical model framework, it is now easy to incorporate visual and audio features. These features can be used to modify the potential functions $\psi(x_i, x_j)$ on the edges. The intuition is that if two images are similar in some feature space, then they are also most likely geographically closer. However, this intuition holds true only when we already have a coarse estimate of their locations and we want to further refine it. For this purpose, we adopt a hierarchical approach. We first obtain a coarse estimate of the query video's location using only the tags in the graphical model. We then choose a subgraph for each query video consisting only of query and training

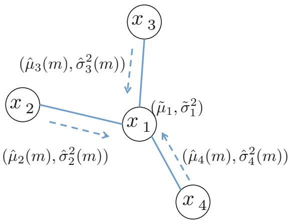

x_3
$(\hat{\mu}_3(m), \hat{\sigma}_3^2(m))$
x_2
$(\tilde{\mu}_1, \tilde{\sigma}_1^2)$
x_1
$(\hat{\mu}_2(m), \hat{\sigma}_2^2(m))$
$(\hat{\mu}_4(m), \hat{\sigma}_4^2(m))$
x_4

Fig. 7.5 Illustration of messages passed along the edges

videos within some particular radius of each other. Visual and acoustic features are obtained for each video using GIST and MFCC features similar to what Friedland et al. use in [16]. The probability distribution of the geographic distance between two videos given the closeness of the videos in the visual and audio feature space is modeled as a mixture of exponentials and is incorporated in the edge potential function $\psi(x_i, x_j)$.

7.5 Experimental Results

We tested our algorithm on the MediaEval 2011 Placing Task data set, which consists of Creative Common-licensed videos manually crawled from Flickr [20]. Videos are not filtered or selected in any way to make the data set more relevant to the task, and are therefore likely representative of videos selected at random. In current online databases (e.g., Flickr and YouTube), videos are not equally distributed over the earth (Fig. 7.2), so downloading a random sample shows a large bias for certain locations. The metadata for each video includes user-contributed title, tags, and description, and may include other data as well. Flickr requires that an uploaded video be created by its uploader and therefore many videos are home-video style short clips. Manual inspection of the data set led us initially to conclude that most of visual/audio contents lack reasonable evidence to estimate the location without textual metadata. For example, unlike the filtered test datasets used in [6], many videos were recorded indoors or in a private space such as the backyard of a house. The data set consists of 10,216 training videos and 5,347 query videos.

Figure 7.6 plots the overall estimation error for all the query videos. The x-axis is the estimation error binned at different error values and the y-axis is the percentage of videos that had an estimation error falling in the corresponding bin.

In order to better understand the performance of the algorithm as a function of the number of training and test data, we carry out experiments for different values of training and test data. We also evaluate the performance relative to a state-of-art

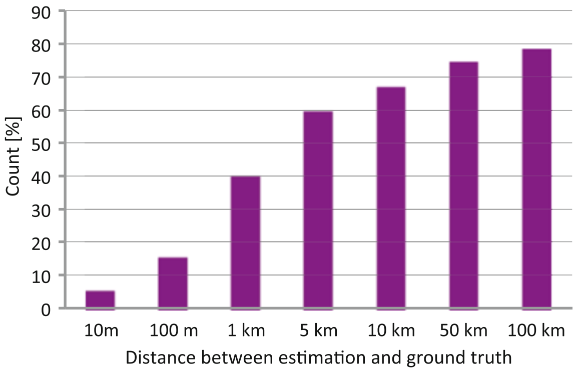


Fig. 7.6 Histogram of the estimation error in geotagging. The x-axis is the estimation error binned at different error values and the y-axis is the percentage of videos that had an estimation error falling in the corresponding bin

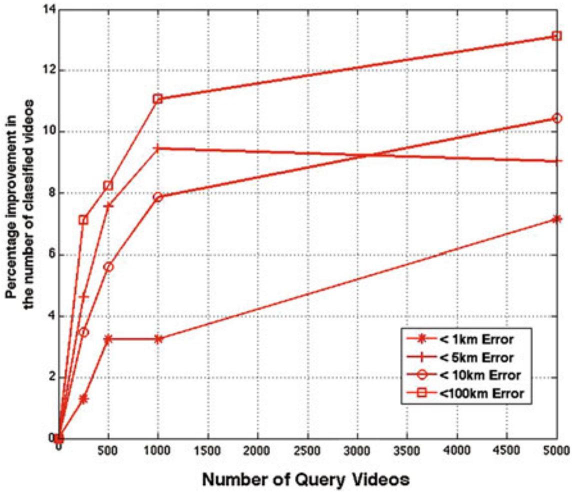


Fig. 7.7 Performance improvement in geo-tagging 5,347 videos using a training set of 500 videos as a function of the number of query videos used in the graphical model

algorithm in the literature. Figure 7.3 shows the performance of [15]. We evaluate the performance of the proposed algorithm for different subsets of the training videos in comparison with this system.

In order to understand the performance improvements obtained in the data sparse case, we use 500 training videos and plot the performance improvement as more and more query videos are added to the system. Figure 7.7 shows the performance improvement in different categories. The number of test videos for this plot is fixed at 5,347. The x-axis shows the number of query videos that were used in forming the graph. For example, the point 1,000 on the x-axis means that, while building the graph for all the 5,347 videos, each video had a neighbor only from these 1,000 videos. The best case is when any video in 5,347 videos could be a neighbor of any other video. The y-axis is the performance improvement over the baseline system. The different curves correspond to different error categories i.e., <1 km error, 10–100 km error etc. The performance improvement in each category is calculated as follows,

$$A = \text{Num. of videos with} < 1\text{km error using our algorithm,} \tag{7.9}$$

$$B = \text{Num. of videos with} < 1\text{km error using the baseline algorithm,} \tag{7.10}$$

$$\text{Perf. improvement in} < 1\text{km error category} = \frac{A - B}{B} \times 100.$$

This extends to the other error categories. The performance improvement can be over 10 % and increases with the number of test videos. However, one can clearly see the diminishing returns with increasing number of query videos.

Figure 7.8 shows the performance improvement from the test videos as the number of training videos increases. In this plot, the underlying graph was generated with all

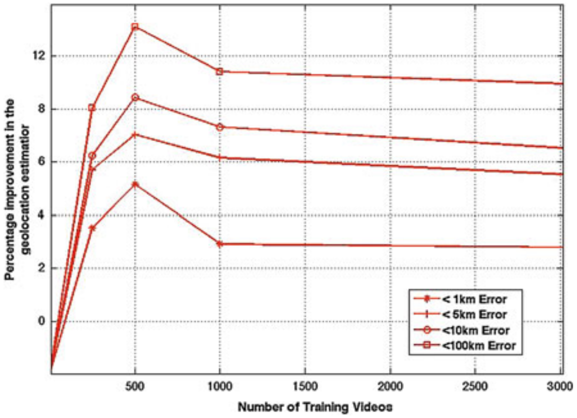


Fig. 7.8 Performance improvement in geo-tagging 5,347 videos as a function of the number of training videos

the 5,347 test videos and the performance improvement was observed as a function of the number of training videos. Though there are gains initially, as the number of training videos increases, the performance improvement obtained by using the query videos decreases. This is to illustrate that in the sparse data case with fewer training videos, the performance improvements can be large. However, with a larger training set, the performance improvement can be very marginal. In practice, given that most of the videos are not geo-tagged, i.e., the test set can be orders of magnitude larger than the training set, one can hope to achieve significant performance gains.

7.6 Conclusions and Future Work

We presented a novel approach to address the problem of geo-tagging multimedia content on the Web like Flickr videos and images. In particular, we addressed the case where the training data set is sparse and explored the possibility of using the test data set to improve the quality of the training database. We proposed a graphical model framework, posed the problem of geo-tagging as one of inference over this graph, and showed that performance improvements can be achieved by processing the test data set in an intelligent way.

Various issues remain to be explored. The modeling of edge potentials in the graphical model is very naive and one can explore richer models such as hierarchical models (e.g., latent dirichlet topic models) to model the correlations on the edges. The node potentials are further modeled as a product of the distributions given each tag individually. However, the distribution of the locations given multiple tags is not independent. For example, the location distributions of the tags “berkeley” and “sathergate” are clearly not independent. Hence a better correlation model needs to be explored for these distributions.

References

1. Mediaeval web site, http://www.multimediaeval.org
2. M. Larson, M. Soleymani, P. Serdyukov, S. Rudinac, C. Wartena, V. Murdock, G. Friedland, R. Ordelman, Gareth J.F. Jones, Automatic Tagging and Geo-Tagging in Video Collections and Communities, in *ACM International Conference on Multimedia Retrieval* (ICMR 2011), April 2011, p. to appear
3. G. Friedland, O. Vinyals, T. Darrell, Multimodal Location Estimation, in *Proceedings of ACM Multimedia*, 2010, pp. 1245–1251
4. G. Schindler, M. Brown, R. Szeliski, City-scale location recognition, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–7
5. W. Zhang, J. Kosecka, Image based localization in urban environments, in *3D Data Processing, Visualization, and Transmission, 3rd Intl. Symposium on*, 2006, pp. 33–40
6. J. Hays, A.A. Efros, IM2GPS: estimating geographic information from a single image, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8
7. J. Luo, D. Joshi, J. Yu, A. Gallagher, Geotagging in multimedia and computer vision-a survey. Multimed. Tools Appl. **51**, 187–211 (2011)
8. T. Rattenbury, M. Naaman, Methods for extracting place semantics from Flickr tags. ACM Trans. Web (TWEB) **3**(1), 1–30 (2009)
9. P. Serdyukov, V. Murdock, R. van Zwol, Placing Flickr photos on a map, in *ACM SIGIR*, 2009, pp. 484–491
10. O. Van Laere, S. Schockaert, B. Dhoedt, Ghent university at the 2011 placing task, in *Proceedings of MediaEval*, 2011
11. L. Cao, J. Yu, J. Luo, T. Huang, Enhancing semantic and geographic annotation of web images via logistic canonical correlation regression, in *Proceedings of the 17th ACM International Conference on Multimedia*, New York, NY, USA, 2009, MM '09, pp. 125–134, ACM
12. A. Gallagher, D. Joshi, J. Yu, J. Luo, Geo-location inference from image content and user tags, in *Proceedings of IEEE CVPR*. 2009, IEEE
13. David J. Crandall, Lars Backstrom, Daniel Huttenlocher, Jon Kleinberg, Mapping the world's photos, in *Proceedings of WWW '09*, New York, NY, USA, 2009, pp. 761–770, ACM
14. P. Kelm, S. Schmiedeke, T. Sikora, A hierarchical, multi-modal approach for placing videos on the map using millions of flickr photographs, in *Proceedings of SBNMA '11*, New York, NY, USA, 2011, pp. 15–20, ACM
15. G. Friedland, J. Choi, H. Lei, A. Janin, Multimodal Location Estimation on Flickr Videos, in *Proceedings of the 2011 ACM Workshop on Social Media*, Scottsdale, Arizona, USA, 2011, pp. 23–28, ACM
16. J. Choi, H. Lei, G. Friedland, The 2011 ICSI Video Location Estimation System, in *Proceedings of MediaEval*, 2011
17. M.J. Wainwright, M.I. Jordan, Graphical models, exponential families, and variational inference. Found. Trends Mach. Learn. **1**, 1–305 (2008)
18. Le Song, Arthur Gretton, Danny Bickson, Yucheng Low, Carlos Guestrin, Kernel belief propagation, in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 707–715
19. N. Vlassis, A. Likas, A greedy em algorithm for gaussian mixture learning. Neural Process. Lett **15**(1), 77–87 (2002)
20. A. Rae, V. Murdock, P. Serdyukov, Working Notes for the Placing Task at MediaEval 2011, in *MediaEval 2011 Workshop* (Pisa, Italy, September 2011)

Chapter 8
Georeferencing Flickr Resources Based on Multimodal Features

Pascal Kelm, Sebastian Schmiedeke, Steven Schockaert, Thomas Sikora, Michele Trevisiol and Olivier Van Laere

8.1 Introduction

The popularity of social media, and location-based services in particular, has led to a vast increase in the number of georeferenced resources on the web. Examples are the large numbers of Flickr photos, Twitter posts (tweets), and Wikipedia articles for which explicit geographic coordinates are currently available. This trend has also led to an upsurge in research into methods for automatically assigning coordinates to web content. Being able to associate coordinates to web content is important in applications such as geographic information retrieval [16], where search results are adapted to the location of the user, and in applications which rely on characterizing places, e.g., for offering personalized travel recommendations [13].

Although there are differences between georeferencing Flickr photos, tweets, and general web documents, the most useful input is usually in textual form, viz. the tags associated with Flickr photos and the contents of tweets and web documents.

P. Kelm (✉) · S. Schmiedeke · T. Sikora
Technische Universität, Berlin, Germany
e-mail: kelm@nue.tu-berlin.de

S. Schmiedeke
e-mail: schmiedeke@nue.tu-berlin.de

T. Sikora
e-mail: sikora@nue.tu-berlin.de

S. Schockaert
Cardiff University, Cardiff, UK
e-mail: S.Schockaert@cs.cardiff.ac.uk

M. Trevisiol · O.Van Laere
Yahoo Labs, Barcelona, Spain
e-mail: trevi@yahoo-inc.com

O.Van Laere
e-mail: vanlaere@yahoo-inc.com

J. Choi and G. Friedland (eds.), *Multimodal Location Estimation of Videos and Images*,
DOI 10.1007/978-3-319-09861-6_8

As a result, methods that have been proposed for georeferencing these types of resources tend to be similar. Broadly speaking, there are three ways coordinates can be assigned to a text document. First, named-entity tagging has traditionally been used to identify mentions of place names in documents, with gazetteers subsequently being used to map these place names to coordinates. The geographic scope of a web document can then be identified with the centroid of those coordinates, or, more commonly, with a probability density of locations. This method is difficult to apply to Flickr photos, since tags lack the context needed to effectively identify named entities. This is exacerbated by the fact that Flickr tags are converted to lower case, which makes it challenging to resolve ambiguities between place names and common nouns (e.g., *nice* vs. *Nice*). Second, we can treat the problem of georeferencing text documents as a classification problem. This approach has been adopted in, among others, [19] for Flickr photos and [29] for Wikipedia articles. Essentially, in this approach, the locations on Earth are discretized into a finite set of areas, using clustering, a fixed-grid representation, or administrative boundaries of geographic regions. Using standard text classification methods, such as Naive Bayes or support-vector machines, the most plausible area is identified, and the centre-of-gravity of that area is used as the estimated location. Third, we can try to find resources with known coordinates which are similar to the resource to be georeferenced, and the location of these resources can be used to estimate the location. A combination of the latter two approaches was advocated in [25], which showed that a two-step approach which first uses a text classifier to find an appropriate area and then uses a similarity search to find a plausible location within that area consistently outperforms either of the two individual methods.

In this chapter, we focus on the case of Flickr photos, and in particular on the question of how the aforementioned text-based methods could be improved by taking into account visual information as well as information from the user profile of the photo uploader. Taking visual information into account efficiently has generally proved challenging in the field. When easily identifiable landmarks such as the Eiffel tower are shown in a photo, methods based on, e.g., SIFT features can be very effective in determining the correct coordinates from the photo alone. However, such cases are rare in practice, and in most of these cases, the correct location can also be obtained from available tags, at a much lower computational cost. We argue that the most effective way to utilize visual features of photos is by extending the aforementioned two-step approach from [25] to a three-step approach. Specifically, an approximate location is first identified using textual information alone (possibly also with some evidence from the owner's user profile), and this location is then refined by comparing the photo to be georeferenced with nearby photos with known coordinates. Given that only nearby photos are considered, lower-level visual features can be especially effective, even though such features are too general to be of use at a worldwide scale.

This chapter is structured as follows. In the next section, we discuss the peculiarities of Flickr photos and describe preprocessing methods which make the subsequent analysis more reliable. Section 8.3 then gives an overview of text-based methods, and visual methods are presented in Sect. 8.4. Finally, in Sect. 8.5, we discuss how the two types of approaches can be integrated.

8.2 Data Selection and Preprocessing

Flickr contains more than 300 million photos with associated geographical coordinates.[1] By analyzing correlations between the locations of these photos on the one hand, and the visual features and textual metadata of the photos on the other hand, we can train a system that estimates the location of previously unseen Flickr photos or videos.

For each uploaded photo, Flickr maintains several types of metadata, which can be obtained via a publicly available API.[2] In this paper, three types of metadata will be relevant. First, photos may be associated with descriptive tags provided by the photos' owners. Many of these tags provide evidence about where the photo was taken (e.g., because they refer to a place name or the name of an event, because they are in a particular language, or because they refer to regional cuisine), so modeling the spatial distribution of photos which have been assigned a particular tag will play a particularly important role in our framework. Second, photos are associated with an owner, whose user profile contains a free-text field describing their home location (e.g., "Ghent, Belgium"). It turns out that this home location is particularly helpful when dealing with photos that do not contain any location-specific tags, or only tags which are ambiguous. All things being equal, photos are more likely to have been taken close to the owner's home location than at the other side of the world. Finally, the photos we consider are also associated with a geographical coordinate, which is considered as the ground truth for the purposes of this work. Note, however, that this is a simplifying assumption, as, for example, photos of a landmark may be associated with the position at which the photo was taken or with the location of the landmark. Indeed, while a small percentage of coordinates come from GPS devices, most coordinates are manually provided by users. For each pair of coordinates, Flickr provides information about its precision, encoded as a number between 1 (world-level) and 16 (street-level), reflecting the zoom level of the map the owners used to assigned the coordinates (in the case of manually assigned locations).

In most approaches for georeferencing Flickr photos, a number of preliminary filtering steps are carried out to clean the training data:

1. Photos that do not have any tags or have invalid coordinates are removed.
2. Photos whose location precision is too low for the task (e.g., 11 or lower) are discarded, retaining only those photos that provide meaningful location information at the sub-city scale.
3. If there are multiple photos with the same upload date, an identical tag set, and identical coordinates, only one of the photos is retained. Users on Flickr can upload content in bulk, i.e., upload multiple photos at once and tag them with the same information; this can skew the analysis, as was first pointed out in [20].

The photos that remain after these filtering steps are used for obtaining clusters of locations and for estimating language models.

[1] http://pressroom.yahoo.co.uk/pr/ycorpuk/flickr-tips-safer-photos.aspx, accessed 31 March 2014.

[2] http://www.flickr.com/services/api/, accessed 28 January 2014.

The tags associated with the photos can also be preprocessed. Flickr normalizes the user-provided tags by converting them to lowercase, removing spaces within tags, and then replacing commas between tags with spaces. For example, the set of tags “Trip 2010, Sagrada Familia, Barcelona” becomes “trip2010 sagradafamilia barcelona”. Some approaches use a number of additional preprocessing steps, e.g., removing diacritics and numbers, separating numerical characters from alphanumeric tags, and/or discarding words from a problem-specific list of stop words (e.g., camera brands and lens types). It should be noted that such forms of preprocessing do not always improve results. For example, we found that the tag “911” (referring to the attacks on 11 September 2001) is strongly correlated with locations in New York City. To give an idea of how many videos with tags tend to remain after these preprocessing steps, in the MediaEval training set of 2012, approximately 40 % of the photos contained at least one tag.

8.2.1 Clustering the Training Data

Most approaches to georeferencing Flickr resources interpret it as a classification problem. To this end, the locations of the photos in the training data are clustered into sets of disjoint areas, which are then interpreted as the class labels. Once a classifier has identified the most likely area where a photo was taken, we may use other techniques to find the most likely location within that area, as detailed in Sects. 8.3 and 8.4.

A number of techniques are available for obtaining a clustered representation of locations. Some of these methods are compared experimentally in [9, 11, 27]. Here, we summarize the main advantages and disadvantages of several popular methods. We will also illustrate each of these methods by using them to cluster the MediaEval Placing Task data set (Figure 8.1 shows the distribution of this dataset over Europe).

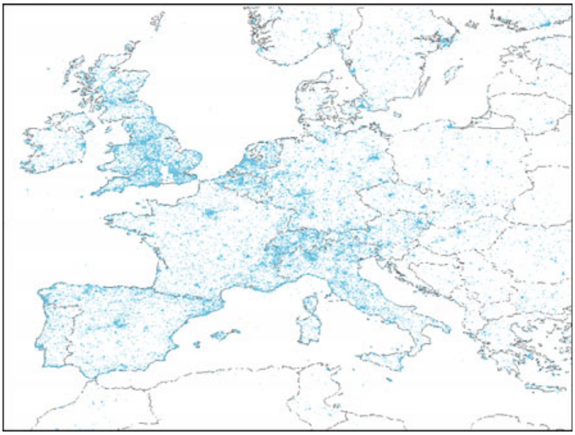


Fig. 8.1 Distribution of the MediaEval Placing Task 2012 data set for Europe

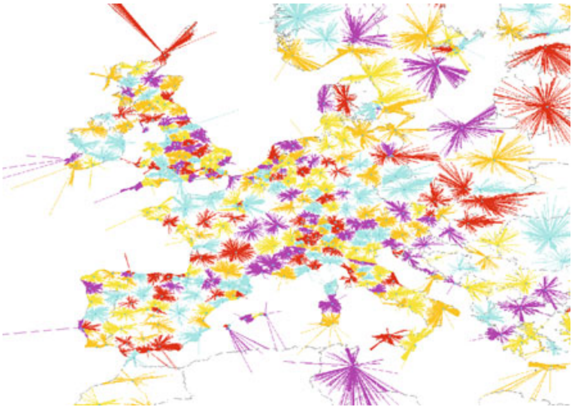


Fig. 8.2 Sample clustering for Europe using k-medoids, k = 1,000 (worldwide)

k-medoids clustering is closely related to the well-known k-means clustering algorithm, differing only in how the center of each cluster is determined. While k-means uses the center-of-gravity for this purpose, in k-medoids, the center is selected as the medoid of the cluster, i.e., the element which minimizes the sum of the distances to the other elements of the cluster. The main advantage of using k-medoids is that the selection of the medoid is more robust to outliers than the center-of-gravity; this is why k-medoids is more commonly used than k-means for clustering sets of coordinates. However, it should be noted that selecting the medoid of a cluster has a time complexity which is quadratic w.r.t. the size of the cluster, in comparison with the linear complexity of selecting the center-of-gravity. For very large training sets, this means that the number of clusters chosen must be sufficiently high (such that the number of photos per cluster is manageable), or that only a sample of the training data can be used to obtain the clusters. Distances are usually calculated using the Haversine metric instead of the Euclidean metric. An example clustering with $k = 1,000$ clusters (worldwide) is shown in Fig. 8.2.

Grid clustering uses a fixed grid of square (or sometimes hexagonal) cells over the surface of the Earth. The main advantage of this method is that it is computationally inexpensive. In the example in Fig. 8.3, grid cells correspond to 4.375 degrees of latitude and longitude; this value results in 1001 clusters worldwide, making it easily comparable with Fig. 8.2. An important distinction from k-medoids clustering is that the size of the grid cells does not depend on the amount of available training data. However, this means we cannot attempt a more accurate classification in areas of the world for which we have abundant training data while being more cautious in areas where training data is sparse. Experimental results described in [27] confirm that using a grid leads to worse performance than using k-medoids.

Mean shift clustering does not require predefining the number of clusters, but instead relies on a scale parameter h. The number of resulting clusters emerges from the choice of the scale factor. An example with $h = 150$ is given in Fig. 8.4. Two points are worth noting here. First, outliers tend to end up in separate clusters, leading

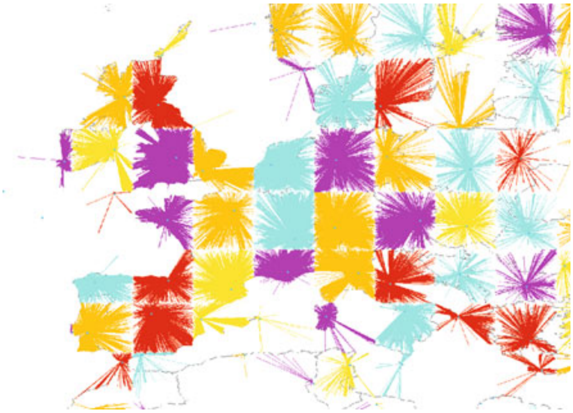


Fig. 8.3 Sample clustering for Europe using a grid of 1,001 cells over the world

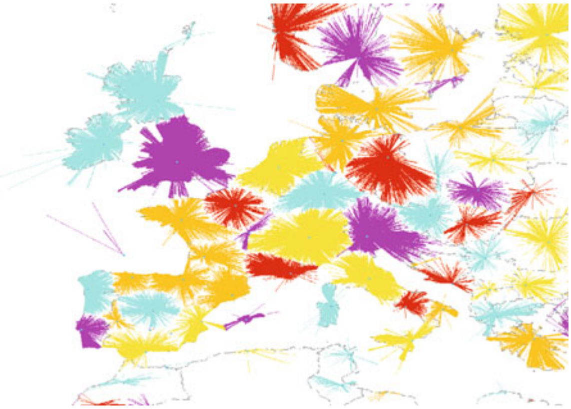


Fig. 8.4 Sample clustering for Europe using the mean shift algorithm ($h = 150$, resulting in 2,349 clusters worldwide)

to a large number of small clusters. Second, as with the grid clustering, the granularity of the clusters does not reflect the amount of training data. As a result, models for georeferencing Flickr photos (see Sect. 8.3) perform worse when using mean shift clustering than when using k-medoids clustering, even if small clusters with outliers are merged with other clusters [27].

Finally, as an alternative to clustering, considered in [9, 10], we could also define the set of *areas based on national boundaries* (as depicted in Fig. 8.5). This method has the advantage that the definition of the areas is likely to be better aligned with the distribution of tags (e.g. the distribution of toponyms and of languages used for tags). As with grids and mean-shift clustering, a disadvantage is that the size of the clusters does not reflect the amount of available training data. However, we could combine the best of both worlds, by considering a two-level hierarchical clustering, where the first level is based on national boundaries and the second level corresponds to k-medoids-based clusterings of photos within the same country. Note that such a two-level approach requires that we can accurately find out the correct country for

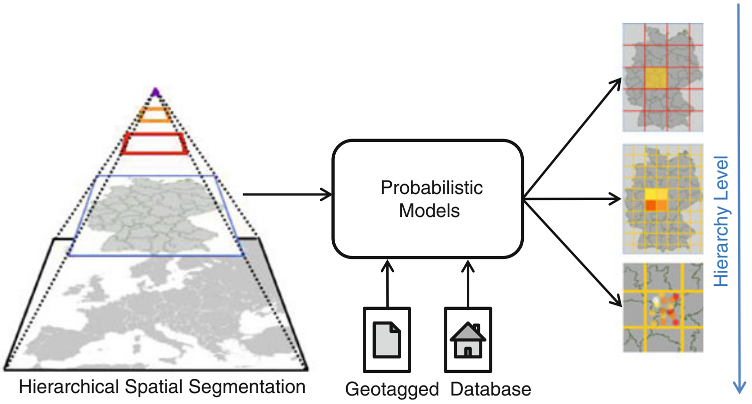


Fig. 8.5 Two-level hierarchical clustering [9]

a given test photo or video. In many cases, this is a realistic assumption, since the determination of a country is usually less problematic than disambiguating the name of a landmark or a city. For example, for a given photo, each associated term (e.g. tag or title word) may potentially refer to a place name. We ca use a gazetteer (i.e., a dictionary of named places, usually linking place names to geographic locations and sometimes a semantic type) to find out which terms may be place names, and where on Earth places with that name occur. One of the most popular gazetteers is the GeoNames database [1] which contains over 10 million geographical names corresponding to over 7.5 million unique features and provides a web-based search engine which returns a list of entries ordered by relevance. The approach from [9, 10] uses GeoNames to create a ranking of the possible countries with which a photo or video can be associated (based on the possible interpretations of the place names associated with its tags). Then, the boundary of the most likely country is determined by querying the Google Maps API [2].

8.2.2 Term Selection

Many of the tags associated with photos and videos on Flickr are not useful for estimating geographic location. To prevent overfitting, it is useful to apply a term selection step, in which all tags that are not deemed geographically relevant are removed. There is a wide selection of methods that can be used for this purpose.

If we consider the areas obtained after clustering, term selection methods that have proven effective for text classification could be applied [30, 31]. Examples of popular methods include χ^2 and information gain. The advantage of such methods is that they are easy to implement and they are based on well-known statistical and information theoretic principles. However, such methods effectively ignore the spatial dimension of the problem. For this reason, [5] introduced a heuristic, location-aware method for selecting terms. The method proposed in [5], called geographic spread, first clusters adjacent grid cells in which a particular tag occurs. Tags are deemed geographically relevant if the number of clusters is sufficiently small and the largest cluster contains a sufficiently large number of tag occurrences. Despite the heuristic nature of this measure, it substantially outperforms methods such as χ^2 and information gain. Finally, methods from the field of geospatial statistics could be considered. For example, using kernel density estimation (KDE) we may model each tag as a smooth probability distribution over the set of locations on Earth. We could then select tags whose associated distribution diverges from a background distribution (reflecting the distribution of Flickr photos and videos). Another possibility is to use methods from epidemiology, such as Ripley's K statistic, to identify tags whose pattern of occurrences deviates significantly from a uniform sampling. Finally, methods for measuring spatial autocorrelation could be used to identify tags whose occurrences tend to be clustered in space.

In [23], an analysis is presented of how different term selection techniques perform in the context of georeferencing Flickr resources. The geographical spread measure, KDE-based methods and a method based on Ripley's K statistic performed comparably. Of these three approaches, the geographical spread measure has the clear advantage of being the easiest method to implement and being the computationally least expensive method. However, it was found to be more sensitive to the number of selected features. Out of a total of about 300 K features, all methods performed optimally when about 50–100K features were selected. However, while the KDE and Ripley's K-based approaches still performed well when only 10 or 25 K features could be selected, the geographic spread measure was not competitive in that range. Overall, the experiments in [23] strongly suggest that measuring spatial autocorrelation is not sufficient; for a term selection method to be effective in this context, it needs to favor terms which only occur in a small number of areas around the world. For example, while the term *beach* is clearly geographically relevant (i.e., spatially autocorrelated), it is not a useful term for deciding in which area a photo was most likely taken.

8.3 Textual Approach

There are two main text-based approaches for estimating the geographic location of a Flickr photo or video. First, we may view the problem as a classification task, in which the classes are geographic areas. For a given photo, the most likely area is determined and the center of that area is used as the estimated location. This approach was proposed in [20]. Second, we may treat the task of assigning a location to a Flickr photo or video as a retrieval task. In this case, we first identify the photos in the training data which are most similar to the considered resource, and use their locations to determine the estimated location. As proposed in [24], we can combine these two approaches by first using a classifier to find the most likely area where the photo or video was taken, and then find similar photos in the training set from that area. Experimental results in [22, 27] show that such a hybrid two-step approach outperforms either of the individual methods. A key factor is the number of clusters considered: the fewer clusters are used, the more emphasis there is on the retrieval step. It turns out that, in general, the more training data is available, the smaller the optimal number of clusters.

In the remainder of this section, we explain in more detail how both approaches operate.

8.3.1 Variations on the Classification Approach

As we explained in Sect. 8.2.1, various methods exist to segment the training data in a set of disjoint areas. We can see each of these areas as a class, and treat the problem of georeferencing Flickr photos as a standard text classification problem. Some authors have used support vector machines [4] or Kullback-Leibler divergence [29] for georeferencing social media documents. The most popular approach, however, seems to be to use a naive Bayes classifier [20, 27], based on a multinomial bag-of-words language model.

In this model, the probability that a Flickr photo d with tags $t_1, ..., t_n$ was taken in area a is estimated as:

$$P(a|d) \sim P(a) \cdot \prod_{i=1}^{n} P(t_i|a) \tag{8.1}$$

As usual, the use of logarithms replaces the multiplication by a summation and prevents the underflow of floating points:

$$\log P(a|d) \sim \log P(a) + \sum_{i=1}^{n} \log P(t_i|a) \tag{8.2}$$

We now go in more detail on how the likelihood $P(t_i|a)$ and the prior probability $P(a)$ can be estimated from Flickr.

8.3.1.1 Estimation of Term Location Distribution

The probability $P(t|a)$ reflects the likelihood of seeing a photo with tag t in photos that have been taken in area a. The simplest way of estimating this likelihood would be to choose $P(t|a) = \frac{N_{t,a}}{N_a}$, where $N_{t,a}$ is the number of photos with tag t in area a and N_a is the total number of photos in area a. However, this would lead to $P(a|d) = 0$ as soon as d has one tag which has not previously been seen in area a. To cope with this problem, usually some form of smoothing is applied.

One possibility for smoothing the probability $P(t|a)$, called Laplace smoothing:

$$P\ (t|a) = \frac{N_{t,a} + 1}{N_a + |V|}, \tag{8.3}$$

where V is the set of all tags (that have been retained after feature selection). Another possible smoothing method is Bayesian smoothing with Dirichlet priors, in which case we have ($\mu > 0$):

$$P(t|a) = \frac{N_{t,a} + \mu\ P(t|V)}{N_a + \mu} \tag{8.4}$$

where the probabilistic model of the vocabulary $P(t|V)$ is defined using maximum likelihood:

$$P(t|V) = \frac{\sum_a N_{t,a}}{\sum_{t' \in V} \sum_a N_{t',a}} \tag{8.5}$$

A final possibility is to use Jelinek-Mercer smoothing, in which case we have ($\lambda \in [0, 1]$):

$$P(t|a) = \lambda \frac{N_{t,a}}{N_a} + (1 - \lambda)\ P(t|V) \tag{8.6}$$

with $P(t|V)$ defined as in (8.5). For more details on these smoothing methods for language models, we refer to [32]. A comparison between these smoothing methods in the context of georeferencing Flickr photos has been given in [27]. Figure 8.6 shows the term-location probabilities $P(t|a)$ of terms t occurring in a video located at the Ground Zero, Mahattan, New York, USA. As shown, single terms do not indicate a specific or right location, only the combination maximizes the likelihood of the right spatial segment. Thus we can conclude that generic terms like '911' do help with specifying the location despite the fact that these terms do not have a geographical relation in the sense of being an entry in a gazetteer.

Note that all the above formulas describe our probabilistic model when using a multinomial distribution with term frequency (tf) weighting. Other weighting schemes are possible, as we will explain in Sect. 8.3.1.3.

8.3.1.2 Estimation of the Prior Probability

There are at least three different ways of choosing the prior probability $P(a)$ that a photo is taken in area a. In some situations, we want to refrain from introducing any bias, and choose $P(a)$ as a uniform probability. However, if photos are randomly sampled from Flickr, some areas of the world are much more likely than others. We ca use a maximum likelihood estimation to include this evidence, in which case we choose:

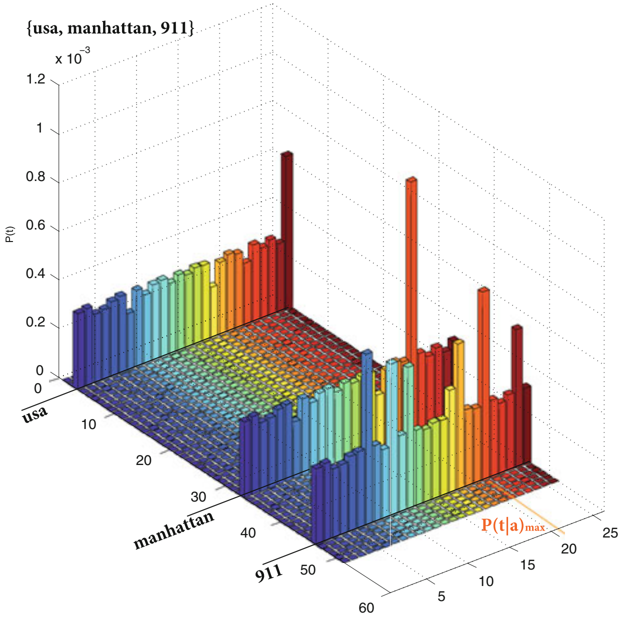


Fig. 8.6 Probabilities $P(t|a)$ of a video containing the terms: 'usa', 'manhattan' and '911'

$$P(a) = \frac{N_a}{\sum_{a'} N_{a'}} \tag{8.7}$$

A third possibility is to take into account prior knowledge about the home location of the owner of the photo, using the intuition that areas closer to where the owner lives are more likely. In [26], the following prior probability was proposed:

$$P(a) \sim \left(\frac{1}{d(a,h)} \right)^{\theta} \tag{8.8}$$

where h is the estimated home location of the user (obtained by georeferencing the corresponding text field in their profile) and a is the centre of area a. Along similar lines, [6] uses a prior probability which is based on population density and a prior probability which is based on climate data (assigning a higher prior probability to more temperate climates). Additionally, it could be worth considering a different prior probability for each user, if sufficient information is available. This was proposed in [22], using a method called *User History*. In particular, when the training set contains previously uploaded photos or videos made by the same user, computing the prior probability based on these locations may help to improve the estimation. Furthermore, it has been investigated in [22] how the user's social network could be used to extend the user-based prior location, following the assumption that the location of a user might be related with the locations of her social connections.

8.3.1.3 Weighting of Term Occurrences

So far, the *term-location probability* and the *prior probability* are estimated by counting the frequency of each term, as applied in (8.3)–(8.6). In this section we will explain how term weighting is used to improve these estimations. Hereby, the count $N_{t,a}$ is replaced with a weight $W(t, a)$ which may reflect other aspects than just the frequency of occurrence of the term. Term weighting is a well know technique in the Information Retrieval domain [15].

As a first method, the *term frequency* (tf) weighting, which has been used so far, is used to distinguish between areas that contain a given tag only a few times and those that contain the tag many times. The use of term frequency corresponds to the assumption that areas which contain more mentions of a given tag are more likely to contain the location of a photo or video with that tag. The number of occurrences of a tag t in area a is usually normalized, by dividing it by the number of occurrences of the most frequent tag, among those tags in photo or video d.

$$W_{\text{tf}}(t, a) = \frac{N_{t,a}}{\max_{t' \in d} N_{t',a}} \tag{8.9}$$

Some studies in [9] have shown that it is not always useful to weight the frequency of a tag, assuming that all that matters is whether it occurs at least once.

$$W_{to}(t, a) = \begin{cases} 1, & N_{t,a} > 1 \\ 0, & \text{otherwise} \end{cases} \tag{8.10}$$

In the context of geo-location recognition, certain tags have little or no discriminating power. For instance, the term ‘video’ or ‘photo’ exists almost in every area. The idea is to reduce the tf weight of a given tag by a factor which is increasing in the number of areas in which the tag occurs. For this reason the *inverse document frequency* (idf) of a tag t is defined as follows:

$$W_{\text{idf}}(t, a) = log \frac{N}{\sum_a W_{to}(t, a)} \tag{8.11}$$

where N denotes the total number of areas.

The *term frequency-inverse document frequency* (tf-idf) weighting schemes is one of the best known in information retrieval. It combines the *term frequency* and *inverse document frequency*: The $N(t_i, d)$ in (8.9) and $N_{t,l}$ in (8.10) are replaced by the tf-idf scores.

$$W_{\text{tf}-\text{idf}}(t, a) = W_{\text{tf}}(t, a) \cdot W_{\text{idf}}(t, a) \tag{8.12}$$

These three weighting schemes are used for estimating the term-location probability (Sect. 8.3.1.1) and the prior probability (Sect. 8.3.1.2) by replacing the term count N by one of the introduced weightings. The three different weighting schemes

Table 8.1 Correct decision for spatial segments

Top-N grid cells	TO (%)	TF (%)	TF-IDF (%)
1	31.7	44.5	51.4
2	39.7	51.2	57.5
3	44.3	54.3	60.0
4	47.1	56.0	61.6
5	48.8	57.6	62.6
6	50.4	58.6	63.8
7	51.3	59.4	64.8
8	52.1	60.2	65.3
9	53.3	60.7	65.8
10	53.9	61.3	66.4

— *term occurrence* (to), *term frequency* (tf), *and term frequency-inverse document frequency* (*tf-idf*)—are used for term-location probability and prior probability, and then compared against each other. Table 8.1 displays the percentage of correctly predicted cells (i.e., area) in a large grid of 360 × 180 segments (see Sect. 8.2.1), according to the meridians and parallels of the world map. In particular, this table shows how frequently the correct location is within one of the N highest ranked grid cells. The textual model with tf-idf weighting predicts the correct grid cell for the half of the Placing Task 2012 dataset. In 66 % of all cases, the correct locations is among the 10 most likely grid cells.

The *Okapi bm25* is more than a term scoring method, but rather a method for scoring documents in relation to a query. It was introduced in the early 1990s in [17], the *bm25* formula has been widely adopted, and it has repeatedly proved its value across a variety of search domains. One of the first comparisons with *tf-idf* was done by [28], showing that the Okapi *bm25* weighting scheme performed better in many cases. In the context of geo-location recognition, it can be used in line with a Query-Document retrieval approach, such as the one we discuss in Sect. 8.3.2. Alternatively the term weight could also be pre-computed independently from a given query, in which case the *Okapi bm25* weights are defined as follows:

$$W_{\mathrm{bm}}(t,a) = \mathrm{idf}(t) \times \frac{w'_{t,s} \times (k+1)}{w'_{t,s} + k \times (1 - b + b \times \frac{|D|}{\mathrm{avg}_{dl}})} \tag{8.13}$$

where $w'_{t,s}$ is the weight of the term t associated to the area a , avg_{dl} is the average number of tags per training sample, k and b are free parameters usually chosen as $k \in [1.2, 2.0]$ and $b = 0.75$ (see [15]).
The idf part instead is given by

$$\mathrm{idf}(t) = \log \frac{M - N_t + 0.5}{N_{t_i} + 0.5} \tag{8.14}$$

where M is the total number of training photos, and N_t is the number of training photos containing tag t. We refer to [9, 10, 22] for an experimental comparison of these methods.

8.3.2 Variations on the Retrieval Approach

This approach is inspired by the standard information retrieval setting, where the goal is to retrieve the documents which are most closely related to a given input query. In a geo-location context, our query is formed by the text attached to the test photo or video. The text could be the title, description, tags, or any other associated text. In this section we consider only the tags, but similar considerations apply if other text fields are used. In order to interpret georeferencing as a document retrieval problem, we also need to define the document collection. One possibility is to identify documents with geographic areas. In particular, each document then consists of the tags associated with all training photos of the corresponding area. Section 8.3.2.2 will discuss a number of alternative approaches that could be considered. First, in Sect. 8.3.2.1, we discuss a way to compute the geo-descriptiveness of a word. This measure of how much a word is related to a specific location is important to reduce the impact of noise (i.e., tags which are not indicative of a particular location, but whose distribution is not entirely uniform due to chance).

8.3.2.1 Geo-Relevance Filtering

Describing a geographic area or a specific coordinate as a weighted set of tags requires a weighting scheme that reflects the relationship between tags and coordinates. Since we want to consider only tags that are geographically relevant, we first need a way to determine the level of geographic spread (or *geo-descriptiveness*) of a tag. This is similar to the problem of term selection, which was discussed in Sect. 8.2.2. However, whereas term selection requires a hard decision, here we are interested in measuring the degree to which a tag is geographically relevant, so that we can discount, rather than ignore, geographically less relevant tags in the retrieval model.

We can measure the relevance of a tag by jointly considering its frequency of occurrence (or *term frequency*) and the average distance between the locations where it occurs. Therefore, for each tag t we compute its term frequency tf_t in the training data and the average Haversine distance d_t between the locations of the images or videos which contain t. For example, using the data of MediaEval2013 [7, 22] applied the following heuristic approach for measuring the degree of geo-descriptiveness of a tag t:

$$w_t = \begin{cases} -1 & \text{if } \mathrm{tf}_t > 100\,\mathrm{K} \text{ or } d_t < 0.2 \\ 10 & \text{if } \mathrm{tf}_t \geq 200 \text{ and } 10 \leq d_t \leq 50 \\ 5 & \text{if } \mathrm{tf}_t \geq 150 \text{ and } d_t \leq 70 \\ 1 & \text{otherwise} \end{cases}$$

This weighting was designed to assign higher weights to tags representing geographic information, i.e., not only places but also references to locations such as monuments, e.g., Eiffel, Colosseum) or famous people with a geographical correlation, e.g., Gaudi and Barcelona). Table 8.2 illustrates the effect of using these weights.

8.3.2.2 Frequency Matching

When interpreting the task of georeferencing Flickr photos as a document retrieval task, it is intuitive to treat the set of tags associated with a Flickr photo or video that we want to localise as the query. However, there are several possibilities for defining the document collection.

A simple solution is to consider each photo in the training data as an individual document, where the tags associated with the photo represent the terms of the document. However this approach is very noisy in the sense that it results in many documents whose text is identical but which have a different location. For example, the set of tags "france", "pompidou" and "paris" appear in many photos with slightly different coordinates, e.g., (48.8611, 2.3521) and (48.6172, 2.213), whereas we are interested in assigning a single location to a given query. This issue can be tackled by collecting all the coordinates associated to the same set of tags, counting how often each pair of coordinates appears. For the set of tags highlighted before we obtain the coordinates (48.8611, 2.3521) and (48.6172, 2.213) with frequency respectively 12 and 3. The frequency counter suggests that the first one is more reliable than the latter one since it appears more often. Once we obtained the most frequent coordinate we associate it to the set of tags. Alternatively, we could also select the medoid in the set of coordinates. Each document in the collection is then a set of tags, with an associated coordinate, such that no two documents correspond to the same set of tags:

$$< \text{france, pompidou, paris} > \quad \rightarrow \quad (48.8611, 2.3521)$$

A second way to define documents in our context is by grouping all tags that are assigned to documents at the same location. In other words, there is a document for each pair of coordinates, whose text contains all tags of all photos in the training data that have these coordinates:

$$(48.8611, 2.3521) \quad \rightarrow \quad < \text{france, centre, pompidou, centrepompidou, paris}, \cdots >$$

However we have to consider the presence of "duplicates" as photos at slightly different coordinates are often associated with similar tags (as the example highlighted before). In order to have a cleaner and more reliable collection of documents it is possible to merge documents if their location is sufficiently close and their text is sufficiently similar. Alternatively, we could use a clustering method to obtain a large set of areas, and then associate one document to each cluster (Fig. 8.7).

Table 8.2 On the left, we list the most frequently occurring tags. On the right, we list tags t which maximize the weight, i.e., for which $w_t = 10$, ordered by term frequency

Tag	Frequency	Avgerage distance	Tag	Frequency	Average distance
Geotagged	81948	183.600974549	California	34756	17.3347855177
Water	36645	394.109273724	Italy	26432	28.2242237023
Beach	35799	189.702712644	France	25128	30.1802960837
Nature	35791	330.458447429	Australia	22746	48.9602486282
2009	35411	158.518797983	Germany	22060	30.3288003332
2008	34788	163.01576923	Canada	20611	44.1591343262
California	34756	17.3347855177	Spain	20583	33.9538296347
2007	33791	163.774148Z22	Japan	19859	33.8409323297
Sky	32268	500.652309767	Uk	19675	33.246712809
travel	29636	227.985879656	England	19477	25.9678916173
Usa	27933	112.840988813	Espana	14740	36.3307740002
Italy	26432	28.2242237023	Scotland	14418	20.0878818974
Sea	25435	264.814684265	Italia	14266	29.7714336536
France	25128	30.1802960837	Deutschland	12469	26.6710348635
Sunset	24839	530.829445257	Mexico	11248	44.1710727199
Landscape	24291	399.483239901	Washington	10528	29.2185701167
Snow	24251	252.464879232	Texas	9811	26.8342797456
Europe	24173	121.315134058	Florida	9773	19.2311687716
Blue	23187	554.374093842	Newyork	9550	26.7850834626
2006	22802	188.297981051	Portugal	9005	36.4910584785
Australia	22746	48.9602486282	Switzerland	8842	19.0660337254
Night	22745	439.445587846	Sweden	8825	43.4298253824
Germany	22060	30.3288003332	Taiwan	8614	8.58611101877
Winter	21605	264.227771688	Ireland	8277	27.8336567598
Tree	21335	626.460930315	Newzealand	7804	34.7207757884
Canada	20611	44.1591343262	Greece	7734	28.8041197511
Spain	20583	33.9538296347	Ontario	7684	13.5533928009
BW	20475	504.583508417	Oregon	7501	21.9655577751
Architecture	20155	331.577365854	Unitedkingdom	7469	45.7784858581
Japan	19859	33.8409323297	Netherlands	7258	23.4561091895
Green	19822	585.402442074	Austria	6670	17.0680578585
UK	19675	33.Z4671Z809	Colorado	6659	29.053739839Z
Clouds	19597	617.251967179	Thailand	6508	23.5652696046
Flower	19563	631.402098385	Nsw	6231	28.5942670687
England	19477	25.9678916173	Arizona	6041	26.381253062
Park	19245	289.55492473	Sanfrancisco	6021	41.1354079352
Vacation	18870	220.652637728	Nyc	5917	32.7179331764

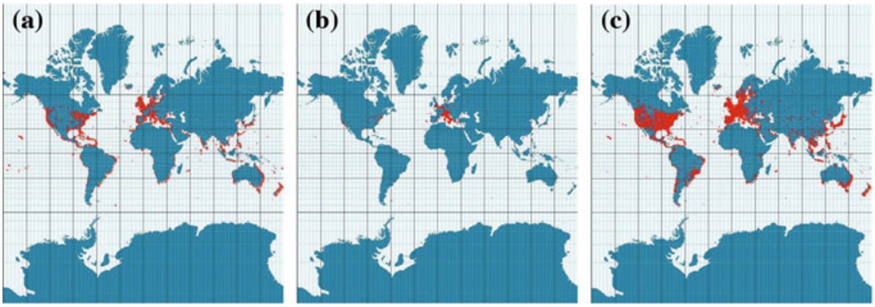


Fig. 8.7 Coordinates of three tags plotted on the world map. The *first* and the *third* figure show the spread of tags that are not bound to specific locations, whereas the middle one shows a tag related to a specific country. At first glance it is clear which is the more meaningful tag that describes a specific location, **a** beach, **b** italy, **c** iphone

In each case, we obtain a collection of documents, defined by a bag of tags and a unique location. To find the location of a test photo, we interpret its tags as the query and then identify the most similar document in the collection, taking its location as the most probably location of the test photo. In particular, we retrieve all documents which have at least one word in common with the query and rank these documents by relevance, using a standard weighting scheme such as tf-idf or *Okapi BM25* (see Sect. 8.3.1.3). According to [25], choosing the top ranked document outperforms taking a weighted average of the top-k documents, although these experiments used a term-frequency weighting, rather than tf-idf or *Okapi BM25*. However, in case of a tie (i.e., when several documents have the maximal relevance score) we could apply other approaches to select a pair of coordinates. In [21, 22], for example, the medoid of all the top ranked documents, weighted by the number of occurrences of each location, is taken as the most representative location.

8.4 Visual Approach

The georeferencing of photos or video recordings with visual features is useful for fine-tuning the result of the textual approaches and for georeferencing Flickr resources without any tags.[3] Several authors have investigated the usefulness of visual content for predicting geographical tags [11]. As expected, visual information is more challenging to use than textual information. Nevertheless visual georeferencing is also useful for other vision tasks by narrowing down the possibilities for further processing (e.g., visual landmark recognition) or refining textual approaches. Imagine a photo depicting a coastal scene or open water for which the geo-coordinates are not known. Many locations like hinterlands or metropolitan areas become very

[3] 13.6 % do not have any tags in the Placing task data set, although some of them can also be georeferenced by taking into account context information such as the home location of the user.

unlikely, and can be excluded from further processing. As in the case of textual methods, we ca use classification approaches and retrieval approaches to take into account visual information.

8.4.1 A Classification Approach to Using Visual Information

The classification approach is about modelling the characteristics of a spatial region in terms of visual features. These visual features can be described by various descriptors, grouped in Colour and edge/texture descriptors. The most prominent are Colour and Edge Directivity (CEDD), Gabor (GD), Scalable Colour (SCD), Tamura (TD), Edge Histogram (EHD), Autocorrelogram (ACC) and Colour Layout (CLD). Colour Histograms, in which the colour distribution is presented in quantised colour component bins, are used to differentiate on the basis of colour perception of the human vision. Edge features, as another class of visual features, help distinguish between natural scenes and scenes containing man-made structures, while texture features might help to discriminate properties such as different terrain types.

These features are applied on each single image or on key frames in case of video sequences, in order to reduce their temporal dimensionality. With these descriptors, a wide spectrum of colour and texture features within images is covered. We are aware that some descriptors address similar image features. If computation time matters, e.g., during a machine learning step, dimensionality reduction techniques like principal component analysis or cross correlation analysis can be applied. In this way, more or less sophisticated machine learning algorithms can be used to generate a model of each spatial area. To this end, all feature can be included in the learning step or several models can be generated, one for each feature. In the latter case, classification performance of the individual models can be used for figuring out the most geo-related visual feature, as shown in Table 8.3. This table contains the results of a nearest neighbour classification for each descriptor and two hierarchy levels (see Sect. 8.2.1). Here, the classification result in terms of accuracy on selected error margins is evaluated per descriptor. Details about the experimental set-up are shown in [12]. This table does not only contain the classification accuracy for different features and different margins of errors, but also for different area sizes (here labelled as ‘Block size’). Block sizes labelled with ‘large’ stands for a spatial level in which the areas are as big as the surface spaned by the meridians and parallels, resulting in a size corresponding to one cell of a grid of 360 by 180 cells. Areas in the spatial level labeled with ‘small’ are only a quarter of that size, that means the world map is segmented in a grid of 720 by 360 cells. This method can iteratively determine the most visually similar spatial area by calculating the Euclidean norm of the respective visual descriptor values.

As can be seen in Table 8.3, the scalable colour descriptor (SCD) consistently outperforms the other descriptors. Although overall textual approaches perform much better than visual approaches, it should be noticed that the best visual model with the scalable colour descriptor achieves an result which is three times more accurate result

Table 8.3 Accuracies on selected error margins (in km) of the visual approach with different descriptors

Feature	Block size	1 (%)	10 (%)	20 (%)	50 (%)	100 (%)	200 (%)	500 (%)	1,000 (%)	2,000 (%)	5,000 (%)
ACC	small	2.3	2.4	2.5	3.3	7.4	12.6	19.9	31.6	43.6	59.7
	large	2.3	2.4	2.5	3.3	7.4	12.6	19.9	31.6	43.6	59.7
CEDD	small	3.2	3.2	3.4	5.1	7.3	11.7	22.1	29.8	44.5	62.9
	large	2.3	2.4	2.5	3.3	7.4	12.6	19.9	31.6	43.6	59.7
CLD	small	1.2	1.3	1.4	2.2	5.9	11.9	18.6	28.5	45.2	60.9
	large	2.3	2.4	2.5	3.3	7.4	12.6	19.9	31.6	43.6	59.7
EHD	small	1.8	2	2.2	3.1	5.2	12	20	30.2	47.3	62.5
	large	2.3	2.4	2.5	3.3	7.4	12.6	19.9	31.6	43.6	59.7
GD	small	1.2	1.3	1.3	2.3	4	7.1	12.5	24.2	37	65.2
	large	2.3	2.4	2.5	3.3	7.4	12.6	19.9	31.6	43.6	59.7
TD	small	0.7	0.7	0.7	1.4	4.7	9.6	15.3	21.6	37	55.9
	large	2.3	2.4	2.5	3.3	7.4	12.6	19.9	31.6	43.6	59.7
SCD	small	5.4	5.6	5.8	6.5	8.6	13.8	24.2	34.9	50.2	63.3
	large	2.3	2.4	2.5	3.3	7.4	12.6	19.9	31.6	43.6	59.7

than the random baseline (12 % at 1,000 km). The nearest neighbour classification can be drastically speeded up by organising the descriptor values in tree structures. A k-d tree has the advantage that the subsequent search for nearest neighbours is speeded up because the search is recursively among the branches of the tree. Finding the nearest neighbour is a $O(\log N)$ operation instead of $O(N)$ in a naive implementation. Here, the problem is the generation a proper 'model' for each area. The simple but fast method of averaging is quite successful, but more sophisticated techniques such as classification methods like support vector machines can be applied, which may enhance the robustness of the model. Applying such a method, a decision $a_{i,j}$ and a probability score $w_{i,j}$ is generated for each geographic areas (i.e., cluster or grid cell) j and each photo or key frame i. Since videos consists of multiple images, these single frames can be classified into different areas a_j, so a subsequent step is necessary to determine the final location decision for the whole sequence. Here, voting leads to a single decision for the complete video sequence. Two voting methods have been previously used are consensus voting and weighted voting. Weighted voting is based on the idea that not all decisions on key frame level are equally accurate. The decision of the predicted area is weighted with the probability score $w_{i,j}$, such that more accurate decisions on key frames give more importance to the final video result. This is in contrast to consensus voting, which assumes that each key frame's decision carries equal weight, i.e., the scores $w_{i,j}$ are effectively ignored. The decision for a spatial area of a video sequence, when using weighted voting, is:

$$a = \arg\max_{j} \sum_{i} w_{i,j} \cdot a_{i,j}, \tag{8.15}$$

where the decision $a_{i,j}$ is set to 1, if key frame i is classified to be in the jth spatial area, otherwise it is 0. The decision rule for consensus voting is obtained by replacing the weights $w_{i,j}$ by 1 in the previous formula.

8.4.2 A Retrieval Approach to Using Visual Information

The retrieval approach is closely related to the field of content-based image retrieval (CBIR). In CBIR visually similar media items are returned for a given query image. These visually similar media items are taken from the training data and are therefore geo-tagged, hence their location can be propagated to the query image or video. We now explain the basic technique for geo-referencing media using their visual content in more detail.

The starting point is an image depicting a outdoor scene for which the geo-coordinates are not known. A CBIR system retrieves images that capture the gist of the query image and returns geo-referenced images of similarly looking outdoor scenes. Those can be used to predict the location of the query image, based on the assumption that similar scenes in images (or key frames from video sequences) correlate to similar geographic areas. In its simplest form, this approach looks for the nearest neighbour of a query item by comparing relatively low-dimensional feature vectors, which is faster than performing a sophisticated classification or object recognition algorithm that has to be computed over many object models. Estimating location using this nearest neighbour method requires a large dataset that covers the entire world. This retrieval approach also has some restrictions; the required database should not only be large, but should also densely cover the world, which is not often the case. As a result, in practice there is a bias towards North American and Europe, and popular tourist destinations due to the large number of photos taken in these areas. However, the greatest limitation is due to visual ambiguity. Images depicting coastal scenes but captured at different places can look very similar. This restricts the ability of approaches based on visual similarity to geo-reference locations that look very different. Georeferencing based on visual features often does not lead to accurate locations, but it can constrain the search space, and may thus be used effectively together with other methods (e.g., based on textual features or context information such as the home location of the owner). However, using visual information is not recommended as a stand-alone approach.

8.5 Centroid-Based Candidate Fusion

We are interested in addressing cases where the training dataset is sparse (e.g., to apply the methods discussed in this chapter to regions of the world where the uptake of Flickr is limited). To this end, we explored the possibility of using unlabelled instances, i.e., photos or videos whose coordinates are unknown. In particular, we used the videos in the test data for this purpose. The Placing Task dataset benefits from a high density of photographs at popular locations and well known places (see Fig. 8.1) across the Earth. Sections 8.3 and 8.4 shows that the retrieval approaches perform very poorly when asked to locate images from undersampled regions. In [8] and [14] it has been shown that the problem of a sparse dataset will not be solved by increasing the size of the dataset. A greater number of training images does not lead to a uniform distribution of the world but mainly to a better description of popular places. Choi et al. [3] proposed a graphical model framework in which geotagging is interpreted as a graph inference problem, and showed that performance improvements can be achieved by smart processing of the test data set. The node potentials in this graph-based framework are modelled as a product of the term location distributions (see Sect. 8.3.1.1), given each tag individually.

Next, we describe a centroid-based candidate fusion method to solve the problem of data sparsity and to enhance the distributions of single candidates in a multimodal manner. The framework also facilitates the fusion of textual and visual features that can further improve the localization performance. One of the biggest problems in the fusion of multimodal features is the different range of features from each of the domains.

This centroid-based candidate fusion approach is based on the sum rule in decision fusion [18]. Since our textual location model produces logarithmic confidence scores Subsequently, these scores are used to generate normalized weights for the candidate fusion:

$$w_n = \frac{P(l_n|d)}{\sum_{i=1}^{N} P(l_i|d)}, \tag{8.16}$$

where w_n is the weight of the nth candidate of a test video d. The equation implies that the sum of weights is 1. Then, the candidate locations GPS($\cdot$) are weighted combined using the sum rule, assuming that the most likely location has the highest weight. The location centroid $\mathbf{x}^{v|t}$ arising from all visual(v) or textual(t) candidates is calculated as follows:

$$\mathbf{x}^{v|t} = \sum_{n=1}^{N} w_n \cdot \mathrm{GPS}(l_n). \tag{8.17}$$

This forms the most likely location for a given video, whereas the centroid does not need to be an existing item within the training data. Then, we determine the location for a specific feature. Since we want to include several different features

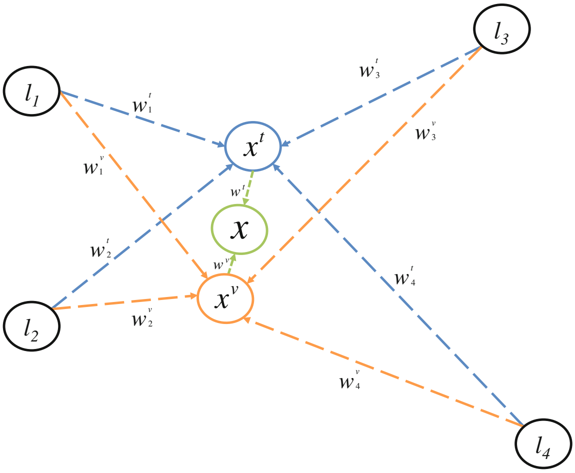


Fig. 8.8 Illustration of fusion of textual and visual candidates

from different modalities in our framework, a confidence score for each calculated centroid is needed for the subsequent multimodal fusion. Here, we choose the standard deviation as inversely proportional weights. The more a feature correlates with a specific spatial location, the closer the likely candidates are located, which implies a small deviation and therefore a high-valued weight. The calculation of the spatial deviation is shown in the following equation:

$$\sigma^{v|t} = \sqrt{\sum_{n=1}^{N} \left(\mathrm{GPS}\,(l_n) - \mathrm{GPS}\,\left(\mathbf{x}^{v|t}\right)\right)^2 \cdot w_n} \tag{8.18}$$

Using these formulas, a centroid is determined for each feature. The final decision for the video location X is specified as shown in Fig. 8.8 using the following multimodal fusion:

$$\mathbf{X} = w^t \cdot \mathbf{X}^t + w^v \cdot \mathbf{X}^v, \tag{8.19}$$

where $w^{v|t}$ are calculated according to (8.16), but with $\frac{1}{\sigma}$ instead of $P(l_i|d)$. Figure 8.9 shows the confidence scores of both modalities for an example video[4] depicting a formula one scene captured in Montreal, Canada. The confidence score is coded in colours as follows; very unlikely areas are depicted in black, where the colour gets lighter with the increasing likelihood of the corresponding areas. As seen, areas around Montreal are more likely than most other areas. The scores of the visual approach using scalable colour as feature as depicted in Fig. 8.9a. Here there are many likely regions in the world: based on the visual information alone, this video

[4] http://www.flickr.com/photos/88878784@N00/4706267893.

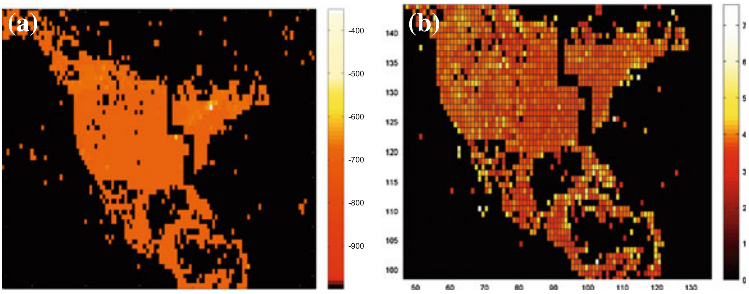


Fig. 8.9 Confidence scores (in log scale) of the textual (**a**) and visual approach (**b**) for North America

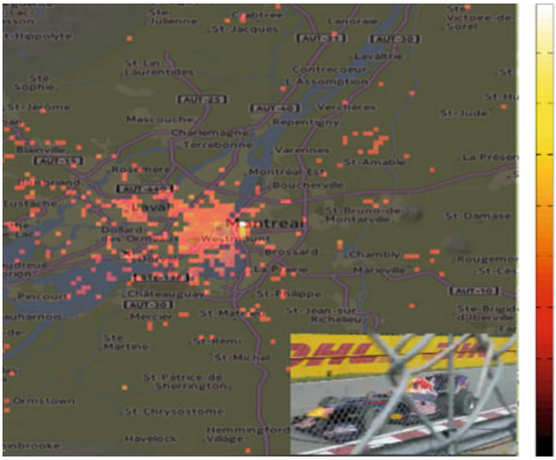


Fig. 8.10 Confidence scores of the visual approach (SCD) restricted to be in the most likely spatial segment determined by the textual approach (tf-idf)

sequence may have been recorded at many locations in the world, hence we need textual metadata to reduce the number of possible candidates.

Figure 8.10 shows such a restriction. The tf-idf text model is based on the hierarchical spatial segmentation method explained in Sect. 8.2.1; it predicts the most likely segment at the highest hierarchy levels and the visual SCD model predicts locations within these segments. As shown, the previous example is correctly assigned to the city of Montreal, Canada. Here, the fusion of textual and visual methods is important to eliminate geographical ambiguities. The candidates of both modalities are combined using our centroid-based fusion as described in Sect. 8.5. The videos of the Placing Task dataset are well tagged, the textual model produced strong candidates and the combination with the visual candidates effect a location gain in small scale. In general, the fusion of several candidates of both modalities is important to eliminate geographical ambiguities. As depicted in Fig. 8.11, the centroid-based fusion improves the results, especially on smaller margins of error, overcoming the sparse nature of the dataset. The strongest gain is achieved by the gazetteer-based national

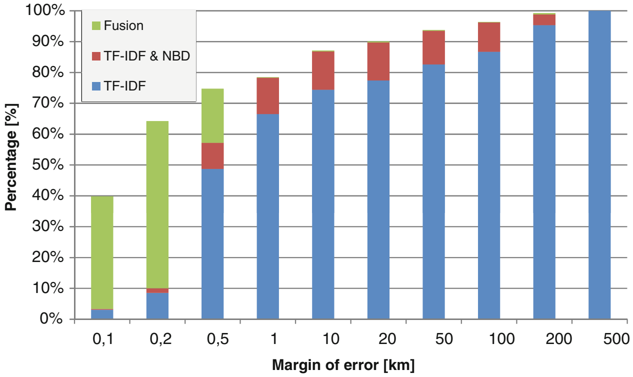


Fig. 8.11 Accuracy achieved by single methods within the hierarchy for selected margins of error

border detection, which eliminates the geographical ambiguity in conjunction with the probabilistic models.

8.6 Conclusion

In this chapter, we discussed a variety of techniques for estimating the geographical location of a Flickr photo or video. We first discussed several methods for clustering the world into a finite number of disjoint geographic areas, as well as the idea of using national borders for this purpose. We also noted that such a spatial segmentation step can be done in a hierarchical fashion, which can have computational advantages (e.g., computationally more expensive methods may be feasible once the range of possible locations has been narrowed down to a particular country, or even to a particular city). Once a set of disjoint geographical areas has been identified, we can treat the problem of georeferencing Flickr photos or videos as a text classification problem, where the geographic areas are the classes and each photo is a document (with its tags as terms). Alternatively, georeferencing can be treated as a retrieval problem. Experimental evidence suggests that optimal performance is achieved by combining both methods, i.e., use a text classification method to find the most likely geographic area, and then use a retrieval method to find the most likely location within that area. Subsequently we discussed the value of visual features. We argued that such features can be very valuable to refine the locations predicted based on textual features, although visual features are not sufficiently powerful to come up with likely locations by themselves (except in very particular cases, such as when the photo contains an easily recognizable landmark). Finally, we discussed a method for centroid-based candidate fusion, which ameliorates the problem of data sparsity and enhances the distributions of single candidates using information from multiple modalities.

References

1. http://www.geonames.org
2. http://code.google.com/apis/maps/index.html
3. J. Choi, G. Friedland, V. Ekambaram, K. Ramchandran, Multimodal location estimation of consumer media: dealing with sparse training data. in IEEE international conference on Multimedia and expo (ICME), 2012 , July, pp. 43–48
4. J. Choi, G. Friedland, V. Ekambaram, K. Ramchandran, Multimodal location estimation of consumer media: dealing with sparse training data. in IEEE international conference on Multimedia and expo (ICME), 2012 , July, pp. 43–48
5. C. Hauff, G. Houben, Wistud at mediaeval, *Placing task* (In MediaEval, CEUR-WS.org, 2011) (2011)
6. C. Hauff, G.-J. Houben, Placing images on the world map: a microblog-based enrichment approach. in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2012) pp. 691–700
7. C. Hauff, B. Thomee, M. Trevisiol, M trevisiol working notes for the placing task at mediaeval 2013. in workshop on MediaEval 2013 (2013)
8. J. Hays, A. Efros, Im2gps: estimating geographic information from a single image. in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. CVPR 2008, pp. 1–8 (2008)
9. P. Kelm, S. Schmiedeke, J. Choi, G. Friedland, V.N, Ekambaram, K. Ramchandran, T. Sikora, A novel fusion method for integrating multiple modalities and knowledge for multimodal location estimation. in Proceedings of the 2nd ACM International Workshop on Geotagging and Its Applications in Multimedia, pp. 7–12, Barcelona, Spain, Oct. 2013. ACM, New York, NY, USA. isbn = 978-1-4503-2391-8 acmid = 2509238
10. P. Kelm, S. Schmiedeke, T. Sikora, A hierarchical, multi-modal approach for placing videos on the map using millions of flickr photographs. in *ACM Multimedia 2011* (Workshop on Social and Behavioral Networked Media Access - SBNMA). ACM, Nov 2011
11. P. Kelm, S. Schmiedeke, T. Sikora, Multi-modal, multi-resource methods for placing flickr videos on the map. in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ICMR '11*, pp. 52:1–52:8, ACM, New York, NY, USA, 2011
12. P. Kelm, S. Schmiedeke, T. Sikora, Multimodal geo-tagging in social media websites using hierarchical spatial segmentation. in *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, LBSN '12, pp. 32–39, ACM, New York, NY, USA, 2012
13. T. Kurashima, T. Iwata, G. Irie, K. Fujimura, Travel route recommendation using geotags in photo sharing sites. in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10*, pp. 579–588, ACM, New York, NY, USA, 2010
14. J. Luo, D. Joshi, J. Yu, A. Gallagher, Geotagging in multimedia and computer visiona survey. Multimedia Tools and Appl. **51**(1), 187–211 (2011)
15. C.D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval* (Cambridge University Press, Cambridge, 2008)
16. R. Purves, C. Jones, Geographic information retrieval. SIGSPATIAL Special **3**(2), 2–4 (2011)
17. S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al., Okapi at trec-3. NIST SPECIAL PUBLICATION SP, pp. 109–109, 1995
18. S. Schmiedeke, P. Kelm, T. Sikora, Cross-modal categorisation of user-generated video sequences. in *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, ICMR '12, pp. 25:1–25:8, ACM, New York, NY, USA, 2012
19. P. Serdyukov, V. Murdock, R. van Zwol, Placing flickr photos on a map. in *Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pp. 484–491, ACM, New York, NY, USA, 2009
20. P. Serdyukov, V. Murdock, R. van Zwol, Placing flickr photos on a map. in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 484–491, 2009

21. M. Trevisiol, J. Delhumeau, H. Jégou, G. Gravier, How INRIA/IRISA identifies Geographic Location of a Video. In *Working Notes Proceedings of the MediaEval 2012 Workshop*, Italy, 2012
22. M. Trevisiol, H. Jégou, J. Delhumeau, and G. Gravier, Retrieving geo-location of videos with a divide & conquer hierarchical multimodal approach. in *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval - ICMR '13*, p. 1, New York, New York, USA, ACM Press, 2013
23. O. Van Laere, J. Quinn, S. Schockaert, B. Dhoedt, Spatially aware term selection for geotagging. IEEE Trans. Knowl. Data Eng. **26**, 221–234 (2014)
24. O. Van Laere, S. Schockaert, B. Dhoedt, Ghent University at the 2010 placing task. in *Working Notes of the MediaEval Workshop*, 2010
25. O. Van Laere, S. Schockaert, B. Dhoedt, Finding locations of Flickr resources using language models and similarity search. in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pp. 48:1–48:8, 2011
26. O. Van Laere, S. Schockaert, B. Dhoedt, Ghent University at the 2011 placing task. in *Working Notes of the MediaEval Workshop*, 2011
27. O. Van Laere, S. Schockaert, B. Dhoedt, Georeferencing Flickr resources based on textual meta-data. Inf. Sci. **238**, 52–74 (2013)
28. J.S. Whissell, C.L.a Clarke, Improving document clustering using okapi bm25 feature weighting. Inf. Retrieval **14**(5), 466–487 (2011)
29. B. P. Wing, J. Baldridge, Simple supervised document geolocation with geodesic grids. in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 955–964, 2011
30. Y. Yang, An evaluation of statistical approaches to text categorization. Inf. Retrieval **1**, 69–90 (1999)
31. Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization. in *Proceedings of the Fourteenth International Conference on Machine Learning*, pp, 412–420, 1997
32. C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems (TOIS) **22**(2), 179–214 (2004)

Chapter 9
Human Versus Machine: Establishing a Human Baseline for Multimodal Location Estimation

Jaeyoung Choi, Howard Lei, Venkatesan Ekambaram, Pascal Kelm, Luke Gottlieb, Thomas Sikora, Kannan Ramchandran and Gerald Friedland

Abstract In recent years, the problem of video location estimation (i.e., estimating the longitude/latitude coordinates of a video without GPS information) has been approached with diverse methods and ideas in the research community and significant improvements have been made. So far, however, systems have only been compared against each other and no systematic study on human performance has been conducted. Based on a human-subject study with 11,900 experiments, this article presents a human baseline for location estimation for different combinations of modalities (audio, audio/video, audio/video/text). Furthermore, this article compares state-of-the-art location estimation systems with the human baseline. Although the overall performance of humans' multimodal video location estimation is better than

J. Choi (✉) · H. Lei · L. Gottlieb · G. Friedland
International Computer Science Institute, Berkeley, CA, USA
e-mail: jaeyoung@icsi.berkeley.edu

H. Lei
California State University, East Bay, CA, USA
e-mail: howard.lei@csueastbay.edu

L. Gottlieb
e-mail: luke@icsi.berkeley.edu

G. Friedland
e-mail: fractor@icsi.berkeley.edu

V. Ekambaram · K. Ramchandran
University of California at Berkeley, Berkeley, CA, USA
e-mail: venkyne@eecs.berkeley.edu

K. Ramchandran
e-mail: kannanr@eecs.berkeley.edu

P. Kelm · T. Sikora
Technische Universität, Berlin, Germany
e-mail: kelm@nue.tu-berlin.de

T. Sikora
e-mail: sikora@nue.tu-berlin.de

J. Choi and G. Friedland (eds.), *Multimodal Location Estimation of Videos and Images*,
DOI 10.1007/978-3-319-09861-6_9

current machine learning approaches, the difference is quite small: For 41 % of the test set, the machine's accuracy was superior to the humans. We present case studies and discuss why machines did better for some videos and not for others. Our analysis suggests new directions and priorities for future work on the improvement of location inference algorithms.

9.1 Introduction

This chapter is based on the work of Choi et al. in [4], and discusses the human baseline of the multimodal video location estimation. Over the recent years, the problem of video location estimation (i.e., estimating the longitude/latitude coordinates of a video without GPS information) has been approached with diverse methods and ideas in the research community and significant improvements have been made. However, approaches have only been compared against each other and there is little intuition on how humans would perform at this task. Some researchers even assume that the automated algorithms would probably always perform better on this task compared to humans. In this paper, we establish a human baseline for video location estimation and present a comparative analysis with automatic location inference systems. The baseline was created by asking qualified humans to perform a total of 9,000 video localizations. With the human baseline in our hand, we are able to analyze different cases of when machines perform better than humans, humans perform better than machines, or when both fail.

This chapter is organized as follows. Section 9.2 provides a brief overview of the existing work in the field and positions our work in comparison to the available literature. Section 9.3 describes the task and the characteristics of the dataset that render the task difficult. Section 9.4 describes the experimental setup for establishing the human baseline using a crowdsourcing platform. Section 9.5 describes our technical approaches to utilizing audio, audio/visual, and audio/visual/textual metadata for automatic location inference. Section 9.6 provides a comparison of the performance of location estimation between machines and humans. In Sect. 9.7, we present case studies and discuss why machines perform better for some videos and not for others. Section 9.8 concludes with a summary of the paper and future research directions based on our analysis.

9.2 Related Works

Crowdsourcing is currently used for a range of applications such as exploiting unsolicited user contributions, for spontaneous annotation of images for retrieval [18], etc. Systematic crowdsourcing platforms, such as Amazon Mechanical Turk, have been used to mass-outsource artificial intelligence jobs [9]. Further, crowdsourcing has also been used for surveying and evaluating user interfaces [12], designs, and other technical approaches.

However, as the name coincidentally implies, platforms such as Mechanical Turk are often best used for mechanical tasks, i.e., tasks that only require simple intuition. Therefore, for a task like the one presented here, where there is a suspicion that humans might perform worse than machines and there is no clear intuition as to how to solve the task, one has to be very careful about how to approach it properly. Apart from [5], there is no previous work on using Mechanical Turk for geo-tagging videos, moreover there seems to be no previous work on how to use Mechanical Turk for a task that is not straightforward to solve.

So far, no human baseline exists for location estimation, and systems have only been compared against each other. At the same time, these systems differ quite dramatically. This paper establishes a human baseline and compares it against two state of the art systems from the literature to draw conclusions for future directions of this research. To ensure the quality of the human baseline, we rely on the qualification methodology described in [5] in combination with redundancy [10].

9.3 Task and Dataset

All experiments described in this article were performed using the dataset distributed for the Placing Task of the 2010 MediaEval benchmark.[1] The Placing Task is a part of the MediaEval benchmarking initiative that requires participants to assign geographical coordinates (latitude and longitude) to each test video. Participants can make use of textual metadata, audio and visual features as well as external resources, depending on the test run.

The dataset consists of 3,185,258 photos and 10,216 videos. All are Creative Commons-licensed and from Flickr.[2] The metadata for each video includes a user-annotated title, tags, and a description among others. The videos are not filtered or selected in any way to make the dataset more relevant to the task, and are therefore likely to be representative of videos selected at random [13]. Figure 9.1 shows the non-uniform distribution of Flickr videos and images due to geographical, economical, and political reasons.

Flickr requires that an uploaded video must be created by the uploader, and thus almost all videos on Flickr are home-video style. The relatively short lengths of each video should be noted, as the maximum length of a Flickr video was limited to 90 s when the dataset was collected. Moreover, about 70 % of videos in this data set have less than 50 s of playtime. Manual inspection of the randomly sampled 150 videos from the dataset shows that if given only the audio and visual contents, 8 % of the videos contained enough information for accurate guesses, and 10 % with rough hints that would lead to city or country-level estimations. The rest of the videos had

[1] http://multimediaeval.org/.

[2] http://www.flickr.com.

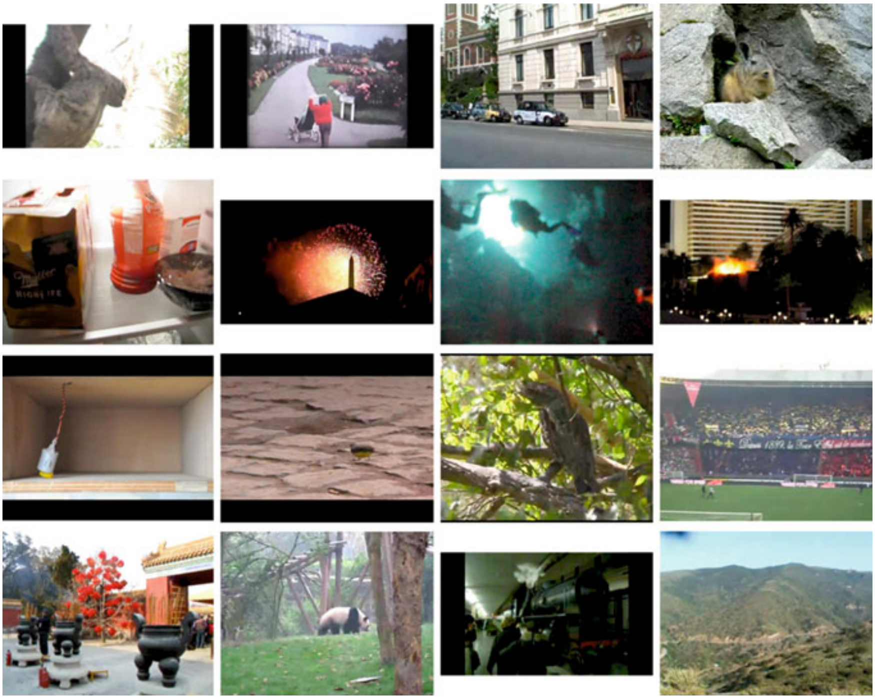


Fig. 9.1 Several frames from the randomly selected videos that were used in the experiment. Most of them are very difficult to specifically geolocate. Figure is from [4]

very little cue for location estimation. Videos of indoor settings comprise 36 % of the videos and about half of them are private space such as one's house or in the backyard. 24 % of the videos contained human speech in various languages including English, Spanish, Swedish, Japanese, etc. The metadata provided by the user often provided direct and sensible clues for location estimation. 98.8 % of videos in the training dataset were annotated by their uploaders with at least one title, tag, or description, which often included location information.

9.4 Establishing a Human Baseline

Collection of human baseline was performed in two steps. In the first step we qualified people, by filtering out incompetent or unmotivated workers and ensuring that the quality of submissions were high enough for the second stage. In the second stage, we collected the human baseline for 1,000 videos. We used Amazon Mechanical Turk as the crowdsourcing platform.

9.4.1 Qualification

The task of location estimation is different from a standard Mechanical Turk task in that it is difficult for both humans and machines, whereas a standard Mechanical Turk task is usually easy for humans and difficult or impossible for machines. There are several notable challenges to finding skilled workers for this task: first, we must find what we term “honest operators,” i.e., people who will seriously attempt to do the task and not just click quickly through it to collect the bounty. Second, we need to develop a meaningful qualification test set that is challenging enough to allow us to qualify people for the real task, but is also solvable by individuals regardless of their culture or location, although English language understanding was required for instructions. For example, in the process of selecting videos, there were videos of tourists in Machu Picchu, which our annotator immediately recognized, however there were no clues to reveal this location that would be usable to someone who had not heard or seen this location previously. These videos were ruled out for the qualification. In the end, ten videos, which we called ‘Ideal10 set’, were carefully chosen and presented to the workers. We created an in-depth tutorial which presented the workers with the basic tools and skills for approaching this challenging task. The workers are allowed to use any applicable resource from the Internet, including Google Maps and Streetview.

Our previous study show that, after the qualification process, workers on the crowdsourcing platform achieved almost equal level of accuracy as internal expert volunteer testers which was composed of highly-educated, well-trained, and motivated researchers. In Fig. 9.2, we have a comparison of the performance results for

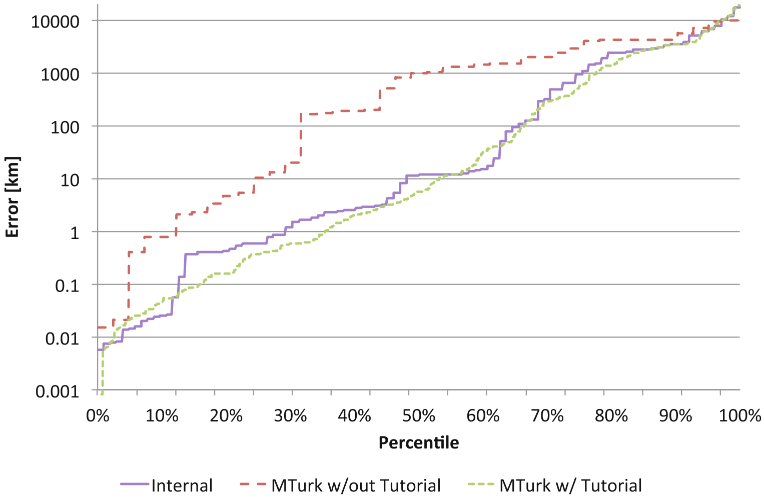


Fig. 9.2 Comparison of performance results for Ideal10 set [5]. Error distance (y-axis) is in a log scale. Figure is from [4]

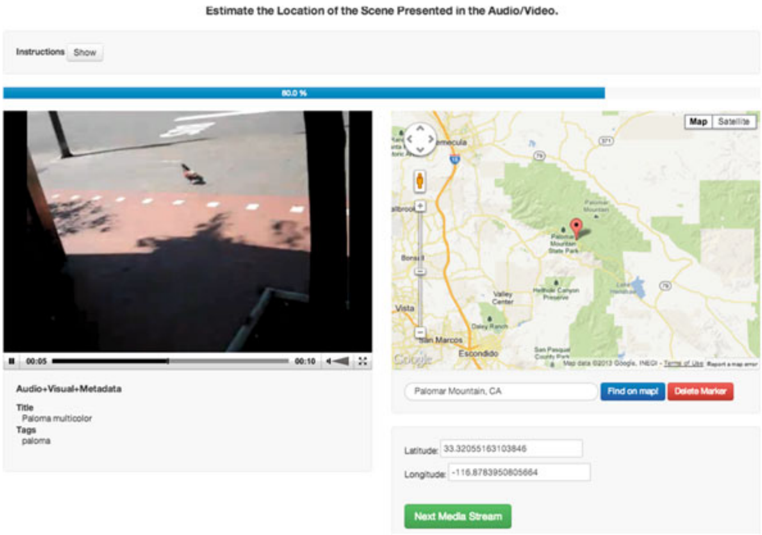
our internal testers, the initial test results of MTurkers, and their results with the tutorial. The internal tester and initial test results were derived by using the classification results from the first round for the videos in the Ideal10 set, as those videos are a subset of our larger annotation. We can see that while the internal testers still perform better than the Mechanical Turk workers, the addition of a tutorial greatly narrowed the performance margin. This led us to conclude that a worker who is adequately trained with a tutorial can be considered reasonably qualified.

However, to select only the highest quality workers, we applied a screening process to filter out not only bad but even workers who performed averagely well. Out of 290 workers on Mechanical Turk who had participated in the qualification task, we qualified only 84 workers (29 % acceptance rate), those who were able to achieve very high accuracy, i.e., were able to put at least 8 of 10 qualification videos within a 5 km radius of the ground truth location. We considered 5 km margin of error as a city-level accuracy. When applied with the same evaluation criteria, internal volunteers showed a similar acceptance rate of 34 %. Additionally, time spent for each human video localization was recorded and often directly correlated with accuracy, giving us a further accuracy indicator for the actual localization test. After successful qualification, we paid USD 1.50 for each HIT (Human Intelligence Task), which consisted of 10 video localizations.

9.4.2 The Web Interface

Here, we describe our user interface which the Amazon Mechanical Turk workers used for the task. We went through several rounds of internal testing and feedback to enhance the usability of the tool.

Figure 9.3 shows the final version of this interface. The instructions on the top of the screen can be expanded and shrunk with a ‘Show/Hide’ button. It was shrunk by default to make the whole interface fit in a normal-sized window to minimize unnecessary scrolling of the screen. A progress bar was shown below the instructions box to let workers know where they are along the progress of a HIT. A video was played automatically once the page was loaded. All videos were re-uploaded to a private file server without the metadata so that simply following the link on the player would not reveal any additional information about the video. A Google Maps instance was placed to the right of the video. A marker would be dropped where the map was clicked, and it could be dragged around the map. The marker’s position was automatically translated to the latitude and longitude and printed to the ‘Latitude’ and ‘Longitude’ boxes. A location search form was placed under the map to aid the search of the location. The form had an auto-completion feature which would help in cases where the worker did not know the exact spelling of the place, etc. At the end of the HIT, we asked participants to leave comments about the HIT. This enabled us to filter out submissions with incidents and other exceptions.


Estimate the Location of the Scene Presented in the Audio/Video.

Audio+Visual+Metadata

Title
Paloma multicolor

Tags
paloma

Palomar Mountain, CA

Latitude: 33.32055163103846

Longitude: -116.8783950805664

Fig. 9.3 Screenshot of web interface used in the Amazon Mechanical Turk experiments described. Figure is from [4]

9.4.3 Collection of Human Intelligence

When we collected the human performance for a different set of modalities, we presented the media with least information to all, i.e., first we gave audio then audio/visual, and lastly audio/visual and textual information (tags). For each HIT, a worker was given five videos with three different media combinations, thus the total of 15 media streams. For the same set of media streams, three identical HITs were generated redundantly for the comparison and to filter out the possible bad results. HITs were assigned with first-come first-served basis to the pool of 84 qualified workers. To distribute the HITs to as many as workers and as evenly as possible, we applied some throttle control to limit the number of HITs that an individual worker can accept. We started the reward for each HIT with USD 0.25 but, based on feedback, quickly increased it to USD 1.50, which resulted in a much faster collection result. Within 18 days, we were able to collect a total of 11,900 localizations (2,900 from qualification, 9,000 from localizing 1,000 videos). Many workers have left comments that the task was challenging, especially for the cases when only the audio was given. At the same time, many of them reported the HIT to be fun and we believe that this motivated people to try to submit better results.

9.5 Machine-Based Location Estimation

In this section, we describe the technical approach and our experimental setup for automatic location inference. We first describe the audio-based approach, then describe how visual feature was added to the modality, and finally, the method that uses all three modalities (audio, visual, and textual metadata).

9.5.1 Audio-Based Location Estimation

We used the city identification system reported in [14] as our machine baseline for location estimation. We describe the main idea of the system as follows. The system involves training a total variability matrix T to model the variability (both city- and channel-related) of the acoustic features of all audio tracks, and using the matrix to obtain a low-dimensional vector characterizing the city where each audio track was from. Specifically, for each audio file, a vector of first-order statistics M—of the acoustic feature vectors of the audio centered around the means of a GMM world model—is first obtained, and can be decomposed as follows:

$$M = m + T\omega \tag{9.1}$$

where m is the GMM world model mean vector, and ω are low-dimensional vectors, known as the identity vectors or i-vectors.

The system then performs Probabilistic Linear Discriminant Analysis (pLDA) [8] and Within-Class Covariance Normalization (WCCN) [6] on the i-vectors. pLDA linearly projects the i-vectors ω onto a set of dimensions to maximize the ratio of between-user scatter to within-user scatter of the i-vectors, producing a new set of vectors. WCCN then whitens the pLDA-projected vectors via a second linear projection, such that the resulting vectors have an identity covariance matrix. For our city identification system, 1,024 mixtures are used for the GMM world model, and a rank of 400 is used for the total variability matrix T, such that the i-vectors ω have 400 dimensions. pLDA projects the i-vectors onto a set of 200 dimensions. The cosine distance is used to obtain the city-similarity score of a pair of i-vectors ω between two audio tracks of user-uploaded videos [19]:

$$\text{score}(\omega_1, \omega_2) = \frac{(A^T\omega_1)^T W^{-1}(A^T\omega_2)}{\sqrt{(A^T\omega_1)^T W^{-1}(A^T\omega_1)}\sqrt{(A^T\omega_2)^T W^{-1}(A^T\omega_2)}} \tag{9.2}$$

where A and W are the LDA and WCCN projection matrices, respectively, and ω_1 and ω_2 are i-vectors from the two audio tracks being compared against. The acoustic features consist of MFCC C0-C19+Δ+$\Delta\Delta$ coefficients of 60 dimensions, computed using 25 ms windows and 10 ms shifts, across 60 to 16,000 Hz.

As we could not train the model to cover all regions of the earth due to the data sparsity, we clustered the distribution of videos into the 40 cities in the training dataset, and reduced the location estimation to a city-identification problem. We trained the system with models for each city using the collection of audio tracks extracted from each city. A video is defined to belong to a city when it is in a 50 km radius of the geographical city center. We then tested the audio tracks extracted from the test videos against the trained models and picked the city with the highest likelihood. For comparability, we converted the city labels to the (latitude, longitude) format with the geo-coordinates of the center of the city. Note, that this creates a slight disadvantage for the machine.

9.5.2 Visual Location Estimation

In order to utilize the visual content of the video for location estimation, we reduce location estimation to an image retrieval problem, assuming that similar images mean similar locations, as in [7]. We used several visual descriptors extracted from sample frames of both query and training videos along with the images given as the Placing Task dataset and ran a k-nearest neighbor search on the training dataset to find the video frame or an image that is most similar. We used Fuzzy Color and Texture Histogram (FCTH) [2], CEDD (Color Edge Directivity Descriptor) [1], and Tamura [20] visual descriptors that were given as a part of the Placing Task dataset. In addition to these descriptors, we extracted Gist features [16] as it was shown to be very effective at scene recognition in [7]. Weighted linear combination of distances was used as the final distance between frames. The scaling of the weights was learned by using a small sample of the training dataset and normalizing the individual distance distributions so that each the standard deviation of each of them would be similar. We used L^2 norm to compare the combination of descriptors and used 1-nearest neighbor matching between the closest pre-extracted frame to the temporal mid-point of a query video and all photos and frames from the videos in the training dataset. In order to handle the large amounts of development data efficiently, we split the reference data set into chunks of 100,000 images, ran 1-NN in parallel on each subset to get intermediate results, and ran 1-NN once again on the intermediate results to get the final nearest neighbor. We used an approximate nearest neighbor library [15] for the experiment.

While videos with soundtrack were shown to the crowd, due to data sparsity, our comparison system did not use the acoustic modality and relied solely on the visual modality.

9.5.3 Multimodal Location Estimation

To integrate textual and visual data, we combined the visual search method with the system reported in [3]. Due to data sparsity, again, while the soundtrack was available to humans, the acoustic modality was not used. The approach based on graphical models is summarized as follows.

Graphical models provide an efficient representation of dependencies amongst different random variables and have been extensively studied in the statistical learning theory community [21]. The random variables in our setup are the geo-locations of the query videos that need to be estimated. We treat the textual tags as observed random variables that are probabilistically related to the geo-location of that video. Figure 9.4 illustrates the idea. The goal is to obtain the best estimate of the unobserved random variables (locations of the query videos) given all the observed variables. We use graphical models to characterize the dependencies amongst the different random variables and use efficient message-passing algorithms to obtain the desired

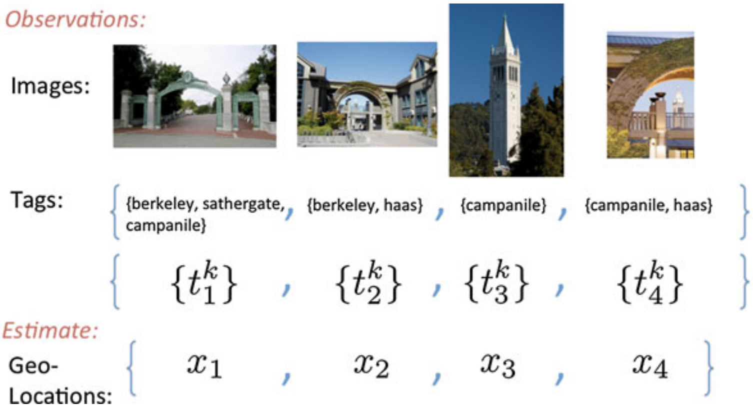


Fig. 9.4 A theoretic viewpoint of the geo-tagging problem. Figure is from [4]

estimates. In order to obtain a graphical model representation for our problem setup, we model the joint distribution of the query video locations given the observed data. We use a simplistic conditional dependency model for the random variables as described below. Each node in our graphical model corresponds to a query video and the associated random variable is the geo-location of that query video. Intuitively, if two images are nearby, then they should be connected by an edge since their locations are highly correlated. The problem is that we do not know the geo-locations a priori. However, given that textual tags are strongly correlated to the geo-locations, a common textual tag between two images is a good indication of the proximity of geo-locations. Hence, we will build the graphical model by having an edge between two nodes if and only if the two query videos have at least one common textual tag. Note that this textual tag need not appear in the training dataset.

Let x_i be the geo-location of the ith video and $\{t_i^k\}_{k=1}^{n_i}$ be the set of n_i tags associated with this video. Based on our model, the joint probability distribution factorizes as follows:

$$p(x_1, ..., x_N | \{t_1^k\}, ..., \{t_N^k\}) \propto \prod_{i \in V} \psi(x_i | \{t_i^k\}) \prod_{(i,j) \in E} \psi(x_i, x_j | \{t_i^k\}, \{t_j^k\}).$$

We now need to model the node and edge potential functions. Given the training data, we fit a Gaussian Mixture Model (GMM) for the distribution of the location given a particular tag t, i.e., $p(x|t)$. The intuition is that tags usually correspond to one or more specific locations and the distribution is multimodal (e.g., the tag “washington” can refer to the State of Washington or Washington D.C., among other locations). To estimate the parameters of the GMM, we use an algorithm based on Expectation Maximization that adaptively chooses the number of components for different tags using a likelihood criterion. Although distribution of the locations given multiple tags is not independent, for this experiment, we start with a naive assumption that different tags are conditionally independent. We take the node potential as follows, $\psi(x_i) \propto \prod_{k=1}^{n_i} p(x_i|t_i^k)$. For the potential functions, $\psi(x_i, x_j|\{t_i^k\}, \{t_j^k\})$, we use a very simple model. Intuitively, if the common tag between two query videos i and j occurs too frequently either in the test set or the training set, that tag is most likely a common word like "video" or "photo" which does not really encode any information about the geographic closeness of the two videos. In this case, we assume that the edge potential is zero (drop edge (i, j)) whenever the number of occurrences of the tag is above a threshold. When the occurrence of the common tag is less frequent, then it is most likely that the geographic locations are very close to each other and we model the potential function as an indicator function:

$$\psi(x_i, x_j|\{t_i^k\}, \{t_j^k\}) = \begin{cases} 1 \text{ if } x_i = x_j, \\ 0 \text{ otherwise.} \end{cases} \tag{9.3}$$

This model is a hard-threshold model and we can clearly use a soft-version wherein the weights on the edges for the potential functions are appropriately chosen.

Further, we propose the following simplification, which leads to analytically tractable expressions for the potential functions and message updates. Given that for many of the tags, the GMM will have one strong mixture component, the distribution $\psi(x_i)$ can be approximated by a Gaussian distribution with the mean $(\tilde{\mu}_i)$ and variance $(\tilde{\sigma}_i^2)$ given by:

$$(\tilde{\mu}_i, \tilde{\sigma}_i^2) = \left(\frac{\sum_{k=1}^{n_i} \frac{1}{\sigma_i^{k2}} \mu_i^k}{\sum_{k=1}^{n_i} \frac{1}{\sigma_i^{k2}}}, \frac{1}{\sum_{k=1}^{n_i} \frac{1}{\sigma_i^{k2}}} \right), \tag{9.4}$$

where μ_i^k and σ_i^{k2} are the mean and variance of the mixture component with the largest weight of the distribution $p(x_i|t_i^k)$. Under this assumption, the iterations of the sum-product algorithm take on the following simplistic form. Node i at iteration m, updates its location estimate $(\hat{\mu}_i(m))$ and variance $(\hat{\sigma}_i^2(m))$ as follows:

$$\hat{\mu}_i(m) = \frac{\frac{1}{\tilde{\sigma}_i^2}\tilde{\mu}_i + \sum_{j \in N(i)} \frac{1}{\hat{\sigma}_j^2(m-1)} \hat{\mu}_j(m-1)}{\frac{1}{\tilde{\sigma}_i^2} + \sum_{j \in N(i)} \frac{1}{\hat{\sigma}_j^2(m-1)}}, \tag{9.5}$$

$$\hat{\sigma}_i^2(m) = \frac{1}{\frac{1}{\tilde{\sigma}_i^2} + \sum_{j \in N(i)} \frac{1}{\hat{\sigma}_j^2(m-1)}}. \tag{9.6}$$

The location estimate for the ith query video $\hat{x}_i$ is taken to be $\hat{\mu}_i(m)$ at the end of m iterations, or when the algorithm has converged. The variance $\hat{\sigma}_i^2(m)$ provides a confidence metric on the location estimate.

So far, only the textual features are used to estimate the location. We performed the same visual search described in Sect. 9.5.2 around the location estimate $\hat{\mu}_i(m)$ with the search boundary set dynamically according to the variance $\hat{\sigma}_i^2(m)$. The intuition is that more we are certain about the location estimate from the graphical framework, we search a narrower range for similar images. Thus, if the variance is low, the search boundary would be set low as well, and vice versa. Since the visual search was limited to much smaller number of images and video frames dynamically adjusted from the confidence of the textual-based location estimate, the results were improved over searching naively across the whole training dataset or searching within the fixed range from the text-based location estimate.

All of the algorithms described above, except for the acoustic system that seems to be unique, achieved among the highest scores in the MediaEval 2012 evaluation and we therefore consider them a state-of-the-art machine baseline.

9.6 Results

To evaluate the performance of both the online workers and the machine, the geodesic distance between the ground truth coordinates and those of the outputs from participants or the machine, respectively, are compared. To take into account the geographic nature of the evaluation, the Haversine [17] distance is used.

Figure 9.5 shows the performance of humans versus the machine given three different combinations of modalities: audio, audio + visual, and audio + visual + text. Human performance was measured by having three different qualified workers redundantly locate the same video. A total number of 1,000 geo-tagged videos were presented in the three different forms (as discussed in Sect. 9.4), resulting in a total number of 9,000 experiments. The results of the qualification experiments are not included in this chart. To obtain a conservative baseline for this chart, the best answer out of the 3 was picked. For representing machine performance, we used the system as described in Sect. 9.5 based on the combination of media tested.

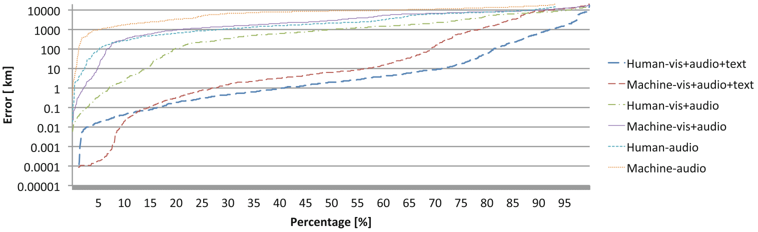


Fig. 9.5 The human baseline of location estimation versus current stat-of-the-art algorithms on all modalities (audio, audio+visual, and audio + visual + text). Figure is from [4]

Overall, humans are better at this task than machines. However, in the very accurate (below 50 m) region, current algorithms outperform humans by about a 12 % margin (number of correctly located videos). The difference between the algorithm and human intelligence is quite low though, since overall only 59 % of the videos are located more accurately by the human than by the algorithm when all modalities are utilized. An unqualified worker base would most certainly have resulted in machine dominance of this task. As a control, choosing the second-best of the three results, resulted in the humans being less accurate than the algorithm in 60 % of the videos. Both the machine and humans do better once more modalities are available. Using only audio, the machine was better for only 16.4 % of the videos. In the visual case, the machine was only better in 25.5 % of the videos.

We qualify the results in the following section.

9.7 Discussion

In this section, we analyze the cases where humans were successful in inferring location while the machine failed and vice versa. We also investigate the videos where humans and computers have both failed to infer location. The threshold for determining the "success" and "failure" was set differently for each of the modalities given. We were more generous when only the audio was given, as it is much more difficult than the two other cases.

9.7.1 Machine Versus Human Using only Audio

For audio only experiments, the threshold to be considered a successful estimate was set at 150 km, and the failure threshold was set at 1,000 km.

9.7.1.1 When Humans Are Better

Out of 1,000 audio-only test videos, there were 62 cases in which the audio provided enough information for humans to infer location while the machines failed. Close analysis of all of these audio tracks revealed that the cases where humans were better than machines can be categorized into the following three classes:

1. Humans were able to identify location based on the kind of language spoken or the distinctive accent or variation of the speakers in the video. With only this information at hand and no other clues, human annotators picked the capital city of the country or region where that language is mainly spoken or was originally from (Paris for French, Glasgow for Scottish accented English, London for British accented English, Lisbon for Portuguese). We have no way to investigate whether the workers were able to use any additional information from understanding the contents of the speech.

2. Humans were able to pick up keywords from the speech that they were able to understand either entirely or at least in part providing a clue sufficient to estimate the location. For example, a Finnish singer saying, “Hello, Helsinki!” at the opening of a concert was the only understandable portion for people who don’t speak Finnish, but this is a sufficient clue for estimating the location.
3. Humans were also able to infer location information from the context of the text spoken. In one video, the speech contained the keywords “California,” “grapes,” “harvest,” and “wine,” which could be inferred to be in the Napa Valley using a Google search or geographical knowledge.

On the machine side, the first category of videos could be localized with an audio-based language or dialect identification system. The second category of videos could be localized as if the textual metadata were given using the keywords extracted from the transcript of the speech obtained from passing the audio track to an automatic speech recognition system. The major challenge would be to deal with the noisy transcript from the “wild” audio. The third kind of inference is the most difficult.

9.7.1.2 When Machines Are Better

While humans were in general better at the task using only the soundtrack, there were 10 cases when machines did reasonably well (estimating location under 100 km, which is the city-level boundary used for the audio-based approach) while all of the human annotators failed. One notable finding from the analysis was that three of these videos were from Prague, CZ. All three videos were from different users and contained different scenes and events, however, a close inspection of the audio revealed that three of the videos contained a musical noise in the background. Also, two of the training videos used to train the model for the city of Prague had music playing in the background. We believe that our city model from the i-vector system picked up the common musical chords that are often played in the touristy locations in Prague.

The system also gave the highest score to Tokyo for a video that shows the Shinkansen train leaving from Tokyo station. Similarly with the above example, we believe that the i-vector system has learned the very specific sound of the Shinkansen train track.

These results are good news for automatic approaches: they show that machine learning approaches can exploit very specific sounds that are hard to spot for humans and use them for location matching.

9.7.1.3 When Both Fail

When a video is edited and the audio track is altered such as when its dubbed with background music, or if there’s no audible speech, both humans and machines usually failed. However, human workers did tend to converge on certain locations for the audio tracks that can be inferred to be a stereotype of a broader category of locations such as "beach," "fireworks," or "farm." For example, for beaches, all three human annotators picked a beach in Los Angeles, CA when the audio track contained the sound of sea gull and the sound of breaking waves, whereas the ground truth was a beach in Liverpool, UK. Two of the annotators picked New York for the audio track that contained the sound of fireworks.

This scheme is actually applicable to machine learning as well. For example, the machine could be trained to classify the scene into a broader category to aid in the estimation of location. Classifying the scene of a beach at night could benefit from using audio as the visual features would not work well due to poor lighting conditions.

9.7.2 Machine Versus Human Using Audio and Video

With the added visual feature, both machines and humans were able to get much better results than using audio alone. For these experiments, the threshold to be considered a successful estimation was set at 50 km, and the failure was set at 1,000 km. We excluded cases where audio-only had already given a sufficient clue for humans to get below the 50 km error range as we could not independently evaluate the effectiveness of the visual feature. We also investigated cases where the audio and visual features complement each other whereas only one modality would have failed.

9.7.2.1 When Humans Are Better

In 179 cases, humans were better than machines. We could separate out about four classes.

1. The majority of cases where human workers perform extremely well (getting under 5 km or even 100 m error) belong to the class that contain textual information in the video. These can be in the form of captions added by the user, signs, or even messages written on buildings or machinery in the video. This category of videos could possibly be located with a video OCR system that extracts textual information.
2. When the visual and audio modalities complement each other but when used separately are less effective, humans are usually better. In other words, multimodal integration in machines is not yet successful. For example, one video showed a TV broadcast of an American Football game with the name of the team and the score shown on the screen. The uploader makes a cheering sound as the game ends and the all three human workers inferred that the uploader is a resident of the winning team's region, which is true.
3. When the scene contains a famous landmark humans perform very well. However, our location estimation system was not specifically trained to recognize landmarks.

4. The atmosphere and context of the scene can be understood by the human workers. For instance, a video taken from a moving car shows the clothing style of the people on the street, the status of the road, and the shape of buildings. In one particular case, all three human workers inferred from this collected information that this could be a specific rural town in India, which was in fact was the right answer.

We did not find reasonable evidence that temporal information of the visual features impacted the location inference of human workers.

9.7.2.2 When Machines Are Better

We found 81 cases where the machine was better than the humans using only visual features. Most of these cases consisted of specific tourist spots where the machine had many training videos the locations are not well known to a lot of people. For example, all three human workers failed to recognize the foggy Machu Picchu (mislabeled as pyramids in Mexico) or a mountain scene of Patagonia (mislabeled as Himalaya or Canadian Rockies).

9.7.2.3 When Both Fail

Most of the cases where both humans and machines failed was when videos were taken indoors such as the inside of a night club (poor lighting, poor audio), videos of babies in a house, and so on. Other cases were some generic scenes such as an unpopular mountain, outskirts of large cities with no landmarks or signs, etc.

9.7.3 Machine Versus Human Using All Modalities

The threshold for success in this case is 5 km as the textual information is very effective at allowing inference of location for both humans and the machine. The threshold for failure is set at 1,000 km. In 39 cases, the machine achieved less than 5 km error while all three qualified human annotators failed to estimate the location with an error of more than 1,000 km. For the opposite case where the human does better than the machine, we found 162 cases.

9.7.3.1 When Humans Are Better

In 162 cases, humans were successful at picking a correct location while the machines failed. Many of the errors were from the system failing to pick up a single keyword that represents the location within the list of tags.

1. We believe the critical advantage in some of these 162 cases was from the misspelling of a tag or that the tag was written in a foreign language (which was not included in the training). Human locators did not have problems with the misspelled words.
2. The bias in the distribution of the training dataset results from the failure of the system to correctly process keywords if they were not seen in the training dataset. Although our system tries to address the problem of sparsity using the graphical framework, it is still bound by the quantity and quality of the data in both the training and test sets. The use of semantic computing-based approaches as done in [11] can be an effective solution in these cases.

9.7.3.2 When Machines Are Better

1. Some of the videos contained multiple tags that were not helpful in inferring location but were repeatedly seen in other videos in the training dataset. For example, "iphone, 3gs, iphone3gs" does not have a specific meaning related to the location in one of the test videos. However, our system was able to pick up the repeated common appearance of these tags in the training data and was able to estimate the location under 0.5 km error. This is due to similar users using the same "language model" when tagging their videos. Keep in mind that test and training set had different sets of users.
2. Humans failed to pick up clues from combination of words when too many tags were given, whereas the machine was able to implicitly incorporate n-grams using the graphical framework.
3. Language barrier: In some cases, the tags were written in a foreign language (not in the training dataset). Worker populations on Amazon Mechanical Turk are mostly English-based, thus the presence of non-English tags presents a language barrier. Some human workers managed to get over this by using translation services such as Google Translate.

9.7.3.3 When Both Fail

There were 26 cases where both the machine as well as all humans failed to get a location even with all possible sources of information present. This is expected because sometimes there are just no useful cues to estimate location. For example, a scene which is a closeup with no distinguishable sounds, and no textual description to indicate location. In the end, we were surprised though that only 2.5 % of all videos fell into this category (where the location was not estimated with under 1,000 km accuracy by either the human or the machine). This indicates a high growth potential for future location estimation research.

9.8 Conclusion and Future Work

In this article, we establish a human baseline for multimodal location estimation of random consumer-produced videos with textual descriptions. Even though algorithms work on low-level statistics, we show that humans outperform the algorithm sometimes and in other cases the algorithm outperforms humans. The difference between human performance and algorithmic performance is so close that we speculate that in a relatively short time, algorithms will become better than the human baseline. Surprisingly enough, only about 2.5 % of the videos could not be located at all. This suggests a huge potential for future research in the field—even though for some of the videos, the algorithm would already pass the “Turing test" as it was already better than humans for 41 % of the videos. The analysis of human versus machine errors suggests complementarity, which implies future work might be very successful when concentrating on interactive systems. For example, for humans the acoustic modality works quite well when language and speech content can be picked up. Machines are better at picking up specific sounds that might be the fingerprint of a location. Similarly, in the visual domain, humans are very good at localization based on written text, signs, architecture, and vehicle styles while machines are very good at finding specific locations that appear often enough in the training data. Humans can remember or search for specific locations based on context, while machines can pick up patterns of tags or textual descriptions that indicate the location of a specific social group unknown to most humans. In summary, all three modalities (audio, video, and text) are very powerful at determining location both for humans and the machine, even though recent research mostly has concentrated on text-based systems with the aid of visual information. Future research on location estimation might gain improved results from including optical character recognition and sign interpretation as well as language identification and language-independent keyword spotting. The use of semantic information from Gazetteers and other knowledge bases will be very helpful for locations with limited training data.

References

1. S. Chatzichristofis, Y. Boutalis, CEDD: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval, *Computer Vision Systems* (Springer, Berlin, 2008), pp. 312–322
2. S. Chatzichristofis, Y. Boutalis, Fcth: Fuzzy color and texture histogram-a low level feature for accurate image retrieval, in *Ninth International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS’08*, pp. 191–196. IEEE (2008)
3. J. Choi, G. Friedland, V. Ekambaram, K. Ramchandran, Multimodal location estimation of consumer media: dealing with sparse training data, in *2012 IEEE International Conference on Multimedia and Expo (ICME)*. pp. 43–48, IEEE (2012)
4. J. Choi, H. Lei, V. Ekambaram, P. Kelm, L. Gottlieb, T. Sikora, K. Ramchandran, G. Friedland, Human vs machine: Establishing a human baseline for multimodal location estimation, in *Proceedings of the 21st ACM International Conference on Multimedia, MM ’13*, pp. 867–876. ACM, New York, USA (2013)

5. L. Gottlieb, J. Choi, G. Friedland, P. Kelm, T. Sikora. Pushing the limits of Mechanical Turk: qualifying the crowd for video geo-location, in *Proceedings of the 2012 ACM Workshop on Crowdsourcing for Multimedia (CrowdMM)* (2012)
6. A. Hatch, S. Kajarekar, A. Stolcke, Within-class covariance normalization for SVM-based speaker recognition, in *Proceedings of ISCA Interspeech*, vol. 4 (2006)
7. J. Hays, A. Efros, IM2GPS: estimating geographic information from a single image, in *IEEE CVPR 2008*, pp. 1–8 (2008)
8. S. Ioffe, Probabilistic linear discriminant analysis, *Computer Vision-ECCV* (Springer, Berlin, 2006), pp. 531–542
9. P.G. Ipeirotis, Analyzing the Amazon Mechanical Turk marketplace. XRDS **17**(2), 16–21 (2010)
10. D. Karger, S. Oh, D. Shah, Budget-optimal crowdsourcing using low-rank matrix approximations, in *49th Annual Allerton Conference Communication, Control, and Computing (Allerton) 2011*, pp. 284–291, September 2011
11. P. Kelm, S. Schmiedeke, T. Sikora, A hierarchical, multi-modal approach for placing videos on the map using millions of Flickr photographs, in *Proceedings of SBNMA '11*, pp. 15–20. ACM, New York, USA (2011)
12. A. Kittur, E. H. Chi, B. Suh, Crowdsourcing user studies with Mechanical Turk, in *Proceedings of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, CHI '08*, pp. 453–456. ACM, New York, USA (2008)
13. M. Larson, M. Soleymani, P. Serdyukov, S. Rudinac, C. Wartena, V. Murdock, G. Friedland, R. Ordelman, G. J. Jones, Automatic tagging and geo-tagging in video collections and communities, in *ACM International Conference on Multimedia Retrieval (ICMR 2011)*, pp. 51:1-51:8, April 2011
14. H. Lei, J. Choi, G. Friedland, City-Identification on Flickr Videos Using Acoustic Features. Technical report, ICSI Technical Report TR-11-001, 2011
15. D.M. Mount, S. Arya, ANN: A library for approximate nearest neighbor searching, in *CGC 2nd Annual Fall Workshop on Computational Geometry*, pp. 153 (1997)
16. A. Oliva, A. Torralba, Building the gist of a scene: the role of global image features in recognition. Prog. Brain Res. **155**, 23–36 (2006)
17. M.C. Palmer, Calculation of distance traveled by fishing vessels using GPS positional data: a theoretical evaluation of the sources of error. Fish. Res. **89**(1), 57–64 (2008)
18. B. Russell, A. Torralba, K. Murphy, W. Freeman, LabelMe: a database and web-based tool for image annotation. Int. J. Comput. Vision **77**, 157–173 (2008). doi:10.1007/s11263-007-0090-8
19. M. Soufifar, M. Kockmann, L. Burget, O. Plchot, O. Glembek, T. Svendsen, iVector approach to phonotactic language recognition, in *Proceedings of Interspeech*, pp. 2913–2916 (2011)
20. H. Tamura, S. Mori, T. Yamawaki, Textural features corresponding to visual perception. IEEE Trans. Syst. Man Cybern. **8**(6), 460–473 (1978)
21. M. Wainwright, M. Jordan, Graphical models, exponential families, and variational inference. Found. Trends Mach. Learn. **1**, 1–305 (2008)

Chapter 10
Personalized Travel Navigation and Photo-Shooting Navigation Using Large-Scale Geotags

T. Yamasaki, A. Gallagher, T. Chen and K. Aizawa

Abstract In this chapter, a geotag-based inter/intra-city travel and photo-shooting navigation system that considers both the personal preference and the seasonal/temporal popularity is presented. For the inter-city travel navigation, similarity among users is efficiently calculated by combining our visit pattern similarity and photo shooting pattern similarity. Accurate intra-city travel navigation is achieved by incorporating the seasonal and temporal information into a Markov model. Photo-shooting navigation by using large-scale geotagged photos is also presented. The effectiveness of the proposed algorithms have been experimentally demonstrated by using more than millions of geo-tags and photos downloaded from Flickr.

10.1 Introduction

Travel planning is fun but sometimes troublesome especially when the travelers are not familiar with the city. Therefore, we often refer to travel guidebooks such as Verlag Karl Baedeker,[1] the Michelin Guide,[2] and Lonely Planet.[3]

[1] http://www.baedeker.com.
[2] http://www.michelintravel.com.
[3] http://www.lonelyplanet.com.

T. Yamasaki (✉) · K. Aizawa
Department of Information and Communication Engineering,
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
e-mail: yamasaki@hal.t.u-tokyo.ac.jp

K. Aizawa
e-mail: aizawa@hal.t.u-tokyo.ac.jp

A. Gallagher · T. Chen
School of Electrical and Computer Engineering, Cornell University,
Ithaca, NY 14853-7501, USA
e-mail: andrew.c.gallagher@gmail.com

T. Chen
e-mail: tsuhan@ornell.edu

J. Choi and G. Friedland (eds.), *Multimodal Location Estimation of Videos and Images*,
DOI 10.1007/978-3-319-09861-6_10

In the Internet era, we can also check user-generated travel assistance WEB sites such as Yelp,[4] TripAdvisor[5] and Yahoo! Travel.[6] However, most of the travel guides mentioned above only provide a ranking list of the landmarks. Namely, they do not provide how (in what order) we should travel around those landmarks and they do not consider the travelers' personal preference, either. In addition, it is often difficult to know detailed travel information such as what the best season or the best time of the day is to visit a certain landmark.

In recent years, travel recommendation and navigation using large-scale user-generated photos has been a hot topic because such photos contain rich meta data such as tags, time, and geo-locations (or geo-tags). Although there are also a lot of photos without such meta data, location of the photos could be estimated using the location estimation techniques in [9, 13, 15, 24, 32, 34] and those introduced in this book.

In this chapter, we propose both travel and photo-shooting navigation based on statistical analysis of such large scale user-generated photos and accompanying meta data. As a result, we can navigate users considering the personal preference and seasonal/temporal information. For the inter-city navigation, collaborative filtering based similar traveler extraction is presented. The similarity among the travelers is calculated by considering several user behavior patterns such as visiting patterns and photo-shooting patterns. For the intra-city travel navigation, the Markov-based behavior model considering seasonal/temporal information is demonstrated. Regarding the photo-shooting navigation, "where to go" and "how to shoot" is recommended based on the location-specified similar image retrieval.

The experiments for the travel navigation were conducted using 6.2 million geo-tag data from 21 famous cities and parks in the world collected from Flickr.[7] Experimental results show that the proposed model drastically outperforms the previous travel navigation systems. In addition, we demonstrate some photo-shooting navigation examples using 2.2 million geo-tags and photos in three cities.

The rest of this chapter is organized as follows. Section 10.2 reviews recent related works. The algorithms for our navigation systems are summarized in Sects. 10.3 and 10.4. The experimental results are presented in Sect. 10.5, followed by concluding remarks in Sect. 10.6.

10.2 Related Works

10.2.1 Travel Navigation

GPS-embedded cell phones and cameras and the popularity of photo sharing sites have enabled us to analyze how people travel around the globe. Geotagged

[4] http://www.yelp.com.

[5] http://www.tripadvisor.com.

[6] http://travel.yahoo.com.

[7] www.flickr.com.

photos are the rich source of information because they contain not only spatial and temporal information of the users but also text tags. It is also possible to analyze the preference of the users by analyzing the number of pictures taken at certain locations. In other words, the user generated geo-tag information can be regarded as human sensors [14]. For instance, it has been demonstrated that Italian travelers and American travelers tend to take different travel routes in Rome [14]. This investigation shows the potential of analysis of the large-scale geotagged data.

By analyzing the geo-tags and the text tags, it is possible to automatically extract landmarks of the city [7, 12, 21, 31, 38], rank the landmarks [18], detect particular events [21, 30], detect object of interest [30], extract representative photos of the landmarks [12, 20, 21], detect panoramic view spots [29], and so on.

Travel route recommendation is one of such applications. Ying et al. [37] recommended travel routes depending on how many hours users can spend in the landmark or the city. [10] summarized the geo-tags of Flickr photos and solved the intra-city travel recommendation as a orienteering problem. Kurashima et al. [22] proposed a probabilistic behavior model by combining topic models and Markov models. Personal preference was implicitly included in the topic model. The system could also suggest travel routes considering user-specified time periods. Arase et al. [2] categorized the geotagged photos into six trip patterns such as landmark visiting, enjoying nature, daily trips in local area, and so on. Users could browse typical photos of the six categories to decide where to go. Cheng et al. [8] analyzed faces in the photos and showed that the travelers' attributes such as gender, age, and race can facilitate better travel route recommendation. Such techniques can also be combined with automatic travel guide generation systems [6] and drive navigation systems considering the scenic attractiveness [39].

There are only a few approaches on inter-city travel recommendation as far as we know [11, 19, 26, 28]. The inter-city recommendation is to recommend places to visit in the city A to those who have traveled in the city B (but never been to the city A). [11, 26] analyzed travelers' similarity by collaborative filtering. Popescu et al. [28] calculated the travel pattern similarity between users based on the kernel convolution using the raw geo-tag data. Jiang et al. [19] introduced more features such as similarity of the tags and visual similarity of the photos between users.

10.2.2 Photo-Shooting Navigation

Obtaining good photos as a memory of the travel has been an important but at the same time difficult problem for us. In fact, we can find a lot of photo-shooting guidebooks in bookstores.

Attractiveness enhancement of photos is one of the solutions and has been one of the main topics of image processing and computer graphics such energy-based impaing/retouching [3, 4, 33], and compositing [1]. Example-based processing using large-scale photos on the Internet [16] is another hot topic in image manipulation. Photo quality assessment [5, 36] and photo editing based on the quality measure [23, 25, 27] can also be used for obtaining good photos.

However, these approaches can be used for post-processing photos, only after coming back from the travel. On the other hand, navigating "where to go" and "how to take" has not been investigated deeply so far.

10.2.3 Contribution of this Work

The contribution of this work is as follows: (1) For the inter-city recommendation, similarity between users is calculated by considering visit pattern similarity, photo shooting pattern similarity and photo preference pattern similarity. In addition, seasonal information is considered. (2) For the intra-city recommendation, seasonal and temporal information is considered and incorporated into the Markov model for better travel recommendation. (3) Photo-shooting navigation such as where to go and how to shoot in order to obtain good photos is addressed by using large-scale geotagged photos on the Internet. Note that this chapter is the extended version of [35].

10.3 Travel Navigation

10.3.1 Inter-city Travel Navigation

The purpose of the inter-city recommendation is to use the user's previous travel patterns in certain cities in order to recommend "must-see" landmarks when the user is visiting another city. The assumption is that the travelers who had similar travel patterns with the current user in a certain city would also have similar travel patterns in another city. The user's previous travel patterns are useful for the inter-city recommendation because the travel patterns implicitly indicates the user's preference. In this chapter, collaborative filtering based approach is used. Here, let us assume that the user has visited the city c previously and wants to visit the city d. In the collaborative filter, travelers who have traveled both in the cities c and d are collected. And those who had similar travel patterns in the city c with the current user are extracted. Then, the landmarks in the city d are re-ranked by their travel patterns. Therefore, how to calculate the similarity between users is a key issue. In this chapter, two similarity measures between users are defined as follows.

Visit pattern similarity
The visit pattern similarity considers whether the user has visited certain landmarks only. Therefore, the vector for the user i in the city c is defined as:

$$\mathbf{v}_i^c = (v_{i1}^c, v_{i2}^c, \ldots, v_{iL^c}^c) \tag{10.1}$$

where L^c is the number of landmarks in the city c. $v_{il}^c = 1$ if the user has visited the landmark l and $v_{il}^c = 0$ if not. Then, the similarity between the user i and the user j

is defined as the cosine similarity:

$$\mathrm{Sim}_v^c(\mathbf{v}_i^c, \mathbf{v}_j^c) = \frac{\mathbf{v}_i^c \cdot \mathbf{v}_j^c}{|\mathbf{v}_i^c||\mathbf{v}_j^c|}. \tag{10.2}$$

Photo shooting pattern similarity

The photo shooting pattern considers how the user liked the landmarks, which is modeled as a function of the number of pictures taken at the landmarks:

$$\mathbf{p}_i^c = (p_{i1}^c, p_{i2}^c, \ldots, p_{iL^c}^c) \tag{10.3}$$

$$p_{il}^c = log(N_{il}^c + 1) \tag{10.4}$$

where N_{il}^c is the number of photos at the landmark l in the city c taken by the user i. The similarity is calculated as:

$$\mathrm{Sim}_p^c(\mathbf{p}_i^c, \mathbf{p}_j^c) = \frac{\mathbf{p}_i^c \cdot \mathbf{p}_j^c}{|\mathbf{p}_i^c||\mathbf{p}_j^c|}. \tag{10.5}$$

The total similarity between the user i and j is calculated by the weighted sum of the visit pattern similarity and the photo shooting pattern similarity:

$$\mathrm{Sim}_{\text{total}}^c(u_i^c, u_j^c) = \alpha \mathrm{Sim}_v^c(\mathbf{v}_i^c, \mathbf{v}_j^c) + (1-\alpha)\mathrm{Sim}_p^c(\mathbf{p}_i^c, \mathbf{p}_j^c) \tag{10.6}$$

where u_i^c represents the user i in the city c.

In this chapter, the travelers whose total similarities are greater than a certain threshold Sim_{th} are regarded as similar travelers. After extracting the similar travelers by Eq. 10.6 from the city c, the score for each landmark in the city d is calculated as follows:

$$Score_{k,s}^d = \sum_m \left(\alpha \mathbf{v}_{m,s}^d + (1-\alpha)\mathbf{p}_{m,s}^d\right) \tag{10.7}$$

where $Score_{k,s}^d$ represents the score for the kth landmark in the city d, which considers the seasonal popularity. $\mathbf{v}_{m,s}^d$ and $\mathbf{p}_{m,s}^d$ represent the visit pattern score and the preference pattern score of the mth user in the city d in the season s, respectively. Here, the users are those who had the similarity greater than Sim_{th} with the user i in the city c as discussed above. Then, the ranking of the landmarks in the city d is calculated based on Eq. 10.7. Note that the score for the landmarks in Eq. 10.7 is dependent on the user and the season. Therefore, the landmarks to visit in the city d are recommended by analyzing similar travelers' travel pattern and seasonal attractiveness.

Different from intra-city recommendation, the temporal information is not considered because the inter-city recommendation is for generating a list of landmarks to visit, not the order of landmarks to visit. Once the user actually starts traveling in the city d, the route can be recommended by the intra-city recommendation.

10.3.2 Intra-city Travel Navigation

Let us assume that we are traveling in NYC. For instance, Rockefeller center is one of the most popular landmarks in NYC, but it becomes particularly popular at night in winter. Because people want to take photos of the Christmas tree, the statue of Prometheus with the ice skate link, and clear night view from the top of the rock. It is also recommended to visit the statue of liberty before Rockefeller center because the statue of liberty is closed at night. These are just intuitive examples of how our algorithm ca help users to find the most appealing landmark in the city. Small festivals or events are not usually introduced in travel guidebooks. Some famous buildings might be closed at night but it might be possible to take pictures from outside. Therefore, a travel recommendation system that can automatically extract temporal/seasonal popularity is required to handle these issues. And also, the system can give us an important information when to visit to enjoy the landmark the best.

Similar to Cheng's model [8], we generate a touring model using a Markov model. Instead of using users' profile such as gender and race as in [8], we introduce seasonal and temporal information ($\mathbf{S}_u$).

$$L^* = \arg\max_{L_j} P(L_j|\mathbf{S}_u, L_i) \tag{10.8}$$

$$\mathbf{S}_u \in (s, t) \tag{10.9}$$

where s is the season and t is the time of the day. Namely, the next destination is dependent only on the current user's location and the other landmarks the user previously has visited do not matter. This model is practical because it can consider the "temporal distance". For instance, even when the landmarks A and B are geometrically far away, there might be a public transportation and would take only a few minutes. In such a case, the transition probability from the landmark A to the landmark B would become high. Although the Markov model can suggest only the next place to visit, it can be used for travel route recommendation because we can repeat the recommendation by eliminating already visited landmarks. In fact, the Markov model-based approaches are often used for the travel recommendation as in [8, 22]

The simplest approach is to separate the geotagged data into different seasons and time ranges and generate the transition probability matrices from the current location L_i to the next location L_j. This transition model works fine if the number of travelers in the city is large enough such as in NYC. On the other hand, in smaller cities, the transition matrix tends to be sparse and might include only a few (or no) travelers in a particular season and time. For instance, when we divide the data into four seasons and four time spans (i.e., dividing a day into 6 h bins: morning, afternoon, night, late at night), the travelers would be divided into 16 different transition matrices, which would prevent proper recommendation.

Another solution is to use the Bayes' theorem as in [8]. The probability that the location L_j to be recommended when the user u with the seasonal and temporal information $\mathbf{S}_u$ is at the location L_i is described as:

$$P(L_{i\rightarrow j}|\mathbf{S}_u) = \frac{P(L_{i\rightarrow j}, \mathbf{S}_u)}{P(\mathbf{S}_u)} \tag{10.10}$$

$$= \frac{P(L_{i\rightarrow j})P(\mathbf{S}_u|L_{i\rightarrow j})}{P(\mathbf{S}_u)} \tag{10.11}$$

$$= \frac{P(L_i)P(L_j|L_i)P(\mathbf{S}_u|L_{i\rightarrow j})}{P(\mathbf{S}_u)} \tag{10.12}$$

Therefore, the Eq. (10.8) will become

$$L^* = \arg\max_{L_j} P(L_j|L_i)P(\mathbf{S}_u|L_{i\rightarrow j}) \tag{10.13}$$

because $P(L_i)$ and $P(\mathbf{S}_u)$ are independent of L_j. If we assume the independence between s and t, the joint probability $P(\mathbf{S}_u|L_{i\rightarrow j})$ can be rewritten as:

$$L^* = \arg\max_{L_j} P(L_j|L_i)P(s|L_{i\rightarrow j})P(t|L_{i\rightarrow j}) \tag{10.14}$$

$P(L_j|L_i)$, $P(s|L_{i\rightarrow j})$, and $P(t|L_{i\rightarrow j})$ can be estimated from the training data:

$$P(L_j|L_i) = \frac{\text{count}(L_{i\rightarrow j})}{\sum_{j\in\mathbf{L}} \text{count}(L_{i\rightarrow j})} \tag{10.15}$$

$$P(s \text{ or } t|L_{i\rightarrow j}) = \frac{\text{count}(L_{i\rightarrow j} \cap \mathbf{S}_u = s \text{ or } t)}{\text{count}(L_{i\rightarrow j})} \tag{10.16}$$

where $\text{count}(L_{i\rightarrow j})$ the total number of travelers who traveled from the location L_i to L_j, and $\text{count}(L_{i\rightarrow j} \cap \mathbf{S}_u = s \text{ or } t)$ is that in a specific season and time of the day.

The model we propose in this chapter is a naive transition model:

$$L^* = \arg\max_{L_j} P(L_j|L_i)P(s|L_{i\rightarrow j})P(t|L_{i\rightarrow j}) \tag{10.17}$$

Therefore, the the transition model is divided into a general model, seasonal transition model, and a temporal transition model.

Note that our seasonal and temporal modeling is orthogonal to the previous Markov model-based approaches [8, 22] and thus can be integrated into them, as well.

10.4 Photo-Shooting Navigation

The travel guides usually tell us what we can see when we visit the landmarks. On the other hand, when we want to take photos of landscape of the city, for instance, we need to look for such landmarks by reviewing all the landmarks in the guidebook.

In addition, taking "good" photos is another difficult problem for the travelers. Therefore, we propose a photo-shooting recommendation system that can navigate the users where to go and how to take in order to shoot photos they really want to take.

Assume that the user is now in the city c has a landscape photo which was either taken by the user himself/herself or retrieved from the Internet by the keyword search. Here, we do not care about where the landscape photo was taken (It might have been taken in the city c or in another city). Then, we extract the GIST descriptor from the landscape image. GIST descriptors are extracted for the photos taken in the city c, as well. Since the GIST descriptor is good at retrieving similar scene/composition images, we can retrieve similar landscape images taken in the city c and show the locations of the retrieved photos on the map. This scenario is applied not only to landscape photos, but also to buildings, streets, crowds, nature, and so on. In this manner, the system can navigate where to go to take photos of a certain taste.

The example above is photo-to-photo-similarity-based recommendation. Similarly, user-to-user similarity-based recommendation can also be achieved. When the user inputs several photos they like, regardless of where they were taken, we can retrieve the users who have taken similar photos in the city c and visualize where such similar users visit and what kind of pictures they take.

It sometimes happens that the photos we have taken at landmarks are not as good as those on the guidebooks. Therefore, we also present a "how to shoot" navigation system using large-scale photos on the Internet. Once the user's location is identified, the photos taken around that area are retrieved and sorted by the order of photo attractiveness. To evaluate such attractiveness, we use a model in [17]. Reference [17] demonstrated that the attractiveness of the photos can be roughly estimated as a function of the number of views, the number of favorites, and the number of days the photo is reveled on the Internet as follows:

$$\text{(Attractiveness score)} = \frac{(\text{\# of favorites})^2}{(\text{\# of views})} + \gamma((\text{\# of views}) - m) \quad (10.18)$$

$$m = (\text{average number of views of the photos uploaded on the same day}) \quad (10.19)$$

The advantage of this model is that the attractiveness score can be estimated only from the meta data. When the users want to use more accurate attractiveness evaluation, image-based approaches introduced in 10.2.2 can be used.

10.5 Experimental Results

10.5.1 Experimental Setup

The datasets used in the travel navigation were collected by crawling Flickr using its public API by ourselves. The cities crawled were 21 cities and parks in the world (Boston, Chicago, New York, Philadelphia, San Diego, San Francisco, Seattle, Toronto, Washington D.C., Yosemite, Yellowstone, Niagara, Honolulu, Las Vegas, Taipei, Dallas, Praha, Kyoto, Vancouver, Firenze, Brisbane). All photos were taken between January 1, 2004 and June 31, 2011. The geo-tag information was clustered by using mean shift by following previous papers [12, 22] to find landmarks effectively. The bandwidth was set as 100m. The landmarks with less than 100 unique users were eliminated. As a result, the data consists of 6,253,865 photographs and their associated meta data, which were taken by 219,390 unique users.

We evaluated the performance by using all the landmarks, the top 25 landmarks, and the top 10 landmarks. For instance, if the user traveled

$$L_1^5 \rightarrow L_2^{24} \rightarrow L_3^1 \rightarrow \cdots, \tag{10.20}$$

and if we look at only the top 10 land marks, the travel route is regarded as

$$L_1^5 \rightarrow L_2^1 \rightarrow \cdots, \tag{10.21}$$

where L_i^k means the popularity of the ith location is ranked at the kth place. When a user is traveling from the landmark A to the landmark B, the user tends to visit other landmarks on the way to B even if they are not very popular (i.e., Charging Bull on the way from Ground Zero to Battery park in New York). This process neglects such minor landmarks.

The recommendation was evaluated by the leave-one-user-out method. Namely, the recommendation model was generated for each user using all the other travelers' history in each city.

For photo-shooting navigation, we have collected 2.2 million geotagged photos from Flickr, which were taken in New York, Paris, and Tokyo. The photos were taken from January 1, 2008 to December 31, 2012. Since it is difficult to evaluate the photo-shooting recommendation accuracy, only the results are shown below.

10.5.2 Inter-city Travel Navigation

For the inter-city recommendation, the users who have traveled two or more cities (out of the 21 cities listed above) and have visited two or more landmarks in each city were extracted. In addition, the city/park pairs that contained at least 500 travelers were considered. As a result, 17,016 users and 36 city/park pairs were detected.

The accuracy was evaluated by the mean average precision (mAP) of the recommended landmarks:

$$mAP = \frac{1}{Q}\sum_{q=1}^{Q} AP_q \tag{10.22}$$

$$AP_q = \frac{1}{P_q}\sum_{p=1}^{P_q} \frac{p}{\text{(the order of } p\text{th hit)}} \tag{10.23}$$

where Q is the number of travelers, AP_q is the average precision of the qth traveler, and P_q is the number of landmarks the traveler q visited. Namely, the order of landmarks to visit was not considered and only the ranking of the recommended landmarks are evaluated. Sim_{th} and α were set to 0.1 and 0.6, respectively, in this experiment.

The mean average precision values are shown in Fig. 10.1. Our proposed method is compared with two baselines: the simple popularity-based recommendation, which is generated solely from the popularity ranking, and seasonal-popularity-based recommendation, which is based on season-aware landmark popularity ranking. It is demonstrated that our proposed inter-city recommendation model is better than the simple popularity-based recommendation and seasonal-popularity-based recommendation when top 10/25 landmarks were considered. For instance, mAP is better by more than 0.02 when the top 10 landmarks are considered. It is observed that the proposed model with the seasonal information is worse than that without the seasonal information. It is because we do not have enough number of travelers in constructing the model. It is expected that the proposed model with the seasonal information would be improved if more travelers' data are accumulated. On the other hand, our proposed model gets worse than the baseline methods when we consider all the landmarks. This is also because of the lack of the number of travelers. When we increase the dimension of the similarity pattern vector (the number of landmarks), the similarity score would become sensitive to noise especially when the number of travelers is small.

The impact of changing Sim_{th} and α is demonstrated in Fig. 10.2. Sim_{th} should be 0–0.2, which indicates that the similarity-based thresholding does not contribute to better recommendation. However, we think that this is because the number of similar travelers becomes too small when we set Sim_{th} larger. It is advised to set α as 0.4–0.8, showing that visit pattern similarity is more informative than photo shooting pattern similarity.

10.5.3 Intra-city Travel Navigation

For the intra-city recommendation, all the 6,253,865 geo-tags by 219,390 unique users were used. The performance was evaluated by measuring how accurately the system estimated the next location L_j when the user was at the location L_i. As described in Sect. 10.5.1, the recommendation accuracy was calculated by the leave-one-user-out method. Namely, one user is used as test data and the other users' travel history were used to build the model. In the experiment, we compared our approach with the following probabilistic models:

- Multinomial model: predicts the next landmark based on its popularity. The most popular landmark except for the current and already visited ones is recommended. This model does not consider the user's current location.
- Markov model: recommend the next landmark based on the user's current location using $P(L_j|L_i)$. This model considers the user's current location but does not consider the seasonal or temporal information.

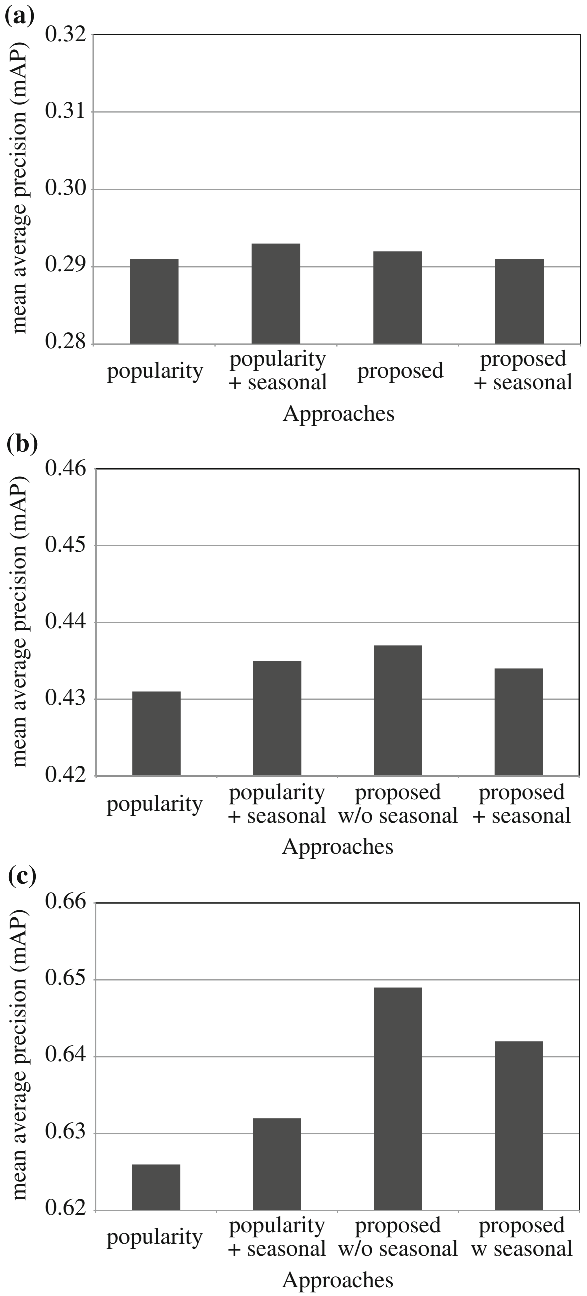


Fig. 10.1 Mean average precision of the inter-city recommendation: **a** all the landmarks, **b** top 25 landmarks, and **c** top 10 landmarks

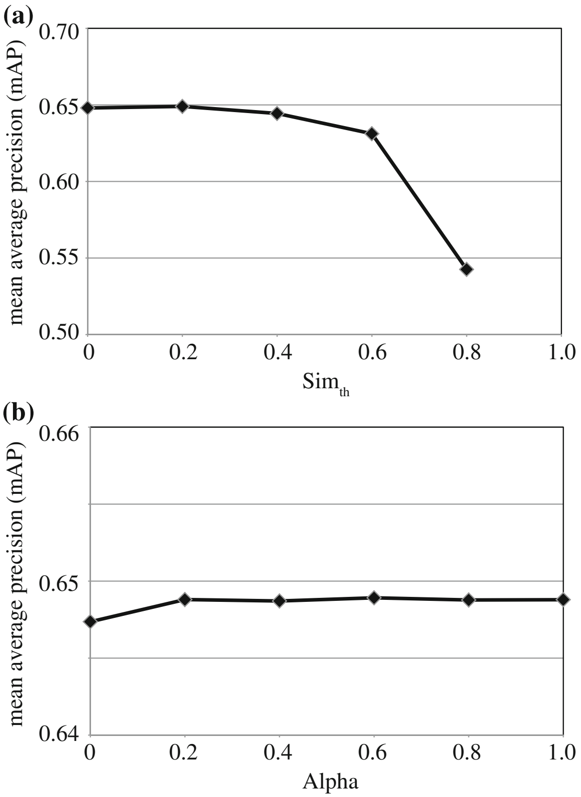


Fig. 10.2 Mean average precision as a function of **a** Sim$_{th}$ **b** α. Top 10 landmarks were considered

- Seasonal/Temporal Markov model: recommend the next landmark based on the user's current location using $P(L_j, (\text{s or t})|L_i)$. This joint model that considers the user's current location and either of the seasonal or temporal information.
- Seasonal&Temporal Markov model: recommend the next landmark based on the user's current location using $P(L_j,\text{s},\text{t}|L_i)$. This joint model that considers the user's current location, the seasonal information and the temporal information.

For each seasonal, temporal, and seasonal & temporal (proposed) model, two different seasonal information (4 seasons and 12 months) and two different time divisions (every 6 h and every 3 h) were trained and tested.

The detailed analysis of the recommendation accuracy is demonstrated in Fig. 10.3. Although slight difference can be observed from each other, the recommendation model considering the seasonal and temporal information always works better than the original Markov model. It is also demonstrated that the temporal information contributes more in the travel navigation when we look at the third to sixth columns in the figures. In addition, our Bayesian-based model is always better than the joint model in which the geotagged data are separated into different seasons and time

Fig. 10.3 Detailed mean accuracy of the next location recommendation: **a** all landmarks, **b** top 25 landmarks, and **c** top 10 landmarks

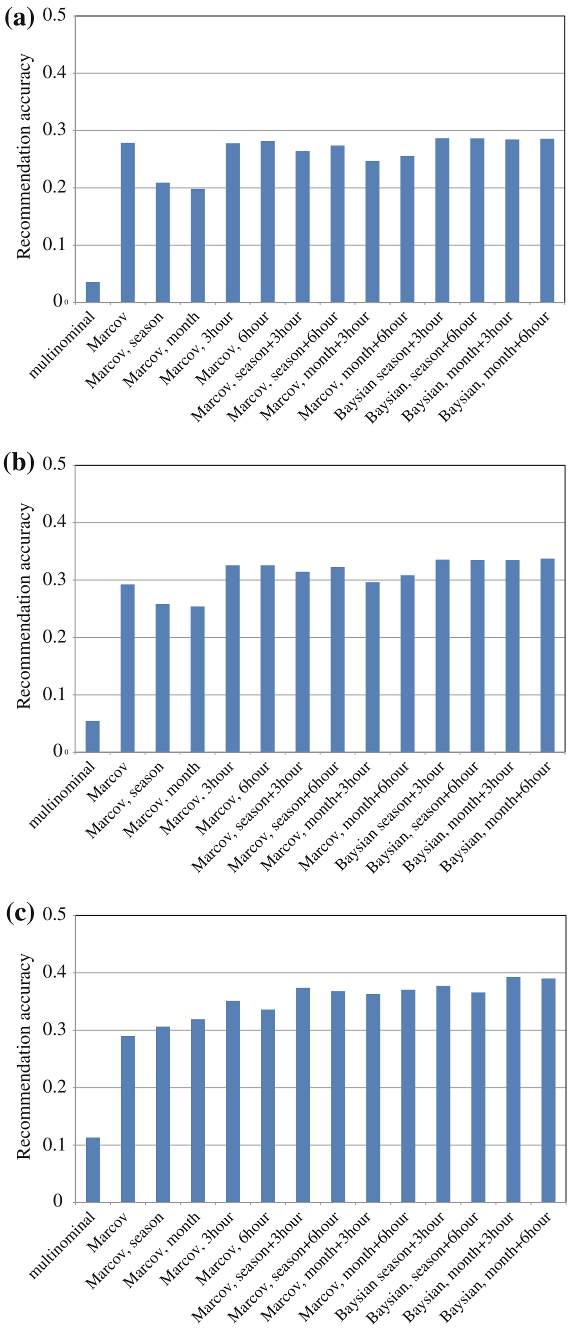


Table 10.1 Accuracy of intra-city recommendation

	Naive Markov (%)	Their results (%)	Improvement from Markov (%)
Eastern cities in US [22] (bw = 50m)	35.4	37.0	+1.6
Western cities in US [22] (bw = 50m)	24.7	27.2	+2.5
New York [8] ($H(L_{ij}) \geq 2$)	28	28	+0.03
	Naive Markov (%)	Our results (%)	Improvement from Markov (%)
21 cities in the world (all landmarks)	25.5	26.4	+0.9
21 cities in the world (top 25 landmarks)	29.3	33.4	+4.1
21 cities in the world (top 10 landmarks)	29.0	39.3	+10.3

ranges and the transition probability matrices from the current location L_i to the next location L_j are generated independently from each other season and time.

Table 10.1 shows the accuracy comparison of the intra-city landmark recommendation. Although the proposed model is simple, it is demonstrated that the performance improvement is much larger than the other approaches [8, 22]. Note that the datasets are different from each other because there is no open dataset. Cheng's work [8], in particular, requires that faces can be detected in the photos, therefore they require different dataset from [22] and ours. It is also observed that the performance becomes better when the number of landmarks to consider is smaller. This observation does not necessarily mean that travel navigation becomes easier with less number of landmarks to consider. In fact, if we look at the results by the naive Markov approach, the navigation accuracies are almost all the same regardless of the number of landmarks. This indicates that the difficulty of travel navigation is almost independent of the number of landmarks but popular landmarks tend to have stronger seasonal/temporal dependence in their popularity.

Two interesting examples from specific cities (New York and Taipei) are shown in Fig. 10.4 and Table 10.2. Here, the results considering the top 25 landmarks are presented. It is shown that the recommendation accuracy is improved in both cities. The accuracy becomes much better in New York (from 12.1 to 22.8 %) because some landmarks are popular at a certain season or time of the day. For example, the Rockefeller center gets more popular and the Statue of liberty gets less popular, respectively, in winter though these landmarks are always popular in all seasons. On the other hand, it can be observed that landmarks in Taipei have less such season or time specific popularity, indicating that the popularities of the landmarks in Taipei are independent

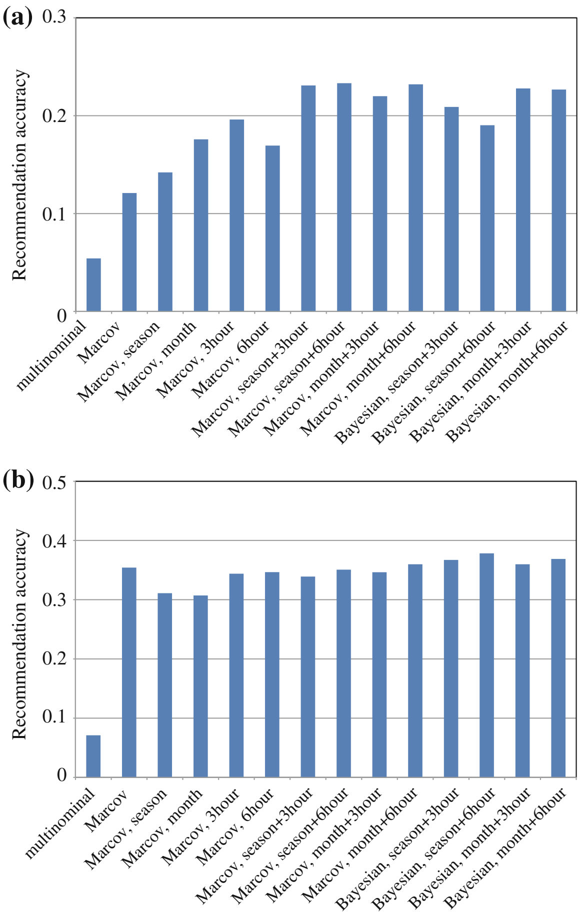
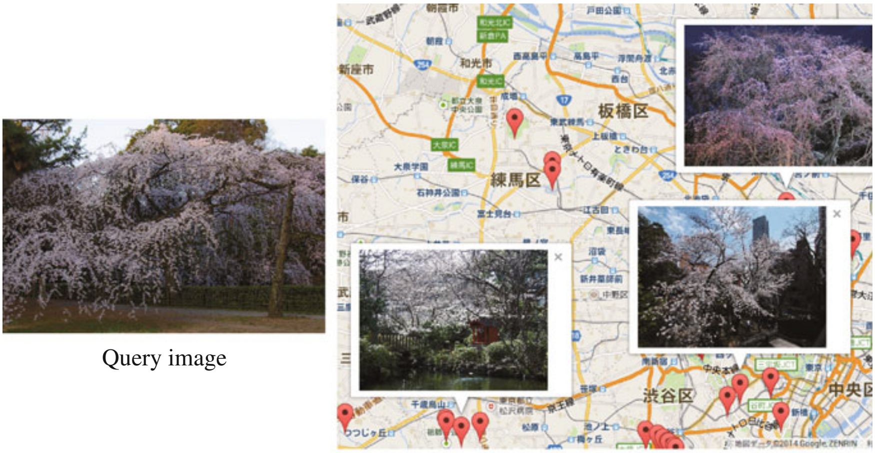


Fig. 10.4 Recommendation accuracy for specific cities: **a** New York **b** Taipei. The top 25 landmarks were considered

Table 10.2 Accuracy of intra-city recommendation in specific cities

	Naive Markov (%)	Our results (%)	Improvement from Markov (%)
New York	12.1	22.8	+10.7
Taipei	35.4	37.8	+2.4



Query image

Suggested landmarks in Tokyo

Fig. 10.5 "Where to go" navigation for photo-shooting

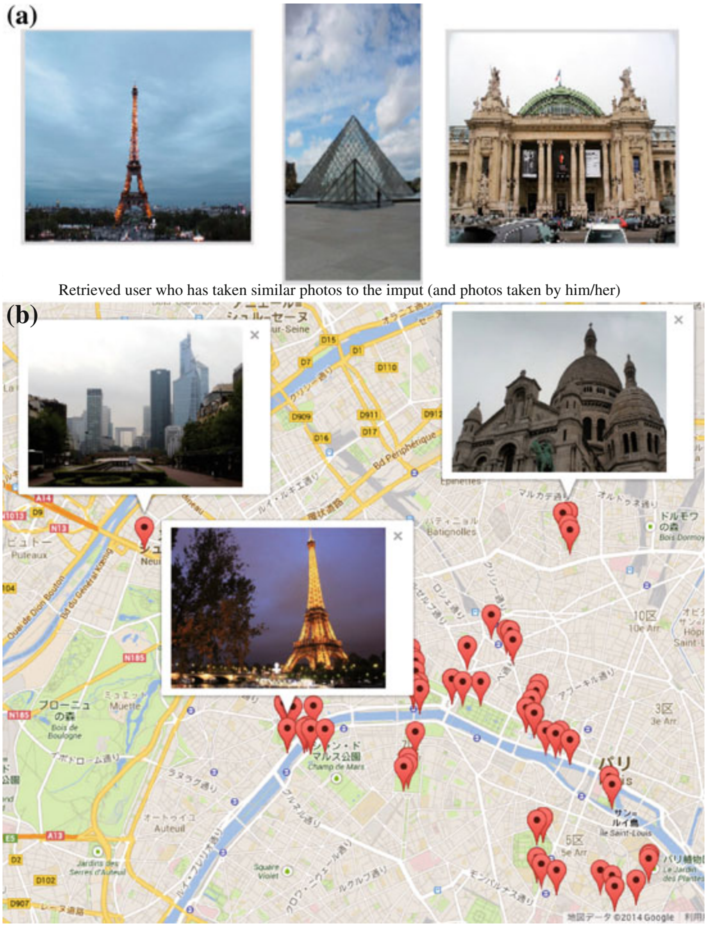
of time and season. In fact, popular landmarks in Taipei such as Taipei 101, National Palace Museum, etc., are indoor attractions and less sensitive to season and time.

10.5.4 Photo-Shooting Navigation

Figure 10.5 demonstrates the "where to go" navigation examples. In Fig. 10.5, a landscape photo taken at the Tokyo tower is used as a query and the recommended landmarks in Paris where similar photos can be taken are displayed on the map along with the photo examples. In Fig. 10.5, two different cherry blossoms are used as queries and the navigation system properly suggests where to visit in order to take similar photos.

In Fig. 10.6, results of user-to-user-similarity-based navigation are shown. Here, the user inputs three photos he/she has taken in Paris and the system retrieves the users who have taken similar photos. In Fig. 10.6, the query images and a retrieved user and his photos are shown. Then, the other photos that the retrieved user has taken in Paris are displayed on the map. In this manner, the system can navigate where to visit depending on the user's photo-shooting styles.

Examples of the "how to shoot" navigation are demonstrated in Fig. 10.7. The photos of the Eiffel tower in Paris and the Rockefeller center in New York with higher attractiveness score are shown in Fig. 10.7a and b, respectively. By showing such photo examples, the users can be advised how to take good photos by themselves.



Retrieved user who has taken similar photos to the imput (and photos taken by him/her)

Fig. 10.6 “Where to go” navigation based on similar user retrieval based on photo preference: **a** query and retrieved photos and **b** suggested landmarks and photos

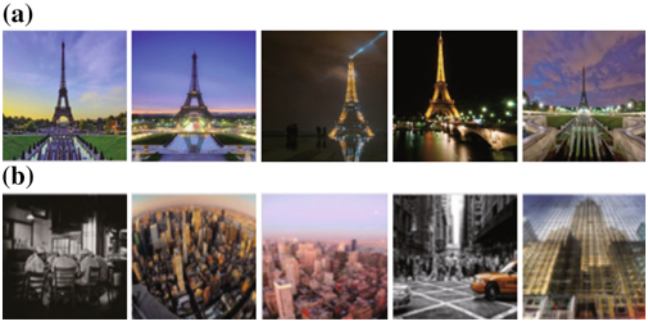


Fig. 10.7 “How to shoot” navigation: **a** Eiffel tower in Paris. **b** Rockefeller center in New York

10.6 Conclusions

This chapter presented a personalized travel and photo-shooting navigation algorithms. Our travel navigation model featuring seasonal and temporal information can improve the recommendation accuracy better than previous approaches. It is also possible to combine our proposed algorithm with the previous travel navigation approaches. Experiments using more than 6.2 million geo-tag data demonstrated the validity of our proposed algorithm. In addition, photo-shooting navigation examples using large-scale geotagged photos have also been presented.

References

1. A. Agarwala, Efficient gradient-domain compositing using quadtrees. ACM Trans. Graph. **26**(3), 94 (2007)
2. Y. Arase, X. Xie, T. Hara, S. Nishio, Mining people's trips from large scale geo-tagged photos, in *ACMMM*, pp. 133–142 (2011)
3. C. Barnes, E. Shechtman, A. Finkelstein, D.B. Goldman, PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph. (Proc SIGGRAPH) **28**(3), (2009)
4. M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00*, pp. 417–424 (2000)
5. S. Bhattacharya, R. Sukthankar, M. Shah, A framework for photo-quality assessment and enhancement based on visual aesthetics, in *ACMMM*, pp. 271–280 (2010)
6. M. Birsak, P. Musialski, P. Wonka, M. Wimmer, Automatic generation of tourist brochures. Computer Graphics Forum (Proceedings of EUROGRAPHICS 2014) 33 (2014)
7. W.C. Chen, A. Battestini, N. Gelfand, V. Setlur, Visual summaries of popular landmarks from community photo collections, in *ACMMM*, pp. 789–792 (2009)
8. A.J. Cheng, Y.Y. Chen, Y.T. Huang, W.H. Hsu, H.Y.M. Liao, Personalized travel recommendation by mining people attributes from community-contributed photos, in *ACMMM*, pp. 83–92 (2011)
9. J. Choi, G. Friedland, V. Ekambaram, K. Ramchandran, Multimodal location estimation of consumer media: Dealing with sparse training data, in *ICME*, pp. 43–48 (2012)
10. M.D. Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, C.Yu ,Constructing travel itineraries from tagged geo-temporal breadcrumbs, in *WWW*, pp. 1083–1084 (2010)
11. M. Clements, P. Serdyukov, A.P. de Vries, M.J. Reinders, Using flickr geotags to predict user travel behaviour, in *SIGIR*, pp. 851–852 (2010)
12. D. Crandall, L. Backstrom, D. Huttenlocher, J. Kleinberg, Mapping the world fs photos, in *WWW*, pp. 761–770 (2009)
13. Q. Fang, J. Sang, C. Xu, Giant: Geo-informative attributes for location recognition and exploration, in *ACMMM*, pp. 13–22 (2013)
14. F. Girardin, F. Calabrese, F. Fiore, C. Ratti, J. Blat, Digital footprinting: Uncovering tourists with user-generated content. IEEE Pervasive Comput. **7**(4), 36–43 (2008)
15. J. Hays, A. Efros, Im2gps: estimating geographic information from a single image, in *CVPR*, pp. 1–8 (2008)
16. J. Hays, A.A. Efros, Scene completion using millions of photographs. ACM Trans. Graph. **26**(3), 87–94 (2007)
17. N. Ishihara, Y. Itoh, K. Takashima, F. Kishino, Attractiveness-based image scoring using similar images on websites, in *IPSJ Interaction (in Japanese)* (2011)

18. A. Jaffe, M. Naaman, T. Tassa, M. Davis, Generating summaries and visualization for large collections of geo-referenced photographs, in *ACM MIR*, pp. 89–98 (2006)
19. K.Jiang, P. Wang, N. Yu, Contextrank: Personalized tourism recommendation by exploiting context information of geotaggedweb photos, in *ICIG*, pp. 931–937 (2011)
20. L. Kennedy, M. Naaman, Generating diverse and representative image search results for landmarks, in *WWW*, pp. 297–306 (2008)
21. L. Kennedy, M. Naaman, S. Ahern, R. Nair, T. Rattenbury, How flickr helps us make sense of the world: context and content in community-contributed media collections, in *ACMMM*, pp. 631–640 (2007)
22. T. Kurashima, T. Iwata, G. Irie, K. Fujimura, Travel route recommendation using geotags in photo sharing sites, in *CIKM*, pp. 579–588 (2010)
23. Li C, Loui AC, Chen T (2010) Towards aesthetics: A photo quality assessment and photo selection system, in *ACMMM*, pp. 827–830
24. J. Li, Z. Qian, Y.Y. Tang, L. Yang, T. Mei, Gps estimation for places of interest from social users' uploaded photos. IEEE Trans. Multimedia **15**(8), 2058–2071 (2013)
25. Z. Luming, S. Mingli, Y. Yi, Z. Qi, Z. Chen, N. Sebe, Weakly supervised photo cropping. IEEE Trans. Multimedia **16**(1), 94–107 (2014)
26. Mamei M, Rosi A, Zambonelli F (2010) Automatic analysis of geotagged photos for intelligent tourist services, in *Sixth International Conference on Intelligent Environments*, pp. 146–151 (2010)
27. Nishiyama M, Okabe T, Sato Y, Sato I (2009) Sensation-based photo cropping, in *ACMMM*, pp. 669–672
28. Popescu A, Grefenstette G (2011) Mining social media to create personalized recommendations for tourist visits, in *COM.Geo*
29. A. Popescu, G. Grefenstette, P.A. Moëllic, Mining tourist information from user-supplied collections, in *CKIM*, pp. 1713–1716 (2009)
30. Quack T, Leibe B, Gool LV (2008) World-scale mining of objects and events from community photo collections, in *CIVR*, pp. 47–56
31. T. Rattenbury, M. Naaman, Methods for extracting place semantics from flickr tags. ACM Trans. Web **3**(1), 1:1–1:30 (2009)
32. H. Shu, C. Chen, T. Chen, Landmark recognition: A unary approach, in *ICIP*, pp. 2645–2648 (2010)
33. J. Sun, L. Yuan, J. Jia, H. Shum, Image completion with structure propagation. ACM Trans. Graph. **24**(3), 861–868 (2005)
34. J. Wu, J. Rehg, Centrist: A visual descriptor for scene categorization. IEEE TPAMI **33**(8), 1489–1501 (2011)
35. Yamasaki T, Gallagher A, Chen T (2013) Personalized intra- and inter-city travel recommendation using large-scale geo-tagged photos, in *Geomm*
36. Yeh CH, Ho YC, Barsky BA, Ouhyoung M (2010) Personalized photograph ranking and selection system, in *ACMMM*, pp. 211–220 (2010)
37. Yin H, Lu X, Wang C, Yu N, Zhang L (2010) Photo2trip: an interactive trip planning system based on geo-tagged photos, in *ACMMM*, pp. 1579–1582
38. Yunpeng L, Crandall D, Huttenlocher D (2009) Landmark classification in large-scale image collections, in *ICCV*, pp. 1957–1964
39. Y. Zheng, S. Yan, Z. Zha, Y. Li, x Zhou, T. Chua, R. Jain, Gpsview: A scenic driving route planner. ACM Trans. Multimedia Comput. Commun. Appl. **9**(1), 3:1–3:18 (2013)