Multimedia Annotation Through Search and Mining

A Dissertation submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Emily K. Moxley

Committee in Charge:

Professor B. S. Manjunath, Chair
Professor Shivkumar Chandrasekaran
Professor Kenneth Rose
Dr. Wei-Ying Ma

September 2009
The Dissertation of
Emily K. Moxley is approved:

________________________
Professor Shivkumar Chandrasekaran

________________________
Professor Kenneth Rose

________________________
Dr. Wei-Ying Ma

________________________
Professor B. S. Manjunath, Committee Chairperson

July 2009
Multimedia Annotation Through Search and Mining

Copyright © 2009
by
Emily K. Moxley
To everyone that makes me smile
Acknowledgements

This work represents the culmination of an entire army of people, who have kept me humored, interested, and interesting. Thanks foremost goes to my advisor Professor B.S. Manjunath whose example has taught me the importance of balance and of considering all aspects when making decisions. It is with your guidance that I’ve slowly developed the skills and learned to ask the questions to become a researcher. I also greatly appreciate the efforts by Dr. Wei-Ying Ma, who provided me with a life-changing experience in Beijing at Microsoft Research Asia. And Jim, I wouldn’t be here without your valuable collaboration, reminders of the big picture, and general acceptance of my stress level. You can stop writing my thesis now; this manuscript represents its completion. We’ll always have Beijing.

Nhat, Mike, and the others in the VRL lab have always been able to commiserate with me and provided great advice for every situation I wiggled my way into. You have given me valuable lessons on how to handle them all. Luca and Ibrahim, your patience, encouragement, and practice exams before orals helped get me through my first summer. Swapna and Melissa, you’ve helped me stay sane in the lab and around the water cooler, and I greatly admire your gentle and supportive approach.

My roommates Ian, Nada, Ken H., and (honorary) Ken S. have made the whole process fun, along with countless others that cannot all possibly be named. Thanks for the memories, the quotes, and the advice on literally everything. I will always think
of these years in Santa Barbara with warmth and nostalgia. To the A-Team: you’ve provided priceless long-distance support and TLC. Thanks for the reminders of life outside graduate school, for humoring my terrible, nerdy jokes, and for putting up with my constant barrage of e-mails.

And finally, thanks to my brothers and parents who know when to build me up, but also when to tear me down. Jerry, your love of life is an inspiration; and Joel, your crazy antics and yet good sense have given me great guidance. Mom and Dad, your comforting words always came at the right times. I owe the richness of my life entirely to your selfless parenting.

All of you have contributed greatly to the process reflected by this thesis. As Barack Obama declared during his inauguration, “Our stories are singular, but our destiny is shared.” Thanks goes to all of you who have shared this experience with me and so positively contributed to our collective destiny.
Curriculum Vitæ

Emily K. Moxley

Education

September 2009  Doctor of Philosophy
Department of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, California

June 2007  Master of Science in Engineering
Department of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, California

June 2005  Bachelor of Science in Engineering
Department of Electrical Engineering
Princeton University
Princeton, New Jersey

Participation

2005-2009  NSF IGERT Trainee in Interactive Digital Multimedia
Department of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, California

Selected Publications


Brent Hecht and Emily Moxley. “Terabytes of Tobler: Evaluating the First Law in a Massive, Domain-Neutral Representation


tion, Augsburg, Bavaria, Germany, pp. 84-88, Sep. 2007.


Abstract
Multimedia Annotation Through Search and Mining
Emily K. Moxley

Multimedia annotation represents an application of computer vision that presents the recognition of objects or ideas related to a multimedia document as a text label. Typically, annotation algorithms depend on complicated feature extraction and matching algorithms that attempt to learn individual annotation models. This work, however, reveals that it is possible to achieve effective annotation of large datasets without specific models by combining information from low-level visual features with annotation mining of the data. This technique is referred to as annotation by mining. The method is especially effective in the presence of aliased, redundant data, a characteristic feature of social media sites and content available on the web. By using this formulation, we are able to address the problem in a way that is highly scalable and fast regardless of dictionary size.

The work places particular emphasis on learning using graph theory. Such an approach can lead to algorithms that effectively combine disparate feature metrics through examination of the stability and smoothness of a graph constructed in any metric space. Specifically, a concept of “graph smoothness” is formulated that reflects the distribution of different attributes in the graph. This smoothness measurement allows us to
extract visual annotations and geographic place annotations, as well as find weighting parameters for disparate similarity modalities. Analysis validates the approach on two different sets of videos, one a collection of TRECVID news videos and another a set crawled from the online repository hosted by YouTube, and two different image databases crawled from the set of Flickr geotagged photos. The approach is proven to be successful at mining accurate annotations out of noisy transcripts and noisy tagged social media data while scaling to dictionary sizes of more than 430,000 words.
Contents

Acknowledgements v
Curriculum Vitæ vii
Abstract x
List of Figures xv
List of Tables xvii

1 Introduction 1
   1.1 Why Annotation? Why Mining?  3
   1.2 Challenges in Modern Multimedia Annotation  9
      1.2.1 Multimedia Annotation  9
      1.2.2 Social Media  10
   1.3 Summary of Contributions and Organization  14

2 Multimedia Annotation and Graph Theory 17
   2.1 Visual Feature Models for Images  18
   2.2 Correlation Mining for Images  21
   2.3 Associated Text and Web Mining for Images  22
   2.4 SVM Models for Video Annotation  24
   2.5 Correlation Mining for Videos  25
   2.6 Graph Theory Techniques  27
      2.6.1 Graph Theory in Image Annotation  29
      2.6.2 Graph Theory in Video Annotation  30
      2.6.3 Multiple Graph Reinforcement  33
## 3 Collaborative Annotation of Geotagged Photos

### 3.1 Related Work

### 3.2 Geotagged Image Annotation Algorithm

### 3.3 Tag Suggestion By Mining

#### 3.3.1 Tag Suggestion by Geographic Prior

#### 3.3.2 Tag Suggestion by Visual Features

### 3.4 Tag Reranking by Georelevance

### 3.5 Performance of Georelevance Reranking

#### 3.5.1 Relevance/Coverage

#### 3.5.2 Groundtruth Annotation

#### 3.5.3 Geographic Baseline

#### 3.5.4 Visual Baseline

#### 3.5.5 Georelevant Annotation

### 3.6 Probabilistic Formulation

### 3.7 Bayesian Annotation Suggestion

#### 3.7.1 Non-Parametric k-NN Density Estimation

#### 3.7.2 Regional Representation Using a Quadtree

#### 3.7.3 Baseline Methods

#### 3.7.4 Smart Fusion

### 3.8 Performance of MAP Georelevant Tag Suggestion

#### 3.8.1 Precision-Recall of Tag Suggestions

### 3.9 Conclusions

## 4 Mining Tag Semantics

### 4.1 Related Work

### 4.2 Semantic Identification of Tags

#### 4.2.1 Visual Term Extraction

#### 4.2.2 Place Extraction

#### 4.2.3 Landmark Detection

### 4.3 Conclusions

## 5 Video Annotation Using Search and Mining

### 5.1 Annotating Videos

### 5.2 Mining Annotations in Similar Videos

#### 5.2.1 Proof-of-Concept

### 5.3 Annotation Using Graphs

#### 5.3.1 Singular Graph Reinforcement: Semi-Supervised Learning

#### 5.3.2 Singular Graph Reinforcement: Unsupervised Learning

#### 5.3.3 Multiple Graph Reinforcement: Semi-Supervised Learning

#### 5.3.4 Multiple Graph Reinforcement: Unsupervised Learning
5.4 Graph Reinforcement as a Steady-State Markov Chain Problem ................................................. 128
5.5 Fitting a Zipf Curve .................................................................................................................. 131
5.6 Conclusions ............................................................................................................................... 133

6 Case Studies: Graph-Based Video Annotation ................................................................. 134
  6.1 Transcript Mining Using Graphs ............................................................................................ 134
    6.1.1 Unsupervised Learning vs. Semi-Supervised Learning ..................................................... 135
    6.1.2 Annotation Discovery ......................................................................................................... 138
    6.1.3 Evaluation of Graph Size \( N \) ......................................................................................... 139
    6.1.4 Evaluation of Weighting Parameter \( r \) ........................................................................ 141
  6.2 Collaborative Video Annotation Using Graphs ......................................................................... 142
    6.2.1 Experimental Setup ........................................................................................................... 145
    6.2.2 Unsupervised Learning vs. Semi-Supervised Learning ..................................................... 146
    6.2.3 Collaborative Annotation Extension and Filtering ............................................................ 147
    6.2.4 Annotating Without Initial Keywords ............................................................................ 151
    6.2.5 Evaluation of Graph Size, \( N \) ....................................................................................... 151
    6.2.6 Evaluation of Weighting Parameter, \( r \) ...................................................................... 152
  6.3 Conclusions ............................................................................................................................... 154

7 Conclusions and Future Directions ....................................................................................... 156
  7.1 Multimedia Annotation .............................................................................................................. 156
    7.1.1 Redundancy ........................................................................................................................ 157
    7.1.2 Annotation-Dependent Fusion .......................................................................................... 157
  7.2 Social Media .............................................................................................................................. 158
    7.2.1 Tag Classification .............................................................................................................. 158
    7.2.2 User Analysis ...................................................................................................................... 159
    7.2.3 Analysis Alongside Traditional Media ............................................................................... 160
  7.3 Conclusions ............................................................................................................................... 163

Bibliography ................................................................................................................................. 165
# List of Figures

1.1 Redundancy in Flickr .................................................. 4  
1.2 Tag cloud from Wordpress ........................................... 6  
1.3 Redundancy in search returns from YouTube database ........... 7  
1.4 Redundancy in del.icio.us ............................................. 8  
1.5 Image from single-label dataset ..................................... 11  
1.6 Image from a multi-label dataset ................................... 12  

2.1 Diagram of Zhou’s smooth manifold ranking ...................... 30  
2.2 Multi-graph learning diagram ...................................... 34  

3.1 Tag cloud for Flickr data ............................................ 40  
3.2 Workflow for geotagged image suggestion .......................... 41  
3.3 Local vs. global tag frequency ...................................... 48  
3.4 Tag suggestion performance by geographic radius ............... 53  
3.5 Tag suggestion by size of contributing similar set ............... 55  
3.6 Example tag suggestions for georeferenced image data ........... 56  
3.7 Physical representation of quadtree boundaries ................... 65  
3.8 Map showing geographic distribution of test images ............. 69  
3.9 Dual method performance to visual and geographic baseline ..... 70  
3.10 Visual information’s influence on images in Pisa ................. 72  
3.11 Visual information’s influence on images in the Bronx .......... 73  
3.12 Visual information’s influence on images in Australia .......... 74  
3.13 Performance of dual method and baselines for visual keywords only . 76  
3.14 Performance using smart fusion .................................... 77  
3.15 Example tags using probabilistic formulation ...................... 78  
3.16 Performance when including reverse geocoder ..................... 79  
3.17 Performance by photo density ...................................... 80  
3.18 Performance gain by using quadtree ............................... 81
3.19 Georelevance reranking performance for Los Angeles and Southern California ............................... 83

4.1 Example use of tag semantics ............................................. 86
4.2 Mutual information vs. graph smoothness of visual terms .......... 95
4.3 Tag frequency by descending mutual information score .......... 96
4.4 Visual terms using MI vs. graph smoothness using 5 features .... 97
4.5 Visual terms using proportional MI vs. graph smoothness ....... 98
4.6 Identification of place tags .............................................. 99
4.7 Identification of place terms ............................................ 101
4.8 Examples of detected landmarks ....................................... 102

5.1 System workflow for basic search and mining approach .......... 110
5.2 Comparison of search modalities for search and mining approach .... 114
5.3 System workflow for graph reinforcement approach ............. 117
5.4 Constructing a graph .................................................... 118
5.5 Graph reinforcement as a steady-state Markov problem ........ 130
5.6 Transcript word frequency compared to theoretical best-fit zipf curve . 132

6.1 Graph reinforcement annotation performance ....................... 137
6.2 Example annotations using graph reinforcement approach ........ 138
6.3 Annotations discovered by graph reinforcement ................... 140
6.4 Performance of graph reinforcement by graph size $N$ .......... 141
6.5 Performance of graph reinforcement by weighting parameter $r$ .. 143
6.6 Collaborative tagging for YouTube videos using graph reinforcement . 148
6.7 Example annotations for YouTube videos using graph reinforcement . 149
6.8 ROC for tag extension and tag filtering .............................. 150
6.9 Performance on unlabeled video ....................................... 152
6.10 Performance by graph size $N$ and weighting parameter $r$ .... 155
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Graph theory analogy for unsupervised video annotation</td>
<td>28</td>
</tr>
<tr>
<td>3.1</td>
<td>Manually-selected visually-relevant keywords</td>
<td>75</td>
</tr>
<tr>
<td>4.1</td>
<td>Mutual information between tags and visual features</td>
<td>91</td>
</tr>
<tr>
<td>4.2</td>
<td>Example smoothness statistics</td>
<td>104</td>
</tr>
<tr>
<td>4.3</td>
<td>Statistics for landmark extraction algorithm</td>
<td>105</td>
</tr>
<tr>
<td>5.1</td>
<td>Graph theory analogy for unsupervised video annotation</td>
<td>120</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Multimedia annotation is the application of text labels to images, videos, or other multimedia. The annotations can be used for browsing, search and retrieval, or organization by a document owner, a friend of that owner, or anyone that happens to peruse the collection either privately or online. The labels are applied using a variety of rationales and intentions. While some are purely for personal use, others are for the purpose of sharing with friends or strangers browsing the collection. As applied to an image or photo, for instance, annotations range from general objects in the image, to descriptions of the setting of the photo, names of individual people in the image, qualities of the photo, or general as well as specific locations of the photo. This information may or may not be apparent to a general viewer without information beyond the photo.
Chapter 1. Introduction

This thesis aims to tackle the problem of image and video annotation with an emphasis on data mining techniques. A mining approach, a definition of which is provided in Chapter 2 but which we roughly define as exploiting near neighbors for annotations rather than building feature-based models, more effectively utilizes the full information available to annotate multimedia, especially in the presence of similar independent annotation instances. Mining represents a complementary approach to traditional computer vision techniques that focus on feature selection and modeling, and eventually vision techniques may be combined with mining in the annotation task. Social media communities, e.g., del.icio.us, Flickr, Webshots, YouTube, Panaramio, etc., are a particularly relevant field for annotation by mining because they provide multiple independent tagging instances of similar or duplicate documents, a situation where mining can be especially powerful, and these communities are dealt with extensively in this thesis.

This introductory chapter extends motivation to the annotation problem and to using a mining scheme to annotate automatically. A brief overview of social media communities is given, followed by a summary of the contributions to the field and the organization of the remainder of this thesis.
1.1 Why Annotation? Why Mining?

The exponential increase of multimedia data in recent years demands effective organization for efficient user retrieval and browsing. Flickr has 3.3 million geotagged photos contributed per month, as well as multiplicatively more that are not. In April 2009, about 5000 photos were uploaded each minute to Flickr. While Flickr will not give out current statistics on its size, we estimate that at the writing of this thesis in July 2009, the repository has approximately 7.5 billion images. Similarly huge in scale, YouTube contained 144 million videos as early as August 2008.

While the Internet is rich with huge image and video databases, the ability to use these databases is limited by our ability to bridge the semantic gap between what a user seeks into a form that is understood by the computer. An effective method for dealing with the gap between user desires (expressed through text queries) and database content (a collection of images/videos) is to assign tags to the data, a process called annotation. An example of the tags applied to an image can be seen in Figure 1.1. The owner of the top image, for instance, has applied the keywords “sanfrancisco,” “california,” “nikon,” and “goldengatebridge.” Given a particular query, a multimedia search algorithm can scan for or intelligently interpret the query words to find matching tags in the database, returning the images or videos that most closely satisfy the interpreted query.
Chapter 1. Introduction

**Figure 1.1:** Top results for “golden gate bridge” on Flickr. Similar images can be found in results, with repetition of tags and annotations. In this set, extending the “goldengatebridge” tag from the 1st return may be helpful and complementary in describing the bottom two images which do not have that tag.

As of writing, all large multimedia repositories, including YouTube [95], Flickr [19], and Facebook [17], have user-generated tags that enable querying. However, manual annotations are incomplete, noisy, and ripe for exploitation. For instance, an
advertiser may upload a video of his product, but in order to increase its viewership give it currently-popular tags (e.g., “Britney Spears”) so that it is returned in the top hits for a large number of queries, thus giving his product more exposure. This tactic is particularly rife in sites that use tag clouds such as Wordpress [88]. An example of a tag cloud can be seen in Figure 1.2. Application of popular tags has been directly incentivized at Wordpress as viewers can browse by clicking on frequently-used tags on the main page, so adding a popular tag to your article will increase its number of hits. Outside malicious cases, tags are subject to user error. It is quite likely that a user would not initially think of all relevant keywords, but rather merely a subset, as can be seen in Figures 1.1, 1.3, 1.4. Also worth consideration is the frequency of careless spelling errors. An automatic approach that minimizes these errors is desirable to correct mistakes as well as facilitate the upload and organization process for users.

Motivation for a mining approach to automatic annotation becomes explicitly clear with the Flickr example in Figure 1.1, the YouTube example shown in Figure 1.3, and the del.icio.us example in Figure 1.4 which demonstrate the redundancy of modern information especially in social media sites. As evidenced by these examples, due to increased bandwidth and cheap information storage the Internet contains a lot of redundant information. It seems foolish to ignore the possibility that the work may have already been partly done. A new multimedia document, or poorly-tagged one, can be annotated by collecting and filtering the tags of similar documents. An automatic
Chapter 1. Introduction

Figure 1.2: Tag cloud from Wordpres. Bigger words represent tags that have been applied to more blog entries. Tagging spam is incentivized since authors that use popular keywords will receive more hits.

This technique would have less sporadic error and would also be able to correct malicious or naively error-prone tagging. It would be able to filter incorrect annotations and extend correct ones to a particular target document. This thesis therefore proposes an annotation method that mines redundant and relevant information from noisy databases.
Chapter 1. Introduction

With the appropriate data, this technique mines correct annotations out of a collection of possible annotations found for similar documents.

![YouTube search results](image)

**Figure 1.3:** Top results for “etrad commercial” on YouTube. Duplicates and similar videos can be found in top results, with repetition of tags and annotations. In this example, extending the “superbowl” tag from the 3rd and 4th returns may be helpful and complementary in describing the first two returns.
Chapter 1. Introduction

**Figure 1.4:** Top results for “http://www.whenisgood.net” on del.icio.us. 153 people have tagged this identical website with keywords, a subset of which are shown on the right side of the page. Using these independent tagging instances, an effective system can glean a trustworthy, complete set of keywords.

Next this introduction will present an overview of the annotation challenges in order to motivate an automatic *mining* approach. A discussion of the challenges generated by social media communities in Section 1.2.2 motivates the collaborative filtering applications found in this thesis, before we describe the specific contributions of this thesis to conclude the chapter.
1.2 Challenges in Modern Multimedia Annotation

1.2.1 Multimedia Annotation

Annotation is a challenging task for several reasons. Firstly, the recognition of objects in images using computer vision techniques is not at all a solved problem. As the brain’s recognition of elements is not fully understood, computer vision algorithms are unable to mimic effectively the vision/recognition process. State-of-the-art classification algorithms have accuracy rates of 63% even when dealing with clean, object-oriented datasets with high-quality pictures and limited, known classification possibilities [22]. In addition, most images are not object-oriented. Most photos, for instance, consist of a large amount of clutter and occlusion which may or may not be of interest. Partially as a result of this complication, segmentation is an extensively studied field in multimedia. Multi-instance annotation takes the segmented media as a collection of annotation instances.

The aforementioned case deals with classification, an instantiation of annotation with a single label. However, this thesis deals exclusively with the multi-label case where an unknown number of annotations or labels are to be applied. The distinction between multi- and single-label annotation can be seen clearly in the examples in Figures 1.5 and 1.6. Challenges in single label annotation, as in Figure 1.5, include segmentation of the dominant object. Multi-label annotation, as in Figure 1.6, is com-
Chapter 1. Introduction

Structured by cluttered backgrounds and segmentation of the individual objects. Many multi-label annotation instantiations are far more complicated than even that found in Figure 1.6, where all of the annotations are visually present in the image. This thesis deals with more complicated instances such as the Flickr-type photos in Figure 1.1. In such databases instances are encountered where annotations like “October” or “scary,” that are not visually-oriented, have been applied to the image.

1.2.2 Social Media

The definition of social media is a developing concept, but largely refers to user-driven information published in a highly scalable way. Weblogs, social blogs, wikis, podcasts, pictures, and video are all forms of social media. One important characteristic is that it is published in a way that allows a many-to-many dialog among community members. A large number of websites have arisen that provide online repositories for users to share media of interest, such as links, images, or videos. Typically, publishing sources in social media are not verified and the repositories are largely unstructured. In these communities unstructured annotation of objects, known as tagging, discards the inflexibility inherent in structured labeling, but this benefit has associated costs. The freeform nature of tags brings new challenges such as how to extract a useful signal or harvest knowledge in the presence of semantic uncertainty and noise in labeling.
Chapter 1. Introduction

Figure 1.5: Example image from Caltech-101, an object-oriented dataset typically used as a benchmark for single-label annotation (classification). Correct classification is “elephant.”

Research into online media communities has motivated studies on the effectiveness of user annotation and the learning possible from independent annotations offered prolifically by community users. Shirky [69] emphasizes the importance of the annotations in online media websites as they allow for a dynamic, evolving understanding of the
Chapter 1. Introduction

Figure 1.6: Example image from a dataset meant for multi-label classification, used in [100]. “Complete” annotation given by the owner of this dataset is “desert,” “sunset,” “trees.”

world, and such databases have been used for purposes as diverse as finding lexical distances [89] and identifying images of landmarks of geotagged Wikipedia entries [64]. A moderate body of literature exists identifying statistics, trends, and taxonomies for annotation of online media databases [11], [37], [2], [70], [28], [46].

Analysis of tag statistics in online communities motivate problems such as combating spam [28], [24], [12], [49]. The repeated tagging instances afforded by aliased data in online media sites [8] enable tag filtering in social media computing to prevent annotation spam, when a user uploads advertising media and tags the disguised ad with popular search terms to attract hits. Simultaneously, tag extension has been emphasized for such sites with the observation that a large number of keywords can be generated only by a large group of users [23]. The question of effective learning us-
Chapter 1. Introduction

ing user-supplied information has not been explicitly tested. This thesis harnesses the large amount of freely-contributed overlapping annotation to address the issues of consistency and spam in an automated learning process. The synthesis of user-contributed tags is often referred to as “collaborative tagging.”

A further challenge presented by social media communities lies in translating owner and user motivations. Even if segmentation and object detection algorithms were perfect, the issue of interpreting viewer and owner intentions still exists. Particular examples of complicating circumstances are the use of annotations that indicate place or time (and thus, are typically not visually contained in the photo), annotations that have multiple meanings such as “apple,” and difficulty interpreting queries such as “views golden gate” which could indicate views of or from. Geotags are another instance where intent is complicated; some geotags represent the location that a photo was taken, others indicate the location of the building, object, or other aspect of the scene in a photo.

Ignorance of tag categories and tagging motives in social media databases limits the effectiveness of any automatic tagging system. If a system can know the motivation or type of a particular tag, it can more accurately weight appropriate features and metadata. This thesis furthermore digs into automatic extraction of tag semantics, an issue identified by the literature [28], [49], and explored outside this thesis only to the extent of place and event tag identification [65]. This thesis presents research that sim-
Chapter 1. Introduction

Similarly attempts automatic semantic categorization of certain types of tags that aid the annotation task.

1.3 Summary of Contributions and Organization

Broadly speaking, the primary aim of this thesis is to address the annotation problem with a mining approach as opposed to the typical visual feature modeling approach. It furthermore tackles issues that arise from aspects characteristic of social media, issues such as redundancy, trustworthiness of information, and informative use of independent tagging instances. The contributions of this thesis are summarized below.

- **Annotation by mining:** In comparison to much of the state-of-the-art in multimedia annotation, this work does not rely on a training set that limits the annotation set to those labels for which there is sufficient training data. Rather, it discovers the appropriate annotations in tagging instances of similar documents using techniques that model the visual, speech, or geographic space around a focus image without annotation modeling from feature primitives. This is covered for image annotation in Chapters 3 and 4 and for video annotation in Chapters 5 and 6. Video annotation is formulated heuristically at first, and then extended to a graph theory framework and a collaborative tagging algorithm.
Chapter 1. Introduction

- Collaborative tagging: The collaborative tagging goal, gleaning trustworthy keywords from the tagging synergy of multiple online community users, is a timely problem put forth by the research community [24, 12, 23]. This thesis performs annotation on documents generated from communities where independent annotations exist on both identical documents and thematically similar documents. The work tackles collaborative tagging by using both the Flickr image community in Chapters 3 and 4 and the YouTube video community in Chapter 6. Collaborative tagging is attempted using a graph theory approach to smooth annotations as well as using a probabilistic approach that finds the posterior of a tag given its visual features and location. The collaborative tagging on Flickr data draws from on work done in collaboration with Jim Kleban.

- Annotation of geotagged photos: An algorithm explicitly built for annotating geotagged photos has not been attempted before. Work on geotagged photos has been limited to identifying landmarks, designating canonical views, and neighborhood extraction. This thesis presents work that for the first time attempts to optimize annotation for photos with an accompanying geotag, and does this on a world scale. This contribution, described in Chapters 3 and 4 was done in collaboration with Jim Kleban.
Chapter 1. Introduction

- **Social media community analysis**: This thesis contains research on tag semantic analysis, a crucial element for future effective collaborative tagging. Chapter 4 provides analysis of identifying tag semantics mined from Flickr data that can be incorporated into effective annotation algorithms.
Chapter 2

Multimedia Annotation and Graph Theory

This chapter presents an overview of the recent advances in multimedia annotation. The primary focus lies in image and video annotation with an emphasis on graph theoretic methods for annotation. The chapter presents an overview of the major movements in multimedia annotation, beginning with the case of image annotation and building to video annotation. The trajectory of image annotation can generally be seen as a movement from a small database where expensive techniques can be used, to web-scale annotation methods that must be fast and can exploit information on the Internet. The trajectory has also been largely from single-label annotation (also called “classification”) to multi-label annotation, where more than one label can apply to an image.
Chapter 2. Multimedia Annotation and Graph Theory

Video annotation has followed a similar movement, often simply applying image annotation techniques to a collection of keyframes from the video. Graph techniques are one theme in annotation literature that have shown promising results and that are used extensively in this thesis, and an exploration of this area follows general discussion of the image and video annotation problem.

2.1 Visual Feature Models for Images

The traditional approach to image annotation relies on computer vision modeling using visual features, treating annotation as a detection problem for each individual word or phrase. A model is created for each possible annotation, $b$, that represents the “canonical” representation of annotation $b$. It attempts to distill for each annotation a set of feature primitives which indicate suitability for an annotation. The major benchmarking dataset for the single-annotation case has typically been the Caltech 101 collection, which contains 101 categories consisting of between 31 and 435 object-oriented photos. Work began with classification (i.e., single-label annotation) using global features [13], then regional features derived from blocks, regions, “superpixels,” or subimages [32, 44, 10, 33], and ultimately, local features [48, 21], which are described in the following paragraphs.
Chapter 2. Multimedia Annotation and Graph Theory

**Global features** for multimedia analysis constitute a singular, typically low-dimensional, feature that characterizes the entire color, edge, or texture space, for instance, of an image. The MPEG-7 standard [67] defines a commonly-used set of global features that fit this description. Their performance is extremely limited since they extract characteristics of an entire image, but they are fast to search and compute, reducing the distance complexity from $O(3M_1N_1 \times 3M_2N_2)$ to $O(c)$.

**Subimage feature** approaches typically formulate annotation as a multiple-instance labeling problem. Each region, or “instance,” is characterized by a particular feature, and an annotation applies to an image if it applies to any of its instances. These techniques are called “bag-of-features” techniques if structure or spatial information of the regions is discarded, and tend to outperform global techniques at the cost of complexity. Boutell *et al.* [7] test a multi-label classifier that treats the multi-label problem as a series of single-label SVM detection problems. These subimage techniques are effective but are very sensitive to subimage definition, that is, the segmentation algorithm used to create regions or the size of the blocks used as subimages. Yixin Chen *et al.* [10] also have a subimage approach, but do not strictly implement a bag-of-features model. They show good performance by using SVM modeling to select important features.

Algorithms using **local features** collect features centered around identified keypoints, and then perform pairwise matching to identify similar keypoints. The most famous of these approaches is found in the original scale-invariant feature transform
(SIFT) implementation [48]. In some cases local approaches find configuration correspondences, as found in work by Frome et al. [21]. Frome’s work represents the current state-of-the-art for classification on Caltech 101. Local feature-matching algorithms tend to suffer from the curse of dimensionality, as comparing hundreds or even thousands of features and incorporating spatial configurations or deformation modeling is extremely computationally expensive.

Model-based approaches, which attempt to reduce an annotation to feature primitives, are certainly important to developing an automatic annotation system. However, using one-keyword, one-model algorithms is computationally expensive since it requires that the system first glean representative features from a training set, and then compare this to the feature primitives of every image where the annotation possibly applies. Such an approach necessitates significant training information for every annotation. They are also extremely sensitive to feature selection (e.g., choice of color feature, or texture feature, or SIFT signature). Indeed, most of the literature focuses on finding and defining the best features for annotation. Research on learning applied in order to mine correlations from these feature-based annotations is explored in the next section.
2.2 Correlation Mining for Images

Building from the works described in the previous section which apply image annotations is a body of literature where correlations are used to amplify positive relations among annotations and diminish negative ones. Positively correlated annotations can be said to collaborate, that is, reinforce the confidence that each annotation is present, and are found in pairs such as “ocean” and “boat.” Negatively correlated annotations, on the other hand, can be said to compete, often resulting in a decision to discard one of the competing labels, as you might if an image had been automatically annotated with a pair such as “ocean” and “desert.”

The literature focusing on correlation mining addresses almost exclusively the multi-label problem where each image may have more than one annotation, as opposed to single-label annotation found in most of the citations in Section 2.1. Godbole et al. [27] first introduce the concept of correlation learning in a text-based framework. Kang et al. [39] take this concept of correlation mining and apply it to image annotation. Their work exploits correlations among annotations, allowing the simultaneous propagation of multiple labels. Their work shows an improvement over the previous models that motivates further research in correlation mining. Zhou and Zhang [100] present two methods for what they term “multi-instance multi-label” learning, where a bag of instances is used to find a set of labels. One method adds weak classifiers in an Adaboost-
Chapter 2. Multimedia Annotation and Graph Theory

like method to determine classifications. The other method first performs fuzzy clustering on the instances, and then does constructive clustering while preserving structure information in order to label the image. Parikh et al. [62] use a segmentation-based approach followed by modeling of each segment’s context, using relative location, scale, and co-occurrence in order to assign multiple labels to the image. Their results motivate an approach that incorporates spatial information over traditional “bag-of-words” models, of course at the expense of model complexity. However, their results are largely dependent on the performance of segmentation. In another multi-instance multi-label method, Zha et al. [97] develop a hidden conditional random field that formulates in a single step the interaction among multiple regions in an image as well as the multiple competing or collaborating annotations. The approaches referenced here deal with small dictionaries and small test sets. In the next section we explore the literature that deals with image annotation on web scale-sized datasets.

2.3 Associated Text and Web Mining for Images

The Internet has offered a large amount of new information that can be leveraged in learning. In addition to providing billions of images, it also offers context for many of them in the form of surrounding text, owner annotation, titles, and alt-text to name a few. Work in the literature has attempted to exploit this web information to improve
image annotation. Wang et al. [86] perform a sort of clustering on words in text surrounding web images that results in good annotation performance. Using this technique they are able to find correlated sets of annotations effectively, but their work is limited to a relatively clean dataset of largely landscape-style professional photographs. Torralba et al. [74] show that with a sufficiently large database (80 million images) single-label annotation can be done using a dictionary of over 75,000 words using very simple nearest neighbor techniques on low-resolution versions of images. Their approach shows extreme scalability and is very effective; its major limitation lies in that it is only capable of applying singular labels. It relies heavily on the principle that with large enough databases, the closest photo is likely to be of the exact same thing. Xu et al. [93] develop a model that learns weights for the different sources of Internet text (e.g., surrounding text, captions, and file names). Their work also uses the web to find related tags learned on the Flickr database. The concept of their work, using multiple textual cues to annotate, is novel but improvements to more basic algorithms are not overwhelming.

This thesis addresses the problem in the realm of geographically-tagged (i.e., geotagged) images for which there is an associated latitude/longitude coordinate pair indicating where the picture was taken. It deals with image annotation exclusively in the area of social media databases, another area which has not been previously explored. It
formulates an annotation by mining method specifically suited to images with geotag metadata, but generalizable to other methods. This research is described in Chapter 3.

### 2.4 SVM Models for Video Annotation

Video annotation has traditionally been treated largely as an image annotation problem, using keyframes as instances and then collecting these keyframe annotations to create the complete video annotation. Typical methods for video annotation developed through the TRECVID collaboration [78] began using supervised learners, specifically Support Vector Machines (SVMs), to learn a pre-defined and small set of concepts from labeled training images. SVMs emerged from this benchmarking dataset as the “learner-of-choice” for video data, proving more effective than many other types of learners that were initially explored. Extending from this single SVM for annotation, work performed by Yan et al. [71] trains two SVMs using different features, adding the most confident predictions to the labeled training set and then training and classifying using the other SVM iteratively until all targets have been annotated. They show significant improvements using this combination as compared to just one SVM.
Chapter 2. Multimedia Annotation and Graph Theory

2.5 Correlation Mining for Videos

Much of the literature in video annotation extends the single annotation SVM models described above to mining for correlations in the initial SVM model predictions, by allowing associated words to reinforce each other (e.g., “boat” and “water”) and negative correlations to suppress annotations (e.g., “ocean” and “airplane”). For example, Natsev et al. [55] focus on the correlation mining that can occur using model vectors created using a low-level visual method. Their work shows the marked improvement accomplished by allowing annotations to interact over independent annotation decisions. Lavrenko et al. [44] utilize both vision and mining to construct a joint probability of visual region-based words with text annotations, incorporating co-occurrent visual features and co-occurrent annotations. Their approach is largely effective though its scalability is unproven. Ghoshal et al. [26] present an annotation scheme for a limited number of concepts driven by labeled training data and a Hidden Markov Model where each state corresponds to a particular annotation. Performance rivaled the contemporaneous state-of-the-art for TRECVID data, though for certain annotations its performance is much worse. Another supervised approach that relies heavily on data mining methods can be found in [63], which mines correlated annotations, such as “mountain” and “outdoor,” as well as cross-modal correlations between visual and textual features. Yan et al. [94] develop a supervised learning algorithm that uses decision trees to mine
correlations in annotating around 30-40 concepts. Their work shows the power of using an ensemble of classifiers for annotation. Tang et al. [72] use a neighborhood label propagation technique to spread one of a few concept annotations to videos. Their work is novel in incorporating consistency assumptions over unlabeled neighborhoods. Tseng et al. [79] learn annotations by fusing speech models, visual models, and frequent sequential pattern models in order to detect 130 concepts. Their work notably incorporates a hybrid method for annotation and shows great improvement over other models for annotation.

Each of these supervised methods use data mining for correlation discovery within a small annotation set. They require a model to be built on training data for each annotation, so it is impossible to “discover” new annotations for which there is not training data. Mining is limited to correlation analysis within a fixed dictionary. On the other hand, Velivelli et al. [82] use the Automatic Speech Recognition (ASR) results to mine a corpus for annotations. However, the mining step is limited to vocabulary creation for the entire database, rather than in finding a video-specific vocabulary for a target. Xing et al. [92] focus on topic discovery in a video dataset using multi-wing harmoniums, where each harmonium is derived from a different mode. Xing’s work does not explicitly address the accuracy of annotations associated with the general topics they discover, though it does indicate that certain topic words can be extracted. The specificity of these words is limited, however, in that only general topics with a
large number of positive examples are extracted. This thesis presents for the first time work that emphasizes new annotation discovery that is specifically tailored to the video of interest. An approach is tested on two unique datasets: one dataset uses transcript mining to determine appropriate annotations for news video, and the other uses user-generated tags to reinforce and filter video annotations from a social media dataset.

As we have seen in this chapter, there are many existing multimedia annotation techniques. A few major trends include bag-of-words models, LDA models, and Markov processes. In the next section, we explore how graph theory techniques have been used in the field, which have been gaining popularity as learning tools in multimedia annotation, as they offer an effective way to combine the different modalities available for multimedia data (e.g., image, text, time).

## 2.6 Graph Theory Techniques

Graph theory is an area of research dedicated to using graph structures to model relationships between objects. These graphs are characterized by a collection of nodes or vertices that have associated attributes, and edges that may have associated weights and may be directed or undirected. Graphs can be used for annotation by creating a structure with the objects to be annotated as nodes, the annotations as attributes of these nodes, and then edges that connect those nodes as some measurement of affinity between the
Chapter 2. Multimedia Annotation and Graph Theory

<table>
<thead>
<tr>
<th>Physical Analogy</th>
<th>Graph Analogy</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>vertex or node</td>
<td>video</td>
<td>$x_i$</td>
</tr>
<tr>
<td>weighted edges</td>
<td>similarity between videos $i$ and $j$</td>
<td>$W_{ij}$</td>
</tr>
<tr>
<td>initial attribute $b$ at node $i$</td>
<td>frequency of word $b$ in transcript or indicator that owner gave annotation $b$</td>
<td>$Y^b_i$</td>
</tr>
<tr>
<td>subsequent attribute $b$ at node $i$</td>
<td>score for word $b$ after updating with frequency in neighboring video</td>
<td>$f^b_i$</td>
</tr>
<tr>
<td>attribute $b$ relevance over all nodes</td>
<td>relevance of annotation $b$ for all videos in graph</td>
<td>$f^b = [f^b_1, f^b_2, f^b_3, \ldots]^T$</td>
</tr>
<tr>
<td>stable attribute over nodes</td>
<td>annotation relevance for stable graph</td>
<td>$f^{b*}$</td>
</tr>
</tbody>
</table>

Table 2.1: Analogy between graph theory terminology and unsupervised video annotation system, with notation.

nodes. Ultimately, the point is to find some sort of stabilized graph structure that shows a level of consistency in the annotations over the different nodes. The notation and a summary of the analogy between the annotation problem and graph theory is given in Table 2.1.

Graph techniques used for annotation are typically considered to be semi-supervised learning, a special case of supervised learning where the distribution of unlabeled points is incorporated into the learning process. Zhu [101] provides a survey of different types of semi-supervised learning, with emphasis on graphs, many of which are explained here. The semi-supervised nature of graph theoretic structures results in performance that typically beats out supervised methods.
2.6.1 Graph Theory in Image Annotation

Graph theory has arisen as a way to propagate and reinforce image annotations initially suggested by low-level feature techniques [73, 39, 47, 66]. Zhou et al. [99] present a regularization framework for finding the best annotations for a collection of text documents. This work is described in more detail in the next section. Tong et al. [73] extend this framework to image data using two graphs, and Liu et al. [47] build further on this work. Kang et al. [39], rather than independently propagating each label, use correlated label propagation to spread multiple labels simultaneously which gives effective results. Rui et al. [66] address annotation using graphs on a web scale, by using a bipartite graph model to annotate web images using surrounding text, and thus show effective use of graphs on a web-scale dataset. The graphs serve to find strong correlations between annotations, regardless of image information. Recently graphs used for multimedia annotation have taken the form of a hidden conditional random field [97], which formulates in a single step the interaction within and between visual primitives and annotations, showing effectiveness in a small dataset. This thesis extends work on image annotation using graph theory by exploring ways smoothness measured on an image graph can be used to identify tag types in Chapter 4.
2.6.2 Graph Theory in Video Annotation

As video annotation is often invoked as a collection of image annotation problems, graph theory has been used to address this problem as well. The foundation of such a framework is presented by Zhou et al. [99]. He presents a smooth manifold ranking theorem on a single graph to solve the problem on text data, in essence creating a diffusion kernel for the labels. The diagram in Figure 2.1 outlines the basic flow. The system is initialized with some set of labels $Y$, that are then spread over the connected nodes until convergence at $F^*$. This formulation will be shown mathematically in the next subsection.

\[
F(0) = Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix}, \quad F(1) = (1-\alpha)Y + \alpha SY, \quad F(T) = (1-\alpha)^{T-1}Y + (\alpha S)^{T-1}Y, \quad F^* = (1-\alpha)^{T-1}(I-\alpha S)^{-1}Y
\]

**Figure 2.1:** Diagram for Zhou’s smooth manifold ranking on a single graph. System is initialized with the original labels and with some set of edge weights derived from similarity. System iterates until convergence at a stable, smooth set of labels. This system models basic single-graph reinforcement.

Tong et al. [73] provide a solution for annotating using two graphs; their algorithm provides a coherent solution for combining graphs with the same nodes but connected by different edge weights. Wang et al. [85] extends this method to a larger number of
graphs and shows its use for video annotation. The basics of the multi-graph problem are described in Section 2.6.3.

**Single Graph Reinforcement**

First consider the technique described by Tong [73] using a single graph. The problem is formulated by a set of vertices (or nodes) and edge weights. Vertices, as applied in this thesis, are some form of multimedia document, and the edges between those vertices represent the similarity between the documents, as shown in Figure 2.1. This similarity can be measured using any singular feature distance; it may be affinity in text, visual, audio, or concepts/semantic space, for instance. A table summarizing the analogy between the graph and the video data is given in Table 2.1. The sequence will be described in the following paragraphs and can be seen visually in Figure 2.1.

Once the vertices and edge weights have been defined, the problem becomes finding the most stable graph structure, which represents a balance between *sticking* to the initial graph and *smoothing* the graph such that node attribute differences are proportional to the edge weight that connects them. Before stabilization, this may not be the case as the attributes are the result of some noisy initial state. Therefore, each node is to influence its neighbors such that the node attributes vary proportionally with distance in space defined by edge weight. Finding the stable graph structure designates the most stable feature attributes, \( f_i^{ba} \), at each of the vertices, \( x_i \), that vary proportionally with
edge weights. In finding a “stable” structure, the idea is to smooth the attributes (annotations), of each node over the space: the attributes of one node should be similar to those of a node nearby.

This graph reinforcement formulation can also be seen as combining a collection of weak classifiers to produce a stronger one. Each document can be seen as an error-prone annotator. These weak classifiers from the individual documents are then combined using a graph stabilization technique to produce a stronger annotation.

The similarity matrix of the graph, $W$, has elements $W_{ij}$ that are the edge weights on the fully-connected graph between vertices $i$ and $j$. Typically this distance is normalized, or shaped, using a radius parameter $\sigma$. The elements of $W$ are given by

$$W_{ij} = \begin{cases} 
\exp \left( -\frac{d(x_i,x_j)}{\sigma} \right) & \text{if } i \neq j \\
0 & \text{else}
\end{cases} \quad \text{(2.1)}$$

where $d(x_i, x_j)$ is some distance function between $x_i$ and $x_j$. This affinity metric, proposed in [99] and often used in the literature, is strictly positive and has the quality that close features receive a high weight.
2.6.3 Multiple Graph Reinforcement

Multi-graph learning has been used for situations where more than one feature, mode, or distance measure can typify similarity. For instance, some media forms can quantify visual similarity and audio similarity, but it is not intuitively obvious how to effectively combine these two disparate metrics when producing an annotation. Combination of multiple metrics or features can be done through a weighted combination of the stable graphs generated by each metric, as seen in Figure 2.2. In some situations, this amounts to a weighted bagging technique, where each graph represents a classifier and they are combined to create a stronger prediction. The problem in multi-graph learning becomes solving for $\alpha_g$, the weighting terms. In a totally naive case, $\alpha_1, \ldots, \alpha_G$ can be set to equal values, or in a case of perfect knowledge set to 1 for the best graph and 0 for all others. Wang et al. provide an EM-style maximization framework for finding ideal weights. This thesis presents a smoothness measurement that can be calculated on each of the graphs to provide an estimate of the quality of the graph. Intuitively, smoother graphs which show a great degree of intra-similarity in term space for a great degree of visual feature similarity, are better models of the particular node $x_i$ or the particular term $b$. The formulation proves to effectively handle many of the challenges of multimedia annotation and social media using graph theory, as will be illuminated in the following chapters.
Figure 2.2: Diagram for multi-graph learning. Each graph is created using a particular modalities, and then combined using weights $\alpha$ for final annotation.
Chapter 3

Collaborative Annotation of Geotagged Photos

This chapter addresses the multimedia annotation by mining problem applied to geotagged images found on social media sites. Specifically, it addresses the invocation of the problem found in personal photo labeling, formulating an effective mining algorithm that can be used to extend annotations and glean representative terms of a geographic area. Advancements in affordable cameras, bandwidth, and storage have allowed digital photo sharing to boom, much of which has been done at online repositories such as Panoramio [61], Flickr [19], or Webshots [87], where individual contributions have aggregated into massive, unorganized datastores. In these communities unstructured annotation of objects, known as tagging, is an important data source to be
analyzed along with conventional metadata. At many of these sites, users can provide
a geotag by dragging the photo onto a map interface, or extract the geotag from EXIF
data provided by a GPS-enabled camera or camera phone. This chapter attempts to
annotate geotagged photos from a social media site by learning from the community
as a whole, and, as a consequence, provides understanding about urban and regional
locations through the utilization of community annotations. It appears in large part in
[43] and [50].

The proposal takes a geotagged photo, and then using visual information along
with the location that the photo was taken, suggests annotations for the image. Section
3.3 describes general tag suggestion using mining techniques. Section 3.4 explains
a method for filtering and reranking for geographically-aware tag suggestion. This
formulation is compared to an image similarity baseline, using only visual features
to detect annotations, and a geographic baseline driven exclusively by geographic tag
prior. These experiments are provided in Section 3.5. The algorithm for geographically-
informed tag suggestion is then reformulated as a probabilistic maximum a posteriori
(MAP) problem using a global annotation dataset in Section 3.6. The mathematical
formulation is given in Section 3.7. This work is tested in Section 3.8 and discussion in
Section 3.9 concludes the chapter.
Chapter 3. Collaborative Annotation of Geotagged Photos

3.1 Related Work

Interesting applications have emerged which utilize the owner-supplied annotations in geotagged image data. Notable works include extraction of canonical landmark views [42] and geographical image features [16], and automatic landmark identification [64] and [98]. Also related to this chapter, Naaman et al. developed a system that suggests a geographic or text annotation by searching a communal database [54]. Other work has mined geotag traces for event identification [96].

Directly relevant to the aim of using georeferences to infer labels, Joshi and Luo [38] have presented work that quantifies the probability of a particular activity or event that has a relevant geographic footprint (i.e., events such as “hiking” but not “party”) by learning the likelihood of the event conditioned on geotags, text tags, and visual features. Their experiments on a manually cleaned Flickr dataset show that fusion of geographic and visual information can improve results for classification of geotagged photos in some cases. The challenges elucidated by this work include the difficulty of automatic tagging for a large variety of words as well as the difficulty of noisy image-sets derived from social media. Hays presents a system for predicting geocoordinates by clustering nearest neighbor results from a very large database in a process called “IM2GPS” [31]. Crandall et al. extend this work by adding text features (from tags) to the IM2GPS algorithm for predicting image geocoordinates [15]. Kennedy et al. use
community annotations and clustering to generate geographic (e.g., landmarks) and temporal (e.g., events) labels [41]. These labels are employed in World Explorer for visualization [1]. Boutell and Luo perform a study on incorporating camera metadata, including exposure time, use of flash, and focal length, into classification [6], but their work stops short of incorporating GPS metadata.

As explained in Section 2.1, in contrast to typical annotation systems based on computer vision models, the system described in this chapter does not rely on learning particular vocabulary terms. It is therefore scalable to any dictionary size. It is also the only work to our knowledge which specifically addresses the problem of efficiently tagging images on a worldwide scale. The technique of applying tag propagation by mining large datasets to annotate is related to work in image annotation by mining [86] though this work approaches the problem using a different algorithm and with a different sort of dataset.

While authors such as Sheng et al. have explored collaborative tagging using synthetic data [68], the question of effective learning using user-supplied information has not been explicitly tested. This chapter harnesses the large amount of freely contributed overlapping annotation to address consistency and spam issues [49, 12, 24] in social media in an automated learning process.
Chapter 3. Collaborative Annotation of Geotagged Photos

3.2 Geotagged Image Annotation Algorithm

The annotation algorithm described in this chapter, which will be referred to as the “dual visual and geographic method” or its name used on the web, “SpiritTagger,” suggests tags which capture the spirit of a location. As an illustration, Figure 3.1 shows a cloud of representative tags for Los Angeles with font size proportional to local importance. While a simple neighborhood finder or geographic information system (GIS) could discover tags such as “Santa Monica,” those systems would not necessarily suggest tags such as “sunglasses” or “shopping” that are likely to be present in a photo from a particular area with certain characteristic visual features. SpiritTagger accomplishes this by leveraging the local prevalence of tags compared to a global distribution, either explicitly by reranking, or implicitly by evaluating annotations as relevant or not given a particular geographic location.

The workflow for a collaborative annotation method that mines for annotations can be found in Figure 3.2. This system addresses tag filtering and tag extension by employing geographic pattern mining. The process begins with a user uploading an image along with an accompanying latitude-longitude coordinate pair. These can typically be provided by GPS-enabled devices or by dragging a photo onto a map interface, a feature found at [19]. Based on the new image, the system assembles visually relevant photos weighted by geographic distance from the input image. A candidate set of tags
Chapter 3. Collaborative Annotation of Geotagged Photos

Figure 3.1: Cloud of prevalent tags extracted for the urban region of Los Angeles. Font size is proportional to the learned tag “importance” in the region. Important tags are not limited to place names. There are also relevant scene categories like “freeway” and “crosswalk” and objects like “skateboard” and “lifeguard.”

is then collected and scored from these photos, using either a reranking method that boosts relevance scores for \textit{geographically representative} tags or a probabilistic algorithm that maximizes MAP of a tag given a photo’s visual and geotag features. The highest scoring tags can then be suggested to the user for annotating her photo. The reranking methodology is described first in this chapter and largely appears in [50], followed by the probabilistic formulation much of which appears in [43].
Figure 3.2: Workflow for georelevant tag suggestion. A similar set of images is identified that are close to the visual content and geotag of the upload. The system mines the similar set of photos for geographically relevant annotations and suggests them to the user, using one of two formulations. In the first formulation, tags are scored and then reranked by georelevance. In the second, tags are ranked using a probabilistic formulation that incorporates geographic information.
3.3 Tag Suggestion By Mining

In Section 2.1 we discussed many algorithms that annotate images by applying a learned computer vision model. However, these algorithms are most successful for clean and object-oriented image sets since they are largely detection-based algorithms that treat annotations singularly. Photo-sharing websites, populated by real-world tourist photos, will consist primarily of images with cluttered, natural scenes which pose significant problems (see Section 1.2.1).

Well-known computer vision algorithms, such as that found in [7], do not utilize the large number of community-provided annotations which can provide a way to mitigate some of the aforementioned difficulties. This work proposes to annotate a photo by mining a database for similar photos that offer geographic and visual relevance. The collection itself provides the source for tag suggestions which can then be offered to the user. It's possible to incorporate this into an upload tool that allows for annotation with a simple mouse click, as can be tried at [84]. While annotation by mining has been suggested in the literature by researchers in image annotation [86], this system also uses geographical context information in addition to visual image similarity to annotate photos. We begin the discussion of creating an annotation-by-mining tool in the next subsection with a description of a geographic baseline that uses the geotag
only to annotate a photo, and then move into systems that incorporate visual features and georelevance reranking.

### 3.3.1 Tag Suggestion by Geographic Prior

Perhaps the simplest suggestion tool for geotagged photos considers those images within a certain radius of the candidate photo. Finding a geographic prior amounts to collecting the frequency at which an annotation is seen for other images within a certain radius \( r \). One pitfall is that without filtering, a single user’s annotations over multiple photos can often erroneously share the same GPS coordinate. This commonly occurs when a group of photos are dragged from an album onto a single location in a map interface for geotagging. When this occurs close to a target photo, these noisy tags may dominate the annotation scoring. Therefore, a limitation is introduced, requiring that a particular user \( U_i \) “votes” for a particular annotation no more than once. Consider the set of users \( U = \{U_1, U_2, ... U_{|U|}\} \) who have at least one image in the set of geographically close images for the target photo. Then, the collection of annotations for similar images from each user \( U_i \) is used in suggestion, call them \( A_i = \{b_1^{(i)}, b_2^{(i)}, ...\} \). The annotation is given a score based on the number of users that provided that tag. A tag suggestion tool using geographic radius scores an annotation \( b \) with the formulation:

\[
S_b = |b \cap \{A_1, A_2, ... A_{|U|}\}|
\] (3.1)
Annotations with a high score, meaning many users in geographic proximity have applied those annotations, can be supplied as suggested tags to the user.

### 3.3.2 Tag Suggestion by Visual Features

Simple geographic mining does not use the full power of a georeferenced photo set; in fact, it is really just a description of a particular area as gleaned from image annotations. Building on the geographic mining formulation, image similarity can be integrated to pare down candidate photos to a visually similar set that can be mined for annotations. Such an attack has been proposed and shown to return relevant photos in systems such as MediAssist [58].

In a visual formulation, global color, texture, edge features and local SIFT [48] features are extracted for the set of images within a certain geographic radius from the target photo. The N-nearest neighbors in visual feature space are retained, and a scoring system similar to that in the Section 3.3.1 is performed. The set of unique users, \( U \), are collected with the associated user-annotation sets, \( \{A_1, A_2, ..., A_{|U|}\} \). Frequent annotations are offered as suggestions to the uploading user. An additional similarity term can be included that weights visual neighbors by their visual distance from the target photo. Using the notation \( \alpha_v \) for this visual similarity term which is a function of the visual feature distance between the target and image \( i \), the annotation score is now:
Chapter 3. Collaborative Annotation of Geotagged Photos

\[ S_b = \sum_{i=1}^{n} \alpha_v(i) \mid b \cap A_i \mid \quad (3.2) \]

The dual method our work proposes proceeds from this visual method. As described in the next section, tags are boosted when found to be relevant semantically in a wider location such as a city or region.

### 3.4 Tag Reranking by Georelevance

Reranking has been used in the literature to re-order a list of search results for a particular query. Hsu et al. [34], for instance, developed the “Information Bottleneck” principle to re-order video results for queries, using the notion that the top results are more likely to be correct, and videos similar to them should be boosted to the top of the ranked list. In the following section, we describe how reranking can be used to emphasize the georelevance of certain terms.

The premise of this formulation is that there exists for geographic regions a set of representative tags which can be derived by comparing their frequency of use in an area. In order to compensate for general terms, we do the comparison as the frequency within a certain area, as compared to the frequency that tag is used anywhere. Using this principle we can rerank tag suggestions, given an image query, in a way that reflects frequently-used terms in the area, giving a sense of the local spirit of a place.
Chapter 3. Collaborative Annotation of Geotagged Photos

While previous work [1] utilized tag distributions in a geographic area in order to find representative tags for visualization on a map, this work goes further to incorporate georelevant terms into the actual annotation algorithm.

Tag georelevance is primarily calculated from the ratio of tag frequencies between the region of interest and globally as measured by the number of unique users of a tag. To illustrate, Figure 3.3 compares unique user frequencies for a set of tags crawled from Flickr in an area restricted to Los Angeles and a set of tags crawled globally. Tags with high frequency in LA but lower frequency globally, such as “cars,” “freeway,” and “palm,” will be granted higher georelevance.

To further ensure the usefulness of the tag ratio information, two terms are added to the georelevance equation which serve to filter out tags that are incorrect or unuseful. The first term penalizes tags that do not occur very often both globally and locally by the minimum in number of users of a tag in the two compared areas. The second term penalizes tags which correspond with very specific geographic locations by considering the ratio of the standard deviation of geographic coordinates among the set of geographic coordinates for each use of the tag to the maximum such standard deviation for the area.

One formulation of tag georelevance, \( \alpha_g \), as a function of tag \( b \), consists of three linear terms:
\[ \alpha_g (b) = \log \left( \frac{f_{\text{local}} (b) + 1}{f_{\text{global}} (b) + 1} \right) + \lambda_1 \cdot \varsigma (b) \]
\[ + \lambda_2 \cdot \left( \frac{\sigma_{\text{geo}} (b)}{\max (\sigma_{\text{geo}})} \right) \]  

with \( \varsigma \), the minimum use penalty term, given by:

\[ \varsigma (b) = \log (\min (F_{\text{local}} (b), F_{\text{global}} (b)) + 1) \]

\( f \) is the frequency as measured by the normalized number of unique users per tag, \( F \) is the unique user tag frequency without normalization, \( \lambda_1 \) and \( \lambda_2 \) are weighting factors set at values 0.25 and 0.15 respectively as found to work well in experiments, and \( \sigma_{\text{geo}} (b) \) is the standard deviation of the GPS coordinates of all images that have tag \( b \) (taken as a sum of latitude and longitude statistics). The first term will be high for tags that are used frequently in an area compared to globally and low for those that do not. The minimum use penalty term will be high for tags that are used frequently and 0 for those not used. The final term will be high for those terms that are spread out geographically and low for those that apply to images located at a singular point.

This local tag frequency information is used to further improve the scoring formulation given in the previous section and in Equation (3.1). In particular, the scores for
Chapter 3. Collaborative Annotation of Geotagged Photos

3.5 Performance of Georelevance Reranking

The goal is to determine how well the reranking algorithm for tag suggestion performs compared to baseline methods, and how factors like geographical radius (which can be a user control or system setting) and number of nearest neighbors vary the reranking.

Figure 3.3: Twenty ordered tags shown to demonstrate tag frequency differences between Los Angeles region and globally. Tags such as “getty”, “cars”, “freeway”, and “palm” with a higher normalized frequency in Los Angeles are weighed more.

annotation $b$ are reranked using the formulation:

$$S_b = \alpha_g(b) \sum_{i=1}^{\left| U \right|} \alpha_v(i) \left| b \cap A_i \right|$$

(3.5)
Chapter 3. Collaborative Annotation of Geotagged Photos

To do so, two regions are first selected with good coverage: a dense urban section of Los Angeles and the larger region of Southern California. A total of 116,281 geotagged images from Flickr using their API were then crawled. 25,988 of these images were randomly selected from anywhere globally, 31,361 were limited to the Southern California geographic region (between $32.5^\circ$ and $35^\circ$ latitude and $-120.6^\circ$ and $-114.6^\circ$ longitude), and 58,932 were selected from within the Los Angeles geographic area (between $33.7^\circ$ and $34.3^\circ$ latitude and $-118.5^\circ$ and $-117.9^\circ$ longitude). The set of images contained over 48,000 unique tags.

As a testset 99 images from the Los Angeles city data set and 100 images from the Southern California region were selected as candidates for tag suggestion. The images were randomly selected while rejecting images that were overexposed, blurred, or possibly containing privacy concerns.

3.5.1 Relevance/Coverage

A standard precision-recall metric does not accurately reflect the performance of annotation, as annotations do not fit neatly into a true/false categorization. Rather, they fit into a range between “relevant” to “irrelevant,” as well as “incorrect.” Incorrect words are those that are certainly wrong, perhaps a place name that is not where the image was taken or an object that is not contained in the image. An irrelevant annotation is one which is unuseful (e.g., “geotagged”) or its correctness cannot be gauged.
Chapter 3. Collaborative Annotation of Geotagged Photos

(e.g., names that identify people in the image). An evaluation metric is adopted which provides a tag \( b \) with a score, \( c_i \), of +1 for a relevant tag, 0 for an irrelevant tag, and −1 for an incorrect tag. “Relevance” and “irrelevance” are judged based on whether a typical user would use that word in a query seeking that image. Thus, a modified precision metric, called relevance, is defined as the average score of the \( K \) extracted tags, 
\[
P = \frac{1}{K} \sum_{i=1}^{K} c_i.\]
A similar three-class scoring method has been adopted elsewhere in image annotation [86], though it diverges in assigning an uninformative score of 0.5.

Additionally, the set of appropriate tags is not limited, and therefore a standard recall metric cannot be used. Instead, a running list is kept of all “relevant” annotations for an image encountered using any method incorporated in this chapter. Then, a recall-like metric is adopted, called coverage, that indicates the percentage of all seen positive annotations \( A \) covered by the method: 
\[
R = \frac{|S \cap A|}{|A|},\]
where \( S \) is the set of tags extracted using the particular method. The best metric has the greatest area under the relevance/coverage curve, exhibiting high relevance without expending coverage. An example set of tags using the methodology described in this section, as well as the annotations the owner applied, can be examined in Figure 3.6.

3.5.2 Groundtruth Annotation

The groundtruth annotation was done by a team of 15 annotators, nearly all of whom were totally unfamiliar with the algorithms or problem description. To collect
Chapter 3. Collaborative Annotation of Geotagged Photos

groundtruth, a web-based tool presented a random photo from the test set along with a Google street map centered at the test image’s GPS coordinate. Annotators were then asked to score 10 tag suggestions provided by SpiritTagger, the baseline methods, or the user-supplied annotations. The 15 annotators were instructed to label each suggested tag as either “relevant,” “irrelevant,” “incorrect,” or “unsure.” Instructions to the annotators included guidelines such as: a) place names can be relevant if correct b) phone numbers and people’s names are irrelevant. Annotators could reference the web in order to determine if an annotation is correct. Tags labeled as “unsure” are not used in scoring the experiments. In total, 16,540 groundtruth annotations were collected; this meant 89% of the suggestions made by SpiritTagger have been manually delineated as “correct,” “incorrect,” or “irrelevant” for the photo.

In order to judge the algorithm’s ability to suggest tags that are geographically relevant, the tag suggestions are compared to those generated by a geographic baseline and a visual baseline constrained to the same geographic area.

3.5.3 Geographic Baseline

The geographic baseline collects photos within a certain geographic radius, as described in Section 3.3.1. Experiments are performed over a radius of 10km, 1km, 100m, and 10m. Results in Figure 3.4 show a tradeoff between relevance and coverage when varying the radius. A large radius provides greater coverage, since many keywords can
be discovered when using a greater geographic range. On the other hand, a small radius such as 10 or 100 meters shows larger relevance since the included photos are likely to contain applicable tags. However, a smaller radius will not cover the breadth and variety of keywords that a large radius over a more diverse set of images will. Furthermore, using a large radius in an urban area can lead to incorrect top results that refer to popular but distant neighborhoods, such as labeling a photo from nearby Venice Beach with “Santa Monica.” This observation appears as low relevance/low coverage points to the left side of the $r = 1$km and 10km curves.

A formulation that scores tags with an exponentially decreasing weighting element for candidate tags based on increasing geographic distance, rather than taking all tags equally within a radius, is also tested. Surprisingly, experiments find that this causes a slight decrease in performance. This may be due to noise in the degree of exactness of geotags.

### 3.5.4 Visual Baseline

The visual baseline finds close visual neighbors within a certain geographic distance, as described in Section 3.3.2. Similarity is formulated using a metric exponentially decreasing with increasing image feature space distance, namely,

$$s_{ij}^f = \exp\left(-\frac{d_f(x_i, x_j)}{\sigma_f}\right).$$

(3.6)
Figure 3.4: Graph showing performance of geographic baseline by size of radius. As expected, using a small radius (10m) has high relevance but does not show the coverage larger radii would. A large radius (1km or 10km) provides better coverage. Large radius shows mainly incorrect or irrelevant top returns, however, shown by its low precision at low coverage.

using a late fusion of feature vectors, $f$, that have been normalized to unit value standard deviation along each dimension. For the similarity measure, the decay constant $\sigma_f$ is set to the standard deviation of the distance metric used for that feature as seen in the experimental data. Overall similarity between images $i$ and $j$ is given by a linear combination of feature similarities as $s_{ij} = \sum_f \alpha_f s^f_{ij}$.

Three global features and one local SIFT feature are used, with weights $\alpha_f$ set to 0.033, 0.033, 0.033, and 0.9 by order of enumeration:
Chapter 3. Collaborative Annotation of Geotagged Photos

1. **Edge Distribution Histogram** The **EHD** is an 80-dimensional feature consisting of histograms of gradient orientations computed from the image tiled in a 4x4 grid, as described in the MPEG-7 standard [67]. Each histogram contains 5 bins and consists of the magnitude response of a filter.

2. **Homogeneous Texture Descriptor** The **HTD** feature captures the statistics (mean, variance) computed across the image from the response of a bank of 24 oriented Gabor filters [67]. The resulting descriptor has 48 dimensions.

3. **Color Layout Descriptor** The **CLD** is characterized by an 18-dimensional descriptor, consisting of three 6-dimensional coefficients from the DCT of each color channel in YCbCr space [40].

4. **SIFT Signature** The **SIFT** feature represents the SIFT descriptors [48] extracted at 5000 random keypoints [57] and pushed through a vocabulary tree with 4 levels and a branching factor of 10, as advanced by Nister and Stewenius [56].

The MPEG-7 descriptors (EHD, HTD, CLD) are computed using slightly altered code available from [67], and a modification of the code provided by Vedaldi [81] is used for the SIFT signature extraction. The SIFT signature, which showed the best performance when results are casually examined, is weighted the most. This visual baseline is tested using various similar image set sizes; as formulated in Equation (3.2),
we vary $N$ over 20, 10, 5. Evaluations show that performance improves when a large number of images, $N = 20$, is kept.

**Los Angeles, SpiritTagger Performance by NN**

Figure 3.5: Graph showing performance by number of images, $N$, contributing tags for suggestion. Better performance for higher number of images kept, though some performance loss at points of high precision/low coverage.

### 3.5.5 Georelevant Annotation

The results for annotating the two testsets as described above indicate the usefulness of learned georelevance in an urban area but also show it may be less so for larger, less densely-photographed regions. Some examples of the tags supplied by the algorithm, using the scoring formulation shown in Equation (3.5), are shown in Figure 3.6. The
examples show that many of the tags provided are ones actually used by the owner. Additionally, many keywords not used but deemed relevant are suggested. Thus, the geo-aware tag suggestion tool aids the annotation process both by making annotation faster (clicking rather than typing) and also by improving coverage of the keywords attached to a photo (i.e., giving it more relevant, or a more “complete,” annotations).

<table>
<thead>
<tr>
<th>Image</th>
<th>Owner tags (after upload)</th>
<th>SpiritTagger suggestions (before owner annotation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>trevor gordon, shortboarding sandspit</td>
<td>Barbara, Santa, santa barbara, surf, Pier, surfboard, Lemons, surfing, harbor, Surfer, Wednesday, channel islands, al merrick, boarding, maassen, bags, kneeboard, offshore, white, cockail, longboarding sandspit, hama, tyler anderson, dammcool, oysters, person, Media, Hurricane, picture, dean, wave, fun...</td>
<td>California, USA, Beverly Hills, Stores, Archipel, 90210, flagship stores, N. Rodeo Drive, boutiques, designer boutique, high end retail stores, 433 N Rodeo Drive, La Perla, La Perla boutique, <a href="http://www.laperla.it">www.laperla.it</a>, Rodeo Collection, Hermes of Paris, 2007, Vacation, Shopping, architecture, travel</td>
</tr>
<tr>
<td>USA, Vacation, travel, California</td>
<td>Anaheim, California, Angels, baseball, Twins, MLB, majorleaguebaseball, major, league, baseball, ballpark, Edison, field, California, geotagged</td>
<td>Anaheim, California, Angels, baseball, Twins, AngelStadium, MLB, major, Motorcycle, jones, Edison, majorleaguebaseball, kennedy, figgins, supercross, pierzynski, ama, league, davanon, ballpark, Racing, geotagged, minnesota, field</td>
</tr>
</tbody>
</table>

**Figure 3.6:** Example Flickr uploads, the tags the owner ultimately applied for the image, and the tags that could have been suggested by the reranking algorithm (bold for correct, plaintext for irrelevant, italics for incorrect). Note that the algorithm has no knowledge of the owner’s tags but suggests many of the annotations the owner eventually gives. The left example shows the ability to properly weight terms particularly applicable to Southern California, such as “surf” and “surfboard.” Middle example reveals a recognition of the frequency of upscale shopping in Los Angeles. Surfing and shopping are two associations that go beyond place or neighborhood labeling.

A plot of the relevance/coverage curves for the baseline methods as well as reranking are shown in Figures 3.19(a) and 3.19(b). One observation is that for the Southern California database, while the performance is slightly better than the visual baseline,
Chapter 3. Collaborative Annotation of Geotagged Photos

it is slightly worse than a simple geographic baseline. The performance loss may imply that the geographic relevance reranking performed by the method does not work as well in larger areas. Perhaps this is due to the term that suppresses tags not widely distributed geographically in the region. Additionally, the high performance of the geographic baseline in the Southern California study may be due to high learning of nonspecific but correct tags, such as “LA” or “California” or “USA.”

The advantage of georelevance reranking is primarily realized in the bump in the relevance/coverage curve in Figure 3.19(a) that occurs beginning around $P = 0.1$. This level represents the place where tags with score $S_b = 0.9$ begin to be included, and is therefore the point where annotations that occur only once in the similar set $V$ begin to be included. This point on the curve represents the location where the score formulation reduces to $S_b = \alpha_g(b)\alpha_v(j)$, simply the multiplication of the georelevance term with the visual similarity with one other photo. For images close enough, $\alpha_v$ will be similar, and the georelevance term $\alpha_g$ can dominate. Thus, the improvement occurs because the algorithm effectively boosts georelevant tags above irrelevant ones in the set of tags that appear just once.
Chapter 3. Collaborative Annotation of Geotagged Photos

3.6 Probabilistic Formulation

The reranking method in the previous section shows encouraging results particularly as seen in Figure 3.19(a). The local area in [50] is defined as a box bounded by a set of handpicked geographic coordinates, and the system is only tested for two general areas, “Los Angeles” and “Southern California.” How to effectively utilize geotag metadata to inform annotation decisions on a worldwide scale has yet to be demonstrated. This section reformulates the same problem on a worldwide scale and explores the effect of dataset density on the results. The remainder of this chapter presents the system shown in Figure 3.2, using georelevance formulation (b). With an uploaded photo and associated geotag, the system offers relevant annotations predicted by ranking estimated posterior probabilities derived from the geocoordinate and visual features of the 1.2 million+ global images in the database.

In this section, we will extend the concept of georelevant term propagation to a worldwide database. The system offers a method for choosing smartly which annotations of geotagged photos have visual relevance and for effectively combining them with geography-based annotations. These annotation decisions are made in a probabilistic framework of maximizing posterior annotation probabilities given a geographic and visual feature.
Chapter 3. Collaborative Annotation of Geotagged Photos

In order to learn and test an annotation system for geotagged photos, we first crawled 1.75 million georeferenced images covering the globe using the Flickr API and the methodology from [31]. Of the 1.75 million images, we are able to retain and extract features for 1.2 million images found to have suitable resolution and aspect ratio. Additionally, for each image we retrieve the following metadata: owner id, Flickr id, time taken, time of upload, title, tags, latitude, longitude, geotag accuracy as given by zoom level of map when geotagged, and public license information. The dataset has 65,679 unique users and 436,506 unique tags. The high number of photos per user compared to the Flickr average likely results from the rejection of photos during the crawl with tags indicating personal use. There are 29,652 tags which are employed at least 25 times and by more than one user.

To the four descriptors described in Section 3.5.4, we add a fifth:

- **Gist**: The GIST descriptor describes the spatial layout of an image using global features derived from the spatial envelope of an image. It is particularly powerful in scene categorization. The final descriptor is 512-dimensional [59]. We employ a C implementation of the MATLAB code provided by Torralba [75] for GIST extraction.

The rubric for annotating a geotagged photo using the probabilistic framework follows in Section 3.7. Experiments on these algorithms and conclusions wrap up the chapter in Sections 3.8 and 3.9.
3.7 Bayesian Annotation Suggestion

While previous work utilized tag distributions in a geographic area in order to find representative tags for visualization and knowledge extraction [1], we extend the idea by framing the problem in terms of optimal Bayesian Maximum A Posteriori (MAP) probability estimation for a set of tag candidates.

The image is described by a set of visual feature primitives, described in Section 3.6, which we will call $x$, and geographic information $g$ indicating the location that the photo was taken. Thus, for each tag $b$ we can derive a probability that the tag $b$ is applicable to the image as:

$$p(b|x, g) = \frac{p(b, x, g)}{p(x, g)}$$  \hspace{1cm} (3.7)

Several methods exist for calculating the posterior $p(b|x, g)$. We prefer non-parametric techniques which extend flexibility by allowing us to avoid expensive model calculations. Density estimation using k-nearest neighbors allows a direct calculation of the posterior using a fixed number of closest data points rather than by searching over a fixed volume [76]. We adopt k-NN density estimation to calculate the posterior imposing a kernel on each candidate within the search space. This method is described in the next section.
3.7.1 Non-Parametric k-NN Density Estimation

Non-parametric density estimation using k-Nearest Neighbors (k-NN) provides a way to estimate the posterior $p(b|x, g)$ and has been used in segmentation [76], video motion classification [3], and object classification in images [5, 18]. We can reformulate the posterior as:

$$p(b|x, g) = \frac{p(b, x, g)}{p(x, g|b)} = \frac{p(b, x, g)}{p(x|b) + p(x, g|b)} \quad (3.8)$$

Simple k-NN algorithms in the literature treat the $k$ closest images identically and derive a probability from the number of k-NN that are of class $b$, for instance, $p(b|x, g) = \sum_{i=1}^{k} \frac{I_b(X_i)}{k}$ where $X_i$ is the $i^{th}$ closest image to $(x, g)$ and $I_b(X_i)$ is an indicator function denoting whether image $X_i$ is of class $b$. More sophisticated algorithms apply a penalty, parameterized by cost function $K$, for the $k^{th}$-nearest neighbor’s distance from the given parameters $x$ and $g$, leading to a formulation:

$$p(b|x, g) = \frac{\sum_{i=1}^{k} I_b(X_i)K(x, X_i)}{\sum_{i=1}^{k} K(x, X_i)} \quad (3.9)$$

$K(x, X_i)$ is often formulated as $K(\frac{x-X_i}{h})$ where $h$ is a bandwidth parameter.
Non-Parametric k-NN Density Estimation with Geotags

Our algorithm first employs a rectangular window, $K_g = \hat{I}_g(X_i)$, whose size is a function of dataset density, around $g$ indicating whether $X_i$ is within a certain distance of $g$. The geographic region of influence, $\hat{g}$, is determined by the quadtree described in Section 3.7.2. A Gaussian with mean zero and unit variance is another commonly used kernel in feature similarity research, and we use a Gaussian kernel, $K_{x}(x, X_i) = \frac{1}{\sqrt{2\pi h}} e^{-\frac{(x-X_i)^2}{2h}}$ around the visual feature space $x$ to apply a contribution to the density estimate which drops with the distance between $x$ and $X_i$, where $X_i$ is the $i^{th}$ nearest neighbor. We formulate the multivariate Gaussian for each feature isotropically, denoted as $K(x) = \prod_{j=1}^{d} K(x_j)$ if $x = [x_1, ..., x_d]^T$. This leads to $K_{x}(x) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi h_j}} e^{-\frac{x_j^2}{2h_j}}$, and finally, the formulation becomes:

$$p(b|x, g) = \frac{\sum_{i=1}^{k} I_b(X_i) I_g(X_i) \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi h_j}} e^{-\frac{(x_j-X_{i,j})^2}{2h_j}}}{\sum_{i=1}^{k} I_g(X_i) \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi h_j}} e^{-\frac{(x_j-X_{i,j})^2}{2h_j}}}$$ (3.10)

Each dimension $j$ is scaled by the standard deviation of that dimension. We attempted to learn optimal $h_j$ per feature by employing gradient descent designed to minimize tag selection error on a held-out validation set. The $h$ parameters vary the “bandwidth” of each feature, and therefore attempt to appropriately scale the different visual feature distances. However, we did not see much difference in the various
local maxima that resulted from different starting conditions. The actual values of $h$
used are 4.5 for the EHD, 6.0 for the HTD, 12.0 for the GIST, 2.35 for the CLD, and
18.2 for SIFT signature. It is noted that $h$ is the same over all dimensions of a feature,
since the dimensions have already been normalized by standard deviation. Formulated
with these parameters, the result is that smoothing is somewhat proportional to feature
dimensionality.

### 3.7.2 Regional Representation Using a Quadtree

In order to efficiently retrieve the image objects when querying with geocoordinates, we employ a quadtree. A quadtree is a data structure formed by recursively
dividing data into four regions until a stopping condition is met. A quadtree adapts
to the source data, growing in areas where the data is dense and terminating where
data is sparse. Quadtrees have been used previously for watermarking [80] and object
recognition through inexact image matching [14]. Wu et al. [90] present a system that
performs content-based image retrieval by searching an updating quadtree that effec-
tively represents segmented region features. Grady and Schwartz [30] use a quadtree to
segment medical images. In most of these works, quadtree decomposition is used for
sub-image definition, that is, to determine regions in the image.

We build a quadtree on the worldwide image database of 1.2 million+ geotagged
Flickr images, crawled using the process described in Section 3.6, using the geoco-
ordinate tags for branching. The quadtree is grown by branching a central node into four equal-sized geographic quadrants until a stopping condition is met. We specified a minimum-support level of 100 images as the stopping condition: if a node contains fewer than 100 images with unique (user, latitude, longitude) triples, subdivision stops. This triple formulation is used because it usually corresponds with a unique set of text tags. Often, a user will apply the same text tags to a collection geotagged by dragging them onto the same location on a map. Each of the leaf nodes, then, represents a space that is inversely proportional to the density of photos taken in that area. Figure 3.7 shows image density and the quadtree overlaid on a map of the eastern United States. Denser regions, like New York City, have deeper nodes than sparse areas. The quadtree allows the system to quickly identify dense regions where the system can generate tag semantic scores with higher confidence. Each of these terminal nodes is considered to have enough images to robustly characterize the geographic space it covers, even in the presence of noisy geo- and text-tagging that results from use of voluntary user content. The geotag, \( \hat{\mathbf{g}} \), used in this chapter refers to the region covered by the terminal node that contains \( \mathbf{g} \), and represents a discretization of a previously continuous quantity.

Having presented the fundamental components of the posterior calculation, we provide a summary of the algorithm for the calculation using k-NN density estimation below in Algorithm 1.
Figure 3.7: A quadtree efficiently indexes the geotagged image distribution for efficient search over 1.2 million+ images. The tree’s node boundaries are the black lines mapped over the light blue dots representing image locations for an area in the eastern United States. Smallest rectangles represent the lowest-level nodes, and are found at dense, primarily metropolitan, areas.

3.7.3 Baseline Methods

We will compute two baseline methods for assessing the quality of annotation suggestions. A visual baseline will employ content-based analysis alone and a geographic baseline will employ the prior distribution of tags present in the node specified by \( \hat{g} \).
Chapter 3. Collaborative Annotation of Geotagged Photos

Algorithm 1  Algorithm for MAP estimation of a particular annotation given visual features and a geotag.

Input: a query image $Q$

1. Extract image features $x$ for $Q$
2. Identify the appropriate quadtree node and geotag $\hat{g}$ for $Q$ by pushing geotag down the quadtree to find terminal node
3. Collect the set of images $I$ that share $\hat{g}$
4. Collect the set of tags $B$ associated with $I$, and compute over each feature $f$ and each tag $b$,

$$p_f(b|\mathbf{x}, g) = \frac{\sum_{i=1}^{k} I_b(X_i)I_\hat{g}(X_i) \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}h_j} e^{-\frac{(x_j-X_{i,j})^2}{2h_j}}}{\sum_{i=1}^{k} \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}h_j} e^{-\frac{(x_j-X_{i,j})^2}{2h_j}}}$$

5. For each $b \in B$, compute $p(b|\mathbf{x}, g) = \prod_{f=1}^{5} p_f(b|\mathbf{x}, g)$

Output: a list of tag scores

Visual Baseline

The visual baseline assumes that the tags can be predicted by visuals alone. This formulation reduces to:

$$p(b|\mathbf{x}) = \frac{p(b\mathbf{,} \mathbf{x})}{p(\mathbf{x})} = \frac{\sum_{i=1}^{k} I_b(X_i) \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}h_j} e^{-\frac{(x_j-X_{i,j})^2}{2h_j}}}{\sum_{i=1}^{k} \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}h_j} e^{-\frac{(x_j-X_{i,j})^2}{2h_j}}}$$ (3.11)
Chapter 3. Collaborative Annotation of Geotagged Photos

The visual baseline amounts to a tag suggestion agent that is ignorant of the image’s geotag. It thus reduces the problem to the standard image annotation approach where it is treated as a detection event.

**Geographic Baseline**

This baseline assumes that the tags can be predicted by geography alone. This formulation reduces to:

\[
p(b|g) = \frac{p(b, g)}{p(g)}
\]  

(3.12)

The geographic baseline amounts to offering tag suggestions ranked by their prior for an area. While such an algorithm provides mainly place names, common objects of photographic interest can be well represented.

### 3.7.4 Smart Fusion

As the results from Joshi and Luo [38] indicate, a fundamental problem in annotation of geotagged images is that some tags are relevant to visual features (e.g., sunset, beach), while others are not (e.g., vacation, California). A smart way to deal with discrepancies in tag types has yet to be established. We propose finding the mutual information between the distribution of tags in visual feature clusters as a way to determine if visual features help assign the tag. A description of visual term identification extraction will be discussed later in Section 4.2.1. We apply *late fusion* of the dual
Chapter 3. Collaborative Annotation of Geotagged Photos

method, governed by maximizing $p(b|x, g)$ as described in Section 3.7.1, with the Geographic Baseline in Section 3.7.3 to make an effective system for incorporating visual information. Late fusion is governed by the piecewise function: if tag $b$ has one of the 1,250 highest values of $MI(b, x)$ as estimated by the mutual information $MI(b, c)$, we choose to use the score from the dual method for tag $b$. Otherwise, we choose the geographic baseline score, $p(b|g)$. In the next chapter, alternate ways of making decisions about “visual tags” are explored.

3.8 Performance of MAP Georelevant Tag Suggestion

A series of experiments are performed to examine explicitly the performance of the system as a tag suggestion agent. They are performed on the data and features described in Section 3.6 using the annotation algorithm described in Section 3.7. An analysis of smart fusion as well as the choice of $\hat{g}$, the geographic search region, are also considered.

3.8.1 Precision-Recall of Tag Suggestions

For the experiments on a probabilistic formulation for tag annotation, 230 images are reserved as a testset. These images and all images from the same owner are removed from the collection and the learning used in these experiments is done without the
Chapter 3. Collaborative Annotation of Geotagged Photos

Figure 3.8: Distribution of the 230 test images randomly selected from the 1.2 million+ worldwide photos. Most examples are in the United States and Europe, corresponding with the most frequently photographed areas.

benefit of their 104,000 images. A web interface is provided for a team of judges to select correct tags for the test images from a randomized subset of the tags suggested by any of the methods. The judges were unfamiliar with the algorithms. The web interface provided the analyst with the test image along with the owner-supplied tags and a map centered at the geotag of the image. The analyst could then click on the relevant tags and submit. Tags not clicked but that had been offered are considered incorrect. In the following subsections various experiments on the database are presented that judge the performance of an annotation method by precision/recall. Precision is taken to be the number of tags provided by the algorithm that are judged correct, while recall is
Tag Suggestion using MAP

![Precision vs. recall curve comparing dual method to two baselines. Geographic prior is found to outperform dual method, suggesting visual information is best employed only for selected tags.]

**Figure 3.9:** Precision vs. recall curve comparing dual method to two baselines. Geographic prior is found to outperform dual method, suggesting visual information is best employed only for selected tags.

the number of total relevant tags discovered using *any* method that are covered by the *particular* method. Thus, the scoring mirrored that in Section 3.5.1 but with a two-class scoring metric rather than three; here, the “irrelevant” option has been discarded. From previous empirical evidence it is gauged that “irrelevant” and “incorrect” are too similar to be distinguished, so the two-class method is employed here.
Dual Method vs. Baselines

In this experiment, we compare the annotation probabilities offered by the dual algorithm to the visual and geographic baseline methods. The visual baseline compares values of $p(b|x)$, and the geographic baseline values of $p(b|g)$, as compared to the suggestions offered by the dual formulation $p(b|x,g)$. Figure 3.9 presents the discouraging results from this experiment, which show that the dual method using $p(b|x,g)$ performs near or worse than the geographic baseline, which formulates the problem as only maximizing $p(b|g)$. As is expected in an extremely noisy, non-object oriented dataset such as tourist photos, the visual-only baseline performs extremely poorly. Indeed, most “correct” tags are place names and a formulation that maximizes only $p(b|x)$ covers few place names. Research from Joshi [38] shows similar difficulties in seeing gains by fusing visual information with a geographic baseline for words such as “vacation,” “university,” and “stadium.”

In Figures 3.10, 3.11, and 3.12, we take a closer look at what visual information affords us even by isolating/forcing pictures to have the same geotag, $g$. Here the difference in scoring is exclusively from visual information, and we can see that it leads to more relevant results. It is true that the visual information plays a part in distinguishing what we are viewing in a particular geographic region.
Figure 3.10: Two images of different views in Pisa, annotated using the same geotag, $g$. Middle columns show the scores using the dual formulation, and right-hand column shows the score if using the geographic baseline. The relevance of visual information is explored in the table. Scores show that tags “river,” “sunset,” and “landscape” are not relevant to the left image but are to the right. On the other hand, tags such as “Leaning Tower of Pisa,” and “architecture” are more relevant and have higher scores for the lefthand image.
| Tag            | \( p(b|x_1, g) \) | \( p(b|x_2, g) \) | \( p(b|g) \) |
|----------------|------------------|------------------|--------------|
| New York City  | 0.40873          | 0.459295         | 0.247006     |
| Yankee Stadium | 0.222122         | 0.355396         | 0.043413     |
| New York       | 0.323473         | 0.32641          | 0.244012     |
| Bronx          | 0.295210         | 0.260333         | 0.091317     |
| NYC            | 0.286841         | 0.265928         | 0.209581     |
| baseball       | 0.157694         | 0.264744         | 0.026946     |
| New York Yankees | 0.141298        | 0.25674          | 0.025449     |
| Bronx Zoo      | 0.190247         | 0.117621         | 0.133234     |
| animals        | 0.140927         | 0.0818207        | 0.106287     |
| Manhattan      | 0.118774         | 0.120228         | 0.115269     |
| zoo            | 0.147916         | 0.114231         | 0.092814     |
| catcher        | 0.0              | 0.110997         | 0.002994     |
| polar bear     | 0.0              | 0.0              | 0.002994     |
| bear           | 0.0              | 0.0              | 0.001282     |

**Figure 3.11:** Two images of different views in the Bronx, New York City, annotated using the same geotag, \( g \). Terms such as “Yankee Stadium,” “New York Yankees,” “baseball,” and “catcher” are given more of a boost in the righthand image taken of a baseball game at Yankee Stadium. On the other hand, the lefthand picture taken of a polar bear at the Bronx Zoo does not identify the actual animal but does correctly identify that the terms “Bronx Zoo,” “zoo,” and “animals” are more relevant than for the image at the baseball game.
Figure 3.12: Two images of different views in Australia annotated using the same geotag, g. The dual method boosts “surfing,” “beach,” and “wave” terms for the righthand image. For the lefthand image, the picture is not similar to other images taken at the location and few correct terms are identified beyond location terms.
Chapter 3. Collaborative Annotation of Geotagged Photos

Visual Tags

A subset of tags was selected manually to verify that at least in some contexts a dual visual and geographic approach would outperform a geographic baseline. The tags are listed in the box in Table 3.1. Results of algorithms on only these keywords are given in Figure 3.13.

Table 3.1: Manually-selected keywords expected to be visually relevant.

| nature, outdoors, mountain, sky, tree, water, sea, bridge, beach, jungle, park, animals, tower, flower, river, trees, boats, ship, architecture, wildlife, clouds, palm, overcast, door, desert, highway, house, street, city, building, skyline, ocean, snow, steam, forest, sunset, tropical, church |

Figure 3.14 shows the results of using this mutual information to fuse smartly the dual method with the geographic baseline, and Figure 3.15 provides examples of the tags suggested using this method compared to the visual and geographic baseline. A performance gain, while not pronounced, is seen in the higher ranked tags. Tags with visual information occur in low frequency compared with place names and hence there is a limit to improvement.

Geographic Baseline vs. Reverse Geo-Coding

An experiment is done that performed a reverse geo-coding on the query geocoordinates to examine the performance of the geographic baseline. The reverse geo-coder uses the Flickr API [20] to look up a geotag and then suggest the associated city, re-
Chapter 3. Collaborative Annotation of Geotagged Photos

Performance on "Visual" Tags

Figure 3.13: Performance of various algorithms on a subset of hand selected visual tags such as “beach,” “sunset,” “nature,” “wildlife.” This subset of 9% of the tags performs best when using only a visual based coder, and accordingly, the proposed visual/geographic algorithm shows improvements for this subset of tags as compared to the geographic prior baseline.

...gion (e.g., state or province), and country as annotations with maximum score. The performance increase that can be seen from the geographic baseline to one that has undergone this reverse geocoding is evident in Figure 3.16. It shows that the use of a reverse geocoder significantly improves even the geographic baseline.
Tag Suggestion using MAP and Smart Fusion

**Figure 3.14:** Performance of smart fusion to identify visually-relevant tags against geographic baseline. Point for owner annotations, which system is ignorant of, is also provided. Both techniques provide automatic annotations that represent a good start towards providing the annotations the owner ultimately applied.

**Definition of Geographic Search Region, \( \hat{g} \)**

A comparison is made between the performance of the algorithms based on the area of the geographic footprint of the node. Performance is measured for two groups based on density, split at the median of geographic footprint area. We examine the performance of the baselines with the composite algorithm to determine their performance as a function of image density. The results are shown in Figure 3.17. The dual method
Chapter 3. Collaborative Annotation of Geotagged Photos

<table>
<thead>
<tr>
<th>FlickrID</th>
<th>Image</th>
<th>visual baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>994514</td>
<td>Orlando, florida, United States, themepark, night, cathedral, view, Seoul, nad conference, national association for the deaf, Landscape, Great Salt Lake, stars</td>
<td>bridge, California, nocal</td>
</tr>
<tr>
<td>536489</td>
<td></td>
<td>beach, OIF, africa, vacation, water, boat, sea, trip, ocean, clouds</td>
</tr>
<tr>
<td>147754</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.15: Example tags suggested by each method. Suggestions are in order of decreasing confidence. Bold red for correct, black plaintext for incorrect.

performs better initially for denser regions, suggesting the system can be tuned to offer high ranking visual-based annotations in such areas.

Additionally, a comparison is made between using the quadtree to find an appropriate region and a fixed geographic area. This study analyzes whether the use of a quadtree, which terminates at a level of minimum support in a geographic area, is a robust way of determining a geographic area of relevance. Figure 3.18 shows: (1) the
geographic baseline using the quadtree, and (2) a formulation of $p(b|\hat{g})$ that fixes a radius of 7km around the geocoordinates of the target. If the image was taken in a dense area (e.g., New York City), then method (2) will include more images in calculation $p(b|\hat{g})$; if it was taken in a less popular area (e.g., Antarctica) then method (2) will include fewer images in measuring the probability of a particular tag. The fixed radius of 7km is chosen because it is the average of the corresponding radii taken at the level of minimum support for the geographic baseline. The results show that indeed the quadtree is an effective way to localize the region used for tag suggestion.

Figure 3.16: Improvement on geographical baseline by providing location terms from reverse geocoder.
Chapter 3. Collaborative Annotation of Geotagged Photos

Algorithm Performance by Photo Density

Figure 3.17: Comparison of performance based on density of images at geocoordinate of image. While the dual method and the geographic baseline are significantly affected by the density of images in the area (they perform better in dense areas), the visual baseline exhibits a mixed tradeoff when we separate by image density.

3.9 Conclusions

In this chapter we have presented two annotation systems that incorporate GPS metadata increasingly available for photos that can be used alongside visuals for annotation. The first uses a manually-defined region to evaluate the georelevance of a term, and then reranks the annotations for a particular query image based on these values. The best results use a localized urban area (e.g., Los Angeles) to determine the relevance of
Chapter 3. Collaborative Annotation of Geotagged Photos

Quadtree Performance

Figure 3.18: Comparison of performance using a quadtree to determine geographic area versus a fixed radius. The quadtree defines an area of influence based on sample distribution, while a fixed radius allows for many cases with too few images.

a particular annotation to a region, keep a large number of similar images, and weight similar images by visual similarity, though not necessarily geographic similarity.

The second formulation presents a world-scale tag suggestion system which employs a database of 1.2 million+ geotagged images in order to provide annotations to photographs taken anywhere in the world. Relevant annotations are found to be highly geographically dependent, as seen by the performance of a baseline derived from representing the geographic tag distribution on a quadtree. Tag suggestions which are aided by visual analysis can be determined via estimating mutual information, and we found
Chapter 3. Collaborative Annotation of Geotagged Photos

that visual methods hold the most promise for densely sampled regions. Further analy-
sis of the text metadata on this type of data is provided in the next chapter, followed by
explorations of similar annotation algorithms on video in the final chapters.
Figure 3.19: (a) Performance of dual method against the geographic baseline and the visual baseline for Los Angeles dataset. Relevance/coverage shows significant performance improvement over the baseline methods, indicating successful evaluation of an annotation’s georelevance in order to rerank annotations. (b) Performance of dual method against the geographic baseline and the visual baseline for Southern California dataset. Relevance/coverage shows slight performance improvement over the visual baseline, but does not beat geographic baseline, indicating a limitation in learning annotation reranking for a large region.
The nature of modern media is such that social media repositories have only a very loose structure to their tagging systems. The media owners have many different uses and motivations when they apply tags. Sometimes, the use is for personal retrieval or organization, and has limited meaning to the casual user (e.g., “Thursday”); other times, the meaning of a tag is universally understood (e.g., “Eiffel Tower”). The tags may be visual objects in the photo (e.g., “Eiffel Tower”), and other times may not be visual objects (e.g., “Thursday”). In this chapter we attempt to define several types of tag usage and automatically extract such tag types from georeferenced photo collections. We refer to this study of tag types and usage as tag semantics since it attempts to understand the meaning of a tag. A better understanding of tag semantics would benefit many information-based applications, such as automatic extraction of visual examples.
Chapter 4. Mining Tag Semantics

of events/landmarks [42, 64] and tag-driven image annotation [50, 43]. Analysis by mining a large dataset of photographs for time, location, co-occurrence, and visual information over multiple geographical scales provides valuable knowledge about how tags are applied.

The importance of tag semantics in the annotation problem arises directly from the work presented on geotagged image annotation in Chapter 3. Without adapting the annotation methodology for terms that are visually-based, we saw no improvement over a simple geographic baseline in annotation performance. However, by selectively using visual information only for tags that had high mutual information with visual features, some improvement is seen, and manually-selected test images show a clear improvement from incorporating visual information.

Figure 4.1 shows how learned tag semantics can be employed in the multimedia annotation problem. If tags can be identified as objects in the image, for instance, it’s reasonable to use visual features when judging if a particular term is relevant. Other tags, such as ones that indicate time of year the photo was taken, or the location where it was taken, may not be visually relevant and visual feature metrics should be ignored when determining relevance of the tag. In Figure 4.1, tag suggestions are calculated on a previously unlabeled input image from visual features and world location. These tags are redundant as many refer to the same object (variations on “Eiffel Tower”) and others can be pruned by examining time or location metadata alone. In addition to
Figure 4.1: Knowledge of tag semantics allows for better annotation. Appropriate features can be applied by learning how a tag is employed. For instance, if a tag is deemed a landmark, the algorithm can use visual features to determine if it is appropriate. If it is a place, a geotag alone might be the most effective.

adapting the relevance evaluation for different tags, a semantic post-filtering step which categorizes tags into place, timed event, landmark, and visual description provides a cleaner suggestion list to a user or retrieval system.

This chapter will expand on existing methods for automatically extracting tag semantics and report results on a worldwide database of 1.7 million+ Flickr images. Much of this chapter appears in [51]. The next section outlines related work on semantic knowledge extraction from tags. Section 4.2 describes methods for extracting semantics and describes their effectiveness, and the chapter concludes in Section 4.3.
4.1 Related Work

Exploring the purpose and use of tagging has been an area of active research [28], [49]. Golder and Huberman [28] outline seven functions tags can serve for bookmarking which are a superset of functions served by specifying location and information content of a media document. They enumerate the seven functions of tagging as:

1. **Identifying What or Who it is About.** Nouns and proper nouns. E.g., Golden Gate Bridge, water, sky.

2. **Identifying What it is.** What *kind* of thing is being tagged. E.g., photo, video.

3. **Identifying Who Owns It.** E.g., Emily, emoxley777.

4. **Refining Categories.** Rather than establishing categories, qualify existing characteristics. E.g., europe7.

5. **Identifying Qualities or Characteristics.** Typically adjectives. E.g., scary, funny, blackandwhite.


7. **Task Organizing.** E.g., toedit, toshare.

Marlow *et al.* [49] delineate an alternate set of user incentives for tags. They expect tags to be used for:
Chapter 4. Mining Tag Semantics

1. **Future retrieval.** To facilitate the owner’s future use of the item. E.g., thesisreferences.

2. **Contribution and sharing.** To create groups for audiences. E.g., vacationspots, bridges.

3. **Attracting attention.** Using common tags to attract views of an item. E.g., barackobama.

4. **Play and competition.** E.g., in the ESP Game [29], to offer the same tag as another user.

5. **Self Presentation.** To leave a mark on a particular resource. E.g., seen live, emoxley777.

6. **Opinion Expression.** To pass judgment on the item. E.g., greatvideo.

Attempts have been made to extract tag semantics through secondary web repositories. Overell *et al.* [60], for instance, develop a classifier that maps Flickr tags to semantic categories, specifically, WordNet classes. Semantic categories include “Act,” “Animal,” “Artifact,” etc. Their work employs Wikipedia to classify terms, but they do not consider tag usage statistics and image data. Quack *et al.* [64] identify potential landmarks by tight visual clusters, and then mine Wikipedia images and text to associate the pictures with a world landmark. Rattenbury [65] attempts to automat-
Chapter 4. Mining Tag Semantics

...ically extract tag semantics, specifically, identifying “place” and “event” tags, using Flickr geotag and time metadata. They establish an entropy-based technique for automatically identifying these semantic categories and analyze results for a set of roughly 50,000 images with 803 unique tags in the San Francisco Bay Area. As the number of available geotagged photos has increased considerably to an estimated 125 million (as of July 2009), a worldwide analysis is now possible. This chapter will present research that extends automatic tag semantic extraction by analyzing results over a large worldwide database. We identify visually descriptive tags and place tags using techniques like mutual information and graph smoothness, consider the addition of co-occurrence information, and identify sets of tags which correspond to landmarks.

4.2 Semantic Identification of Tags

Three categories are considered which may be detectable using the signal present in the photo collection data: places, visual descriptors, and landmarks. In order to explore aspects of tag usage over a world-encompassing dataset, 1.7 million+ georeferenced images are crawled from Flickr. For visual analysis two common descriptors are employed, GIST [59] and SIFT signature [56]. The crawl process, dataset, metadata, and feature extraction are the same as that described in Section 3.6. The data was stored and retrieved using a quadtree structure as described in Section 3.7.2.
Chapter 4. Mining Tag Semantics

4.2.1 Visual Term Extraction

As discussed briefly in Section 3.8.1, certain tags refer to qualities which are visually identifiable from the photo’s content like “sunset,” “sky,” and “beach.” It is estimated that on average however less than 30% of the 436,506 unique tags belong to this category, qualified as “identifiable from photo content.” This estimate is based off of a random sampling of the tags that were manually classified. These are the tags that could be suggested from analysis of visual features. To find tags which represent visual terms mutual information is considered between a visual feature random variable, $x$, and a tag variable, $b$. To generate $x$, K-means clustering is used to discretize the computed image feature. The mutual information, $MI$, is estimated pointwise as:

$$MI(b, x) = \sum_{b, \bar{b}} \sum_k p(b, x_k) \log \left( \frac{p(b, x_k)}{p(b)p(x_k)} \right)$$  \hspace{1cm} (4.1)$$

$p(x)$ is the cluster prior, $p(b)$ is the tag prior, $p(\bar{b}) = 1 - p(b)$, and $p(b, x_k)$ is counted as the number of photos with tag $b$ in cluster $k$ divided by the total number of photos.

The GIST feature with K=950 clusters is selected to estimate $p(x)$. This number of clusters was chosen because it was the largest number that could be feasibly used to create a clustering on a typical server machine. This number also showed the best performance when we did other image matching and search functions. Table 4.1 shows a list of tags with high mutual information discovered in the dataset. Of the top 100
scores for $MI(b, x)$, 57% were manually judged visually relevant by several independent judges, indicating the ability of mutual information to identify visual tags.

<table>
<thead>
<tr>
<th>Tag</th>
<th>MI</th>
<th>Tag</th>
<th>MI</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunset</td>
<td>0.0088</td>
<td>(C)</td>
<td>2.7402e-06</td>
</tr>
<tr>
<td>clouds</td>
<td>0.0073</td>
<td>mashup</td>
<td>3.9652e-06</td>
</tr>
<tr>
<td>flowers</td>
<td>0.0071</td>
<td>work trip</td>
<td>5.2887e-06</td>
</tr>
<tr>
<td>sky</td>
<td>0.0070</td>
<td>sounds</td>
<td>5.935e-06</td>
</tr>
<tr>
<td>beach</td>
<td>0.0063</td>
<td>psychiatry</td>
<td>6.0322e-06</td>
</tr>
<tr>
<td>nature</td>
<td>0.0053</td>
<td>SFW</td>
<td>5.9390e-06</td>
</tr>
</tbody>
</table>

**Table 4.1:** Tags with high (expected to be visually relevant) and low (not expected to be visually relevant) mutual information with visual features.

**Visual Term Extraction Using Graphs**

As a comparison to mutual information-based fusion, we calculate an estimation of the visual relevance of a particular term using graph-based methods. In Section 4.2.1, we have deemed words to be “visual” terms if they have high mutual information between GIST feature clustering and presence of the tag. Here, we judge terms based on the smoothness of a graph constructed in a particular visual feature space.

A graph is formed using the analogy delineated in Table 2.1, using images as nodes and similarity in particular feature space for edge weights. For a particular tag, $b$, a graph can be constructed using randomly selected images that have tag $b$ and randomly selected images that do not have tag $b$. A typical similarity metric, given in Equation (4.2), can characterize the edge weight $W_{ij}$ between nodes $x_i$ and $x_j$. $d_{feat}$ is also
Chapter 4. Mining Tag Semantics

formulated in the typical way, as the sum of the $L_2$ distance between each dimension:

$$d_{\text{feat}}(x_i, x_j) = \sum_{n=1}^{N} (x_{i}^{n} - x_{j}^{n})^2$$

where $x_{i}$ is the feature associated with $x_i$. $\sigma_{\text{feat}}$ is the standard deviation of $d_{\text{feat}}$.

$$W_{ij} = \exp\left(-\frac{d_{\text{feat}}(x_i, x_j)}{\sigma_{\text{feat}}}\right)$$  \hspace{1cm} (4.2)

A smoothness measurement can be extracted for such a graph that provides an idea of how the visual features are distributed. The smoothness formulation is formulated using the equation:

$$S(b) = \left(\sum_{i,j} W_{ij} |f_{i}^{b} - f_{j}^{b}|\right)^{-1}$$  \hspace{1cm} (4.3)

where $f_{i}^{b}$, $f_{j}^{b}$ represent the attributes of the nodes as described in Section 2.6. A node’s attributes are the annotations associated with it. Thus, a node has multiple attributes but there is only one value for each $f_{i}^{b}$. In this instantiation of graphs where we are attempting to find the smoothness of the graph for a particular annotation $b$, the attributes are simply binary indicator variables reflecting if the image at $x_i$ has annotation $b$. The idea, of course, is to give high smoothness scores for graphs in which images that are very similar in feature space have similar annotations, and in which images that are dissimilar in feature space have different annotations.

In order to fairly compare a smoothness formulation with the mutual information formulation in Section 4.2.1 for identifying “visual tags,” we construct the graph first in
Chapter 4. Mining Tag Semantics

GIST space and compare the list of tags sorted in descending smoothness with the list created by sorting in descending mutual information. A comparison of performance is given in Figure 4.2. It is apparent that the graph smoothness-based definition outperforms the mutual information-based definition, showing an increase in the area-under-the-curve statistic of 8%. Put another way, in the top 100 scores, the graph smoothness statistic had five more correct than the mutual information statistic.

By examining the lists, a hypothesis is made that mutual information is biased towards common terms; many of its top returns are some of the most frequently occurring words in the dataset. In Figure 4.3, we show a graph comparing the rank in descending mutual information against tag frequency. It becomes clear from this figure that the tags with the highest mutual information tend to be the most common terms. So, beyond avoiding lossy quantization necessitated in the mutual information formulation, the graph smoothness-defined operation avoids this bias and leads to superior recognition rate. We explore a modification of the original MI formulation below.

We then examine what happens when we compute graph smoothness statistics in 6 different spaces: five visual feature spaces (described in Section 3.5.4) and one geographic space, with edge weight typified by closeness in straightline distance between geotags of the nodes. That is, for the graph in geographic space, $d_{feat}(x_i, x_j) = R \times 2 \times \arcsin \left( \min \left( 1, \sqrt{\sin^2 \left( \frac{\text{lat}_i - \text{lat}_j}{2} \right) + \cos(\text{lat}_i) \times \cos(\text{lat}_j) \times \sin^2 \left( \frac{\text{lon}_i - \text{lon}_j}{2} \right) } \right) \right)$. Graphs with smoothness statistics above the median are considered “smooth,” and each
Chapter 4. Mining Tag Semantics

tag is given a score between 0 and 5 indicating the number of smooth visual feature graphs. A score of 5 indicates that the tag is smooth in all 5 visual feature spaces; a score of 0 that the tag is unsmooth in all visual feature spaces. Within the quantized scores $[0, \ldots, 5]$, the tags are further ordered in ascending geographic smoothness, since visual terms should not correspond to any particular geographic location. A chart summarizing the scores for some tags is given in Table 4.2, and the precision/recall graph is given in Figure 4.4. The most interesting thing about this graph is that incorporating more visual features does NOT improve performance. The AUC for GIST-only graph smoothness is about the same as using graph smoothness on five visual feature metrics. We see a mixed tradeoff where using all five visual features results in better precision at high recall but lower initial precision. Thus, it has a moderating effect on the curve, and lends further justification to the idea that the GIST feature alone is very powerful as also cited by [31].

One final comparison is done that attempts to compensate for the bias revealed in Figure 4.3. Rather than using simply mutual information score over the clustering for each tag, we now scale this by that tag’s maximum mutual information. That is, since $MI(X, Y) = H(X) - H(X|Y)$ and $H(X|Y) \geq 0$, the maximum $MI(X, Y)$ for any $Y$ is $H(X)$. Therefore, we now score tags ordered by: $\frac{MI(X,Y)}{H(X)}$. Results are given in Figure 4.5, and they show this particular formulation slightly outperforms graph smoothness.
"Visual Term" Identification

Figure 4.2: Graph smoothness in GIST space outperforms mutual information in identifying visual terms. Further analysis shows that mutual information results in a bias towards more frequent terms.

4.2.2 Place Extraction

The authors in [65] introduce Scale-Structure Identification (SSI) for identifying tags associated with places and events. For each tag they consider the entropy over connected subcomponents on a graph with vertices as photos labeled with that tag. A connection criteria, a maximum distance for which an edge appears, controls the scale. Tags with a tight distribution on a location (or time in the case of events) will generate
Chapter 4. Mining Tag Semantics

Figure 4.3: This figure shows the average tag frequency in order of mutual information. The x-axis represents rank of tag in order of descending mutual information; high mutual information scores are to the left, low ones to the right on the x-axis. It is easy to see that higher frequency terms tend to have more mutual information.

large clusters over multiple scales. A decision variable, $\lambda$, for tag $b$ is computed as a sum of entropy over multiple scales:

$$
\lambda_b = \sum_{k=1}^{K} \sum_{Y \in \Psi_{r_k,b}} -\frac{|Y|}{|N_b|} \log \frac{|Y|}{|N_b|}
$$

(4.4)

where there are $K$ scales, $Y$ is a set of photos connected by distance $r_k$ containing tag $b$, $\Psi_{r_k,b}$ is the collection of connected component sets $Y$ for $b$ and $r_k$, and $N_b$ is the set
"Visual Term" Identification Using 5 Visual Features

![Graph smoothness outperforms mutual information in identifying visual terms. Using 5 features as opposed to GIST only has a sort of moderating effect. Initial precision is lower, but ultimate precision is higher. AUC comparison gives AUC=0.5208 when 5 visual features are used, and AUC=0.5212 if the Gist feature alone is used.](image)

Figure 4.4: Graph smoothness outperforms mutual information in identifying visual terms. Using 5 features as opposed to GIST only has a sort of moderating effect. Initial precision is lower, but ultimate precision is higher. AUC comparison gives AUC=0.5208 when 5 visual features are used, and AUC=0.5212 if the Gist feature alone is used.

Each level of the quadtree can be seen as representing a scale space, and thus we can develop an alternate formulation of SSI using the quadtree. To serve as a comparison to this existing method, we employ a similar idea but using the quadtree for scale space definition. Entropy is calculated for each level of the quadtree, as in Equation (4.4): $\Psi_{r_k,b}$ becomes a quadtree node at level $r_k$. We calculate the entropy of a tag at each level...
Figure 4.5: We can improve the mutual information formulation by scaling by the maximum mutual information for each tag, $H(b)$. This shows a slight improvement over graph smoothness with AUC=0.5309 compared to AUC=0.5212 for graph smoothness.

in the quadtree in order to determine if tag $b$ is a place term. Furthermore, the influence of using co-occurrence information is explored. Many place names co-occur with other place names, such as the same place in a different language or a less-specific location (e.g. “Santa Barbara” with “California”). This information is used to influence decisions on a tag. Using the Jaccard coefficient on the image sets containing compared tags (size of the intersection divided by the size of the union), a normalized co-occurrence of tags can be measured and a prediction made using a weighted sum of the scores of tags above a certain threshold of Jaccard similarity: 

$$
\lambda_b' = w_0 \lambda_b + \frac{1}{w_0 + \sum_i J(i, b)} \sum_i J(i, b) \lambda_i.
$$
Chapter 4. Mining Tag Semantics

Figure 4.6: Graph showing precision/recall for place identification of Flickr tags. SSI approach does not exploit co-occurrence information and does not use a quadtree for scale-space definition. Co-occurrence information provides better initial precision, when co-occurrent terms are indicative. However, when co-occurrence fails, it reduces precision, as evidenced by lower precision at high recall.

This formula takes the original score for tag \( b \) and adds the scores for co-occurrent tags \( i \), scaled by its Jaccard similarity with \( b \), \( J(i,b) \). Figure 4.6 shows the results of using a quadtree and incorporating co-occurrence for place identification over 5 scales ranging from 1.1 to 11,100 kilometers. Better initial precision is observed, but when co-occurrence fails (that is, tag \( b \) lacks indicative co-occurrent terms), the system provides lower precision as seen at points reflecting higher recall.
Chapter 4. Mining Tag Semantics

It is notable that these methodologies can easily be extended to another semantic classification: events. For event detection, we simply must additionally consider periodic time events (e.g., “august”) by examining the structure of clusters and appropriately recomputing the entropy on modulo time as is done in [65].

Place Extraction using Graphs

A graph can be constructed using geographic distance between images that contain a certain tag. Geographic terms are expected to have a high value for the smoothness of a graph constructed with these edge weights. Figure 4.7 shows a comparison of evaluating tags using this formulation against the SSI method described above and in [65]. The results show that place extraction is largely a solved problem and additionally that graph smoothness outperforms the SSI method used in the literature.

4.2.3 Landmark Detection

The strongest evidence for visually descriptive tags is when they occur exclusively on photos taken of the same geographically fixed object. Along these lines Kennedy and Naaman[42] are able to identify representative results for landmarks via clustering, and Quack et al. [64] use matching on Wikipedia images for improved accuracy.

To detect tags used to describe landmarks, agglomerative clustering is first employed on image sets consisting of the members of dense nodes in the worldwide
Figure 4.7: Geographic smoothness outperforms SSI in identifying place terms. Geographic term identification is somewhat a solved problem, as evidenced by the AUC approaching 1.

Quadtree. Distances between images are computed using $L_1$ distance on the SIFT signature, and images join a cluster when the members are within a distance threshold $\sigma$ using complete linkage. For each set, clusters exceeding a minimum membership of 5 images are considered possible landmarks. Since the dataset consists of many similar images taken by the same user, membership is limited to one image per user.

To extract the proper name of the landmark, a database is queried which lists georeferenced Wikipedia entries [25]. A matching score is calculated for each landmark.
Figure 4.8: Examples of detected landmarks. Tags from images in a cluster generate a name estimate from a list of georeferenced Wikipedia articles. Stricter clustering yields better naming results, as evidenced by the incorrect guess in the last row.

name in the database within 0.01° latitude and longitude of the median coordinates of cluster members. The score is generated by considering the string similarity between the landmark name and the set of tags and titles for images in the cluster. In an annotation system, if an image is matched to a representative cluster, the landmark title can substitute for similar tags.

By searching breadthwise on the six deepest levels of the quadtree the system is able to automatically extract views for 62 identified landmarks. Figure 4.8 shows an

<table>
<thead>
<tr>
<th>Images in cluster</th>
<th>Ranked Labels from Images</th>
<th>Name Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Images" /></td>
<td>London, bridge, Tower, england, river, riverthames, UK, architecture</td>
<td>Tower Bridge</td>
</tr>
<tr>
<td><img src="image2" alt="Images" /></td>
<td>Prague, Czech Republic, Praha, Astronomical, Clock, 2008, Old Town Square</td>
<td>Old Town Square (Prague)</td>
</tr>
<tr>
<td><img src="image3" alt="Images" /></td>
<td>Pisa, Italy, 2007, of, 2008, Tower, Leaning, Piazza</td>
<td>Leaning Tower of Pisa</td>
</tr>
</tbody>
</table>
example of some of the images correctly matched to a proper name. Table 4.3 shows how the minimum distance threshold and node coarseness affect the results of landmark identification for the data. It is notable that the algorithms described in this section for landmark extraction are very similar to a simultaneous publication from Google Research [98].

4.3 Conclusions

In this chapter, the collection of tags from a collaborative community has been paired with geographic information and visual information to extract knowledge about tag usage. Previous work has been expanded to also consider visually descriptive tags using mutual information and graph smoothness, as well as to identify geographic tags using co-occurrence information and graph smoothness and to identify sets of tags referring to landmarks. By evaluating tag usage semantics automatically, the freeform flexibility inherent to the tagging process is maintained, but valuable information is gained that will better enable future tagging algorithms.
### Table 4.2: Visualization of Tag Semantics

In the top section we see the “visual terms” we would expect to have high visual smoothness, and low geographic smoothness. Next we see somewhat “specific” geographic terms - while they are not considered “visual” tags, they are specific enough locations that it is believable that the pictures coming from those areas are visually smooth. In the third section we see what may or may not be considered visual terms, but that are general enough that they would be distributed in feature space, e.g., “building.” In the final section are more general geographic terms that are smooth in geographic space but not very smooth in visual feature spaces.

<table>
<thead>
<tr>
<th>tag</th>
<th>geo. smoothness</th>
<th>visually smooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowers</td>
<td>0.19275</td>
<td>5</td>
</tr>
<tr>
<td>sunset</td>
<td>0.19296</td>
<td>5</td>
</tr>
<tr>
<td>moon</td>
<td>0.19319</td>
<td>5</td>
</tr>
<tr>
<td>boat</td>
<td>0.19406</td>
<td>5</td>
</tr>
<tr>
<td>dusk</td>
<td>0.19411</td>
<td>5</td>
</tr>
<tr>
<td>fishing</td>
<td>0.1944</td>
<td>5</td>
</tr>
<tr>
<td>door</td>
<td>0.19471</td>
<td>5</td>
</tr>
<tr>
<td>window</td>
<td>0.19506</td>
<td>5</td>
</tr>
<tr>
<td>cloud</td>
<td>0.1951</td>
<td>5</td>
</tr>
<tr>
<td>Mali</td>
<td>0.94536</td>
<td>5</td>
</tr>
<tr>
<td>Flanders</td>
<td>0.94994</td>
<td>5</td>
</tr>
<tr>
<td>Easter Island</td>
<td>0.95006</td>
<td>5</td>
</tr>
<tr>
<td>building</td>
<td>0.19946</td>
<td>1</td>
</tr>
<tr>
<td>horses</td>
<td>0.20294</td>
<td>1</td>
</tr>
<tr>
<td>vacation</td>
<td>0.20416</td>
<td>1</td>
</tr>
<tr>
<td>sport</td>
<td>0.20545</td>
<td>1</td>
</tr>
<tr>
<td>university</td>
<td>0.20732</td>
<td>1</td>
</tr>
<tr>
<td>Ecuador</td>
<td>0.99373</td>
<td>1</td>
</tr>
<tr>
<td>Bangkok</td>
<td>0.99633</td>
<td>1</td>
</tr>
<tr>
<td>Australia</td>
<td>0.99801</td>
<td>1</td>
</tr>
</tbody>
</table>
**Chapter 4. Mining Tag Semantics**

<table>
<thead>
<tr>
<th>d</th>
<th>Coherence</th>
<th>Precision</th>
<th>NAcc</th>
<th>Num/Uniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0.86</td>
<td>0.80</td>
<td>0.63</td>
<td>146/51</td>
</tr>
<tr>
<td>15</td>
<td>0.90</td>
<td>0.90</td>
<td>0.61</td>
<td>96/36</td>
</tr>
<tr>
<td>16</td>
<td>0.90</td>
<td>0.93</td>
<td>0.81</td>
<td>32/11</td>
</tr>
<tr>
<td>17</td>
<td>0.86</td>
<td>1.0</td>
<td>1.0</td>
<td>11/5</td>
</tr>
<tr>
<td>18</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1/1</td>
</tr>
</tbody>
</table>

**Table 4.3:** Landmark extraction as a function of quadtree depth $d$ for a fixed clustering threshold $\sigma$. Coherence is the fraction of images in a cluster belonging to the correct landmark. Precision is the fraction of clusters which are landmarks. Naming accuracy (NAcc) is the fraction of clusters correctly named. Num/Uniques is the number of landmark clusters found and number of unique landmarks found.
Chapter 5

Video Annotation Using Search and Mining

The mining approach has a recent history in image annotation literature, but it is absent in video annotation literature thus far. In video annotation, we seek to describe the content or characteristics of a video using text labels or annotations. Video is unique from images in that it has a temporal quality and an embedded audio track. Annotation is thus more complicated than images but there is also additional information available in the form of audio features. Audio enables the direct extraction of automatic speech recognized transcripts (ASR) as well as machine translated (MT) transcripts that can be analyzed in order to annotate without using any side information. However, ASR transcripts are very noisy since untrained speech recognition is not perfectly accurate,
Chapter 5. Video Annotation Using Search and Mining

so annotation directly from original ASR is also prone to error. This chapter presents robust techniques that avoid errors associated with mining noisy audio tracks by finding and mining other similar videos during annotation. Other video collections have some form of initial annotation supplied by the owner, but these annotations are lacking for reasons akin to those explored in image annotations previously. Several techniques will be presented to address the video annotation problem from a search and mining point of view. This chapter gives a basic search and mining scheme that shows the gains possible by mining a redundant dataset for similar videos before annotating, serving as a proof of concept for the approach. It then extends this basic scheme to incorporate graphs, a learning technique particularly suited for multimedia data. This chapter appears in large part in [50] and [53].

5.1 Annotating Videos

Video annotation is typically directly extended from image annotation. Keyframes are selected, usually incorporating aspects of shot segmentation, analysis of camera motion, and k-means clustering algorithms to find salient, representative frames. The collected keyframes are expected to be representative of the video as a whole. Each keyframe can be annotated using algorithms such as those described in Chapter 2, and the video is said to be annotated by the collected labels from the keyframes. Even
Chapter 5. Video Annotation Using Search and Mining

attempting to annotate using only a very few words, most automatic video annotation algorithms exhibit precision of less than 5% [77]. As a result of this problem, online video repositories, ranging from almost entirely user-contributed content such as YouTube or Vimeo [83] to entirely professional such as Hulu [36], rely on manually-applied tags and content description.

The automatic video annotation problem has generally been approached as a detection problem where SVM annotation models are applied to determine the presence of certain annotations. The SVM emerged as the choice learner in the TRECVID benchmarking competition. The individual SVMs can be improved by correlation mining (see Section 2.5), which allows annotations to reinforce or punish each other. Correlation mining affords large gains over more elementary means, but all tests have been performed on small dictionaries of typically fewer than 40 words, as they are limited by their basis in supervised techniques that must learn a model for each keyword. On the other hand, Velivelli [82] presents a method that annotates using transcripts, but the transcript corpus as a whole is used to limit the number of dictionary terms rather than trying to use each video to find a video-specific vocabulary.

Existing annotation methods usually suffer from two problems: (1) most video annotation approaches are supervised techniques and do not address tag discovery, since previously unseen annotations lack training data. As a result, they are limited to a pre-defined concept set; and additionally, (2) existing unsupervised approaches fail to use
mining when annotating a specific video, limiting the mining step to initial dictionary creation before a target is chosen [82] or general topic annotation as in [92]. The technique described in the following sections presents a method for video annotation that uses mining in an unsupervised way to label videos with an unlimited dictionary size.

5.2 Mining Annotations in Similar Videos

A schematic of a simple video annotation system driven by search and mining is shown in Figure 5.1. First, a query video is issued to search a database for similar videos, and then the ASR of similar videos is mined to identify keywords for this query video. The search can be done using any one of several modalities. This approach addresses the shortcoming of limited vocabulary in previous video annotation methods since the process is not restricted to machine learning of certain annotations. Instead, a general unsupervised process is used where visual, text, and concept features are used to find similar videos, and then textual analysis used to mine the similar videos for annotations.

The ASR transcripts of the similar videos can be mined for video annotations, similar to what is done for text annotation of written documents [82, 92]. The noise and errors in current ASR/MT technology make keyphrase extraction impossible, since nearly any relevant phrase has an error in at least one of the words, so this work focuses
**Figure 5.1:** System workflow. System first collects similar videos using various search algorithms. The ASR/MT transcript of the similar videos are used in a mining step to generate tag annotations for the query video.

on keyword extraction. After stemming, in which words are reduced to their root form, and stop-list application [4], in which common words are thrown out, a term frequency vector is created for each video clip representing the number of times each word appears in the clip. We call this term vector that counts the frequency of words \( b_1, b_2, \ldots \), in video \( i \), \( f_i = [t b_1, t b_2 \ldots]^T \). The most common words, with highest \( t b_i \), are kept as annotations. For example, in the case of news videos, these tend to be names of leaders or countries, or concepts that identify the topic of the news story. Analogously, we can do a similar process after collecting annotations from the similar videos.

However, annotation from this term frequency vector \( f \) is prone to errors for two reasons: (1) transcripts are noisy and words are misspelled or misheard, and (2) a particular keyword may not be used frequently in the speech for a video. We make annotations more robust by supplementing the video’s ASR with the transcript from
similar or related videos. These similar videos are found by comparing image features, for instance, finding videos with the closest color autocorrelogram. This supplementary ASR provides new annotations that perhaps did not exist in the original video and also reinforces the annotations that are extracted from the original ASR. The algorithm annotates using a term vector \( f^* \) that is a weighted combination of \( f \) from \( N \) similar videos: 
\[
 f^* = \sum_{i=1}^{N} \alpha_i f_i 
\]
A heuristic formation of this sum that uses ad-hoc decisions for determining the value of \( N \) is first explored as a proof of concept in the next section. An alternate formulation of \( f^* \) that will be explored in Section 5.3 uses graphs to combine the transcripts of similar videos to create \( f^* \). Experiments on this graph formulation are given in Sections 6.1 and 6.2.

### 5.2.1 Proof-of-Concept

A simple proof of concept is performed to validate the proposed approach of combining similar videos and mining ASR transcripts to annotate videos. We perform a basic search using multiple modalities as described in [35]. The transcripts from these \( N \) similar videos are then added to the original to create \( f^* \). Two methods of weighting the ASR supplemented by search are examined. Case 1 simply weights similar video \( i \) equally with the original clip \( q \), meaning \( \alpha_i = 1 \) \( \forall i = 1, ..., N \). Case 2 weights original query \( q \) with \( \alpha_q = 1 \), and the similar clips proportional to its similarity to \( q \). The similarity values, \( S_i \), are provided by the modalities described in [35]. The resulting term
frequency vector $f_q^*$ for query $q$ is formulated as

$$f_q^* = \sum_{i=1}^{N} \alpha_i f_i$$  \hspace{1cm} (5.1)

where for case 1, $\alpha_i = 1$ for $i = 1, \ldots, N$, and for case 2,

$$\alpha_i = \begin{cases} 
1 & i = q, \\
\frac{S_i}{\sum_{i=1}^{N} S_i} & i \neq q
\end{cases}$$  \hspace{1cm} (5.2)

Experiments are conducted over TRECVID 2005 corpus [77], which consists of 169 hours of news video (137 development videos and 140 evaluation videos) in three languages, i.e., English, Arabic, and Chinese. A shot is adopted as the basic unit for video annotation in order to provide enough videos for adequate search and mining. In total, there are 89,673 clips in the database. Throughout this section, reference to a “video” indicates one of these 89,673 clips. 112 shots are chosen for querying based on the belief that clip content overlaps with that in other videos. These test shots are selected to be representative of the general content of the database, including commercials, international and domestic news stories. 64 test shots are in English, 25 in Chinese, and 23 in Arabic. The overlapping video in the database is not necessarily expected to be in the same language. Although the video database used in this experi-
ment is far smaller than the image database used for search-based annotation of images in [86], the significant overlap in video content [91] allows search-based annotation to be effective on such a small data set. It is notable that exact duplicates in the dataset are not particularly useful, as the associated ASR/MT text will be identical to the original. The same relevance/coverage metric used in Section 3.5.1 is used for these experiments on videos.

Evaluation of Search Modality

In this analysis the performance of different search modalities for the annotation task is sought. Each search modality returns a ranked list of shots for a query based on similarity in that particular mode (image/query-by-image-example (QBE), text, concepts as defined by scores for SVM models, or some fusion of them). Intuitively, it seems that image-based querying should perform the best, since text querying returns shots with similar transcripts, and therefore just re-emphasizes the original text. Concept querying is expected to work reasonably as well, since concept querying uses a 36-dimensional vector that is derived from image features only. Figure 5.2 shows the six different modalities as compared to annotation without search supplementation, using only the original transcript.

Most notable in Figure 5.2 is the significant improvement using simple QBE search over annotation without supplementing with similar videos. Using QBE search results
in precision improvement of 20% to 25%. Concept and fusion of QBE, text, and concept performed similarly well. As expected, text alone performs quite poorly and is actually worse than annotation without search, that is, identifying keywords using only the original transcript $f_q$. Additionally, text fusion with either QBE or concepts greatly decreases annotation performance as compared to either QBE or concept modality search alone.

**Figure 5.2:** Search modality comparison. Querying by image and by concept greatly outperforms text search, as text search re-emphasizes the noisy transcript.
Chapter 5. Video Annotation Using Search and Mining

Evaluation of Selection of Shots and Weighting

An attempt is made to analyze the sensitivity to the number of similar included videos, \( \mathcal{N} \). The reader is referred to [52] for more in-depth analysis, but the conclusion remains that the performance is best when using fewer similar videos. However, this effect is likely due to the weighting scheme that does not decrease quickly enough from the most similar video to the least similar video. This would result in irrelevant returns at the end of the inclusion list being given a weight that is not significantly different from the (relevant) top return. With irrelevant returns having similar weight to relevant ones, the annotations become less relevant on the average, especially because typically the irrelevant returns are from another common story and therefore have common language elements. However, no matter the inclusion rate, all annotations using search outperform annotations without search. In addition, it is gathered that weighting the image shots by their similarity to the query always outperforms equal weighting. Un-weighted inclusion of similar shots results in a decrease in performance as compared to without-search annotation.

These examples extend a proof of concept to automatic video annotation through search for similar videos and mining of related transcripts. The approach is not restricted to any pre-defined vocabulary. Experimentally the best methods weight the text of similar videos proportionally to similarity with the target video, and then extract annotations. The best annotations resulted from instantiations of this method that search
based on visual or concept similarity, as these are modalities independent of the annotation space (i.e., text). In the following, we present a more principled approach to annotation, using graph-based models.

### 5.3 Annotation Using Graphs

Graphs offer a promising avenue for multimodal data as a graph can be formed for each mode individually and then the graphs combined. They are thus particularly well-suited for multimedia problems, where multiple modes characterize the similarity between objects. An overview of several graph learning problems is given in Section 2.6. In this section we explore an instantiation of graph theory that allows us to reinforce and propagate annotations. Graphs are used in this section to stabilize annotations over different feature spaces, and then the individual graphs are combined using a weighting scheme derived from graph smoothness. These concepts of “stable graphs” and “graph smoothness” are explored below.

The workflow for a multimedia annotation system that uses graphs is shown in Figure 5.3. First, a query is issued to a database, searching using a modality independent of the annotation type. Since text annotations are sought, the search is done using a non-textual feature such as image characteristics. Then a stable graph is found that emphasizes frequent labels in the result set. Various definitions of this stable graph cre-
Figure 5.3: System workflow, using a video annotation example. The system first collects similar videos and generates small graphs for different visual features. For example, the visual features may be Autocorrelogram (AC), Color Moment (CM), Edge Distribution Histogram (EDH) features, and so on. A stable graph is found by iterative diffusion over the connected nodes and then the annotations mined from the weighted ASR/MT transcripts of each stable graph. A zipf-based cutoff is used to determine the relevant annotations in a particular video.

Annotation can be seen in Section 2.6.2, and we describe one definition below. The stability enforces a smoothness on the annotations of the videos, so that similar videos have similar annotations. The resulting stable graph can be directly analyzed for keywords or combined with other graphs to annotate the target. This approach addresses the shortcoming of limited vocabulary size in previous methods (described in Sections 2.4 and 2.5) since the process is not restricted to machine learning of certain annotations. Instead, our approach represents a general unsupervised process where visual features are used to find similar videos, and then textual analysis of transcripts used to determine
Chapter 5. Video Annotation Using Search and Mining

annotations. The mining of transcripts of similar videos improves the completeness and accuracy of the annotation process.

The technique represents an inductive learning process that uses the weak predictions afforded by each video to create a stronger prediction of appropriate annotations for the set. By allowing correlations in near neighbors to reinforce each other, better annotations can be extracted for the videos. New terms and keywords that did not exist previously as annotations can be discovered. A term that appears infrequently in a video, but frequently when a collection of similar videos is considered, may appear as a “new” annotation.

Figure 5.4: Construction of a graph. First, a collection of similar videos are found by searching using a visual feature. These are the nodes $x_i$ and they lie at a particular point in space, the location of the visual feature associated with that video. The closeness between these nodes is measured and assigned an edge weight, symbolized by the thickness of the line in the rightmost visualization.

Figure 5.4 gives an outline of how a graph is constructed. A graph is formed using near neighbors of a video, found by using a particular feature to determine the distance between a target and database videos, and then keeping the closest videos. These videos
form the nodes, $x_i$, which reside at a particular point in the feature space. These nodes are connected by edges, $W_{ij}$, that are weighted by the distance between the nodes in this feature space. Defining the most stable graph is different in (1) the case where the attributes (or annotations) of some nodes are known, in which case supervised or semi-supervised learning is preferable, and (2) the case in nearly all practical annotation instances, where the vocabulary is so extensive that groundtruth for all annotations does not exist. Rather, a few videos have a limited number of annotation predictions that positively identify only a few feature attributes but do not negatively identify any annotations.

We will next describe graph reinforcement via semi-supervised and unsupervised learning. First we mathematically formulate stable graph reinforcement of a single graph. The well-established, state-of-the-art semi-supervised case is presented in Section 5.3.1, and then this work’s contribution, an extension of the state-of-the-art to the unsupervised case, in Section 5.3.2. It then moves to multiple graphs in Sections 5.3.3 and 5.3.4 where similarly both the semi-supervised and unsupervised cases are explained. In the multi-graph case a definition of “smoothness” is incorporated to appropriately weight each graph.
5.3.1 Singular Graph Reinforcement: Semi-Supervised Learning

Semi-supervised learning is a technique designed to optimize learning in the case where only a handful of annotations exist in a large collection of data. In semi-supervised learning, assumptions made on the large set of unlabeled data are built into models, and by incorporating these assumptions it typically outperforms the classification of supervised learning methods [9, 102]. Table 5.1 is supplied here as a reference for notation in this section.

The authors of [99] phrase the annotation problem, that is, finding the optimal classification for a feature at each node, in terms of a regularization framework. Their formulation amounts to solving:

\[
 f^{b*} = \arg \min_{f^b} \left\{ \sum_{i,j} W_{ij} \left| f^b_i \sqrt{D_{ii}} - f^b_j \sqrt{D_{jj}} \right|^2 + \mu \sum_i \left| f^b_i - Y^b_i \right|^2 \right\}
\]  (5.3)
where $Y^b_i$ is the initial labeling of node $i$: +1 for positive, −1 for negative, and 0 for unlabeled regarding annotation $b$ or in the case of transcript mining, $Y^b_i = tf_i^b$, the frequency of $b$ in the ASR; $D$ is the diagonal normalizing matrix given by $D_{ii} = \sum_j W_{ij}$; $\mu$ is a customizable parameter indicating how much the original annotation should “stick”; and $f_i^b$ is the current label value of node $x_i$. The first term in Equation (5.3) can be considered a “smoothness constraint” that implies a cost for labels that change too quickly over the space. The second term is a “stickiness constraint” that implies a cost for changing the initial labeling. The closed-form solution to Equation (5.3) is found to be [99]:

$$f^b_* = (I + \frac{1}{\mu}L)^{-1}Y^b$$

We can also think about the problem sequentially. Applying the update:

$$f^b_{t+1} = \frac{1}{1 + \mu}(I - L)f^b_t + \frac{\mu}{1 + \mu}Y^b$$

iteratively results in convergence at $f^b_\infty$ [85]. The reinforcement of the initial labeling $Y$ is apparent in this equation, and may not be ideal in the case where initial labels are
unknown. In the next section, we explore an unsupervised approach that extends from the semi-supervised case.

### 5.3.2 Singular Graph Reinforcement: Unsupervised Learning

In the case of unsupervised learning where a training set has not been given, the iterative process described in Equation (5.5) reduces to a trivial case. Our contributions to the graph theoretic approach to annotation lie in this case. Without initial labeling, $Y_i^b = 0 \forall i, b$, so Equation (5.3) reduces to:

$$f^b = \arg\min_{f^b} \left\{ \sum_{i,j} W_{ij} \left| \frac{f^b_i}{\sqrt{D_{ii}}} - \frac{f^b_j}{\sqrt{D_{jj}}} \right|^2 + \mu \sum_i |f_i^b|^2 \right\}$$  \hspace{1cm} (5.6)

The second summation, the “sticking constraint,” becomes a constant positive value, and minimizing the expression over $f^b$ results in a trivial solution where $f_i^b = f_j^b$.

Furthermore, “supposing” or “imposing” an initial labeling $Y^b$ results in a repeated reinforcement of noisy, often incorrect, labels which is apparent when considering the iterative approach in Equation (5.5). “Supposing” an initial labeling using the owner annotations results in incorrect reinforcement since the lack of an initial label does not necessarily imply that label is inappropriate, as it may have only been overlooked or the cause of a machine translation error. Experiments in this study show that supposing
either a negative annotation, \( Y_i^b = -1 \), or an unlabeled value, \( Y_i^b = 0 \), for missing annotations underperforms a more explicitly unsupervised graph propagation technique.

Instead, a different tack must be taken to find a stable graph over the feature space. The values in \( W \) are used to update the nodes according to the properties of its neighbors. Consider \( W' \) the row-normalized \( W \), that is \( W'_{ij} = \frac{W_{ij}}{\sum_j W_{ij}} \). \( W' \) is a non-generative diffusion kernel on a fully-connected graph. By “non-generative,” we simply mean that the final weighted number of annotations of the graph at convergence is the same as the initial number of annotations, but has been redistributed. Effectively, the potential annotations of each node diffuse to the other nodes, weighted by affinity between them.

The algorithm considers the frequency \( f_i^b \) of a particular term \( b \) for the video \( x_i \), and updates it according to that term’s frequency in close neighbors. Additionally, a term is included such that each step does not completely diffuse its current labels to neighbors, but retains them weighted by some factor \( \mu \). For each possible term \( b \) for the transcript, the distribution of the term over the graph nodes after one iteration is described by the term frequency vector \( f^b \):

\[
f_{r=t+1}^b = [(1 - \mu)W' + \mu I]^T f_{r=t}^b
\]  

(5.7)
where $I$ is the identity matrix and $f^b = \begin{bmatrix} f^b_1 & f^b_2 & \ldots & f^b_N \end{bmatrix}^T$ is the feature vector indicating classification annotation $b$ for each of the $N$ videos.

As the update matrix, $\left[(1 - \mu)W + \mu I\right]$, is row-normalized, the total number of annotations does not grow but rather redistributes. The ergodic theorem indicates that the process will converge [45]. Intuitively, if a particular keyword is only found for the target, it will be distributed to the nodes and its importance minimized (tag filtering). On the other hand, if the other similar videos have an annotation that is not contained in the target, that keyword will be added to the attributes of the target node (tag extension). Additionally, the graph will reinforce a particular term in the target that is also present in other videos.

The formulation thus far has been limited to discussion of reinforcement using a single graph. It is preferable to be able to use multiple graphs so that several features, modalities, and distance metrics can characterize the reinforcement. Extension to this case is discussed in the next section.

5.3.3 Multiple Graph Reinforcement: Semi-Supervised Learning

This section presents the graph reinforcement algorithm using multiple graphs, providing a way to weight the single graphs effectively and improve annotations of the node videos more accurately than a single graph. In terms of Zhou’s regularization
framework [99], the semi-supervised multi-graph problem, for the case of $G$ graphs requires solving the following, assuming $\{\alpha_i\}$ are known:

$$f^b = \arg \min_{f^b} \sum_{g=1}^{G} \sum_{i,j} \left\{ \alpha_g W_{g,ij} \left| \frac{f^b_i}{\sqrt{D_{g,ii}}} - \frac{f^b_j}{\sqrt{D_{g,jj}}} \right|^2 \right\}$$

Equation (5.8)

In practice, however, we must solve for both the set of $\{\alpha_i\}$, which are unknown, and $f$. Wang [85] shows that optimizing Equation (5.8) for both $f$ and $\alpha$ results in a trivial solution where $\alpha_{g_{\text{best}}} = 1$ for the smoothest graph ($g_{\text{best}} = \arg\min_g \{f^b L_g f^b\}$) and $\alpha_g = 0$ otherwise. This weighting amounts to doing single graph reinforcement on only the smoothest graph. Instead, Wang suggests that by relaxing $\alpha_g$ to $[\alpha_g]^r$, we can solve for $\alpha_g$, in the case of fixed $f^b$, with:

$$\alpha_g = \left( \frac{1}{\sum_{g=1}^{G} \left( f^b L_g f^b + \mu f^b - Y \right)^2} \right)^{1/r} \left( \frac{1}{f^b L_g f^b + \mu f^b - Y} \right)^{1/r}$$

Equation (5.9)

and solve for $f^b$, in the case of fixed $\alpha$, with

$$f^b = \left( I + \frac{1}{\mu} \frac{1}{\sum_{g=1}^{G} \alpha_g^r L_g} \right)^{-1}$$

Equation (5.10)
Chapter 5. Video Annotation Using Search and Mining

An iterative EM algorithm that alternates updating $\alpha$ according to Equation (5.9) and then $f^b$ according to Equation (5.10) converges to a unique solution [85]. Our contributions do not lie here but in the next section where we alter the formulations in Equations (5.9) and (5.10) for scenarios where the initial annotation does not exist or is unclear.

5.3.4 Multiple Graph Reinforcement: Unsupervised Learning

This weighting formulation is limited in the case where negative samples do not exist. Equation (5.9) stresses that close nodes should have similar annotations with the term $f^b \mathbf{L}_g f^b$. However, this relies on a normalization of the feature attributes $f^b_i \in [-1, 1]$ so that diverging decisions provide a negative contribution to $f^b \mathbf{L}_g f^b$.

Building on the idea of smoothness formulated in Equation (5.9), a smoothness function is derived in this section for the unsupervised case where negative examples do not exist.

Using the same principle as the semi-supervised learning scenario, our algorithm stresses that close nodes, signified by high edge weight, should also be close in annotations. The basic idea is that labels should not change too quickly over the space. Therefore, a greater “cost” should be assigned when annotations differ for close neighbors. A cost function is assigned per annotation that scales difference in annotation by
similarity between nodes:

\[
C_g(f^b, W_g) = \sum_{i,j} W_{g,ij} |f^b_i - f^b_j| \tag{5.11}
\]

For each of the \( G \) graphs, it makes the most sense to average the graph’s smoothness over the possible annotations. The graph smoothness for a particular graph builds on the cost function from each word found in Equation (5.11), including the relaxation parameter \( r \). By summing the cost over all annotations and then normalizing, weights are normalized to sum to 1:

\[
\alpha_g = \left[ \frac{1}{\sum_b C(f^b, W_g)} \right]^{\frac{1}{r-1}}
\]

\[
\sum_g \left[ \frac{1}{\sum_b C(f^b, W_g)} \right]^{\frac{1}{r-1}} \tag{5.12}
\]

This smoothness measurement can be thought of as the “goodness” of the graph, and can be used to deemphasize poor (“unsmooth”) graphs. Intuitively, as \( r \to \infty \), \( \alpha_g \to \frac{1}{G} \), equally weighting all graphs. As \( r \to 1 \), only the smoothest graph with small \( \sum_b C(f^b, W_g) \) is kept.
5.4 Graph Reinforcement as a Steady-State Markov Chain

Problem

Markov chain processes are used in many fields of research and therefore are easily grasped by a broad range of people. Thus, we present here the graph reinforcement formulation in a more physical, intuitive way as a steady-state Markov chain problem, or a random walk with restarts. Here, our method of graph reinforcement, described in Section 5.3.2 and 5.3.4, is framed as finding the steady-state solution to a Markov chain. In this formulation, each state in the system is a particular video that is “visited” by a collection of labels (see Figure 5.5). The affinity matrix $W$ represents the transition matrix between states, which will be called $P$, with $P_{ij}$ representing the probability that an annotation currently visiting $x_i$ will next visit video $x_j$. $P$ is explicitly formulated for this work’s unsupervised case as $P = (1 - \mu)W + \mu I$. Each of the labels at a node travels to the other nodes governed by this transition matrix. That is, each annotation label $b$ has a certain probability of visiting each other video, stepping
through a stochastic process governed by the equation:

\[
\begin{bmatrix}
  f_{b1}^{t}\tau+1 \\
f_{b2}^{t}\tau+1 \\
\vdots \\
f_{bN}^{t}\tau+1 
\end{bmatrix}
= \begin{bmatrix}
p_{00} & p_{01} & \cdots & p_{0N} \\
p_{10} & p_{11} & \cdots & p_{1N} \\
\vdots & \vdots & \ddots & \vdots \\
p_{N0} & p_{N1} & \cdots & p_{NN}
\end{bmatrix}
\begin{bmatrix}
f_{b1}^{t}\tau \\
f_{b2}^{t}\tau \\
\vdots \\
f_{bN}^{t}\tau 
\end{bmatrix}
\]

This equation can be formulated for all annotations \( b \). For a single graph, then, the problem can be solved by combining the final probability, \( \pi_{0}, \pi_{1}, \ldots \pi_{N} \), a term has of entering a certain “state,” that is, video, with the initial starting state \( f_{\tau=0} \). The initial state has each video visited by the terms in its initial transcript or in the owner’s labelling. Thus, the steady-state reduces to:

\[
f_{b}\ast = \begin{bmatrix}
f_{b1}^{t=\infty} \\
f_{b2}^{t=\infty} \\
\vdots \\
f_{bN}^{t=\infty}
\end{bmatrix}
= \begin{bmatrix}
\pi_{0} & \pi_{0} & \cdots & \pi_{0} \\
\pi_{1} & \pi_{1} & \cdots & \pi_{1} \\
\vdots & \vdots & \ddots & \vdots \\
\pi_{N} & \pi_{N} & \cdots & \pi_{N}
\end{bmatrix}
\begin{bmatrix}
f_{b1}^{t=0} \\
f_{b2}^{t=0} \\
\vdots \\
f_{bN}^{t=0}
\end{bmatrix}
= \Pi f_{\tau=0}^{b} \quad (5.13)
\]
When dealing with chains governed by multiple features and transition matrices, $P_1, P_2, \ldots P_G$, each process must be weighted. That is,

$$ f^* = \sum_{g=1}^{G} \alpha_g f_g^* = \sum_{g=1}^{G} \alpha_g \Pi_g f_{g,\tau=0} $$

(5.14)

Figure 5.5: Graph reinforcement approach as a steady-state Markov problem.
where \( \alpha_g \) is formulated as in Section 5.3.4. Now that a stable state for the target annotation video has been found, annotation selection out of this stable graph or state can be described.

### 5.5 Fitting a Zipf Curve

Given the term frequency vectors \( f^* \), a zipf curve can be used to extract important keywords. The zipf curve models a typical distribution of word frequency in language [4]. Using a threshold based on a zipf curve allows the algorithm to keep a variable number of keywords, rather than, say, a fixed number \( \bar{K} \) most important keywords. It is not dependent on a pre-specified threshold or a pre-specified number of annotations, which is important because some videos require more annotations than others. The curve is defined using a shape parameter \( s \) which uniquely defines the zipf distribution, defined for the \( k^{th} \) most frequent word in the case of a dictionary of size \( N \) as

\[
f_z(k; s, N) = \frac{1/k^s}{N \sum_{n=1}^{N} 1/n^s}
\]  (5.15)

A best-fit zipf curve is found by solving for the shape parameter \( s \) for a vector of the target node’s attributes or annotations, \( f^* \). Then, keywords appearing more often than the theoretical \( K^{th} \)-ranked word, \( f_z(K'; s, N) \), are flagged as annotations. A range of
Chapter 5. Video Annotation Using Search and Mining

$K'$ provides instances that can create a precision/recall-like curve illuminates tradeoffs between fewer (greater precision) or more (greater recall) keywords. Figure 5.6 shows that using the zipf curve for annotation cutoff is appropriate. The term frequencies over the TRECVID 2005 corpus used for experiments have been plotted against a theoretical zipf curve.

**Actual Word Distribution vs. Zipf Curve**

![Graph showing actual word distribution vs. zipf curve](image)

**Figure 5.6:** This figure shows motivation for using a zipf curve to model term distribution. Solid line represents a theoretical zipf curve, points represent actual data points of term frequency from TRECVID 2005 video corpus which is used for experiments.
5.6 Conclusions

In this chapter we have presented the framework for a video annotation by mining scheme. The method finds similar videos, and then uses these videos to create more robust annotations for the original video. The robust annotations can be extracted from automatic speech recognized transcripts or from existing annotations in similar videos. A proof of concept has been presented that simply combines the transcript of similar videos to the original. Furthermore, we motivate an approach using graph theory, which allows us to effectively combine the similar videos with the original. Annotations can be identified as those words that occur more frequently than certain theoretical values on a zipf curve, allowing us to identify keywords not simply as words that appear with a minimum frequency but rather have a certain level of importance when compared to general speech.
Chapter 6

Case Studies: Graph-Based Video Annotation

In this chapter we present several case studies of video annotation that use the fundamentals presented in Chapter 5. Both the case of ASR/MT transcript mining is presented as well as the instantiation of collecting manual annotations in a process of collaborative tagging.

6.1 Transcript Mining Using Graphs

In this section experiments are described that evaluate the effectiveness of the proposed approach that uses graph theory described in Section 5.3. Specifically, a compari-
son is made between the unsupervised approach, a modification of the semi-supervised approach described in Sections 5.3.1 and 5.3.3 and in [85], and an approach without graph reinforcement. An analysis of annotation “discovery” of the proposed technique follows. The section concludes with a study of the effect of the parameters, graph size, $N$, and the weighting parameter, $r$. The data and relevance/coverage performance metric used in these experiments are identical to that used in Section 5.2.1.

6.1.1 Unsupervised Learning vs. Semi-Supervised Learning

An experiment is done to compare the proposed approach to a modification of the state-of-the-art approach from [85]. While [85] is for the case when certain labels are known, this work extends it to a dataset where labels are not known but predictions or indicators exist for those labels. The extension, specifically, uses the transcript term frequency as initial labels, that is, $Y_i^b = f_i^{b,\tau=0}$.

For these studies, only visual modalities are used, and only an $L_2$ norm is used for distance measurement between features. The following visual features have proved effective for video annotation [35], and are used in our experiments:

- **Edge Distribution Histogram** (EDH) 75-dimensional,

- **5x5 Color Moment** (CM5x5) 225-dimensional, based on 5x5 block division of images in Lab color space,
Chapter 6. Case Studies: Graph-Based Video Annotation

- **3x3 Color Moment** (CM3x3) 81-dimensional, based on 3x3 block division,

- **Wavelet Texture** (WT) 128-dimensional,

- **Color Autocorrelogram** (AC) 144-dimensional, based on 36-bin color histogram and four distances,

- **HSV Color Histogram** (HSV) 64-dimensional,

- **Co-occurrence Texture** (CT) 75-dimensional.

The single graph case creates a graph with nodes \(x_i\) formed by shots, and edge weights, \(W_{ij}\), the linear combination of the normalized distance between two nodes in each modality. For the single-graph learning case,

\[
W_{ij} = \begin{cases} 
\sum_{\text{feat}} \exp\left(-\frac{d_{\text{feat}}(x_i,x_j)}{\sigma_{\text{feat}}}\right) & \text{if } i \neq j \\
0 & \text{else}
\end{cases}
\]

(6.1)

for \(\text{feat} \in \{EDH, CM5 \times 5, CM3 \times 3, WT, AC, HSV, CT\}\). The multi-graph learning case takes each of these features and creates a separate graph, and combines them using the weighting scheme found in Section 5.3.4.

Figure 6.1 shows the results of this study. The proposed approach performs better than the state-of-the-art found in [85] where prior labels are assumed from the noisy initial transcripts. This results from the repeated emphasis of initial incorrect labels,
explained as problematic in Section 5.3.2. Both the semi-supervised approach and the unsupervised approach significantly outperform an annotation method that does not leverage mining of similar documents. Figure 6.2 shows an example set of tags from mining without supplementing with graph reinforcement (using only the initial transcript for annotation, and adopted as a baseline for comparison), mining after supplementing with several types of graph reinforcement, and the associated ASR.

**Figure 6.1:** Graph showing proposed unsupervised approach vs. state-of-the-art semi-supervised approach as well as the the annotations resulting from using only initial transcript. Unsupervised approach performs as well as state-of-the-art semi-supervised approach, and all approaches significantly outperform the approach that does not use similar documents.
He also pointed out that new system is other nuclear powers no problems and make efforts to make up for world women today no revealed that he is referring to what system 6 in the future of Russia is facing what reply it was learned that Russia is developing a new generation of missiles and heavy and those in as many as 10 A nuclear warhead. 

"and for the new peace plan would ever be good enough to last decade of his life was more like the Palestinian suicide bombings."

first all his wife thailand theresa hance in pennsylvania a church dropped objectives her husband for the votes !! now participate said a next wonder other countries to the negotiating table of armed services commander-in-chief has not been system voters in the world of people in both closely watching the united states election. 

and for the new peace plan would ever be good enough to last decade of his life was more like the Palestinian suicide bombings."

"for the new peace plan would ever be good enough to last decade of his life was more like the Palestinian suicide bombings."

The main contribution of this annotation process that leverages similar multimedia documents, of course, is in supplying new annotations that could not have been derived.
from the original. We observe that, on average, less that 11% of the annotations can be derived directly from the original document. Almost 90% of the annotations are mined from the neighbors of the document. This ability to find new annotations is unique to an unsupervised labeling system, since supervised systems must have training data for each annotation. Therefore, a study is done on the above methods to determine the method’s ability to discover new annotations. Figure 6.3 shows the average number of new annotations that are discovered using each of the proposed processes, by the average number of keywords per document. These discovered annotations represent terms not found in the original document and therefore impossible to discover without mining similar documents. Our unsupervised technique outperforms the semi-supervised technique in annotation discovery, indicating a limitation found in “stickiness” constraint of the modified semi-supervised technique.

6.1.3 Evaluation of Graph Size $N$

An experiment is performed to analyze the effect of graph size on annotation results. “Graph size” refers to the number of near neighbors that construct each of the graphs formed from different modalities, denoted as $N$ in Section 5.4. Intuitively, the multimedia annotation technique should be relatively robust to graph size, as the graph diffusion step incorporates document similarity into the annotation decision process.
Chapter 6. Case Studies: Graph-Based Video Annotation

Figure 6.3: Number of keywords “discovered” through graph reinforcement. Proposed unsupervised methods (“UL SGR,” “UL MGR”) discover significantly more keywords than state-of-the-art semi-supervised method (“SSL”). This factor indicates that the semi-supervised technique’s reinforcement of initial labeling (“stickiness”) hinders ability to discover new annotations.

The results using graph sizes 3, 5, and 20 for the unsupervised multi-graph case are shown in Figure 6.4. The plot reveals the tradeoff between graph size and relevance/coverage. The larger the graph size, the steeper the slope of the relevance/coverage curve. This observation seems reasonable, since large graphs have many uncorrelated videos and therefore performance drops more quickly despite its ability to discover
correct general terms initially. Thus, graph size offers systems a design choice about whether to optimize for high relevance or high coverage.

![Unsupervised Multi-Graph Learning: Study on Graph Size](image)

**Figure 6.4:** Effect of graph size on performance. Large graphs have high precision with low coverage; small graphs have lower precision with high coverage. Large graphs are able to find general, correct terms, but small graphs discover more specific keywords.

### 6.1.4 Evaluation of Weighting Parameter $r$

The value of parameter $r$ can be used to vary the gradation between smoothness differences on the multiple graphs, as described in Section 5.3.4. As $r \to 1$, the $\alpha_g$
values render an effect where only the smoothest graph is considered, $\alpha_{\text{smoothest}} = 1$
and $\alpha_{\text{otherwise}} = 0$. As $r \to \infty$, the weights $\alpha_g$ have the same value, $\frac{1}{G}$.

The higher the visual correlation of a video, the more its transcript should correlate. As the transcript and visual feature similarity are independent sources of information about the video, this serves as a sort of cross-validation that the choice of visual feature is appropriate. Results from an experiment which evaluated choice of parameter $r$ can be seen in 6.5. Using an intermediate value of $r$ which incorporated all graphs at varying weights is found to perform best. This validates the notion of using graph smoothness to derive weights in multi-graph learning.

### 6.2 Collaborative Video Annotation Using Graphs

In this section we extend the graph reinforcement approach presented in Chapter 5 to video data from a social media site. As thoroughly explained in Section 4.1 for the case of a geotagged image repository, at social media sites the owner’s annotation is vulnerable to user interpretation, intention, and the current vocabulary for the subject. This vocabulary may change over time, as, for instance, in the case where “swk” became a nickname for the “star wars kid” who became an Internet phenomenon. These annotations are typically incomplete and noisy since they result from one-time annotations from single users. The annotation set typically contains incorrect keywords and
Unsupervised Multi-Graph Learning: Effect of $r$ Parameter

![Graph showing effect of $r$ parameter.](image)

**Figure 6.5:** Graph showing effect of $r$ parameter. A biased weighting ($r = 2.5$) that does not emphasize only the smoothest case ($r = 1.1$) but does not equally weight all graphs ($r = 10000$) is best. Shows that smoothing cost function derived from graphs is effective for combining graphs.

is missing relevant ones. An automated method that provides both high recall and precision of tags is needed before these large databases can be effectively searched and viewed.

The experiments presented here motivate a new form of collaborative annotation afforded by an online media community where similar variations of the same video frequently exist and user annotations are noisy and incomplete. Since online media
networks are driven by large communities with similar interests, the content tends to grow virally and a video can be easily tagged by collecting and filtering the tags of similar videos. In sites such as YouTube, each video is uploaded and tagged independently, but using the overlap in such a site, better annotations are learned that improve retrieval and browsing of the site. This work leverages graph theory techniques on information from individual community users for annotation of videos with an unlimited vocabulary. By using the annotations provided from independent tagging instances, it furthermore addresses an opportunity uniquely afforded by today’s online media communities and one that has not been explicitly tested. The technique amounts to mining correct annotations out of a collection of possible annotations found from redundant data. Research on online media communities has motivated an exploration of the effectiveness of user annotation and the learning that can be done using the annotations independently generated by the users. The annotation scheme tested in this section addresses a timely problem put forth by the research community [24, 12, 23], that is, gleaning trustworthy keywords from the tagging synergy of multiple users of an online media community, that has not been previously attempted.

Experiments are conducted to evaluate the effectiveness of the unsupervised graph theoretic approach in Section 5.3.4 for automatic collaborative tagging. Specifically, a comparison is made between the unsupervised approach, the semi-supervised approach described in [85], and the original tags provided for a YouTube video by an individual
user. An analysis of annotation “discovery” of the proposed technique follows. The section concludes with a study of system parameters graph size, $N$, and weighting parameter, $r$, similar to the studies done in Sections 6.1.4 and 6.1.3.

### 6.2.1 Experimental Setup

Experiments in this work use YouTube videos as the annotation targets since such a database contains the necessary qualities of (1) repetitiveness, and (2) independent tagging instances. The repository was crawled to extract 728 videos. It was ensured that some of the videos had overlapping and similar content in the total set. For instance, several e-trade commercials featuring a talking baby are extracted, along with the clips YouTube has identified as “duplicates.” These duplicates and similar videos are not explicitly marked in the database. The database was grown to 728 videos in order to provide a significant number so that performance gain deemed from the graph reinforcement collaborative tagging was not random (resulting from inclusion of the most common annotations). The inclusion of similar videos is reasonable since it is believed that 85% of videos in YouTube are expected to have such overlap in the media database. Complicating instances were explicitly included, such as commercials with similar themes or revisions/edits of an original video, along with other randomly crawled YouTube videos. These instances are expected to make the task more difficult, but representative of YouTube videos which often undergo iterations or edits.
A keyframe was extracted on average once every 10 seconds. They are not regularly spaced in time but are the frames closest to the centroids of clustered CLD features. Using the centroid frames allows the keyframes to capture different scenes or views from the video. The similarity between two videos is considered to be the maximum of any pairwise keyframe similarity between the two videos. Certainly, a more sophisticated method for similarity estimation can be adopted. The annotation dictionary is limited only by the total set of user-supplied labels. The total lexicon dictated by the 728 videos consisted of 3326 words. It is expected that the interaction of the tags from a collection of similar and duplicate videos could be discovered and mined from the graph reinforcement process described above to filter and extend the tags. The same relevance/coverage metric used in Section 6.1 is also used in this section.

### 6.2.2 Unsupervised Learning vs. Semi-Supervised Learning

An experiment is done to compare the proposed approach to the semi-supervised approach from [85]. In this implementation, the system uses the owner’s initial labels for the videos as training data for the semi-supervised case; that is, \( Y^{b}_{i} = 1 \) if the keyword \( b \) appeared in the annotations in the database. If a label \( b \) is not present, results are only shown for giving it an “unlabelled” value of \( Y^{b}_{i} = 0 \), though an experiment was done using \( Y^{b}_{i} = -1 \) which showed similar or worse performance.
Figure 6.6 shows the results of this study. The proposed approach performs better than the state-of-the-art found in [85] where prior labels are assumed from the incomplete initial tags as explained above. This results from the repeated emphasis on initial incorrect labels from owners of the videos. The unsupervised approach significantly outperforms the original annotations, even at the same recall level. Figure 6.7 shows an example set of initial tags, as well as tags from both semi-supervised and unsupervised graph reinforcement and mining.

### 6.2.3 Collaborative Annotation Extension and Filtering

In the introduction to this section, the possibility of tag extension and tag filtering afforded by online communities is emphasized. Here, the performance of the algorithm in [85] and the performance of the unsupervised algorithm described in Section 5.3.4 are explicitly tested in each of these areas separately, which affords a unique opportunity to supply *new* annotations. Considering all relevant annotations found in this study, on average 69.6% are discovered only after mining neighbors; that is, on average, the initial labeling had 30.4% of the relevant annotations that can be discovered by collecting the annotations of similar videos. The ability to correctly find new annotations from weakly labeled training data without building a distinct model for a particular annotation is a unique feature of the system.
**Proposed Graph Learning vs. State of the Art**

![Graph showing proposed unsupervised approach vs. semi-supervised approach and the relevance/coverage point of the owner tags. Unsupervised multi-graph reinforcement performs the best. Unsupervised singular graph reinforcement offers higher relevance but lower coverage than semi-supervised learning since it considers fewer videos.](image)

**Figure 6.6:** Graph showing proposed unsupervised approach vs. semi-supervised approach and the relevance/coverage point of the owner tags. Unsupervised multi-graph reinforcement performs the best. Unsupervised singular graph reinforcement offers higher relevance but lower coverage than semi-supervised learning since it considers fewer videos.

Figure 6.8 plots the receiver operating characteristic for tag extension. A false alarm is a tag that is extended that is incorrect, and a miss is a correct tag that is not extended. These extended annotations represent terms not found in the original document and therefore impossible to discover without mining similar documents. Both the unsupervised and semi-supervised technique are successful at annotation discovery, though the
**Chapter 6. Case Studies: Graph-Based Video Annotation**

<table>
<thead>
<tr>
<th>ID: Title</th>
<th>Frame from Video</th>
<th>Initial Tags</th>
<th>OMG-SSL</th>
<th>Proposed Unsupervised Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>BJDdwZjB4z8: “star wars kid”</td>
<td><img src="image1.png" alt="Frame from Video" /></td>
<td><strong>star, wars, kid</strong></td>
<td><strong>star, wars, kid, starwars</strong></td>
<td><strong>star, wars, kid, lightsaber, starwars, fat, jedi, funny</strong></td>
</tr>
<tr>
<td>35p4PIxeMMU: “Women know your limits”</td>
<td><img src="image2.png" alt="Frame from Video" /></td>
<td><strong>comedy, dinner, fun, limits, women</strong></td>
<td><strong>women, know, your, limits, fun, dinner, comedy</strong></td>
<td><strong>women, know, your, limits, funny, woman, comedy, dinner, fun</strong></td>
</tr>
<tr>
<td>6q3HP4vFd7g: “E-Trade Superbowl XLII (42) 2008 ‘Baby’ Commercial !!!”</td>
<td><img src="image3.png" alt="Frame from Video" /></td>
<td>42, ad, advert, advertisement, baby, clown, commercial, e, england, etrade, giants, new, patriots, superbowl, trade, xlii, york</td>
<td><strong>trade, baby, commercial, e, superbowl, clown, etrade, xlii, ad, 42, advertisement, england, giants, york, patriots, advert, new</strong></td>
<td><strong>e, trade, baby, superbowl, commercial, super, bowl, clown</strong></td>
</tr>
</tbody>
</table>

**Figure 6.7:** Example test videos and annotations, with red bold text for correct, plain black text for irrelevant, and italics for incorrect. The left two examples exhibit the tag extension afforded by unsupervised and semi-supervised learning. The right example shows the tag filtering possible using unsupervised learning, that is not done properly by the semi-supervised algorithm. Collaborative tagging done using graph mining techniques generally improves the tags, providing new tags and filtering irrelevant ones.

The unsupervised technique shows a slight advantage arising from the “sticking” constraint of the semi-supervised technique.

In an effort to combat spam, owner annotations can be filtered by the algorithm.

**Figure 6.8** shows the receiver operating characteristic for tag filtering. A false alarm is an incorrect initial tag that is kept, and a miss is a correct initial tag that is discarded by the algorithm. The unsupervised technique has a better ROC since it does not repeatedly reinforce noisy, incorrect initial labels as explained in Section 5.3.2 and as is clear from...
Chapter 6. Case Studies: Graph-Based Video Annotation

Figure 6.8: ROC curve for each algorithm’s treatment of tag extension and tag filtering. **Tag extension**: TP for annotation extended by algorithm to video and judged correct, FP for extended but judged incorrect, TN for not extended and judged incorrect, FN for not extended but judged correct. Unsupervised annotation has better tag extension qualities because of elimination of “stickiness” on initial labels. **Tag filtering**: TP for annotation judged correct and kept by algorithm, FP for judged incorrect but kept, TN for judged incorrect and discarded, FN for judged correct but discarded. Unsupervised learning has more area under ROC and therefore better treatment of owner annotations.

Equation 5.5. The unsupervised learning algorithm proposed here is better suited for gleaning signal from noise in cases of with large amounts of noisy initial data.
6.2.4 Annotating Without Initial Keywords

A special study is done to examine the performance of the algorithm when annotating a new video that had not already been given initial keywords by the owner. This scenario corresponds to annotations at the time of upload that could be suggested to the user. As shown in the relevance/coverage graph in Figure 6.9, both algorithms perform quite well annotating totally unlabeled documents, a consequence of the presence of aliased or similar data. When annotating without initial keywords, the “stickiness” constraint explained in Section 5.3.2 has been removed. Thus, the proposed algorithm only offers a slight advantage specifically when the video has not already been annotated but performs on par with OMG-SSL.

6.2.5 Evaluation of Graph Size, N

An experiment is conducted to find the effect of graph size on annotations. The results using graph sizes 3, 5, and 10 for the unsupervised multi-graph case are shown in Figure 6.10(a). The plot reveals the tradeoff between graph size and relevance/coverage. The smaller the graph size, the steeper the slope of the relevance/coverage curve, an observation also made in 6.1.3. This result arises because smaller graphs have fewer uncorrelated videos and therefore will have higher precision. However, the smaller graphs are limited in learning all appropriate annotations, and therefore do not exhibit the same coverage that larger graphs are able to achieve. Each node introduces potential
Figure 6.9: Performance of proposed algorithm and state-of-the-art when presented with an unlabeled video. Superior performance results from redundancy in social media sites. OMG-SSL performs on par with proposed method since the sticking constraint of the initial labeling is not a hindrance in this scenario.

keywords, so more potential keywords can be found with more nodes. As in annotation through transcript mining, graph size offers systems a design choice about whether to optimize for high relevance or high coverage.

6.2.6 Evaluation of Weighting Parameter, $r$

The value of parameter $r$ can be used to vary the gradation between smoothness differences on the multiple graphs, as described in Section 5.3.4. As $r \to 1$, the $\alpha_g$ values render an effect where only the smoothest graph is considered, $\alpha_{\text{heat}} = 1$ and
otherwise $\alpha_g = 0$. Increasing $r \to \infty$ results in equal weighting, $\alpha_g \to \frac{1}{G} \forall g$. The higher the visual correlation between videos, the more their tags should correlate. As the tags and visual feature similarity are from independent sources but both provide information about the video, tag correlation serves as validation that the choice of visual feature is appropriate.

Results from an experiment which evaluated choice of parameter $r$ can be seen in Figure 6.10(b). Effectively using only one graph with $r = 1.1$, as expected, results in the worst performance. Using a small value of $r = 1.4$, which incorporated all graphs at varying weights is found to perform best. However it is only slightly better than using all graphs at equal weights because all of the visual features chosen are known to be relevant in video search. The performance would be more profound if using features that are only relevant for some videos; for instance, audio features are very relevant in annotating music videos but less relevant perhaps in commercials. Still, the slight improvement offered by using weights rather than equal weighting validates the notion of using graph smoothness to derive weights.

This section presents the gains achievable through collaborative tagging in a community video site. Sites such as YouTube are populated by repeated, duplicate, and related documents because of the viral nature of Internet media. This section presents a robust method for automatically annotating documents in such a database. Stable graphs found in one medium (visual) are used to supplement the annotations in another.
relevant media form (text). This method is robust and trainable to particular qualities of the target data as well as annotation goals, with strong performance from a range of graph sizes and weighting parameters.

6.3 Conclusions

In this section we have presented video annotation experiments using a graph theoretic search and mining scheme. Experiments are conducted on video datasets that had associated transcripts as well as user-generated videos from YouTube. In both experiments, great gains are afforded by first identifying similar videos and using them to supplement the annotation. The technique is capable of identifying correct annotations even when using a large number of other videos. It successfully discovers new keywords while filtering out incorrect ones, providing the most robust method for automatic video annotation to date. It also presents the first experiments on collaborative tagging using only user-generated video.
Figure 6.10: a) Effect of graph size on performance. Performance tends to improve with larger graphs because more nodes result in positive reinforcement of commonalities without reinforcing irrelevant annotations from the additional nodes. b) Effect of $r$ parameter. A biased weighting ($r = 1.4$) that significantly weights the smoothest graphs rather than equally weighting all graphs ($r = 1000$) has the most area under relevance/coverage graph. Smoothing cost function derived from graphs effectively combines the graphs for multi-graph reinforcement.
Chapter 7

Conclusions and Future Directions

The work in this thesis contributes a new dimension to multimedia annotation that can work alongside traditional feature-based annotation methods, and it opens up a new field of social media annotation research and analysis which has not been previously studied. The algorithms presented are the beginning of a research field with significant opportunities.

7.1 Multimedia Annotation

The contributions of this thesis to the deep field of multimedia annotation lie primarily in the use of a retrieval model to annotate multimedia that works effectively with large databases and in the presence of metadata. The work in this thesis concludes
that the redundancy of Internet-scale annotation allows it to be successfully used in image and video annotation. However, the extent of redundancy that exists and that is required to employ these algorithms has not been conclusively explored. Furthermore, room still exists to research incorporation of tag types into the annotation problem. These two areas are explored below.

7.1.1 Redundancy

The analysis done on geotagged images make clear that the level of redundancy in a dataset is a contributing factor to annotation quality when using a search and mining technique. We found more success in annotating dense quadtree nodes and small regions such as Los Angeles. However, the necessary level of redundancy has not been explicitly addressed and is an area that can be more rigorously explored. Future work lies in the area of evaluating performance using an increasing database size, akin to the work done by Hays and Efros on geolocating an image using various database magnitudes [31].

7.1.2 Annotation-Dependent Fusion

We have used graph smoothness to find weights for combining similar videos. The weights are found over the attributes (that is, the initial annotations) as a whole, finding the average smoothness over each attribute. However, it’s possible that the weights can
be extracted *per annotation*. Rather than averaging smoothness over the annotations associated with the collection of nodes, we can find the smoothness of the graphs for each annotation. This makes intuitive sense, because annotations such as “music video” for instance will depend on different features than perhaps “funny.” This would amount to reformulating the equation for the weighting terms to be a function of $b$ by eliminating the summation over $b$.

### 7.2 Social Media

Social media, as a brand new media form, offers fertile research ground due both to its novelty and to the complications that arise from its lack of defined structure. This thesis has addressed at least a few of the opportunities offered by social media sites, but as is typical, it leaves us with more questions unanswered than when we began.

#### 7.2.1 Tag Classification

We have made clear that before effective annotation on social media data can be addressed, the tags must be classified into groups. Initial attempts at this are described on three tag types, termed “visual,” “geographic,” and “landmark.” However, improvements are still possible and other categories can be identified, such as event tags from
temporal information. Additionally, actual incorporation of these tag types into the annotation assessment has not been extensively explored.

### 7.2.2 User Analysis

Many opportunities for future work exist in incorporating more metadata about users, including incorporation of human factors such as owner hometown. Previous attempts to geolocate tagged photos have focused on using visual features only [31] or combining visual and text tags to predict the geocoordinates of the photo [15]. However, this work neglects incorporation of human factors and pathologies in geocoordinate prediction. Rather than initializing geographic prior as a uniform distribution, one consideration is to incorporate a biased prior based on owner hometown or other learned behaviors (e.g., a person from Santa Barbara is more likely to have photos from that area).

Additionally, we note that all existing research on Flickr focuses on the photos themselves or on geographic locations. Work may address questions such as: can we identify groups of users? What are the different kinds of groups formed in Flickr? Can this information be used to “smooth” or inform the prior distribution for other analyses (e.g., the geolocation prior described in the previous section)?
7.2.3 Analysis Alongside Traditional Media

There has been little research on the value of georeferenced user-generated content in more traditional media forms. In particular, the ways in which social media can inform and improve traditional geospatial analysis have not been addressed. Investigating novel ways to integrate social media data with remotely sensed imagery, just as an example, may boost signals in both corpora and allow robust extraction of information using machine learning. This new form of data fusion has the potential to enhance the value of satellite imagery and aerial photos in the Earth and social sciences, as well as augment social media analysis and applications by robustly improving data quality.

No research has been done investigating the potential of this data to aid the solving of large problems in the Earth and social sciences. This investigation could be done by experimenting on the fusion of georeferenced user-generated data with remote sensing data. It may also be possible to improve the quality of user-generated content by integrating signals from remotely sensed imagery. Specifically, the work may investigate ways in which georeferenced user-generated content can inform three specific image analysis problems: 1) land-use/land-cover (LULC) classification, 2) road/trail detection, and 3) geospatial object recognition.
Land-use/Land-cover Classification

Remotely sensed imagery is a primary source of data for performing land cover classification (identification of one of a very limited set of land types, e.g. urban, evergreen forest, water, etc.) and land use classification (which provides information about the human use of the Earth’s surface, e.g. parkland, residential areas, commercial zones). This information has important implications, from the understanding and prediction of climate change to resource management and city planning. Geotagged content has not previously been used to inform LULC classification. Geotagged text may take the form of Wikipedia articles, geotagged tweets (desert tweets will draw from a different vocabulary than tweets at bodies of water, for example), and Flickr tags, for instance. These can be used to construct a non-uniform prior for the image labeling process. Machine learning using volunteered geographic information and graph theory will afford a sort of “hybrid intelligence” that realizes large improvements by folding human intelligence in with traditional artificial intelligence techniques.

Map Improvement

Remotely sensed imagery is also the primary source for creating road maps. Automatically detecting smaller roads and trails is a challenging problem particularly in undeveloped regions. The proposed work investigates how user-contributed global positioning satellite (GPS) tracks can inform automated road detection in remotely sensed
Chapter 7. Conclusions and Future Directions

imagery. GPS tracks are readily available from open-source mapping projects such as OpenStreetMap. While the raw tracks are too noisy for delineating the roads directly, they represent an excellent source for seeding image-based road extraction from remote sensing data.

Geospatial Object Recognition

Using remotely sensed imagery to automatically detect complex geospatial objects such as schools and airports remains an appealing but elusive research problem. Current geographic databases—gazetteers—are frequently out-of-date or lack critical information, such as the accurate spatial extent of the objects, as most gazetteers simply provide single-point locations. It is possible to investigate how geotagged text from social media data combined with remotely sensed imagery can inform the detection of novel object instances, as well as help estimate the spatial extent of known structures and instances, an application which certainly has national security implications.

Using Remote Sensing to Improve Social Media

The automated detection of visually distinctive objects, such as athletic stadiums, harbors and mobile home parks, is possible using today’s sub-meter resolution in remotely sensed imagery. A further area of investigation lies in detecting objects in remotely sensed imagery that can be used to annotate georeferenced terrestrial images
such as those in Flickr. For example, the set of candidate annotations from object detection can be further refined using the remotely sensed image content with well-developed machine learning and data fusion techniques. Integrating aerial data to social media applications represents a new, independent source of information that can greatly extend current capabilities.

**Further Applications**

Using social media databases for learning that instructs defense, industrial, or commercial applications has yet to be accomplished, but in the future the incorporation of volunteered information seems unavoidable. Already, reliance on volunteered content from non-experts is used to quickly disseminate information during emergencies, such as maps indicating fire or flood areas. Aerial image analysis can be combined with volunteered content to address a host of applications such as surveillance, climate change analysis, and emergency management, besides improving existing commercial applications such as location-based advertising and personalized recommendation systems.

**7.3 Conclusions**

In this thesis we have addressed the multimedia annotation problem from a search and mining perspective that leverages the large scale databases available on the Internet.
Chapter 7. Conclusions and Future Directions

We have successfully shown the power of such a system in analyzing and correctly identifying the content in both images and video. Furthermore, we have performed the annotation using a larger dictionary size than ever considered before, and done it in a highly scalable way, while leaving future opportunities in the area of map improvement, user analysis, and social media improvement open for further exploration.
Bibliography


Bibliography


166


Bibliography


[64] T. Quack, B. Leibe, and L. Van Gool. World-scale Mining of Objects and Events from Community Photo Collections. In *Proceedings of the Conference*


[69] C. Shirky. Ontology is Overrated – Categories, Links, and Tags. 11


[76] T. Tran, R. Wehrens, and L. M. Buydens. KNN Density-Based Clustering for High Dimensional Multispectral Images. *Computational Statistics and Data Analysis*, 2006. 60, 61


[84] VRL. http://cortina.ece.ucsb.edu/. Website. 42


[95] YouTube. http://www.youtube.com/. Website. 4


