

LEVERAGING GEO-REFERENCED DIGITAL PHOTOGRAPHS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Mor Naaman

July 2005

© Copyright by Mor Naaman 2005  
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

Hector Garcia-Molina Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

Andreas Paepcke

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

Terry Winograd

Approved for the University Committee on Graduate Studies.

# Abstract

Given automatically captured metadata such as time and location about photos in a personal collection, we devised a series of methods for supporting photo management. These methods allow an enhanced level of semantic interaction with photo collections, while requiring little or no effort from the collection’s owner.

This work describes how we automatically organize such geo-referenced collections into semantic location and event hierarchies that can greatly improve the user’s browse and search tasks. We present an evaluation of our browsing system, PhotoCompass, including a comparison to a map-based photo browsing approach. We also show how browsing geo-referenced collections can be augmented with other types of context that is derived from time and location and is useful for retrieval in the context of personal photo collections.

In addition, we show two ways in which time and location data, coupled with a minimal amount of user annotation, can effectively suggest some of the semantic content of non-annotated photographs.

First, as the user annotates some of the identities of people in their collection, patterns of re-occurrence and co-occurrence of different people in different locations and events emerge. Our system uses these patterns to generate label suggestions for identities that were not yet annotated, effectively guessing the “content” of photos in terms of the people that appear in them.

Second, we describe LOCALE, a system that allows users to implicitly share labels for photographs based on location. For a photograph with no label, LOCALE can assign a tentative label using knowledge about other photographs that were taken in the same area. The system thus allows automated label suggestions and text-based

search for unlabeled photos. LOCALE effectively guesses the “content” of photos in terms of landmarks that appear in them.

# Acknowledgements

Many people have contributed in various ways to make this work possible. First, I would like to express my deep gratitude to my advisors, Hector Garcia-Molina and Andreas Paepcke.

I thank Hector for infecting me with his contagious excitement for research. I greatly appreciated Hector's simple and bright research style. In particular, Hector excelled at forcing me to distill, organize and understand my thoughts: from the sentences I was saying, to the claims I was making, the papers I was writing, and the work I was doing. In addition, Hector is one of the most pleasant, outgoing, unassuming people that I have ever met; his responsiveness as an advisor, despite serving as department chair and on many different corporate boards and government committees is unrivaled. Finally, Hector had taken some of best photos of me. I do not think I could have chosen a better advisor for my Ph.D.

I thank Andreas, my second advisor, who I consider a mentor as well as a close personal friend. Andreas was there to provide assistance in everything from research problems to personal dilemmas. I enjoyed our endless hours of helpful discussions and fun conversations, which I will cherish and miss greatly.

I would like to thank Terry Winograd for helpful comments throughout my Ph.D. work, and especially for his reading of this thesis. It has been a pleasure to observe and interact with Terry for the last few years. I also grateful to the final two members of my oral's committee: Scott Klemmer and Barbara Tversky.

I was fortunate to work with a wonderful group of co-authors: Susumu Harada, Yee Jiun Song, QianYing (Jane) Wang and Ron B. Yeh. Not only have they made a crucial contribution to the research, but most of all, they became my friends. All

the research and leisure time we spent together was characterized by fun atmosphere and great interaction, and without any conflicts.

I would like to thank the numerous individuals that facilitated various parts of this thesis. First I would like to thank Jichun (James) Zhu for his devoted programming work on early PhotoCompas prototypes. Kentaro Toyama and Ron Logan of Microsoft Research, the creators of WWMX, have my thanks for giving us access to their tools and data. I am grateful to Marti Hearst and Kevin Li of UC Berkeley’s SIMS, who made the Flamenco system available to us; Kevin even made code modifications to adapt the program to the needs of our experiments. I would also like to thank Alan Steremberg and Lawrence G. Pawson from Weather Underground for granting us access to weather data for the entire planet. The experiment in Chapter 7 was made possible by the help of the Stanford Visitor Center, whose staff was extremely helpful; I would especially like to thank Sean Fenton, Andrea Pazfrost and Lisa Mendelman for their efforts. I greatly appreciate the help of Meredith Williams, who supported this work with guidance around the complicated world of Geographical Information Systems, guidance that quickly become a fun friendship. Finally, I want to thank all the participating users in our various experiments, some of whom made a serious time investment with little reward.

Andy Kacsmar, our system administrator had a large part in enabling this work. Andy supported our system requirements and solved all the issues in a kind and effective way. He bravely suffered through various problems, incidents, requirements and failures, and always replied promptly and with great patience. I thank Andy for that.

I have immensely enjoyed interacting with the members of the Stanford Database Group (now the Stanford InfoLab) and the HCI group. I will not list them all here. I *will* list my officemates during these years, to whom I am grateful for their friendliness and helpfulness: Beverly Yang, Zoltán Gyöngyi, Sep Kamvar, Teg Grenager, and Bob Mungamuru. They all courageously endured my — shall we say, ‘interesting’ — taste in music.

Marianne Siroker, the group’s dedicated administrator, has literally taken care of everything. Marianne made my life here at Stanford as free as possible from

bureaucracy and administrative considerations. Above all, Marianne did it all calmly, professionally, and while smiling. Sarah Weden was always there to offer her assistance as well.

Many important people did not contribute directly to this thesis work, and were not officially present in my “work life.” However, they still paved the way for my Ph.D. becoming a reality.

Dr. Tamir Tassa is responsible for planting the seed of my interest in pursuing a Ph.D. In addition, Tamir has accompanied the entire process of my Ph.D. as a friend and mentor: from the application, through admission and into the actual “ride” through the program. Tamir was always there with invaluable advice and strong encouragement. It is appropriately symbolic that my stay at Stanford is bounded by two trips to India with Tamir.

My family, of course, was always with me — despite being in Israel, 24 hours of travel and 10 timezones away. Their love and unconditional support and pride have traveled across the ocean and helped greatly throughout my Ph.D. as it does, always, in life.

I would also like to acknowledge all my friends at Stanford, Israel, and everywhere else in the world. They make it all worthwhile.

Last but not least, to Dr. Tonya Putnam, who had a significant share in my life during these Stanford years. I cannot claim I could not have written this thesis without Tonya’s support. However, I sure loved writing it with Tonya at my side (and slightly ahead).



# Contents

<b>Abstract</b>	<b>iv</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Related Work</b>	<b>8</b>
<b>3 Automatically Organizing Photo Collections</b>	<b>15</b>
3.1 Discovering the structure of an image collection . . . . .	17
3.1.1 Output Requirements . . . . .	20
3.1.2 A Three-Pass Algorithm for Generating Location and Event Categorization . . . . .	25
3.2 Naming Geographic Locations . . . . .	31
3.2.1 Naming a set of Geographic Coordinates . . . . .	32
3.2.2 Naming Location Clusters and Events . . . . .	39
3.3 Experiments and Results . . . . .	41
3.3.1 Evaluation of Event Segmentation . . . . .	43
3.3.2 Evaluation of Location Hierarchy . . . . .	49
3.3.3 Evaluation of Naming . . . . .	51
3.4 Related Work . . . . .	53
3.5 Conclusions and Future Work . . . . .	55

<b>4</b>	<b>Interacting with Photo Collections</b>	<b>56</b>
4.1	Two Experimental Browsers . . . . .	59
4.1.1	PhotoCompas . . . . .	59
4.1.2	WWMX Browser . . . . .	62
4.2	Experiment . . . . .	63
4.2.1	Participants . . . . .	64
4.2.2	Procedure . . . . .	65
4.2.3	Other Procedural Considerations . . . . .	66
4.3	Results . . . . .	67
4.4	Discussion . . . . .	71
4.4.1	Measured and Questionnaire Results . . . . .	72
4.4.2	Results from Debriefing Session . . . . .	73
4.5	Fixed vs. Ad Hoc Hierarchies . . . . .	75
4.6	Conclusions and Future Work . . . . .	76
<b>5</b>	<b>Enhancing the Context Metadata</b>	<b>78</b>
5.1	Metadata Categories . . . . .	81
5.2	The Application Interface . . . . .	87
5.3	User Study . . . . .	89
5.3.1	Statistics and Setup . . . . .	90
5.3.2	Method and Results . . . . .	91
5.4	Survey . . . . .	93
5.4.1	Method and Results . . . . .	94
5.4.2	Independent Recall Descriptions . . . . .	99
5.4.3	Recalled Cues vs. Useful Cues . . . . .	102
5.5	Conclusions and Future Work . . . . .	104
<b>6</b>	<b>Resolving Identity in Photo Collections</b>	<b>105</b>
6.1	Related Work . . . . .	109
6.2	Model . . . . .	111
6.2.1	Interaction Model . . . . .	111
6.2.2	System and Parameter Model . . . . .	113

6.3	Generating Label Suggestions . . . . .	115
6.3.1	Basic Estimators . . . . .	115
6.3.2	Estimating Co-occurrence: PeopleRank . . . . .	119
6.3.3	Combining Estimators . . . . .	124
6.4	Evaluation Methods . . . . .	125
6.4.1	Evaluation Goals . . . . .	128
6.5	Results . . . . .	129
6.5.1	Casual Annotator Mode . . . . .	129
6.5.2	Industrious User Mode . . . . .	138
6.6	Conclusions and Future Work . . . . .	143
<b>7</b>	<b>From Where to What: Sharing Metadata</b>	<b>145</b>
7.1	The LOCALE System . . . . .	149
7.1.1	Centralized Mode . . . . .	152
7.1.2	Distributed Mode . . . . .	159
7.2	Experiment . . . . .	164
7.2.1	Experimental Setup . . . . .	164
7.2.2	Experiment Procedure . . . . .	166
7.3	Results . . . . .	169
7.3.1	Individual Collection Scenario . . . . .	169
7.3.2	Global Collection Scenario . . . . .	172
7.4	Automatically Assigning Captions to Photos . . . . .	177
7.5	Capture Location and Object Location . . . . .	179
7.6	Conclusions and Future Work . . . . .	181
<b>8</b>	<b>Conclusions</b>	<b>183</b>
<b>A</b>	<b>Generating Geo-referenced Photographs</b>	<b>186</b>
	<b>Bibliography</b>	<b>191</b>

# List of Tables

3.1	Examples for possible hierarchies when grouping photos by time or location. . . . .	17
3.2	Types of administrative areas and the weights assigned by the system to instances of each. . . . .	35
3.3	Sample datasets used in our experiments. . . . .	41
4.1	Summary of statistically significant differences between WWMX and PhotoCompas. . . . .	72
6.1	The basic estimators and the set of photos each estimator considers when ranking candidates to appear in photo $s$ . . . . .	116
6.2	Statistics for the collections used in our evaluation. . . . .	126
6.3	Summary of the virtual user modes. . . . .	127
7.1	Sample term-score table $TS_M$ for User M . . . . .	161
7.2	Sample photos and terms suggested by LOCALE. . . . .	178

# List of Figures

3.1	PhotoCompas system diagram. . . . .	16
3.2	A sample location hierarchy of a collection, using textual names to illustrate the clusters. . . . .	24
3.3	Processing steps in our automatic organization algorithm. . . . .	27
3.4	A sample sequence of photos and their segment/cluster association. . . . .	27
3.5	Processing steps in naming a set of coordinates. . . . .	34
3.6	Sample set of photos and a subset of the containing features of type “cities” and “parks.” . . . .	36
3.7	A set of coordinates and two nearby cities that may serve as reference points for the set. . . . .	37
3.8	A Map of the first-level clusters for collection $Z$ . . . . .	42
3.9	All nodes in the location hierarchy of Test Collection $Z$ . . . . .	42
3.10	Recall and Precision values for different conditions. . . . .	47
3.11	$P_k$ and $WindowDiff$ values for different conditions. . . . .	48
3.12	Recall and Precision values for different parameter settings. . . . .	49
3.13	$P_k$ and $WindowDiff$ values for different parameter settings. . . . .	50
3.14	Candidate term set for the Yosemite node of collection $Z$ . . . . .	52
4.1	Screen shot of PhotoCompas. . . . .	58
4.2	Screen shot of WWMX. . . . .	59
4.3	Sample PhotoCompas structure. Parts of the location and time/event hierarchies for an actual collection of photos. . . . .	61
4.4	Objective measurements of the Browse Task and the Search Task. . . . .	68
4.5	User subjective evaluation of the tasks in both applications. . . . .	69

4.6	User subjective evaluation of both applications. . . . .	70
4.7	Subjective reports on the helpfulness of Location and Time categories in browse and search tasks. . . . .	71
5.1	The metadata categories generated by our system, as shown in the interface opening screen. . . . .	79
5.2	A subset of the "Sri Lanka dusk photos" from the thesis author's collection, detected using contextual metadata. . . . .	80
5.3	Usage of the different metadata categories within the first two clicks in each trial. . . . .	92
5.4	Average values for how well subjects in different groupings remembered different categories for each of their photos. . . . .	95
5.5	Number of times each metadata category was revealed in photo descriptions. . . . .	100
5.6	Comparing perceived importance of search cues, recall cues, and cues used in photo descriptions. . . . .	102
6.1	Possible interaction mode with the annotation system. . . . .	106
6.2	Relations between people in test collection <i>A</i> . . . . .	120
6.3	5-Hit rate for the different estimators, averaged over all collections. . . . .	130
6.4	5-Hit rate for different geography-based estimators. . . . .	131
6.5	5-Hit rate for different time-based estimators. . . . .	133
6.6	Performance of the PeopleRank Estimators and their combinations, for each collection. . . . .	134
6.7	<i>h</i> -Hit rate (averaged over all collections) for various estimators vs. value of <i>h</i> . . . . .	134
6.8	5-Hit rate (averaged over all collections) for various estimators vs. size of the important people set <i>I</i> . . . . .	135
6.9	5-Hit rate for the different collections, using the Weighted Estimator and varying values of $p_{annotate}$ . . . . .	136
6.10	Comparing different weighting strategies. . . . .	137

6.11	Time sequence analysis of the system’s identity suggestion hit/miss result for individual identities. . . . .	140
6.12	Running window average 5-hit rate over the last 20 identity suggestions (labeling steps) for the <i>A</i> collection. . . . .	142
7.1	Architecture of the LOCALE system. . . . .	150
7.2	Sample Weighted Neighbors computation scenario. . . . .	154
7.3	Sample Location-Clustered computation scenario. . . . .	157
7.4	Map of Stanford campus, with geographical distribution of photographs whose labels contain the term “fountain”. . . . .	158
7.5	LOCALE Search results for “Hoover Tower” query. . . . .	167
7.6	The percentage of individual scenario queries that returned a relevant photo within first three results, for each strategy. . . . .	170
7.7	Average <i>recall at T(t, c)</i> for popular query terms. . . . .	171
7.8	The percentage of queries in each strategy that found a relevant photo in first three results, for global and individual scenarios. . . . .	173
7.9	Average $F_1$ values for least frequent query terms in different strategies, vs. retrieval limit. . . . .	174
7.10	Recall and precision at 15 for each query term. . . . .	176
A.1	Location Stamper screen shot. . . . .	189

# Chapter 1

## Introduction

As the photography world shifted from film cameras into digital cameras, computers now play a significant role in managing people’s photographs or, if you will, memories. Photos are stored, shared, searched and viewed — all in digital format.

Managing large personal collections of digital photographs is an increasingly difficult task. As the rate of digital acquisition rises, storage becomes cheaper, and “snapping” new pictures gets easier, we are inching closer to Vannevar Bush’s 1945 Memex vision [11] of storing a lifetime’s worth of documents and photographs. At the same time, the usefulness of the collected photos is in doubt, given that the methods of access and retrieval are still limited. With digital photos, the opportunity to go “beyond the shoebox” is attractive, yet still not entirely fulfilled.

One of the major hurdles for computer-based photo applications is the *semantic gap*. The semantic gap is defined by Smeulders et al. [82] as “the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.” Given perfect semantic knowledge about the photos, the task of organizing and retrieving from a photo collection would be made much easier. For example, if a system could automatically derive that a photo shows “Kimya drinking with Dylan at Robyn’s birthday, in New York”, this semantic knowledge could go a long way in helping users manage and retrieve from their collection. Sadly, current technology sometimes cannot even reliably detect that there are two people in the photo just described.



The existing approaches towards photo collection management can be categorized into four main thrusts. First, there are tools to enable and ease manual annotation (e.g., [80]). These tools let the user rapidly enter semantic information about the photos, to be used later when viewing or searching the collection. However, annotation is still cumbersome and time-consuming for consumers and professionals alike.

Closer to our approach, some systems (such as [31]) attempt to automatically organize photo collections using photo metadata, most notably the timestamps of photos. These systems often supply an interface and easy tools for the users to enhance and improve the organization manually.

The third approach encompasses methods like zoom and pan operations for fast visual scanning of the images (e.g., [7]). These tools attempt to bypass the semantic gap obstacle by posing the personal collection problem as one of visual search. The zooming thus allows the user to find relevant photos without the system having to discern the semantic content ahead of time. The visual tools, though, may not scale to allow the user manage tens of thousands of images without significant semantic information about the photos.

Finally, other systems attempt to directly address the semantic gap using content-based tools that try to extract semantic information from the visual image (refer to [91] for a survey of the area). These tools are not yet, and will not be in the near future, practical for semantic interpretation of personal photo collections. Indeed, low-level visual features can be easily extracted from the images. However, the semantic gap between identifying the low-level features and recognizing important semantic themes in the photographs, is still wide. For example, reliable face recognition is still not available, although recent improvements show better performance with relaxed requirements (e.g., when faces are directly aligned to the camera). Even more farfetched is the ability to identify semantic themes (such as events or activities) by analyzing visual features.

We expand on these areas and more related work in Chapter 2 and in other chapters that are directly relevant to the specific topic.

In the research reported upon in this thesis, we utilize photo metadata such as time and location to help narrow the semantic gap in digital photo collections.

Location is one of the strongest memory cues when people are recalling past events [93]. Location information can therefore be extremely helpful in organizing and presenting personal photo collections. Lately, technology advancements such as Global Positioning System (GPS) and cellular technology made it feasible to add location information to digital photographs, namely the exact coordinates where each photo was taken. While location-aware cameras are not widely available at the time of writing of this thesis, we project that they will become more common in the future. Even today, cameras that can be extended with a plug-in GPS device are available. Other cameras support a GPS API when connected through an external cable. More significantly, cameras embedded in cellular phones are now abundant — cellular is a location-aware technology whose location accuracy will be rapidly improving in the next few years. There are additional ways to produce “geo-referenced photos” using today’s technology. For a summary, see Toyama et al. [90]. It is our conviction that future readers of this thesis will find the discussion of location-aware camera technology redundant.

We use time metadata in concert with the location metadata described above. All digital cameras available today embed a timestamp, noting the exact time each photograph was taken, in the photo file’s header.<sup>1</sup> The time information is already utilized by commercially available photo browsers (Picasa, iPhotos, Adobe Photoshop Album and others). Novel research systems ([17, 27, 34] and more) also utilize the timestamps, perhaps more aggressively. We discuss those in more detail in Chapter 2.

Given time and location metadata, this research explores various paths to bridging, alleviating or evading the semantic gap in personal collection. Firstly, Chapters 3 and 4 investigate how automatic organization of a photo collection can assist the browse and search tasks. Chapters 5 and 6 look at integrating information from other sources, including user input. In Chapter 7 we expand our settings to allow sharing of information between different users. The discussion explores what additional benefits could be harvested from this type of sharing.

Next, we provide more details on the various parts of this thesis. For each chapter, we note the chapter’s contribution to moderating the photo collection’s inherent

---

<sup>1</sup>The industry-standard EXIF header supports time as well as location (coordinate) fields.

semantic gap.

In Chapter 3 we describe a set of algorithms that execute over a personal collection of photos. Our system, *PhotoCompas*, utilizes the time and location information embedded in digital photographs to automatically organize a personal photo collection. PhotoCompas generates a meaningful grouping of photos, namely browseable location and event hierarchies, from a seemingly “flat” collection of photos. The hierarchies are created using algorithms that interleave time and location to produce an organization that mimics the way people think about their photo collections. In addition, the algorithm annotates the generated hierarchy with meaningful geographical names. In Chapter 3 we also test our approach in case studies using three real-world geo-referenced photo collections. We verify that the results are meaningful and useful for the collection owners.

In Chapter 4 we perform a task-based evaluation of PhotoCompas and compare its performance to that of a map-based application. We constructed a browser for PhotoCompas that employs no graphical user interface elements other than the photos themselves. The users interact with the system via textual menus, created based on the automated organization of the respective photo collection into clustered locations and events. The application we compare against, WWMX, features a rich visual interface, which includes a map and a timeline. WWMX is a third party implementation [90]. We conducted an extensive user study, where subjects performed tasks over their own geo-referenced photo collections. We found that even though the participants enjoyed the visual richness of the map browser, they surprisingly performed as well with the text-based PhotoCompas as with the richer visual alternative. This result argues for a hybrid approach, but it also encourages textual user interface designs where maps are not a good choice. For example, maps are of limited feasibility on hand-held devices, which are candidates for replacing the traditional photo wallet.

Chapter 5 introduces a way to help alleviate the semantic gap by adding additional context information about the photo. The idea is that the context in which the photo was taken can sometimes suggest the content of the photo, or at least provide clues for finding photos in a collection. Fortunately, given time and location information about digital photographs we can automatically generate an abundance of related contextual

metadata, using off-the-shelf and Web-based data sources. Among these metadata are the local daylight status and weather conditions at the time and place a photo was taken. These context metadata have the potential of serving as memory cues and filters when browsing photo collections, especially as these collections grow into the tens of thousands and span dozens of years. For example, a user may remember that a certain photo was taken during a rainstorm. Even if the rain may not be part of the content of the photo (say, the picture was taken indoors), the context may help the user retrieve the relevant photograph.

We describe the contextual metadata that we automatically assemble for a photograph, given time and location, as well as a browser interface (an extension of the interface used in Chapter 4), which utilizes that metadata. We then present the results of a user study and a survey that together expose which categories of contextual metadata are most useful for recalling and finding photographs. We identify among still unavailable metadata categories those that are most promising to develop next.

Chapter 5 shows that the identity of the people who appear in photos is the most important category in personal photo collections: collection owners often remember photos by the identity of people in them, and often want to retrieve photos using identity-based queries. Chapter 6 tackles exactly this problem. In the chapter, we aim at determining the identity of people in photos of a personal photo collection. Recognizing people, or faces, in images is perhaps the most famous example of a computer’s struggle to bridge the gap between the visual and the semantic. Face detection and recognition algorithms have been a major focus of research for the last 15 years, but they still cannot support reliable retrieval from a photo collection, with high recall and precision. Even in the limited circumstances of a personal photo collection, where the number of “interesting” people does not exceed a few dozens, modern face recognition techniques do not perform well enough. A complicating factor in personal collections is that faces are not always well-aligned. In many photos faces are shown in profile, slanted, tilted or even partially or wholly obscured — a fact that makes even detection, let alone recognition, a difficult task.

The system we describe in Chapter 6 suggests identities that are likely to appear in photos in a personal photo collection. Instead of using face recognition techniques,

the system leverages automatically available context, like the time and location where the photos were taken, and utilizes the notions and computation of the event and location groupings of photos, as shown in Chapter 3. As the user annotates some of the identities of people in a subset of photos from their collection, patterns of re-occurrence and co-occurrence of different people in different locations and events emerge. Our system uses these patterns to generate label suggestions for identities that were not yet annotated. These suggestions can greatly accelerate the process of manual annotation. Alternatively, the suggestions can serve as a prior candidate set for a face recognition algorithm. Face recognition accuracy may improve when considering fewer candidates, and when assigning confidence partially based on context. We do not incorporate recognition algorithms in this thesis, and leave it for future work.

We obtained ground-truth identity annotation for four different personal photo collections, and used the annotation to test our system. The system proved effective, making very accurate label suggestions, even when the number of suggestions for each photo was limited to five names, and even when only a small subset of the photos was annotated.

In Chapter 6 we introduce user input, specifically as annotation of identities in photographs. In Chapter 7 we leverage another type of user input, namely free-text captions that users may enter for photos in their collection. We also introduce the concept of implicitly sharing information about photographs between users. Sharing will allow us to build a system that can translate from “where” to “what”. Here, the semantic gap between the visual representation and the object in the photo — a building, a geographical landmark, or a geographical feature, for example — is alleviated.

More specifically, Chapter 7 describes LOCALE, a system that allows users to implicitly share labels for photographs. For a photograph with no label, LOCALE can use the shared information to assign a label based on labels given to other photographs that were taken in the same area. LOCALE thus allows (i) text search over an unlabeled set of photos, and (ii) automated label suggestions for unlabeled photos. We have implemented a LOCALE prototype that supports both these tasks. The

chapter describes the system, as well as an experiment we ran to test the system on the Stanford University campus. The results show that LOCALE performs search tasks with surprising accuracy, even when searching for specific landmarks.

# Chapter 2

## Related Work

Most of the discussion in this chapter focuses on systems that enable browsing of personal collections of photos. The various techniques employed by these systems sometimes parallel the work described in this thesis, but are more often orthogonal to our work, and can be combined with our system and enhance it. We describe some technologies deployed by existing systems that relate to our work, like annotation and labeling. In addition, we touch on work in the realm of browsing geo-referenced photos and multimedia; very little of this work concentrated on personal collections. Finally, we list some related work on visualizing collections of photos, and we touch on current content-based image analysis techniques; the latter are not widely deployed in the settings of personal collections just yet.

### **Photo Browsing Systems**

Since the late 1990s, applications for management of personal collections of photos have been a major focus of research. The goal of most of these projects, much like ours, was to create tools for effective management of photo collections while requiring as little effort as possible from the users. Our research involves aspects of the photo management problem that can augment, enhance or replace certain components of these systems.

The photo browser research projects ([7, 22, 26, 31, 34, 44, 49, 64, 71, 88] to name a few) have developed and improved upon different management techniques for users'

photo collections, including:

- Automatic organization of photo collections
- Annotation and manual organization of photos and collections
- Display and visualization of collection organization
- Efficient and simple querying and retrieval

Some of the concepts developed by these research systems, in particular the labeling techniques and the strong notion of time and sequentiality in photo collections, made their way into major commercial systems, the most popular of which are Google's Picasa, Adobe's Photoshop Album and Apple's iPhoto.<sup>1</sup>

Parallel research thrusts focused on identifying the requirements for photo browsing systems. In particular, research in [25, 73, 75] focused on user studies and interviews, trying to assess how people manage and think about their photographs (often when the photographs in question are not even digital). Most relevant to this thesis, these studies often found that there is a strong notion of "events" in personal photo collections. See Chapter 3 for more detail about this aspect, and a more elaborate survey of related work on the topic.

A number of research papers experimentally studied particular aspects of photo organization. For instance, [16] studied the effectiveness of zoomable interfaces in the context of image browsing; [35] looked at the pile and other organizational metaphors for manual image organization; the researchers in [74] looked into how image similarity improves browsing for photos, although not exactly in the context of personal collections. All these techniques are complementary to the work in this thesis.

### **Annotation and Labeling**

Related research also tackled the problem of photo labeling and annotation. Efficient labeling of photos has been an active research field since 1999. Ease and partial automation of the labeling task were directly addressed in [46, 49, 80, 94, 95]. For

---

<sup>1</sup><http://www.adobe.com/>, <http://www.picasa.com/>, <http://www.apple.com/iphoto>



example, [80] proposed a drag-and-drop approach for labeling people in photos. More recently, the MediaFinder [46] system offered a flexible interface for the user to organize their personal media items spatially in a variety of semantic structures, and annotate multiple photos efficiently using the same framework.

The latest photo browser commercial packages (mentioned above) also attempt to support efficient labeling of photos using techniques developed by researchers. In Photoshop Album, for example, labels are divided into categories including place, event and people. The categories can be further divided into sub-categories. For instance, the people category can be divided into such subcategories as ‘friends’, or ‘family’. The labeling techniques are again orthogonal to ideas presented in this work: in Chapters 3, 6 and 7 we show how to generate location, event and identity labels automatically, semi-automatically (with some user aid) and sometimes based on sharing labels with other users.

More relevant to our work in Chapter 7 is the field of collaborative labeling. In the context of photos, collaboration in labeling has been explicit, and has concentrated on allowing many users to label a shared collection of images. See [50] for an example where participants annotated a public collection of photos from the CHI 2001 conference. While [50] is geared towards an existing community, a broader example can be found in Flickr.<sup>2</sup> In Flickr, users upload photos to a web site. Photos can be labeled by tags or captions, and tags can be assigned by any user, not only the owners of a particular photos. Communities are created ad-hoc, as a product of the tags and labels (e.g., people interested in photos with the “CHI 2001” tag). Finally, the ESP game [92] offers a different version of collaborative labeling. The ESP game randomly matches two physically and virtually separate users (i.e., the two cannot communicate). The ESP system simultaneously presents the users with the same photo. The users score points if they succeed to type in the same textual label. The ESP game works well for semantic concepts in photos (i.e., “woman”, “cat”, “tower”) but cannot hope to correctly contribute to more semantically meaningful labels in a personal collection, where “Kimya”, “our cat, Tiger” and “Hoover Tower” are more likely labels.

---

<sup>2</sup><http://www.flickr.com>

In contrast to these collaborative labeling systems, the collaboration/sharing-based system we present in Chapter 7 is implicit. Sharing of labels is based on the location, and users do not need to share images or make their images public.

More closely related to our work, Davis et al. [18, 77] utilize spatial and temporal context to help annotation of photographs taken with camera-equipped mobile phones. They propose annotation using person, location, object and activity categories, and lay out similar ideas to ours as a possible approach for proposing identity and location labels (Chapters 6 and 7).

Additional work on annotating *identities* in personal photo collections was done in [32, 98] and even earlier in [49]. These systems do not incorporate context, but rely on recognition techniques. Details on these systems are supplied in Chapter 6. They are orthogonal to our approach.

While our work focuses on providing useful textual names for collections of photographs, Lieberman et al. have done work that attempts the reverse. In their work on Aria [55, 56], Lieberman et al. used natural language parsing techniques and a basis of digitally-represented “commonsense knowledge” to suggest relevant photos as a user types a story to describe a series of photographs.

Of course, other work that attempt to automatically produce annotation for digital photographs have been based on image analysis techniques ([24] and more); these are mentioned briefly below.

### **Location-Annotated Photos and Geographic Information Systems**

Associating GPS coordinates with digital photographs is a fairly recent development. While some of the photo browsing applications mentioned above use location as a navigation facet ([31, 44, 49, 89] and even a commercial system like Photoshop Album), very little work has been done on photo collections in which the exact location where each photo was taken is known. There have been few other projects which attempt to exploit the geographic information of digital photographs, amongst them the work of Davis et al. mentioned above [18]. In another notable research project, Toyama et al. built a database that indexes photographs using time and location coordinates [90]. This work, also known as the World Wide Media Exchange (WWMX),

explored methods for acquiring GPS coordinates for photographs, and exploiting the metadata in a graphical user interface for browsing. The WWMX system is described and studied in more depth as part of our discussion in Chapter 4.

In the last year, a number of commercial websites<sup>3</sup> appeared that allow upload of geo-referenced photos; those websites usually include a map interface, which can be used for browsing and searching for photos in a similar fashion to WWMX. Unlike WWMX and our work, these sites do not allow browsing for photos by the time they were taken. Additionally, the sites display a global, public collection of photos and do not support browsing only photos of a personal collection.

More closely related to our work in Chapter 3, Pigeau and Gelgon [69] used clustering in the space and time dimensions to detect groups of geo- and time-referenced photos. However, their system was only used to analyze relatively small sets of photos that represent individual trips, and not complete multiple-years photo collections.

There are several ad-hoc systems that incorporate a map interface for organizing personal photo collections. In [86], for example, the author presents a system called GTWeb, that automatically generates web pages with maps and other annotations from digital photographs and corresponding GPS track data. The points at which the photos were taken are graphically represented as trails on a (non-interactive) map. However, GTWeb focuses on representing photos from long trips, and does not support hierarchical navigation by event or location, for example. In 1999, Smith et al. [83] augmented a simple camera with a GPS device and digital compass. They planned to use the resulting photographs as keys for retrieval of historic data: in their system, clicking on a new geo-referenced photo returns a list of historic photos taken at the same location, and possibly the same orientation.

Still in the context of time- and geo-referenced photos, the PARIS system [51] proposes a temporal and spatial ontology for personal photographs. An ontology-based approach for personal photo collections was also the subject of [57], where the authors focused on designing an event ontology. In our work, we do not use an ontology of events or locations, although the LOCALE system described in Chapter 7 can possibly help in the generation of both ontology types.

---

<sup>3</sup>mappr.com, woophy.com, geobloggers.com

A large number of broader systems have been developed that deal with presenting generic geo-referenced data, especially within the Geographic Information Systems (GIS) community [14, 48, 54, 58] (to list a few) and the digital library community [43, 84, 101]. For example, in Kraak’s work [48], the author characterizes GIS applications along three axes: whether the map is tailored to an individual or to public use; whether the data presented is known or unknown to the user; and whether the user interaction is high or low. Within these axes, Kraak suggests that most of the research before 1996 focused on the (*public, known, low interaction*) combination. Indeed, the (*private, known, high interaction*) combination, that applies to personal collections of photos, is still not explored in depth. It should also be noted that nearly all of the systems developed rely solely on a map based interface, and none of the GIS research efforts cater to browsing and searching in the context of personal photo collections.

For an example for one of the GIS systems, in [14] the application displays geo-referenced photos as points on a zoomable map interface, but the user is unable to see any of the actual photos until a specific point is selected.

Maps may present a problem when the user operates on a small-screen device, as maps are not well suited for this environment. Our work in Chapters 3 and 4 can be easily extended to a small-screen device. While work has been done on summarizing maps for screen constraints [76], most applications of maps on small screen devices, like [72], focus on navigating a restricted area and context (e.g., driving directions).

## Visualization

Visualization of large photo collections is also an active field of research. Methods for fast visual scanning of the images, such as zoom and pan have been developed. These tools (e.g., [7, 8] which introduce zooming and the “Quantum Treemaps”) are quite helpful for viewing several hundreds, or thousands of photographs efficiently. They do not, however, easily scale to manage tens of thousands of images, or scale gracefully down to a small screen device, although [47] is an attempt to do just that in the same framework. Most importantly, these techniques are usually based on some a-priori semantic organization of the photographs (especially in the MediaFinder, [45, 46, 40]). Other approaches to visualization and zooming also depend on some

created structure (e.g., automatically-detected events), as in the work of Drucker et al. [22], and the work of Moghaddam et al. [65] in the context of table-top computers. This structure could be generated, for example, by the algorithms described in this thesis. In this way, our work is complementary to these visualization techniques. In [65], visualization also utilized content-based image similarity, as shown next.

### Content-based Techniques

Content-based approaches are not yet widely employed in photo browsing applications, not even in the research community. The contributions of content techniques to current systems can be categorized into two main aspects. First, similarity-based methods are used for automatic organization and retrieval in the context of personal collections of photos. For example, the system described in [71] performs automatic organization of photos into event clusters based on time *as well as* image similarity. Other systems that use a similarity feature for management of personal collections of photos include [26, 65, 88, 94]. In addition, as mentioned above, the benefits of similarity-based organization were studied in [74] albeit in a slightly different context.

The second aspect of content-based approaches applied to the photo browsing systems is, as mentioned above, annotation of faces. Face detection and recognition techniques are mostly used for aid in annotation, and somewhat in retrieval from photo collections. The content-based techniques for face recognition [32, 49, 98] are far from supplying a satisfying user experience at this point, and could be improved by augmenting some of the ideas we present in Chapter 6.

Other image analysis tools aim to supply content-based search for collections. This approach is very difficult, and semantic retrieval is far from feasible at this point. Image analysis in balance is not yet practical for the meaningful and comprehensive organization of photo collections. To mention some key approaches, some systems attempt to associate content with semantic concepts and labels [6, 24], query by content features [5] (QBIC), and design for content-based retrieval in databases [66]. None of these techniques have been experimented with in the context of personal photo collections. For a survey and summary of content-based image retrieval systems, see [91].

## Chapter 3

# Automatically Organizing Photo Collections

We address the problem of automatically organizing a personal geo-referenced photo collection, in order to facilitate efficient search and browsing for specific photos, or for photos of particular events. Here, and throughout this thesis, we assume that the camera tags each photo in the collection with the location coordinates and time, marking exactly where and when the photo was taken.

In particular, we are looking to automatically generate a structure that will enable browsing of the collection *without* the use of a map. Maps can be extremely inefficient in utilizing screen real estate. In many cases, especially in personal photo collections, pictures may be sporadic in one location, and highly concentrated in another location. Having to pan and zoom a map to the correct low-level location may be cumbersome. This problem intensifies when the user operates on a small-screen device. In addition, many people do not feel comfortable using a map — in a computerized or even a non-computerized environment. Finally, limited input mechanisms (such as cell-phone controls or voice activation, for example) may not be well suited to map-based manipulations. In the next chapter, we describe a user study in which we compared our approach to a map-based application.

Our system, PhotoCompass, performs two major tasks. First, it automatically

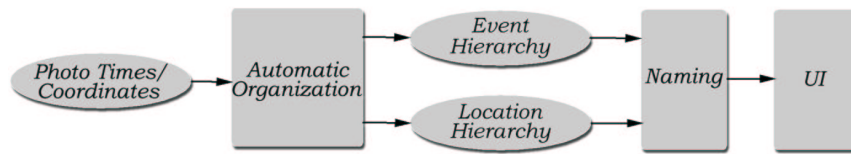


Figure 3.1: PhotoCompas system diagram.

groups the photos into distinct events and geographical locations. Second, PhotoCompas suggests intuitive geographical names for the resulting groups. Figure 3.1 illustrates the processing steps and outputs of the system.

To create the event/location grouping we use a combination of existing time-based event detection techniques [34] and a sequence-based, multidimensional clustering algorithm [30] to group photos according to locations and time-based events. This thesis presents, to our knowledge, the first published algorithm that proposes, implements and experiments with an event-detection algorithm that uses location information in addition to time. Moreover, the location-based grouping enables a new way to access photo collections.

For the second task, naming the groups, PhotoCompas generates a textual caption that describes, in geographic terms, each location or event. It is crucial to generate a good set of captions, since we are not using a map to identify geographic location to the user. Our technique utilizes a geographical dataset of administrative areas (provinces, cities, parks, counties etc.) and a web search engine such as Google [33] to generate these captions.

In this chapter we explore and experiment with the PhotoCompas set of algorithms. We also constructed a simple user interface that uses the algorithms' output. In Chapter 4 we expand on the interface, and on an experiment that compares user task performance on the PhotoCompas-based interface to a map-based browser implementation.

In the rest of this chapter, Section 3.1 describes the algorithms we use to discover the inherent structure of a user's geo-referenced photo collection. Section 3.2 shows how we generate a human-readable geographical name for a set of photos. In Section 3.3 we perform initial evaluation of PhotoCompas by running experiments on

three large sample collections of personal geo-referenced photos.

### 3.1 Discovering the structure of an image collection

The goal of our automatic processing is to group the user’s photos using two different categories, corresponding to the natural way users think about their photos [25, 75]. The first category is location, and the second is event. That is, we wish to group the photos into hierarchies of locations and time-based events. Naturally, these two dimensions interact: photos from a certain event are associated with a location; any location may have pictures taken in it at different times (for example, multiple trips to New York City).

Note that both time and location can easily be subjected to some pre-defined hierarchy. For example, any specific time can be easily categorized into a year, month, day, hour etc. Similarly, a location can be categorized by country, state, county, city etc. On the other hand, both time and location data can be grouped using only their continuous values, without adhering to any imposed hierarchy.

Table 3.1 shows some of the available options in designing time-based and location-based hierarchies. In our system, we *refrain* from using pre-defined rigid hierarchies in the structure-generation stage, for reasons discussed below. Instead, our system creates ad-hoc hierarchies based on events and geographical clustering (an “ad-hoc” hierarchy is one that is based on the particulars of each collection). Note that there is one exception: we are separating both events and locations by country — in our system, no event or location group can span country boundaries.

Table 3.1: Examples for possible hierarchies when grouping photos by time or location.

	<i>Pre-defined Hierarchy</i>	<i>“Ad-hoc” Hierarchy</i>
Time	Year/Month/ Date/...	Events based on time clustering
Location	Country/State/ County/...	Based on geographical clustering



It should be emphasized that we do not rule out using pre-defined hierarchies in the user interface. For example, an application based on the PhotoCompass set of algorithms may still employ a calendar browsing metaphor, possibly alongside an event-based breakdown of time.<sup>1</sup> However, here we are interested in automating *semantic* grouping of photos for the user. We believe that event representation, and ad-hoc location groupings are a better semantic representation of the collection, as explained below.

Other than representation, the semantic grouping also assists with other tasks, for example, annotation of people in the collection, as we show in Chapter 6. In addition, as we show later in this chapter, the time and location groupings will inform each other. For example, two photos taken very closely in time should end up in the same location cluster; similarly, if two consecutive photos clearly belong to different location clusters, they should most likely be assigned to different events even if the photos were taken relatively close to each other in time. We explain this reasoning in the next subsections.

The main argument against using a pre-defined hierarchy is that it is often too rigid, arbitrary, or not well perceived by users. An example that illustrates the problem may be an event that crossed day/month/year boundaries. Consider a time-based event that begins on the 31<sup>st</sup> of October and ends a few hours later in November. We do not want to split these photos into two different days (or worse, months). Another example may be location-proximate photos taken across state or city boundaries. Consider a user who took photos on the Boston/Brookline (Massachusetts) city line. Using a rigid pre-defined location hierarchy, the pictures from the Boston side will be separated from the ones taken on the Brookline side of the city line, while the user may think of the pictures as taken in a single location, whether or not she is aware of the split between the cities. Country boundaries, on the other hand, are generally well perceived and understood by photographers. The country borders are often more apparent, and in addition, taking photos near the country border is a rarer occurrence than across city or state boundaries. We therefore allow rigid country-based

---

<sup>1</sup>One might even consider merging the two representations in some manner; for example, when browsing a simple timeline, the application could “snap” the cursor to the beginning of a new event as detected by PhotoCompass.

breakdown of our sets of photos, as mentioned above.

Another problem is that a fixed hierarchy may be too bushy (many different cities in one state, or picture-taking days in one month). The fixed location hierarchy may have overlapping branches — a certain coordinate may be a part of both a national park and a city, for example. The overlap might cause confusion, and will also contribute to the tree’s bushiness, as more items are needed to represent the hierarchy.

An additional drawback of a fixed hierarchy is that it may not be sufficiently familiar to users. For example, “counties” are one level of the US location hierarchy, yet few users remember the name of the county where Anchorage is located, or the name of the small town five kilometers south of the Grand Canyon. While the states of the US are relatively well known (if not always well perceived), some countries have less known administrative divisions, or do not have them at all.

Some locations can fall outside all nodes of some level of a fixed location hierarchy. For example, a certain location may not be inside any city or park. The location will only be represented using a higher-level node from the hierarchy (say, the state), which may not be specific enough for lookup.

Finally, sometimes we can or want to bypass levels of the hierarchy altogether. For example, Consider a user who has only taken photos in three locations in the United States: New York City, Los Angeles and San Francisco. In this case adding a level of hierarchy for the states New York and California, not to mention the counties in which the photos were taken, is redundant if not cumbersome for browsing.

Instead of using a pre-defined hierarchy, we automatically create a personalized hierarchy for each collection in those two dimensions, using the values for time and location of the photos as real values in a continuous space. To refer to groups of photos in these hierarchies, we define two terms below, to be used throughout this chapter and thesis. Both terms will be defined more precisely in Section 3.1.2:

**A cluster** is a node in the location hierarchy, a group of photos that belong to one “geographic location.”

**A segment** is a *sequential* group of photos. In particular, segments can represent

user events. The terms segment and event are used interchangeably in this thesis.

Our algorithm, then, creates a customized hierarchy for a user’s photo collection based upon the automatically captured coordinates and times when the user has taken photos. For example, looking at the continuous location values only, our system will first group the Boston and Brookline photos together in one cluster. Later the system will decide how to name this cluster. Similarly, our system detects events even if they cross time boundaries (be it day, month or year). In our example, all the photos from the event of October 31<sup>st</sup>/November 1<sup>st</sup> will be grouped together in one segment.

Of course, in order to display the event and location information to a user we must use terms from well-known hierarchies (e.g., year, month, city names, ...). For example, naming a location “Cluster Number 4” is of little value. However, naming it “Boston, Massachusetts” is meaningful, even if the name may not be a strict description of the cluster contents (remember the Brookline photos). However, as shown in Figure 3.1, we only worry about naming the generated clusters and segments *after* the structure for the collection has been generated. In fact, we may merge (as a final step) clusters that occur in the same city — simply because we have no means of making them distinct to the user.

Eventually, PhotoCompass generates two distinct hierarchies for the browsing interface, location and event. In PhotoCompass, the user interface allows filtering based on both hierarchies in turn, so users are able to click through any “virtual” path that interleaves locations and events, but will not be restricted to a specific order of interleaving the two categories if they choose to navigate in a different way.

### 3.1.1 Output Requirements

As Figure 3.1 shows, the output of the automatic organization step are location and event hierarchies. Before we describe the details of the algorithm in Section 3.1.2, we list the requirements and guidelines for each of the outputs.

### Requirements for Event Category

People often think of their photos in terms of events: consecutive photos corresponding to a certain loosely defined theme such as a wedding; a vacation; a birthday etc. [25, 75]. Users inspecting their own photo collection ordered by time can easily draw boundaries between the different photographed events. We wish to mimic this human inspection as accurately as possible. That is, we attempt to group the photos into a sequence of segments, which represents a sequence of events. Hopefully, the segments accurately portray the set of picture-taking events the user had engaged in.

We make use of the “Story Line” assumption: all photos are taken by a single photographer, or alternatively, using a single camera (used by a number of family members). We make this assumption so we can treat the photo collection as a sequence of photos, coherent in space and time (i.e., no two pictures are taken at the same time in two different places). This assumption is not as restricting as it seems when moving on to a family collection with a few contributors and possibly a few cameras. Modern digital cameras insert the camera make and model as part of the photo metadata. If more than one camera appears in the collection, we could use this information together with time/space filtering to separate the different cameras and treat the photos as two separate sequences. Again, this thesis only handles the single-camera scenario.

A second observation, verified in a number of publications [17, 27, 34], suggests that people tend to take personal photographs in bursts. For instance, lots of pictures might be taken at a birthday party, but few, if any, pictures may be taken until another significant event takes place. We take advantage of this “bursty” nature of photo taking in discovering the event structure of the collection.

The event category can be flat or hierarchical. A flat category means we only identify the top-level events — in other words, the points in the stream of consecutive photos where the context has changed (e.g., “A birthday party”, “Trip to Mongolia”, “4<sup>th</sup> of July”). One simple way of doing that is to look at the time elapsed between two consecutive photos. Thus we create a flat grouping of the list of photos into events. In addition, we can use similar techniques to detect sub-events *within* each event. For example, there may be a few “bursts” of photos during a birthday event

that can be detected. Those bursts might be of pictures taken when the cake was cut, when the presents were opened, etc. In his work, Gargi [27] shows that personal picture-taking closely follows a fractal model — i.e., photos are taken in bursts, each burst composed of some finer bursts, and so on.

In this thesis work, we only create top-level events (i.e., flat event hierarchy) as those are the most crucial for browsing a photo collection. The work could be easily extended to support a deeper hierarchy of events, using, for example, techniques from [34].

Our event categorization follows one strict rule:

- i. (“Story Line”) Only consecutive photos can belong to the same event  $\varepsilon$ .  $p_1, p_2 \in \varepsilon \wedge (time(p_1) < time(p_3) < time(p_2)) \Rightarrow p_3 \in \varepsilon$ .

In addition, a number of (sometimes conflicting) observations serve as guidelines for our processing algorithm:

- ii. A gap of  $h$  hours between consecutive photos is often an indication for a new event. The value of  $h$  should change dynamically as informed by knowledge about the geographical clusters: more popular location means lower  $h$  value. For example, consider photos taken in the vicinity of the photographer’s home; they may attend a birthday one evening and participate in a rally the next morning; these are two different events. Compare this to two pictures taken 12 hours apart on a ski trip: a user would probably categorize them into the same high-level event.
- iii. Similarly (“Burst” assumption): within one event, photos are taken at a steady rate; an unusual time gap between photos *may* signal the beginning of a new event. An unusual gap may also signal the split of the original event into two sub-events.
- iv. The physical distance between the locations of two consecutive photos is another possible predictor for event boundaries. This is another way in which we utilize the geographical data to inform our time-based event segmentation.

### Requirements for Location Category

In this section we present the requirements and guidelines we use for the location hierarchy. The location hierarchy is used both for presentation in the user interface, and to inform the event segmentation.

Ideally, we would like users to be able to quickly “drill into” one specific location where they had taken photos. Our location hierarchy could in principle be arbitrarily deep, but our implementation uses a 2- or 3-level hierarchy.

The first level of the hierarchy is pre-defined: the country level. As mentioned above, we use the pre-defined country classification since the world’s division into countries is (generally speaking) well defined, and users are well aware of the “country” context: most people always know what country they are currently taking photos in. Therefore, the system initially splits all photos based on the country in which they were taken.

For the next level of hierarchy, we could consider using the country’s administrative division, such as states for the US. However, we avoid this division for the reasons discussed above. Instead, The second level of the hierarchy, as discussed above, is created by grouping the photos into *clusters* that are expected to make sense to the user. These clusters may vary, both in terms of the size of the area they represent, and the number of pictures in each cluster.

Figure 3.2 shows an example of such a location hierarchy, using textual names to illustrate the general area each cluster represents. This hierarchy is in fact part of the hierarchy created for one of our test collections, *Z* (see Section 3.3), and the node names are a shortened version of the names assigned by our naming algorithm (Section 3.2).<sup>2</sup>

The clusters can be recursively split into lower-level clusters (like the “Around San Francisco” cluster in Figure 3.2). To decide when to split clusters, PhotoCompass utilizes the time information. A cluster is split when the number of occasions the user had visited this location exceeds a threshold. The intuition behind this criterium

---

<sup>2</sup>We remind the reader that the names used for clusters are always just “fuzzy” representations of the area that the clusters cover; for example, the “Berkeley” cluster may include pictures taken outside the city proper.

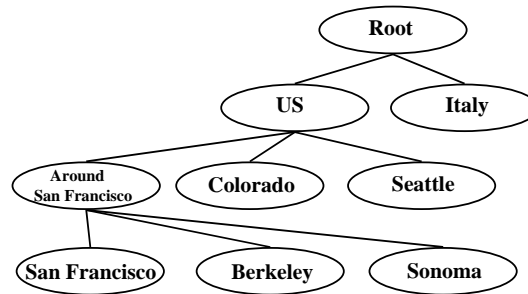


Figure 3.2: A sample location hierarchy of a collection, using textual names to illustrate the clusters.

is that people are more familiar with areas where they have taken photos on many different occasions. Compare this to another possible criterium: the number of photos taken in a certain location cluster exceeds a threshold. This criterium, while similar, may be less desirable: First, people may take a large number of photos on a trip, but they are not necessarily familiar with the area where the photos were taken as they visited it only once. Second, the photo count parameter is more user-dependent as different users will take a different number of photos even if they attend the same events.

To summarize, we want to create a location hierarchy that correctly represents the unique locations the user has visited. The branches of this hierarchy should be distinct enough so the tree correctly depicts the different areas where pictures were taken. At the same time, the tree must not branch excessively at any specific level, since that may be disorienting for browsing. In particular, our hierarchy is created while following these rules:

- v. The tree must not be too bushy:  $|\text{descendants}(\ell)| < n$  for every location node  $\ell$ . We used  $n = 10$  to keep the user interface menu sizes reasonable.
- vi. No redundant inner nodes with only one descendent:  $\forall \ell : |\text{descendants}(\ell)| = 0 \vee |\text{descendants}(\ell)| > 1$ .

Additionally, the hierarchy *tries* to follow these guidelines:

- vii. Location nodes that represent very few photos are merged with other nodes,

unless they show a substantial difference in their geographical location (based on distance and compared to distances between other clusters).

- viii. At any level of the hierarchy, photo  $p$  belongs to the node whose center is geographically closest.  $p \in \ell \Rightarrow \forall \ell_i \neq \ell : \text{samelevel}(\ell_i, \ell) \vee (\text{dist}(p, \ell) < \text{dist}(p, \ell_i))$  where the distance  $\text{dist}$  between a photo and a cluster is the distance from the photo to the cluster center.
- ix. Photos taken in close time proximity should belong to the same location cluster. Here we again use the time data (given the Story Line and Burst assumptions) to inform our geographic hierarchy. This rule will prevent pictures from the same event from being categorized to two different locations, and moreover, is likely to have the benefit of keeping together related locations that otherwise might have been split.
- x. Leaf clusters in the hierarchy are not overloaded, and should represent the level of knowledge the user is assumed to have of this location: if too many segments belong to a single leaf cluster, split this cluster into further geographic sub-clusters.

Often these guidelines can conflict with each other, or be impossible to adhere to given the user's particular set of photos. In the next subsection we present our algorithm that makes use of these guidelines while trying to balance the conflicting rules.

### 3.1.2 A Three-Pass Algorithm for Generating Location and Event Categorization

The algorithm described in this section creates both location and event hierarchies, and assigns all the photos to nodes in each.

As mentioned above, we assume that users are well aware of the countries where they are taking photos. Therefore, we process all photos from each country separately (we have a geographical dataset that enables us to query the location of each photo



to find the country where it was taken). For the remainder of this section, we assume that all photos are taken in one country.

Our goal can be broadly viewed as detecting structure inherent in a sequence of points in a three dimensional space, where time is one dimension, and geography accounts for the other two. Let us outline a few possible approaches to this problem, before describing our chosen implementation strategy:

- Use some pure three dimensional clustering, treating location and time as coordinates in Euclidian space. We experimented briefly with this option, whose main challenge is finding the correct scale for the time vs. location coordinates.
- Perform time segmentation first, then cluster the results using the geographical dimensions.
- Use the geographic coordinates first to detect geographic clusters, then apply time segmentation algorithm to the results.

Naturally, any alternation of these methods is also possible, i.e., an algorithm could iteratively apply each one of these methods in any order.

There is, of course, an abundance of time-series clustering algorithms that can be applied to geographic data. For references to some of them, see [30]. In addition, there are many clustering algorithms and techniques that do not necessarily treat the data as a series (e.g., k-Means); while it is possible to use these for our purpose, we claim below that it is useful to utilize the time series data and the time values to inform the geographic clustering.

Our implementation uses a hybrid time/location technique. The first step treats the photos as a sequence, and looks at the time gap and the geographical distance between each pair of consecutive photos to create an initial grouping into *segments* representing low-level events. Then, we cluster these segments using a geographic clustering algorithm. Finally, we make another time-based pass over the sequence of photos to decide the final breakdown into events (informed by the newly created location clusters). The process is illustrated by Figure 3.3.

Here we define more precisely the two concepts we introduced earlier, a *segment* and a *cluster*. Let  $P = (p_1, p_2, \dots, p_n)$  be the set of user's photos, ordered by the

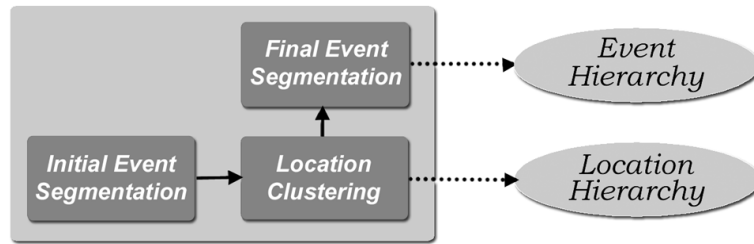


Figure 3.3: Processing steps in our automatic organization algorithm.

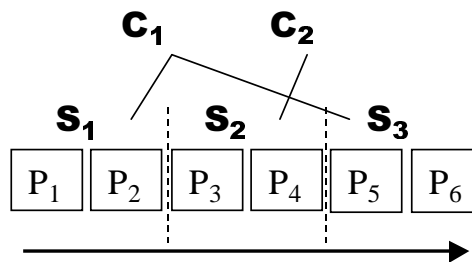


Figure 3.4: A sample sequence of photos and their segment/cluster association.

time the photos were taken. Each photo  $p$  is associated with A segment  $S(i, j)$  is a set of consecutive photos  $(p_i, p_{i+1}, \dots, p_j)$ . A *segmentation*  $M$  of  $P$  is defined by  $k + 1$  indices  $(b_1, b_2, \dots, b_k, b_{k+1})$  that divide  $P$  into  $k$  segments  $S_i \equiv S(b_i, b_{i+1})$  with  $1 = b_1 < b_2 < \dots < b_{k+1} = n$ .

A *cluster*  $C$  is defined as a set of segments that fulfill some predicate  $pred$ :  $C = \cup(S_i | pred(S_i) = true)$ . Semantically, we use clusters to represent segments (and thereby, photos) that occur in the same location ( $pred = \text{“occurs in location } X\text{”}$ ). A cluster can include photos from non-sequential segments. For example, Figure 3.4 shows a sequence of photos, and a possible sets of segments and location clusters. Cluster  $C_1$  in the figure includes photos from  $S_1$  and  $S_3$  as those segments occurred in the same geographic area.

Segments are an approximation to user events, as they divide the photo collection in time. A perfect segmentation  $M_{optimal}$  is the segmentation of  $P$  into ground-truth user events — a segmentation humans will create given their own photo collection. An *over segmentation*  $M_{over}$  of  $P$  is one where the set of indices is a superset of the indices in  $M_{optimal}$ .

The three main steps of the algorithm, as shown in Figure 3.3, can now be expressed in more accurate terms:

1. A linear pass over the sequence of photos, to generate an approximation to  $M_{over}$ .
2. A geographic clustering computation that takes  $M_{over}$  and generates a set of clusters  $C_i$ , each containing one or more segments from  $M_{over}$ ; each segment belongs to exactly one cluster. Each cluster corresponds to a distinct geographic location.
3. Another linear pass over  $P$ , merging adjacent segments from  $M_{over}$  that are likely to be related based on the results of the geographic clustering. This stage results in a final segmentation  $M$ .

Thus, our algorithm produces a segmentation  $M$  and a set of clusters  $C$ , each containing one or more segments from  $M$ . At this point, the clusters  $C$  should represent the geographic locations where the user had taken photos. The segments  $S_j \in M$  represent the different events in the user’s photo collection.

We now present the algorithm in more detail, while highlighting the rules and guidelines that are aided by the different steps.

### Step 1: Computing $M_{over}$

To approximate  $M_{over}$ , we use a variation of our segmentation algorithm presented in [34]. This algorithm is based on the “bursty” nature of photo collection, and is performed in two linear passes. On the first pass, we iterate through the photos in  $P$  — recall that they are ordered by time (rule  $i$ ). During this pass, the index  $i + 1$  is added to a segmentation  $M$  every time two consecutive photographs  $p_i, p_{i+1}$  differ by more than a specified time  $h_{over}$  (guideline  $ii$ ). Our earlier experiments have shown that the algorithm is not very sensitive to the chosen value of  $h_{over}$  when it is between 6-24 hours; we chose  $h_{over} = 12$ .

In the second pass, these initial segments are split into finer segments based on the time differences *and distance* between photographs within each initial segment

(guidelines *iii*, *iv*). The splitting is done by computing the time difference and geographical distance between all consecutive photos in the segment. We then scan this list of differences, and look for outlier values using well-known statistical methods (described in [34]). We split the segment at a point where we find a time difference outlier and a distance outlier at the same point (i.e., between the same two photos). A split between photos  $(p_j, p_{j+1})$  means adding the index  $j + 1$  to the segmentation  $M_{over}$ . The statistical thresholds are conservatively set such that the sequence of photos  $P$  is over-segmented. Although it is not guaranteed that the final result,  $M_{over}$ , will be a superset of our “perfect segmentation”  $M_{optimal}$ , it is likely to be a close approximation to such a superset. We verify that this is indeed the case in Section 3.3.1.

## Step 2: Creating the Location Clusters

Once we have created the segmentation  $M_{over}$ , the next step is to find a grouping based on location for photos in  $P$ . To this end, we use an algorithm we call *SegmentCluster*, a revision of the algorithm described in [30]. We assume the photos in each segment occur in a specific geographic location (guidelines *iv*, *ix*). The problem solved by *SegmentCluster* can be defined as follows: find an assignment of the segments in  $M_{over}$  to a set of location clusters  $C$ .

Define *source locations* as the set of centers of these clusters, denoted by  $\Gamma$ . Given a cluster  $c$ , its source location  $\gamma_c \in \Gamma$  and a segment  $S_j \in M$ , denote the likelihood that the source of  $S_j$  is  $\gamma_c$  (or, that  $S_j$  is assigned to  $c$ ) by  $Prob(S_j|\gamma_c)$ . This probability computation takes into account the location of all the photos in  $S_j$ .

The goal is to find the clusters and the segments associated with each. Mathematically, we wish to pick  $k$  source locations  $\gamma_1, \gamma_2, \dots, \gamma_k$ , and an assignment for each segment  $S_j \in M_{over}$  to a source  $\gamma_{c_j}$ , such that the total probability  $\prod_{j=1}^{|M_{over}|} Prob(S_j|\gamma_{c_j})$  is maximized (guidelines *viii*, *ix*). The algorithm executes using multiple values of  $k$  (the number of different clusters), and applies the Bayesian Information Criterion (BIC) [79] to search for the value of  $k$  that is BIC-optimal (roughly speaking, BIC is maximizing the probability while keeping  $k$ , the number of different locations, as small as possible; this should provide for guidelines *v* and *vii*).

After the execution of *SegmentCluster*, we have a flat list of geographical clusters containing the photos in  $P$ . These clusters can differ in their area size, and the number of segments and photos associated with them. The third row in Figure 3.2 is an example (annotated by textual names) for such a list.

Often, some clusters will have too many segments associated with them, like the “Around San Francisco” cluster in Figure 3.2. As mentioned above, many segments associated with a cluster are an indication that the user is familiar with the geographical area. For each such cluster, we recursively execute *SegmentCluster* on the union of segments associated with the cluster. This step results in further breakdown of the location hierarchy as demonstrated by the bottom row in Figure 3.2, and helps the system along guideline  $x$ .

An alternative to the *SegmentCluster* algorithm, which performs clustering based on groups of photos (the segments), is to simply cluster the set of photos  $P$  using some generic clustering algorithm such as k-Means. However, especially when clusters are not well separated, we find that such algorithms perform poorly, making rather arbitrary divisions of photos into clusters, and possibly violating guideline  $ix$ . On the other hand, *SegmentCluster* uses our additional knowledge of the segments (or events) in which the photos occur to make sure photos that belong together fall into the same geographical cluster (guideline  $ix$ ). The cost of this policy is, possibly, a slight overlap in geographical coverage between clusters (violating guideline  $viii$ ).

### Step 3: Towards $M_{optimal}$

While our first goal of identifying the areas where photos occur is now complete, we do not yet have the grouping of photos into events. In step 1 we created an over-estimate of the event segmentation,  $M_{over}$ . In step 3 we try to obtain an approximation to  $M_{optimal}$ , the ground-truth of event segmentation — the way a user would have split her collection.

Step 3 is a linear pass over the photos in  $P$ , where we merge some adjacent segments in  $M_{over}$  that belong to the same cluster. In other words, if  $S_i = (p_i, p_j)$ ,  $S_{i+1} = (p_{j+1}, p_k)$  and  $S_i, S_j \in c$ , we remove  $j+1$  from the segmentation  $M_{over}$ , creating one longer segment in place of two shorter ones. However, we merge the segments

only if the adjacent segments are less than  $h(c)$  hours apart, where  $h(c)$  is an inverse function of the cluster's popularity — the more segments that exist in this cluster, the smaller the value of  $h(c)$ . This follows the intuition of guideline *ii*, where less-visited clusters should be segmented more conservatively. In our implementation, we used  $h(c) = \text{Max}(H_{min}, \frac{H_{max}}{n^2})$  where  $n$  is the number of segments in that cluster within one year before or after  $S_i$ , and the values for  $H_{min}$  and  $H_{max}$  are set experimentally and are roughly a few hours and a few days, respectively (see Section 3.3.1).

At the end of step 3, we have a location cluster hierarchy, and an event segmentation  $M$ , which is hopefully a good approximation of  $M_{optimal}$ . We verify these results on sample collections in Section 3.3. In the next section we assign geographical names to the different clusters and events, so they can be presented in a user interface.

## 3.2 Naming Geographic Locations

After grouping the photos into event segments and location clusters, we need a way to present these results to users. As mentioned earlier, we are interested in investigating whether users can efficiently navigate the hierarchy without the use of a map. To facilitate this, we need to name each group of photos; i.e., give it some textual caption. We want to give a geographical name to both the location clusters and the different events, as the latter, naturally, also occur in some geographic location. An event's location is often more specific than a location cluster. For example, we may have a location cluster with photos from the San Francisco Bay Area; one of the detected events may be associated with the Bay Area cluster, but in fact had occurred in Stanford, and therefore can be described using a more precise geographical name. Practically, the names are required to be:

- i. Informative and accurate, providing users with a good idea of the location they describe.
- ii. Recognizable. Regardless of how accurate or informative a name is, it is of no use to a user that does not recognize it.

- iii. Unique. No two sets of photos should have the same name unless they represent the same location (this can only happen when we are assigning a geographical name to events).
- iv. As short as possible, to avoid clutter on the user interface, and allow users to quickly scan a list of names.
- v. Picked from a suitable level of granularity. For example, “San Francisco” maybe a better name than “California” because it provides more information, but “North Fillmore St.” may be too fine grained for most purposes.

In the following subsections, we first deal with the general problem of generating a descriptive caption for a set of geographic coordinates. We then show how these techniques can be applied in the context of the event and location hierarchies that our system creates.

### 3.2.1 Naming a set of Geographic Coordinates

This section describes our approach to finding “good name” candidates for a set of geographic coordinates.

In his 1960 book, Lynch [62] listed the components used to create a cognitive image of cities: the Node, the Path, the Edge, the District (or region) and the Landmark. For example, in the context of a city, these components could correspond to intersections, streets, boundaries, neighborhoods, and physical landmarks, although these are just examples of each component. The different types of components are used for reference and navigation around the city.

In this work we are not interested in navigation. We shall focus on two types of components of those described by Lynch, which are useful for reference: regions and landmarks. We will use regions and landmarks, albeit in a different scale than a city scale, when attempting to describe a set of geographic coordinates.

Generally speaking, regions could be cities, countries, neighborhoods, parks, and so forth. Landmarks could be, for example, buildings, monuments, or intersections.

Notice that some geographic features can be referenced as regions but also as landmarks. Especially relevant to our work is the fact that a city can be referenced both as a landmark (“near San Francisco”) and a region (“in San Francisco”).

When considering geographic regions to describe a set of coordinates, people mostly use containing or overlapping operators: “in San Francisco”, “across California and Nevada.” When discussing landmarks, people often use measures of distance like “near the Golden Gate Bridge” or “20 miles north of Auckland.” A place description can be a hybrid of both types, e.g., “in San Francisco, near the Bridge”, as landmarks are often used for navigation and reference inside the city [62].

To automatically generate a name for a set of coordinates, we use both containing regions and nearby landmarks. However, we only consider geographic features of certain types. For regions we use cities, parks, and other such large administrative regions. For landmarks we only use cities. The restricted set of features is due to both the scale of representation (the PhotoCompas location clusters often span multiple cities), the availability of data (neighborhood-level region information, as well as reliable data about low-level landmarks, is difficult to acquire) and cognitive recognition questions (people largely recognize city names, but do they recognize neighborhood names, even if they visited those?). The system described in Chapter 7 has the potential to solve the data availability problem.

Our system’s automatic naming process has three steps, indicated by the rectangles in Figure 3.5 (the data in the figure is represented by ellipses). In the first processing step, the system finds the containing features (such as cities, parks, or states) for each coordinate in the set to be named. In parallel, the system looks for good reference points such as nearby big cities, even if none of the coordinates in the set appear to be inside these cities. Finally, the naming module decides which of the containing features or nearby reference points to use when picking the final name for the given coordinate set. The name can include more than one feature of each type: “Sonoma, Boyes Hot Springs (98kms N of San Francisco)” is one example of a name created by our system. Another such name may simply be “Stanford”.

In the first step, containment mapping, we find for each latitude/longitude pair the state, city and/or park that contain it. This is done using an off-the-shelf geographic



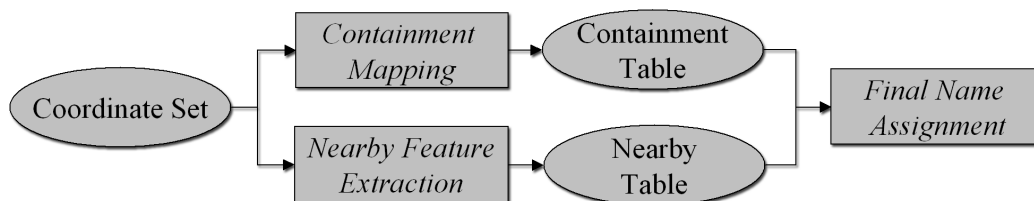


Figure 3.5: Processing steps in naming a set of coordinates.

dataset of administrative regions.<sup>3</sup> For example, a particular coordinate may be inside of California (state), San Francisco (city), and Golden Gate National Recreational Area (park). Another coordinate may be in Washington (state) and Seattle (city), but not in any park. Figure 3.6 shows a sample set of photos and containing features in the San Francisco area.

The dataset we use is based on accurate polygon-based representations for the different administrative features. The data is slightly dated (some of the data goes back to the mid-1990s), but still largely relevant.<sup>4</sup> In addition, the polygon representations can never be perfectly accurate. We estimate the error of the representations to be around a few dozens to hundreds of meters, certainly within reason for this application.

However, if no polygon (or administrative feature) of a certain type matches the queried coordinate pair, we do allow the system to look for relaxed matches by artificially expanding the borders of the polygons of that type. For example, if our dataset suggests that the (latitude, longitude) pair does not fall inside any park, but is 200m away from Yosemite National Park, the algorithm would mark the coordinate as contained in Yosemite park. While the error allowed is relatively strict when looking for a containing park, county or city, we relax the distance requirement significantly when trying to find containing *countries*, making sure we capture coordinates that are a few kilometers off-shore.

Figure 3.6 demonstrates a case where relaxation is required in a city query: there

<sup>3</sup>Regretfully, we only have access to a database of US cities and parks. Thus, we have only tested our naming procedure on US photos.

<sup>4</sup>Up-to-date datasets are available, of course, for a price.

Table 3.2: Types of administrative areas and the weights assigned by the system to instances of each.

Area Type	Weight
Cities	4
County	0
National Parks	5
National Monuments	3
State Parks	3
Other Parks	2
National Forests	0

are two coordinates in the set, seen just off the northern-most tip of San Francisco, that seem to fall just outside the city limits (in fact, they represent photos taken on the Golden Gate bridge). The system will treat these coordinates as if they are contained in San Francisco.

We count the frequency in which each city and park occur in the set of coordinates, building a term-frequency table. We weigh each type differently, with national parks weighed more heavily than cities; and cities weighed more heavily than other parks such as state parks or national forests, for example. The different weights are listed in Table 3.2 and allow us to favor names that are more likely to be recognizable to users.

At the end of this process, we have a *containment table* with terms and their score.

In the second step we look for *nearby* features, or more specifically, neighboring cities. As Figure 3.5 suggests, this step can be done in parallel to the containment mapping step. The neighboring cities can serve as reference for the given coordinate set, in case the containment features of coordinates in the set are not sufficient to represent the set well. This problem can emerge, for example, if coordinates in the set do not fall within any city or park boundaries, or occur sparsely in some area, without any critical mass inside some named locations. See for example Figure 3.7, where a small set of coordinates appears sparsely scattered within a number of cities around Stanford University. By locating cities that are close to the coordinates in this set and computing the distance from the center of the set to the city, the system is able to produce textual names for such nodes, such as “40kms south of San Francisco.”



Figure 3.6: Sample set of photos (circle markers, potentially overlapping) and a subset of the containing features of type “cities” and “parks.”

To compute the center of a coordinate set, we considered two methods. The first is simply the weighted average of the coordinates in the set. This method is satisfactory only if the set of photos is relatively coherent; e.g., represents a single cluster of points (perhaps like the one represented in Figure 3.7). However, in many cases the set of coordinates represents a number of different, yet proximate, locations. For example, Figure 3.6 shows a set of photos that occur around a few key locations in the San Francisco area. A weighted average of the coordinates will result in a center location that does not represent well any of the locations included in the set.

A different algorithm was therefore used to generate the “biased center” of a coordinate set: the coordinate in the set that is most representative of the set. The heuristic algorithm is described in Algorithm 1. In essence, the algorithm attempts to iteratively refine the mean center of the set, by retaining at every step only 50% of the coordinates that are closest to the mean, and recomputing the mean for the reduced set.

After computing the center of a coordinate set, the system finds nearby cities that can serve as reference points. For a city to serve as a good reference point for a given set of coordinates, it must fulfill two requirements:

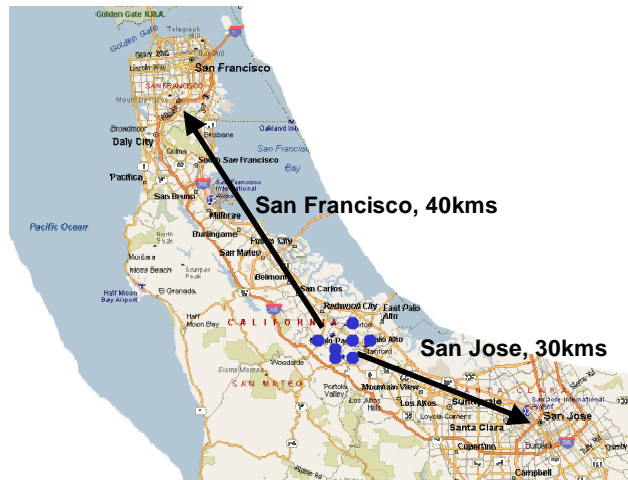


Figure 3.7: A set of coordinates and two nearby cities that may serve as reference points for the set.

---

**Algorithm 1** Computing the “biased center” of coordinate set  $S$ .

---

```

1: while  $|S| > 2$  do
2:    $c \leftarrow$  mean center of coordinates in  $S$ 
3:   Sort  $S$  by ascending distance from  $c$ 
4:    $S \leftarrow$  The first  $\frac{|S|}{2}$  coordinates in  $S$ 
5: end while
6: Return  $c$ 

```

---

- Relevance to the set of coordinates. The reference city needs to be nearby, or within reasonable distance to the set.
- Relevance to the user. The reference city needs to be recognizable to a user.

We again use our geographic dataset to find nearby, large enough cities. The system tries to locate cities that are within 100kms from the center of the coordinate set, and whose population is over 250,000. This search will fail in some cases, for example, when the coordinate set is in a remote area. In such case, we gradually expand the radius (by 25% at each step) and reduce the minimum population requirement (by 40% at each step), until at least one city is found. The rationale behind this method is that in sparse areas even smaller cities or distant large cities are good reference points.

If the search for nearby cities results in more than one city, the system tries to rank them. The ranking attempts to strike a balance between a city being relevant and known to the user, and relevant (in terms of distance) to the set of coordinates. To do that, we use the notion of “gravity”: a combination of population size, the city’s “Google count”, and (inversely) the city’s distance from the center of the set of photos. The “Google count” of a city is the number of results that are returned by Google<sup>5</sup> when the name of the city (together with the state) is used as a search term. We use this as a measure of how well known a city is, and thus, how useful it would be as a reference point. For example, a set of coordinates around the Stanford campus (see Figure 3.7) may be captioned “40kms South of San Francisco”, or “30kms North of San Jose.” The population of these cities is comparable, but since the Google count for San Francisco is much higher than San Jose’s, the former is chosen despite being further away.

To give an example of the application of gravity in our system, refer to the set of coordinates near the Stanford campus, as depicted in Figure 3.7. The center of the set is computed as described above to be somewhere around Stanford campus. A query on the dataset reveals two major nearby cities: San Jose and San Francisco. The nearby reference points can be either “40kms South of San Francisco”, or “30kms North of San Jose.” The population of these cities is comparable: 945,000 residents in San Jose, vs. 800,000 in San Francisco. The distance of both cities from the set of coordinates is also comparable. However, the Google count for San Francisco (search for ‘San Francisco, California’ results in 44,800,000 hits) is much higher than San Jose’s (search for ‘San Jose, California’ results in 13,300,000 hits). San Francisco therefore is ranked higher despite being further away:  $Score(\text{San Francisco}) = \frac{800 \cdot 44.8}{\sqrt{40}} = 5666 > Score(\text{San Jose}) = \frac{945 \cdot 13.3}{\sqrt{30}} = 2294$ .

We experimented with various ways to combine the different metrics to a single score. Specifically, assigning more weight to the distance (i.e., giving the distance an inverse linear weight, or even a quadratic weight) results in inferior results in practice, overestimating the importance of nearby cities.

After this second step, we have a *nearby-cities table*, again with terms and their

---

<sup>5</sup><http://www.google.com>

scores.

The final step, as depicted in Figure 3.5, involves picking 1–3 terms from the nearby-cities and containment tables to appear in the final name of each set of coordinates. For example, a possible caption can include the two top terms from the containment table, and the top nearby city: “Stanford, Butano State Park, 40kms S of San Francisco, CA.” Our method of picking the final terms varies according to the semantics of the set of photos we are trying to name, as we explain in the next subsection.

We have also experimented with the Alexandria Digital Library’s gazetteer [38]. There are two possible ways to utilize Alexandria for our purpose. Firstly, we could use it to map from coordinates to a containing feature, corresponding to our first step described above. However, Alexandria represents geographic features by a rectangular bounding box, which is not accurate enough for our needs. For example, querying with a coordinate in San Francisco returns Canada as one of the containing areas. Our aforementioned dataset is less encompassing but uses much more accurate polygon representations of geographic features. Secondly, we could use Alexandria to map *from* an area (bounding rectangle representing a set of coordinates we wish to name) to a list of contained features like cities, parks and even rivers, waterfalls and mountains. Then we could have used this list to find prominent features that could be used to name the given set of coordinates. In this case, we found it hard to formulate a query that will supply us with a good enough list of contained features — the list was either too noisy, or too sparse. In addition, we did not find a good way to distinguish meaningful features in this set that can be used in a name. Having said all that, we are still hoping to find a way to use the vast information space that is the Alexandria gazetteer in future work.

### 3.2.2 Naming Location Clusters and Events

Picking the final terms from the containment and nearby-cities tables to appear in the caption is done in different ways depending on whether we are naming an event from the event hierarchy; a leaf cluster in the location hierarchy; or an inner node

cluster in the location hierarchy.

Due to the semantics of the final name and what it represents in our PhotoCompass photo browser, the final names must follow two sometimes conflicting guidelines:

- The final name must represent well the different locations and areas covered by the coordinates set.
- The final name must not be too long; users should be able to scan it quickly to get an impression of the area and locations it represents.

Naming a leaf cluster in the location hierarchy, such as the Berkeley cluster in Figure 3.2, is the hardest and most important of the three different types. It must be the most accurate description since there is no lower level where more location details are exposed. In addition, the location may have been visited a number of times, and may represent a heterogeneous set of events occurring in slightly different locations. The rules for naming a leaf cluster are as follows: 1. Use the top term from the containment table, if one exists. 2. Concatenate the second top term from the table only if it has a significant score in this set (e.g., if this term appeared only twice, we do not want to use it). 3. If the number of segments for this cluster is low (suggesting that the user is not very familiar with this area), or if the scores for the top two terms are low, concatenate the top term from the nearby cities table to the name. At the end of this process, we may have anything from 1 to 3 terms that are combined into a textual geographic description for this cluster.

As for the inner nodes in the location hierarchy, some of them represent a country, and are easy to caption. For the other inner nodes, such as the “Around San Francisco” cluster in Figure 3.2, we pick the single top-scoring term from the containment table of each of the node’s descendants, and take the top three terms in this list. The hope is that these three names represent the general area where the current cluster occurs; for a more accurate description of the area, one can turn to the lower level clusters. For example, our “Around San Francisco” cluster may be named “San Francisco, Berkeley, Sonoma.” However, if this cluster is the only one in this specific state, it will be assigned the state’s name.

Finally, when assigning a textual geographic caption to an event, we assume that the user has visited relatively few places over the duration of the event. Furthermore, we assume the user will get more geographical context from the cluster information. For event names, then, we pick only the top term in the containment table. If one does not exist, we choose the top term from the nearby cities table. In any case, we can augment the name with the date and time span of the event, e.g., “Boston, Dec 31<sup>st</sup> 2003 (3 Hours).”

### 3.3 Experiments and Results

Our system produces three different types of output: event segmentation, location hierarchy, and suggested names. Evaluation of each of these outputs poses its own challenges. However, the main challenge in evaluating our system is the current rarity of geo-referenced photo collections. We expect more collections to be available in the future, but today, other than the author of this thesis, we could only find two subjects with a large enough collection of such photos. Our results are then, by necessity, case studies rather than statistically significant analysis.

All three collections were geo-referenced using a separate GPS and camera, in a manner described in Appendix A.

Details of the two collections,  $R$  and  $K$ , together with the author’s collection  $Z$  are listed in Table 3.3. As the evaluation of  $Z$  may be biased by the author’s knowledge, we do not show results for it. However, we do use the results of the performance evaluation for collection  $Z$  to reaffirm the results for the other collections. As our processing is done separately for each country, we only used United States photos from these collections, since both subjects’ collections had too few pictures from other countries to merit evaluation.

Table 3.3: Sample datasets used in our experiments.

<i>Collection</i>	<i>Number of US Photos</i>	<i>Time Span</i>
$R$	2580	27 months
$K$	1192	14 months
$Z$	1823	13 months





Figure 3.8: A Map of the first-level clusters for collection  $Z$ . Two clusters are not shown: “Seattle, WA” and “Philadelphia, PA.” The map is used for illustration.

-San Francisco, Berkeley, Sonoma, CA	876
•Berkeley; Oakland	188
•Glen Ellen; Eldridge (61 miles N of San Francisco)	22
•Petaluma (57 miles NW of San Francisco)	3
•San Francisco; Golden Gate N.R.A	637
•Sonoma; Boyes Hot Springs (52 miles N of San Francisco)	26
-Stanford, Mountain View, Monterey, CA	284
•Monterey (58 miles S of San Jose)	12
•Mountain View (4 miles NW of San Jose)	29
•Stanford	243
-Colorado (219 miles W of Denver)	180
-Long Beach (25 miles S of Los Angeles, CA)	90
-Philadelphia, PA	8
-Seattle, WA	39
-Sequoia N.P. (153 miles E of Fresno, CA)	133
-South lake Tahoe; Bear Valley, CA	96
-Yosemite N.P.; Yosemite Valley, CA	116

Figure 3.9: All nodes in the location hierarchy of Test Collection  $Z$  and the number of photos in each node.

For illustrative purposes, Figures 3.8 and 3.9 show some sample results from PhotoCompass processing of the test collection  $Z$ . Figure 3.8 illustrates the geographical distribution of the clusters created by PhotoCompass for the collection. The figure only shows a subset of the first-level United States clusters for collection  $Z$ , and does not present the sub-clusters that were created for San Francisco and Stanford clusters.

Figure 3.9 shows a textual list of *all* the United States location nodes created by PhotoCompass for collection  $Z$ . The figure also shows the names as assigned to the nodes by the system. In addition, the figure lists the number of photos that belong to each node.

In this chapter, we perform direct evaluation of the system’s output. In other

words, we asked the human subjects (the collection owners) to independently generate equivalent output, and compared it to the system’s output according to various metrics. Alternatively, we asked the subjects to comment on the output generated by our system, and quantified the comments in ways described below.

An alternative approach for evaluation, task-based evaluation, is used to test our system in Chapters 4 and 5. In task-based evaluation, the subjects use the system to perform some task. The assumption is that if the system’s output and interaction are of high quality, it will lead to better performance. In particular, Chapter 4 demonstrates that PhotoCompass performs as well as a map-based browsing application in photos search and browse tasks.

We decided to start here with direct evaluation of the system for two main reasons. First, as explained earlier, few geo-referenced collections are available for testing. A task-based evaluation requires an experiment of a larger scale, and would have involved obtaining and testing with a larger number of collections. Such an effort would have been justified if our initial investigation, using the direct evaluation approach, proves that the system has merit. Second, the direct evaluation would help us study the values of system parameters and their effect on performance. Designing a task-based evaluation from which we can learn directly about the optimal values of system parameters is not trivial, if at all feasible.

### 3.3.1 Evaluation of Event Segmentation

In this section, we evaluate the success of our event segmentation. For this purpose, we asked the owners of our datasets for the “ground truth” segmentation of their collection,  $M_{optimal}$ , as it is defined in Section 3.1.2. As expected,  $M_{optimal}$  demonstrated the difficulty inherent in the task of event segmentation and validated some of our guidelines. On one hand, the subjects often listed multiple picture-taking days as one event (“This was my trip to New York”). On the other hand, subjects often partitioned photos taken in a single day into multiple events: a birthday event closely followed by an unrelated dinner party, for example. Our evaluation goals were as follows:

1. Show that the event segmentation generated by our algorithm is accurate, and is an improvement over time-only techniques.
2. Explore the effect of system parameters on the event segmentation.
3. Verify that  $M_{over}$  is indeed an over-segmentation of the ground-truth events (see Section 3.1.2).

The metrics used in past studies of event clustering for digital photos (e.g., [17, 59]) are the precision and recall for the detected event boundaries:

$$precision = \frac{\text{correctly detected boundaries}}{\text{total number of detected boundaries}} \quad (3.1)$$

$$recall = \frac{\text{correctly detected boundaries}}{\text{total number of ground truth boundaries}} \quad (3.2)$$

The recall and precision metrics are also used in other contexts such as video segmentation and natural language processing (NLP). While we feel these measures are relevant, we also feel they are lacking in capturing the complete semantics of event segmentation. For example, consider a collection of 10 photos and its ground truth segmentation  $M_{optimal} = 1, 6, 10$ . In other words, the collection has two event segments, photos 1–5 and 6–10. Now consider the suggested segmentations 1, 2, 5, 10 and 1, 2, 9, 10. While it is clear that the former is better than the latter, they are scored the same in both recall ( $\frac{2}{3} = .667$ ) and precision ( $\frac{2}{4} = .5$ ).

While designed for evaluating the NLP document-segmentation problem, we adopt two additional metrics to overcome the limitation of recall and precision. In [9], Beeferman et al. suggest the  $P_k$  metric. This metric uses a sliding window to compute a score that is based on the probability that two photos (in our context) that are  $k$  photos apart are incorrectly identified as belonging to the same event, or not belonging to the same event. Pevzner and Hearst in [68] discuss shortfalls of the  $P_k$  metric, and suggest a variation named *WindowDiff* (WD). The WD metric computes an error using a sliding window over the segmented set. At every position WD counts the number of segment boundaries that fall within the window. If the number is different between the ground-truth and the suggested segmentation, the algorithm assigns a

penalty proportional to the difference. The authors suggest that the WD metric grows in roughly linear fashion with the difference between the compared segmentations. The possible values for  $P_k$  and WD range from 0 to 1; for both metrics, lower values are better. We use the  $P_k$  and WD metrics in our evaluation, and propose that these metrics be used in future evaluations of event detection systems.

In Section 3.1.2 we presented the  $h(c)$  function that comes into effect when consecutive events occur within the same geographical cluster. This is the only hand-tuned function that directly reflects on the event segmentation. Therefore, we had to verify the effect its parameters,  $H_{min}$  and  $H_{max}$ , have on the resulting segmentation. We omit this discussion for lack of space but note that, at least for the three test collections, the algorithm seemed insensitive to values that ranged from 1 to 8 hours for  $H_{min}$ , and 100 to 200 hours for  $H_{max}$ . We also confirmed that the chosen  $h(c)$  function performed considerably better than a simple policy of using a fixed time threshold (e.g., 6 hours) to detect consecutive events in the same geographical cluster.

We tested the performance of our event segmentation. The following conditions are compared:

- PC — Our PhotoCompas algorithm as proposed in Section 3.1. PC(8,192) and PC(1,192) represent sample pairs of  $h(c)$  parameters.
- TB — Time-based segmentation algorithm from [34].
- FT — A “fixed” threshold segmentation: we detect a new event every time there is a gap of  $x$  hours.
- FS — Another “fixed” segmentation where we detect a new event if there is a gap of  $x$  hours or  $y$  kilometers.

The parameters for conditions TB, FT and FS were hand tuned to yield a minimal average *WindowDiff* for the two collections. The values in brackets in the following figures are the chosen parameters in hours (or hours and kilometers for FS). Note that the chosen parameter values are set-dependent, and may be different for other collections. However, we wish to show here that even when choosing optimal parameters, our system still performs better than TB, FT and FS.

The recall and precision metrics for the different conditions are presented in Figure 3.10. The rows correspond to the different conditions as listed above. The four bars at each row represent the recall and precision values for each of the two collections  $K$  and  $R$ .

Before we discuss the results for the different strategies, let us look at the recall and precision values for  $M_{over}$ , on the top row of the figure. Remember that, as presented in Section 3.1.2,  $M_{over}$  should be an over-segmentation of the collections (the recall should be 100%). In reality, for both the  $R$  and  $K$  collections the recall is close to 85%, due to a total over both collections of 24 undetected events. When checking the undetected events, we discovered that all of them, except one, contained only a few photos (2.8 on average) and happened on the same day and in the same area as the adjacent event. The only longer missed event started literally minutes away from the end of another event. We conclude that this low recall value for  $M_{over}$  is inevitable, and suggest that these two test collections are inherently hard to segment (the  $M_{over}$  recall for  $Z$  was 97%). While it is possible to tune the parameters in order to get a higher recall for  $M_{over}$ , this will cost significantly in the precision. Since  $M_{over}$  informs our location clustering, we must not be too aggressive in computing it.

We now compare the performance of different strategies as they appear in Figure 3.10. The two PhotoCompas conditions are represented in the bottom two rows in the figure. One can easily see that PhotoCompas balances recall and precision better than all other conditions. We observe that PhotoCompas does better in recall, precision, or both, than all other algorithms with 80%-85% for both metrics and both collections, in both parameter settings.<sup>6</sup>

We manually inspected the events that were undetected (recall) and over-segmented (precision) by the PhotoCompas conditions for both collection. Most of the undetected events, similarly to  $M_{over}$ , were minor events (few pictures) around the subjects' hometowns, within hours or less from other events. One notable exception is a road trip that our algorithm decided not to split, but which the human subject actually preferred to split. On the other hand, when we checked the type of precision errors made by the algorithm, a lot of them involved road trips: a multi-day picture

---

<sup>6</sup>The recall for  $Z$  under the same settings was higher at 90%, while the precision was lower, 76%.

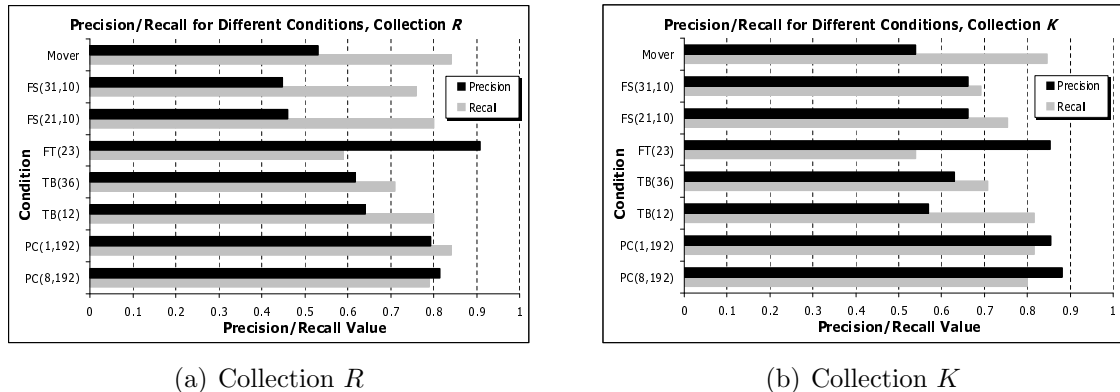


Figure 3.10: Recall and Precision values for different conditions. For both metrics, higher value means better performance.

taking events, that span a very large geographical area. Our algorithm broke these trips apart when the time gap and distance between two consecutive pictures were very high (an overnight gap in picture taking, for example). Another type of events that were over-segmented was events that included photos taken around the subject’s hometown, when two consecutive photos were many hours apart but still “belonged together” as far as the subject was concerned. Notice that automatically distinguishing between these and the undetected events mentioned earlier is impossible.

In Figure 3.11 we show the  $P_k$  and  $WindowDiff$  metrics for the different conditions. We can see that PhotoCompas (two bottom rows, in two different parameter settings) performs better (lower values) than all the other algorithms. Also notice that these metrics bring out the differences in a much more apparent way than precision and recall; compare the results of strategy FT versus PC in Figure 3.11 and Figure 3.10. For collection  $Z$ , the metric values were similar to results for  $K$  and  $R$  —  $P_k = 0.1$ ,  $WindowDiff = 0.2$  for both PhotoCompas conditions.

In Figures 3.12 and 3.13 we show the effect of PhotoCompas algorithm parameters on the different metrics. In Section 3.1.2 we presented the  $h(c)$  function, that comes into effect when consecutive events occur within the same geographical cluster. This is the only hand tuned function that directly reflects on the event segmentation. Therefore, we chose to verify the effect its parameters,  $H_{min}$  and  $H_{max}$ , have on the

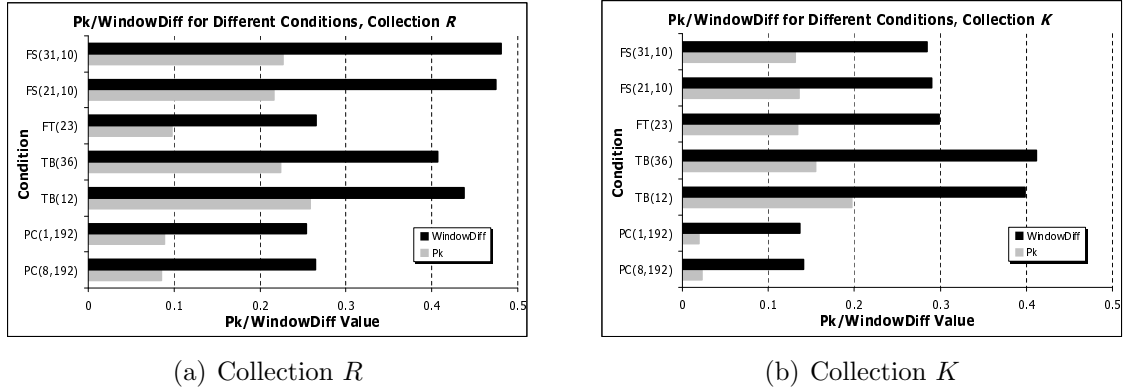


Figure 3.11:  $P_k$  and  $WindowDiff$  values for different conditions. For both metrics, lower value means better performance.

resulting segmentation.

Figure 3.12 shows the recall and precision metrics for our PhotoCompas algorithm as we vary the values for  $(H_{min}, H_{max})$ . To compare, two data points are added where we used a fixed-threshold  $h(c)$  function, splitting consecutive segments in the same cluster if the gap between them exceeds a fixed number of hours. The rows represent different parameter settings. For each parameter setting, we show the precision (the top two bars) and recall (bottom two bars) values for collections  $R$  and  $K$ . Higher values mean better performance. For example, when  $H_{min} = 1$  and  $H_{max} = 100$ , the recall for collection  $K$  exceeds 0.9 (the longest bar in Figure 3.12), and the recall for collection  $R$  is 0.89; the precision for both collection in this setting is 0.74 and 0.75, respectively.

We can see from Figure 3.12 that there are only minor differences between most of the different parameters settings of the dynamic  $h(c)$ . In addition, we see that the dynamic  $h(c)$  function is better than the fixed  $h(c)$  for either recall, precision, or both. Results for collection  $Z$  were similar.

Figure 3.13 shows the  $P_k$  and  $WindowDiff$  metrics for the same different parameter settings of PhotoCompas. The different rows represent the  $(H_{min}, H_{max})$  pairs, or the fixed-time strategy. Again, we can see that while the algorithm is not very sensitive to the value of  $(H_{min}, H_{max})$ , in all cases the dynamic strategy performs better

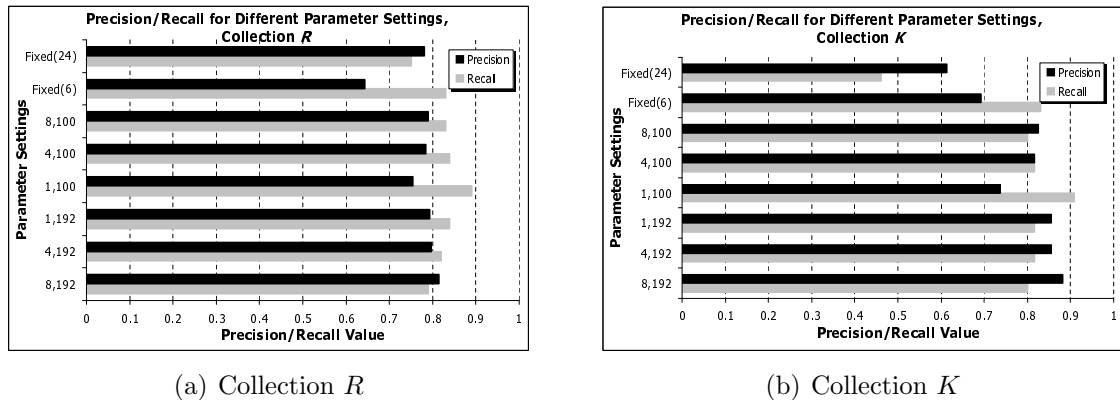


Figure 3.12: Recall and Precision values for different parameter settings. For both metrics, higher value means better performance.

than the fixed  $h(c)$  function. Results for  $Z$  (not shown) reaffirmed this conclusion.

### 3.3.2 Evaluation of Location Hierarchy

We qualitatively evaluated the location hierarchy created by PhotoCompas through interviews with the owners of the  $K$  and  $R$  collections. In both cases, many of the photos were centered around a single metropolitan area, but many others were taken during trips to various other places. Our goal was to check whether the clusters in the hierarchy:

- Are accepted by our subjects.
- Are similar to geographic grouping subjects would have generated themselves.
- Contain erroneous assignments (photos assigned to the wrong cluster).
- Are preferable to a “fixed hierarchy” of state/city.

The PhotoCompas clustering created 5 and 6 initial clusters for the two collections. In both collections, the algorithm selected to split one of these clusters into lower level clusters: 5 in one case, and 10 in the other. We showed the subjects maps of the clusters, drawing the locations of all photos in each cluster. We solicited comments



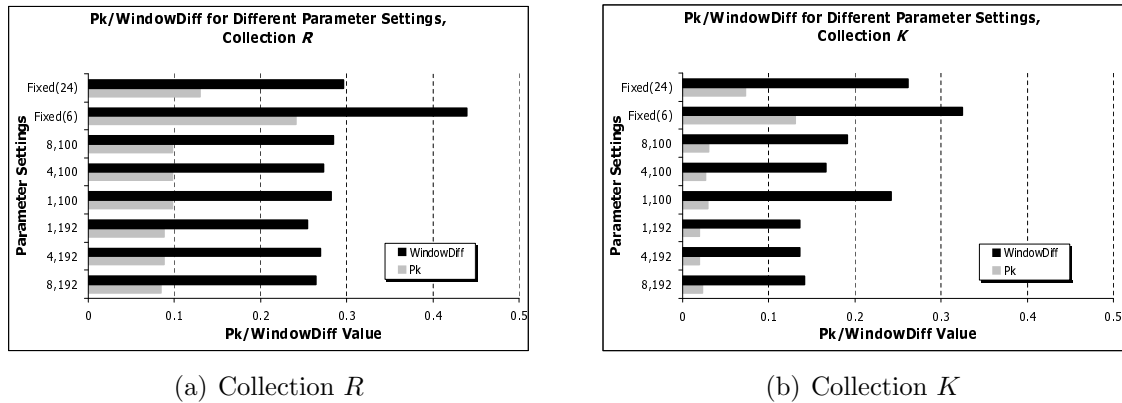


Figure 3.13:  $P_k$  and  $WindowDiff$  values for different parameter settings. For both metrics, lower value means better performance.

about the clusters and photos. We also asked the subjects to suggest edits to the clusters at each level: whether they would like to split or merge any of the displayed clusters.

The clustering results for the high-level clusters were accepted by the subjects, though both had minor edits for these clusters. One subject commented that he would have liked the cluster created for his Northern California / Southern Idaho photos be split into two clusters, by the state. At the same time, though, he liked the fact that photos from Nevada and Utah are grouped together, and that his Central California and Southern California photos were split into two clusters. The other subject did not approve of the algorithm’s decision to keep all his East Coast photos in one cluster. Similar comments were made about the lower level clusters. In summary, the required edits to the clusters were limited. We believe that our first two evaluation goals are met quite successfully.

Checking the third goal, a small issue that arose for both subjects is the occasional overlap of two clusters. This happens, as noted in Section 3.1.2, since our algorithm prefers to keep “related photos” in the same cluster, even at the price of a slight geographic overlap. For example, a road trip that begins at home but the bulk of the photos are taken elsewhere. In such a case, all the trip photos will be assigned to one cluster, possibly creating an overlap with the “home” cluster. The overlap

involves only a small number of photos — otherwise, our algorithm would merge the two clusters. Remember that our clustering has two goals: first, to inform the event segmentation; and second, to be used in the UI presentation to the user. It seems that the UI presentation may suffer slightly from this overlap. This question should be studied in a more extensive user study. Possibly, the clusters should be fixed *after* the event segmentation so photos are re-assigned with no overlap. Alternatively, individual photos can be assigned, after the final segmentation, to more than one location cluster.

Finally, we showed the subjects the breakdown of their collection into a strict administrative hierarchy of states and cities /parks. Clearly, in some cases this hierarchy was useful to the subjects (particularly for areas they know well), while in other cases it was confusing (photos that do not fall into any city/park, or when photos were taken in areas that the subjects are not familiar with). The tradeoffs in using the administrative hierarchy in the UI versus our automatic hierarchy are investigated in a user study in the next chapter, Section 4.5, but still without conclusive results.

### 3.3.3 Evaluation of Naming

We evaluated the PhotoCompas naming algorithm through interviews with the owners of the test collections. We concentrated on names for the geographical clusters and did not look at the geographic names we produced for events. Our evaluation goals were to verify that the produced cluster names are:

- Useful to the subjects, in that a) the name includes terms that are familiar to the subjects and help them understand which geographic area is covered by the cluster, b) the subject is able to tell the cluster apart from other clusters, based on the name and c) the subject can tell which pictures belong to this cluster based on the name.
- Similar to the names that the subjects would have generated themselves.

For each collection, and each cluster, we performed two tests. In the first test, we showed the subjects the list of terms that appeared in the *contained table* and *nearby*



Angels City  
 Arnold  
 Bootjack  
 Calaveras  
 California  
 Fresno  
 Livingston  
 Mariposa  
 Mc Connell State Park  
 Merced  
 Modesto  
 Planada  
 Sacramento  
 San Joaquin  
 San Jose  
 Stanislaus  
 Stanislaus NF  
 Tracy  
 Yosemite NP  
 Yosemite Valley

Figure 3.14: Candidate term set for the Yosemite node of collection  $Z$ .

*cities table* for each cluster as defined in Section 3.2.1. For example, if a certain cluster contains photos from Redwood National Park, the city of Eugene, Oregon and Crescent City, California then our list included all those city, park and state names, plus the appropriate counties. The average length of the lists for the different clusters was 19 place names; it was generally shorter for leaf clusters as they contain fewer photos. Figure 3.14 shows a set of candidate location terms generated for one of the leaf nodes in collection  $Z$ .

For each cluster, we asked the subject to pick at most three place names that represent this cluster best (in fact, they only picked 1.84 place names on average; our algorithm used an average of 2.3 place names for each cluster). As an aid, we showed the subjects maps for all the clusters and offered to show them the pictures as well — that was usually not necessary as subjects had a very good idea what those pictures were. For 76% of the clusters, our algorithm and the subjects picked at least one place name in common. Furthermore, our next test shows that for most of the other 24% of the clusters, the subjects found the given name useful.

In the second test, we asked the subjects to comment on the usefulness of each cluster name. We found that the automatically-produced names were extremely useful (as defined above). Out of the total of 25 clusters, subjects were content with all but one name. In this one case, our clustering algorithm grouped together all photos

from three different US East Coast cities; but the name only represented one of them. Other comments were: a) Three of the cluster names included one park or city name that was unknown to the subject; b) one cluster name was not representative enough of a small subset of its photos. In general, both subjects expressed strong satisfaction with the usefulness of the names.

### 3.4 Related Work

The related work described in this section includes other approaches to automatic organization of photo collections, and work that is related to our efforts to assign names to sets of photos (or geographic coordinates). Additional relevant research on collections of photos is listed in Chapter 2.

Automatically detecting events in a photo collection is an interesting problem that has been well studied in recent years [17, 26, 27, 34, 59, 60, 70, 71, 87]. All these event-detection techniques are based on the times the photos were taken, and some also augment the time data with information about the visual similarities of the photographs. For example, [71] uses both time-based similarity and visual similarity of photos to group photos into meaningful events. To give another example, Gargi shows in [27] that consumer media capture habits can be described along the time dimension as a fractal process. Gargi also devised an algorithm to split photos into meaningful events, utilizing the fractal model properties. Earlier work in our project by Graham et al. [34] used statistical methods to detect outliers in the duration of gaps between photos in a collection, thus identifying points in the sequence where a new event is likely to begin. This latter work was used in the early steps of the PhotoCompass computation, as listed in Section 3.1.

While some of the systems (e.g., [17]) propose that their algorithms can be extended to support location information, only one study was published that utilizes the geographical information for event detection and clustering. This study by Pigeau and Gelgon [69] employs statistical clustering methods to cluster photos in the time dimension and in the space dimension. The clusters are used for collection summary and event-based presentation, and are not used for location-based browsing. The

authors have tested their system on collections that represent a single picture-taking trip.

There are a number of areas of related work for the task of naming sets of geographic coordinates. Several online databases are available that provide valuable sources of geographic data. In our work, we used an off-the-shelf geographic dataset as a source of named location entities, but other sources of relevant information are available. The Alexandria Project is a distributed digital library for materials that are referenced in geographic terms (“gazetteer”) [38, 84]. The resultant database provides a rich resource for named geographic entities. As explained later in this paper, we were unable to successfully make use of this resource, and incorporation of the Alexandria database remains a possible direction of future work. Other publicly available gazetteers include the Geographic Names Information System, the Columbia Gazetteer of the World, and the US Gazetteer operated by the US census bureau.<sup>7</sup> In addition, Microsoft’s MapPoint Web Service<sup>8</sup> provides a programmable interface that supports geographic queries for businesses and other points of interests.

The Geo-SPIRIT project [42] is looking at ways to use the web to describe “imprecise regions” such as Northern California or Mid West that could prove useful in naming a set of coordinates [4]. In [53], Larson and Frontiera proposed and evaluated area-matching algorithms which could provide the matching needed between areas defined by our sets of coordinates to be names, and precise or imprecise geographic regions that may overlap them. It may be possible to use these techniques to better inform our system about the suitability of generated names.

A number of research efforts have been trying to utilize and enhance the web using geographic concepts. The Web-a-Where system [3] disambiguates geographic terms that appear on a web page, and tries to determine the geographic focus of that page. Other projects [21, 81] try to use the textual content as well as the geographic distribution of hyperlinks to a web resource to assess the resource’s geographic scope. Again, systems like these could augment and enhance the naming techniques described in this chapter. For example, once a name was selected for a set of photos, the system

---

<sup>7</sup><http://geonames.usgs.gov/>, <http://www.columbiagazetteer.org/>, <http://www.census.gov/cgi-bin/gazetteer/>

<sup>8</sup><http://mappoint.msn.com/>

can look for web pages containing the name, and verifying that those point to a similar geographic scope.

### 3.5 Conclusions and Future Work

We have shown that PhotoCompas can automatically generate a meaningful organization for personal photo collections. In particular, the system performed well when detecting events in collections; generated location hierarchies that were intuitive to collection owners; and assigned node names that proved useful.

We have built a prototype interface that will support PhotoCompas for desktop and PDA environments. This interface is generated using the HTML-based Flamenco metadata search interface by Yee et al [97]. The next chapter examines the UI presentation aspects of PhotoCompas. For example, is a UI based on our generated location hierarchy more effective than one based on a pre-defined state/city location hierarchy? We also examine the tradeoffs between our approach and a map-based interface approach for geo-referenced photos.

One of the key problems is the lack of multi-year geo-referenced photo collections to experiment with. Once we have obtained more collections, possibly via manual geo-referencing of existing photo collections, we plan to verify that our techniques are also effective for collections that span 20–30 years of photos. In addition, we would like to compare our approach to alternative implementations.

Finally, a more general problem may arise from our naming algorithm. Occasionally, when naming clusters, subjects suggested a cluster name that would be difficult to find on the basis of our data, e.g., “Northern California” or “East Coast.” We will try to address this challenge in our future work; Chapter 7 provides an possible initial direction.

## Chapter 4

# Interacting with Photo Collections

In this chapter we set out to determine experimentally how well two very different applications for browsing personal photos enable end-users to utilize the location and time metadata in concert. Both systems build on the location *as well* as the time metadata of the photo collection. Like the previous chapter, we assume that both the creation place and time of each photo are known for each photo in the collection. Rather than focusing on the computational opportunities that follow from such availability, as we did in Chapter 3, we now ask about the characteristics of user interfaces that take advantage of such per-photo metadata.

Maps are an obvious way to communicate and manipulate location information. A user interface technique that comes to mind is to display photos or their place holders on a geographic map. This approach offers a number of strong advantages. Many users are familiar with maps and can interact with them. Map-based approaches manifest the geographic metadata directly in a manner that exploits users' experiences in their world outside computation. The notions of panning and zooming can presumably be learned quickly.

We hypothesize that maps are indeed a powerful tool and will be useful for many users. However, maps may be impractical in some important situations. First, maps are inefficient in their use of screen real estate. For example, the map based overview of a photo collection that comprises photos from just San Francisco and Paris would occupy much of the screen with (visual) geographic information that is not pertinent

to the collection. The problem intensifies when the user operates on a small-screen device. Such devices seem well positioned to replace the traditional accordion display of photographs that stuffs many travelers' wallets. Yet, maps are not likely to be well suited for this environment. Textual browsers are more likely capable of screen real-estate parsimony. A second shortfall of a map interface is exposed when operating on devices with limited input mechanisms (such as cell phone inputs or voice activation, for example) – such devices may not be well suited to map-based manipulations.

Another important factor to consider is that maps may not be a natural interaction medium for everyone. In particular, maps produce a variety of comfort levels among users [19]. For many users, maps are not easily comprehensible [85]. Some users may exhibit disorientation, feel lost, or become upset.<sup>1</sup>

To explore alternatives to maps, we developed an interface that offers textual navigation of a geographic hierarchy. The browser is based on the PhotoCompas system we describe in Chapter 3. A sample screen shot of the system is shown in Figure 4.1. As a non-map interface, PhotoCompas must provide intuitive structure and names for the different locations as handles for the user to manipulate. Even with a good organization and recognizable geographic names, it is hard for such an interface to support the same level of detail as a map. However, we hypothesize, this textual navigation scheme can serve as a reasonable alternative to the map for our application. Especially, such interface is a viable alternative for users who are not comfortable with maps when they explore geographic relationships between collection items.

We report here on an experiment that compared PhotoCompas' textual approach to browsing and searching geo- and time-referenced photos, with the WWMX application [90] — a mature, effective map-based system that also includes the time dimension in its interface. A sample screen shot of WWMX is shown in Figure 4.2. We chose to compare the two interface methods on a large display because we wished to focus on the cognitive aspects of the approaches, rather than disadvantage the map approach a priori with limited screen real estate, for which current map-based

---

<sup>1</sup>Anecdotally we observe that the book “Geography for Dummies” devotes 11% of its contents to explaining maps. This title in the popular book series leads “Motivating Employees for Dummies” and “Communicating Effectively for Dummies” in Amazon sales rank.



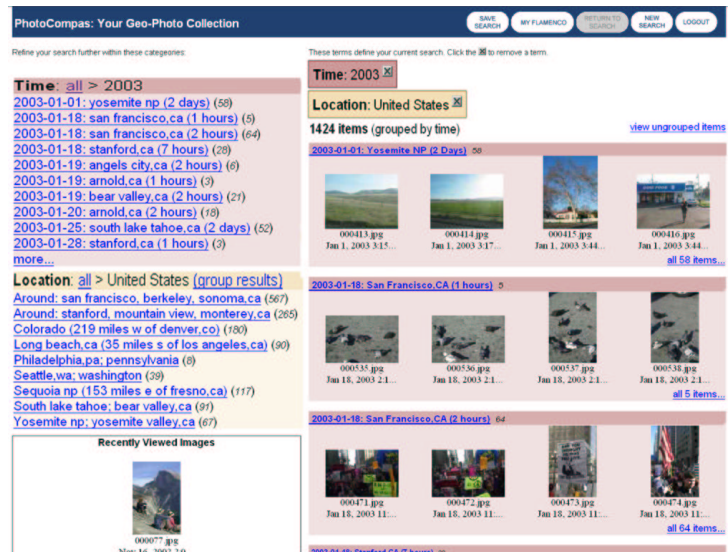


Figure 4.1: Screen shot of PhotoCompas. This view shows the collection restricted to photos taken in the United States during 2003. The events under 2003, and locations under United States are shown.

systems are not designed.

Among the contributions of this work are a methodical assessment of map vs. textual browsing in the context of personal collections of photographs. In addition, the chapter offers an extensive user study, including lessons learned and feedback that pertains to developers of location- and time-based browsers of any type.

Our expectation from the user study was that:

1. users would prefer the visually rich map interface, and that
2. users' browse and search task speed performance in the map interface would exceed their performance with the textual interface.

We thus hypothesized that browser designs whose usage constraints exclude visual maps would incur a significant, if not intolerable performance penalty. In reality, the experiment confirmed hypothesis 1, but we were surprised when we investigated hypothesis 2.

In the subsequent section we provide more details about the two experimental browsers we used in our experiments, including their interface and interaction styles.



Figure 4.2: Screen shot of WWMX. The view is restricted to photos taken in the United States during 2003.

We then describe the setup of our controlled experiment, the resulting data, and our thoughts about the results.

## 4.1 Two Experimental Browsers

As the PhotoCompas system was developed in our project, we describe the interface for PhotoCompas in more detail. We briefly describe the WWMX application as well so the reader will be presented with the full picture regarding our two experimental systems.

### 4.1.1 PhotoCompas

The first browser we studied is based on PhotoCompas (“PhC”), the system we developed and is described in Chapter 3. PhC’s interface is based on automatically generated hierarchical time and location categorization of the photos. The only graphic elements of the interface are the photos themselves. Figure 4.1 shows

one of PhotoCompas' screens. Note that this experimental interface implementation is not optimized to be maximally space efficient or aesthetically pleasing. It is purely HTML-based and inherits HTML's visual and interaction limitations. Our goal, again, was to measure users' understanding and performance of this text-reliant interaction model. The interface was constructed from the *Flamenco* toolkit [97]. The authors of Flamenco kindly made their implementation of a general metadata browser available to us. The Flamenco framework is designed to make use of hierarchical faceted metadata, making it a good complement to our system.

In Figure 4.1, the Time and Location categories are represented on left side of the screen (following [97]'s terminology, categories are called *Metadata Facets*). That portion of the display is subdivided into a Time pane and a Location pane. The header of the Time facet pane orients us in the temporal dimension. The header that is visible in the figure, *Time: All → 2003*, indicates that we arrived at this pane by clicking on *2003* in a previous version of the pane, which had shown all time ranges in years. We could drill down into finer time granularity by clicking on any of the entries in the pane.

Analogously, the Location pane below is organized by where photos in the collection were shot, and its header indicates that we arrived at this pane by clicking on "United States." The first entry, for example, offers us all the photos of the collection that were taken *Around San Francisco, Berkeley, and Sonoma in California*.

A different representation of the categories' subsets is shown on the right side of the screen — a simple, scrollable grid of photo thumbnails. Above the grid we are reminded that the displayed photos were all shot in 2003 in the United States, and shown that the collection contains 1424 images in this time and location. Notice that the thumbnail grid is further subdivided into horizontal regions, each containing four photographs and a header, which is a link, corresponding (in this case) to the Time facet breakdown on the left. The topmost header says "*2003-01-01: Yosemite NP (2 days)*". The four photographs are taken from the collection's subset that is defined by that header. The four photographs, and all other photos one would see after clicking on the header, were taken in Yosemite National Park during that 2-day period.

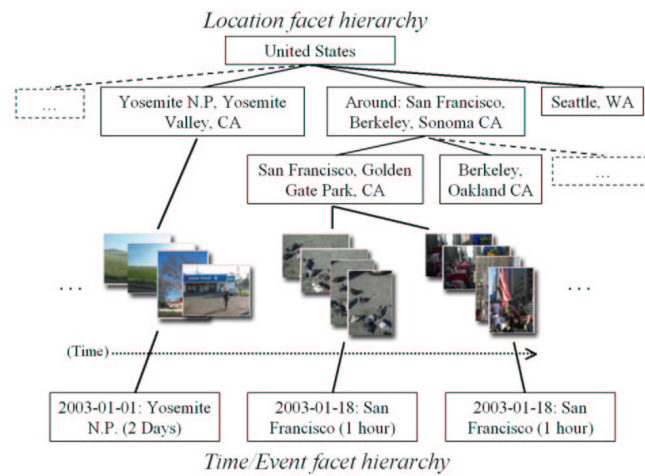


Figure 4.3: Sample PhotoCompass structure. Parts of the location and time/event hierarchies for an actual collection of photos.

The grid’s thumbnails can be grouped by any facet: users can click to have the same photos organized by Location in order to quickly scan representative photos from the different Location sub-categories. In Figure 4.1 this division would correspond to the Location sub-categories as seen on the Location pane on the left.

Users navigate the collection by drilling down in both panes according to their needs. Clicking on a node in the hierarchy constrains the photo collection to only show photos belonging to this node, and expands the respective pane to show the hierarchy breakdown below that node. A possible navigation path is “United States” (Location facet), “2003” (Time/Event facet), and “Yosemite N.P.” (Location again). At that point, the user is presented with all the photos that are part of the Yosemite location, taken at events that occurred in 2003.

As it is based on hierarchical structure, the interface also lends itself to a collapsible tree approach on small screens. See [12] for an example of how hierarchical, textual information can be manipulated efficiently by progressive disclosure and hiding of a nested, tree-shaped data structure. We do not employ this technique in this work.

Figure 4.3 shows a sample schematic view of the Time (bottom) and Location (top of the figure) hierarchies that *underlie* the visible elements of Figure 4.1. The

hierarchies, as well as place names and location designations like *Around: San Francisco...* are automatically constructed from the photos' time and location information as described in Chapter 3. The top nodes in the Location hierarchy are countries. The next level is the breakdown of the country into lower-level location categories.

The top nodes in the Time facet hierarchy are years. The next level, under each year, is simply a flat sequence of events as detected by PhC's event analysis (see Chapter 3). An event is a period of time during which photos were shot, and which the user likely thinks of as one 'occasion.' The granularity of events varies; for example, in Figure 4.1 siblings in the time/event hierarchy are *2003-01-01 (2 days)*, *2003-01-18 (1 hour)*, etc.

Beyond the Location and Time/Event facets described so far, we have explored other types of metadata that can be automatically derived using the time and geographic coordinates of the photos. In particular, given exact time and place where a picture was taken, we are able to use a number of secondary data sources to deduce the actual local time at the shot location; the daylight status (was it night, day, sunset or sunrise); and even the weather conditions and the temperature at the time the picture was taken. All facets can be used to navigate a photo collection. This additional metadata enables, for example, browsing for a photo that was taken in England, on a rainy day, just before the sun would have set.

While we do have this powerful secondary metadata available and have integrated it in PhC, we decided to exclude these facets from the experiment, because they are not available in the experiment's WWMX comparison interface. Exposing subjects to these additional dimensions could thus have given PhC a result-distorting advantage. Instead, we evaluate the usefulness of the additional metadata separately in Chapter 5.

### 4.1.2 WWMX Browser

The World Wide Media Exchange (WWMX) [90] is a state-of-the-art map-based application for digital photos. This application was originally designed by Toyama et al. for a *global* collection of geo-referenced photos (thus "*World Wide Media Exchange*"). Still, WWMX's user interface is also designed to be used for a single-user

photo collection. While the application implementation and interface are described in detail in [90], we summarize it here briefly for completeness.

The WWMX browser uses a graphical map and timeline interface. At any point during the browsing process, the user can view the map, the timeline and a set of photos that occur within both the boundaries of the displayed map and the limits of the displayed timeline. For example, if a map of the United States is shown, and the timeline is set to display the year 2003 only, the photos displayed will be ones that match both these filters. A corresponding sample screen shot of the WWMX application is shown in Figure 4.2 on page 59.

The photos are shown as thumbnails in a dedicated photos pane, but also as dots on the map and the timeline. The dots are consolidated into larger dots when they occur in proximity. Users can pan and zoom the map to show a different area and therefore a different set of photos. An efficient way to do so is to draw a rectangle over the map; the map then zooms into the rectangle. Similarly, users can pan and zoom the timeline to restrict the display to photos from a specific time.

While the interaction in both the location and the time dimensions is different than the PhC interaction, the effect of the interaction is in essence equivalent: each interaction focuses and constraints on either the location dimension or the time dimension. Therefore, comparing WWMX and PhC is relevant and meaningful, and will enable us to evaluate the benefits and drawbacks of a text-menu based interface and a map implementation. The following section describes the experiment we devised for this purpose.

## 4.2 Experiment

A within-subject study was conducted to investigate how users use both systems to perform (i) focused browsing/search for specific photos, and (ii) browsing for photos based on a theme. For each participant, we used the participant's *own personal collection* of photographs. Participants' interaction with the systems was recorded. We looked both at subjective and objective measures to compare users' performance, preferences and attitudes with two photo browsers. We used objective measures

such as task completion time and mouse click count as well as subjective after-task questionnaires.

### 4.2.1 Participants

It was practically impossible to find any geo-referenced personal photo collections at the time of the study, as the geo-photo technology is not widely available yet. In the previous chapter, we used the three “real life” collections we had access to in order to evaluate the output of the PhotoCompas hierarchy-generation step. These were collections whose owners carried a GPS device when taking the photos, such that all photos in the collection had accurate location metadata (see Appendix A).

For the user study, however, we required a much larger subject pool. In addition, we could not reuse subjects from the previous experiments. We solved this problem by using a “location stamping” tool. The Location Stamper<sup>2</sup> allows users to retrospectively mark their photos with a location by dragging and dropping the photos onto a map (see detailed description in Appendix A). Thus, our participant pool was extended to all users with a collection of digital photos, even if those photos were not initially geo-referenced. However, this relaxation of participant selection constraints required our participants to invest extra time in location-stamping their photos. This activity required 1.5 hours per participant, on average.

Even with the location-stamping tool, since digital cameras have only become popular in the last few years, it was difficult to find participants with sizeable photo collections. A further constraint was that participants needed to feel comfortable about giving us access to their photos.

At the end of the search process, and despite the logistical problems, we were able to recruit 15 participants for our experiment; one was used as for the pilot study, and another experienced severe technical problems and was excluded from the result set. We report below the results for the remaining 13 participants.

Participant ( $N = 13$ ) ages ranged from 17 to 49, with the highest representation in the 20s. Five of our participants were male, and eight were female. As both PhC and

---

<sup>2</sup>Available from <http://wmx.org>

WWMX utilize time as well as location information, we did not process photos that had missing time or location information. The average processed collection size was 1,489 pictures. The average time span of the collections was 2.5 years. On average, each collection contained photos from 3.2 different countries. All participants signed consent forms and were compensated for their time.

### 4.2.2 Procedure

For the experiment, we loaded the participant's photo collection onto a desktop PC with 512MB of RAM, dual 2.8GHz Intel Pentium 4 processors, and a 21" flat panel display with a resolution of 1280x1024 (WxH) pixels and 32bit colors. The thumbnails we generated for the photos in PhC were 140 pixels long on their longer edge. The WWMX application used a smaller thumbnail size that fit inside a rectangle of 42x30 (WxH) pixels.

The experiment followed a within-subject design. We exposed each participant to two experimental conditions: the PhotoCompass browser and the WWMX map-based browser. The order of participants' exposure to the two conditions was balanced and assigned in random.

Each participant completed two tasks on each browser. The first was a Search Task. We showed the participants one of their own photos on a computer monitor and asked them to find that photograph in their collection by navigating the application. We set no time limit for this task, but timed it and asked participants to work as efficiently as they could. In order to minimize experimenter bias during the selection of photos for the Search Task, we had a computer randomly select the photos from each participant's collection. The computer presented one random photo after another to one of the experimenters. The experimenter accepted or rejected each photo based on the following criteria: a photo was rejected if (1) the picture was taken at the same event as one that had previously been chosen, or (2) the photo did not display any recognizable context, and the participant would not have been able to identify the photo in the collection. All other photos were accepted. The study in [71] followed a similar procedure and reports positive experience with this approach. We used this



procedure in [37], also with positive results.

The second task was a Browse Task. We asked the participant to select pictures for a collage that represented some portion of the participant’s life. We randomly alternated the collage topic between the two conditions. The two collage topic choices were “friends and family”, and “trips.” We asked participants to select photos from as broad a time span and set of occasions as possible. We did not impose a time limit, but rather asked them to “stop when you feel you found enough photos” (which usually took 4–5 minutes).

For each browser we had participants complete the photo Search Task six times. Then participants performed the Browse Task once for each browser. Each time participants completed both tasks under one of the conditions, they were asked to complete a questionnaire. We asked questions such as the helpfulness of the photo organization, the participant’s degree of satisfaction, the amount of frustration, and adequacy of the allowed time. Answers were encoded on a 10-point Likert scale. We also timed the Search Task, measured the number of mouse clicks for the Search and Browse task, and counted the number of pictures found during the Browse task. Finally, we debriefed all participants at the end of the experiment session.

### 4.2.3 Other Procedural Considerations

There are a few limiting yet unavoidable issues regarding the experiment design. The map-based Location Stamper seems to advantage the WWMX interface, as participants were exposed to the location of their photos on the map when they stamp their photos in the first step of the experiment. When the subjects later use the WWMX interface, they are slightly biased as they were exposed to, and even created, the map locations of their photos. For this reason, we tried to have at least a few days between each participant’s location-stamping session and the experimental session. However, due to time and participants’ personal constraints we could not ensure such a gap for all participants.

On the other hand, WWMX was disadvantaged because we used “manual” referencing of photos, rather than using an accurate location-capturing device. As the

referencing accuracy was determined by each user when they were using the stamping tool, we did not have full-scale accuracy — especially when users were referencing trip-related photos, from locations they did not know as well. Arguably, the added accuracy may benefit a map-based application. It must be noted that the manual referencing may have hurt PhC as well — photos were often marked in locations that did not allow PhC to pick the best name for each set of photos. More accurate GPS based location acquisition would have improved PhC’s naming performance.

PhC was further disadvantaged by the user interface. The Flamenco interface toolkit, while being the correct choice for the fast prototyping facility we required, is limited to HTML-based interactions. Even a mostly-textual interface such as PhC can be visually optimized for a particular application, but Flamenco does not currently provide the necessary control for such optimization. In addition, we could not employ the techniques we developed [34] for selecting summary sets of photos to represent an event or location (like the groups of photos in the grid of Figure 4.1); the Flamenco interface only displays the first few photos of a set, and does not currently allow the application to participate in selecting which photos will be displayed to represent a given collection of items.

Finally, it must be noted that interaction with these two systems is not entirely independent. For example, if a participant was exposed to PhC first, they may during the first part of the experiment become familiar with, or at least be reminded of, the content of their collection. As a consequence, interacting with the second system may be aided by this knowledge. To this end, we took care to balance the order in which participants were exposed to the two conditions as each participant was assigned the first condition at random. On top of that, we checked for the effects of the order of exposure on the users’ performance, and did not find significant differences.

### 4.3 Results

Before the experiment, we hypothesized that the users would perform better using the map, and would like it more than the textual PhC implementation. However, our experiment showed that no significant difference was found between WWMX and

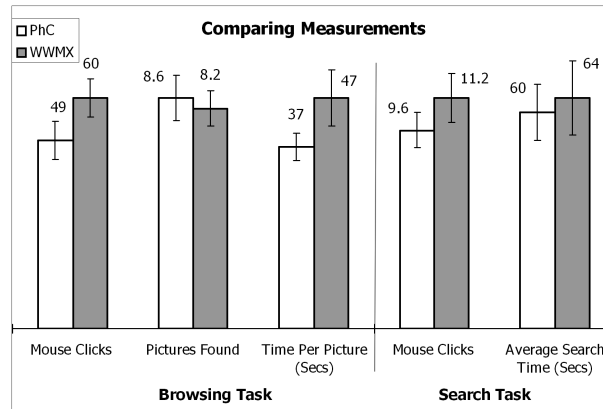


Figure 4.4: Objective measurements of the Browse Task and the Search Task (two rightmost columns).

PhC.

Figure 4.4 presents performance results for the Browse Task and Search Task. For the Browse task, we recorded the number of pictures found, the number of mouse clicks and the average time per picture as performance measures. These measures are shown in the three leftmost columns in the figure. A Repeated Measure ANOVA test was used to investigate the difference between WWMX and PhC. As shown in the figure, the average number of pictures found with WWMX ( $M = 8.6$ ,  $SD = 3.0$ ), is not significantly different from that with PhC ( $M = 8.2$ ,  $SD = 2.4$ ,  $F(12) = 1.32$ ,  $p > 0.1$ ). However, the photos were collected significantly faster using PhC ( $M = 37$  seconds,  $SD = 10.04$ ) than with WWMX ( $M = 47$ ,  $SD = 20.53$ ,  $F(12) = 8.50$ ,  $p < .05$ ). In addition, WWMX ( $M = 60$ ,  $SD = 16.24$ ) required more mouse clicks for the Browse Task than PhC ( $M = 49$ ,  $SD = 16.24$ ,  $F(12) = 10.2$ ,  $p < .01$ ). Contrary to our pre-experiment prediction, participants executed the Browse Task more efficiently with PhC than with WWMX.

For the Search Task, as shown in Figure 4.4 (two rightmost columns), no significant statistical differences were found for average search time (in secs:  $M_{phc} = 60$ ,  $M_{wwmx} = 64$ ,  $p > 0.1$ ), and the number of mouse clicks ( $M_{phc} = 9.6$ ,  $M_{wwmx} = 11.2$ ,  $p > 0.1$ ).

In Figure 4.5, Figure 4.6 and Figure 4.7 we move to report on subjective results

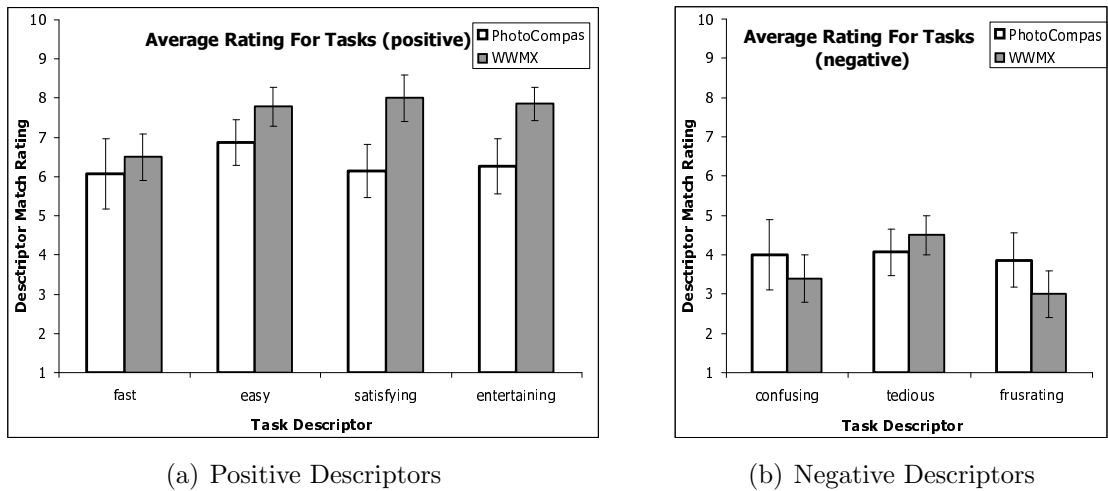


Figure 4.5: User subjective evaluation of the tasks in both applications.

measured by the questionnaire.

Figure 4.5 refers to participants’ responses regarding how they felt about performing both the Browse and Search tasks on each of the interfaces. The ratings were given on a 1–10 Likert scale. A higher rating for a certain descriptor means participants found the descriptor more appropriate for the particular interface. The figure shows both positive descriptors (4.5(a), where a taller bar is interpreted as better) and negative descriptors (4.5(b), a taller bar is interpreted as worse). Participants found the tasks more “satisfying” when performed in WWMX ( $M = 8.0$ ) than they did when performed in PhC ( $M = 6.1$ ,  $p < 0.05$ ). The WWMX task experience had a higher entertainment value ( $M_{wumx} = 7.8$ ,  $M_{phc} = 6.2$ ,  $p < 0.05$ ). Other than these two measures, there were no significant differences between WWMX and PhC for subjective task evaluations.

We also asked the participants for a subjective evaluation of each application. The participants rated terms like “complex”, “efficient”, “helpful”, “novel”, “intuitive”, etc. The results (excluding “complex”) are shown in Figure 4.6. Again, a higher rating for a certain descriptor means participants found the descriptor more appropriate for the particular application. In this figure, taller bars are interpreted as better (all descriptors are positive). Out of these measurements, the only factor which exhibits

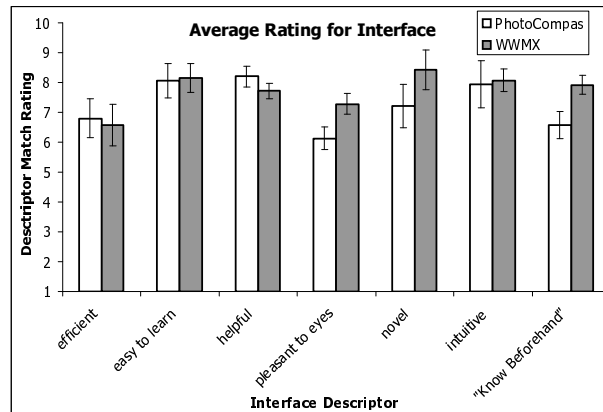


Figure 4.6: User subjective evaluation of both applications.

a statistically significant difference is “know beforehand”. This descriptor represents a subjective measure of how well the participants thought they knew, before embarking on each task in this application, where to look for the photo. WWMX ( $M = 7.9$ ) was rated higher than PhC ( $M = 6.6$ ). However, this subjective measure was not backed by participants’ actual task performance, as we demonstrated above. No other subjective measure was significantly different between WWMX and PhC.

Figure 4.7 compares participants’ subjective sense of whether time and location, respectively, were important for completing tasks on either system. That is, whether the location and time were powerful manipulators in the two applications. The subjects had given a 1–10 rating for the perceived usefulness of each dimension, in each application, for both the Search and Browse Task.

Figure 4.7(a) shows that users had found the time dimension more useful in PhC than in WWMX for the Browse and Search Tasks. In other words, time played a much more important role for PhC during both browsing and searching than it came into play for the WWMX condition. Recall that PhotoCompas attempts to identify and name sets of photographs that were taken at the same event. In WWMX the time element is instead represented via a visual timeline approach. One thing to notice is that the time dimension was more useful in WWMX for the browse task than for the search task, maybe not surprisingly: the browse task often required a range of photos from different occasions.

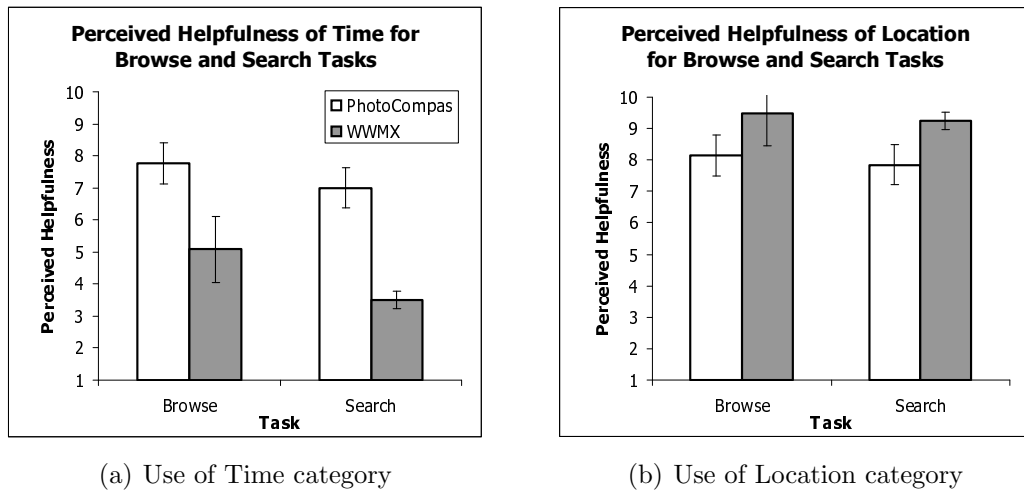


Figure 4.7: Subjective reports on the helpfulness of Location and Time categories in browse and search tasks.

The difference in the importance of *location* between the two applications is shown in Figure 4.7(b). For WWMX, subjects felt that location played a larger role than they thought it did for PhC. The difference was not quite as pronounced as the difference in the perceived usefulness of time, but still significant. Still, Location was clearly important in both interfaces and for both types of tasks. Further analysis reveals that there is a significant interaction effect ( $F(1, 12) = 22.59, p < .01$ ) between interface and manipulators (time and location), which indicates that participants found the time element more useful with the text-based PhC browser, while they found the location information more important with the map-based WWMX browser.

## 4.4 Discussion

Before discussing the results of the previous section, we reiterate that the number of subjects we recruited (given the exacting requirements for subjects' photo collections and time commitment) is only enough to indicate trends in the data. These trends are important to consider for the continuing design of similar applications in this and other domains. However, it is still risky to make broader conclusions without a more

Table 4.1: Summary of statistically significant differences between WWMX and PhotoCompas.

<i>Application</i>	<i>Advantage</i>
WWMX	Found to be more entertaining and satisfying by participants
WWMX	Participants felt more secure beforehand as to where to find photos
PhC	Browse task required less time and mouse clicks
PhC	Event/time dimension more useful for browse/search than WWMX
WWMX	Location dimension more useful for browse/search than PhC

extensive study.

#### 4.4.1 Measured and Questionnaire Results

Table 4.1 more qualitatively summarizes the statistically significant differences between the applications as discovered in the experiment.

The most surprising result of this study is that so little difference emerged in the averaged objective measures of search and browse speed. The WWMX user interface is much more visually oriented than the PhotoCompas interface, which relies predominantly on textual cues and a more analytic mental model. The mouse click count is a particularly suggestive measure, as the function of specifying constraints requires the same number of mouse clicks in both systems. Note that for the Browse Task PhC elicited significantly fewer clicks than WWMX. This fact may be one of the contributing factors to PhC's shorter per-picture browse time (Figure 4.4).

We should mention that the WWMX system suffered some delays as its backend servers were heavily loaded during some of our experiments. We have removed such outliers from the result data, and even discarded all the data from one user as mentioned above due to these delays. In general, both systems exhibited some latency in reacting to user commands. The latency was hard to quantify and roughly equivalent between the two systems.

The visual differences between the applications were exacerbated by the currently unrefined screen appearance of PhotoCompass, when compared to the mature WWMX look. The visual difference was noted by virtually all subjects and is reflected partly in the responses to the *entertaining* and *satisfying* questions, where WWMX shined.

Another surprising result was that there was no difference in the way the two systems were perceived to be *easy*, and (conversely) *confusing*. We had indeed expected that the textual PhC would tend to be less straightforward and more confusing than the much more broadly familiar map-based interface. That hypothesis was not proven.

These results certainly suggest that PhC’s location and event hierarchies were intuitive enough, and together with the automatically generated names, and thumbnails, users had a good grasp of PhC’s navigation mode. Also, the results suggest that users utilized the improved time context (the notion of event) in PhC to compensate for the deficiency of the map-less location navigation. Indeed, users have repeatedly asked for the addition of an “events” feature in WWMX as we report below.

We also noticed, through observing the subjects during the experiment and subjective feedback, that while most subjects liked the map-based interface, some subjects implicitly and explicitly expressed their aversion towards the map-based interface in favor of the text-based hierarchical browsing of PhC. This aversion slightly reflected in their performance measures as well. One question that this observation raises is whether there is a strong bipolar trend in people’s preference for map-based interfaces versus text-based interfaces. Answering this question with statistical significance requires an experiment of much larger scale.

#### 4.4.2 Results from Debriefing Session

We received valuable feedback during the concluding, informal portion of the experiments, where we asked participants for open-ended feedback about both systems. Several issues in particular stood out. They are listed here, roughly ordered from most general to most application-specific:

- The PhC notion of events was popular, and WWMX was lacking in the Time



dimension. Not only did participants find the event metaphor intuitive in PhC, they also requested this feature for WWMX as well. Also, the WWMX timeline interaction was not intuitive to some.

- A calendar view was often requested for PhC, in addition to the flat event listing. Recall that the PhC interface allowed selection of events by year, and displayed a list of all events that occurred in that year, without a month level in between. For some participants, there were too many events in every year, and another level was required. In addition, some participants asked for a “month” or “season” category that is independent of the year: often, they remembered that an event occurred in a specific month or season, but were not sure about the year. Indeed, we use such a category in our full interface version (Chapter 5), which was not investigated in this study.
- In a similar vein, users asked for further breakdown of the high-level events to lower-level events, especially when the high-level event consisted of a few photo-taking days. This feature is certainly feasible, and we have implemented it in some of our other applications [34, 37] as well as subsequent versions of the PhotoCompas system.
- Many subjects requested better abbreviated summaries of image clusters. As mentioned earlier, our implementation platform happened to preclude the necessary operations to intelligently choose representative photos from a set of photos. Procedures for choosing images that ensure good summarization of a photo set are still open to research. In previous work [34] we showed that choosing images that span the time range of the photos in each cluster is effective for summarization. Other considerations like location and image content features could also help in choosing summary images.
- For both applications, participants asked for additional ways to add and manipulate the metadata. One example is renaming events in PhC to reflect the actual content of the event (e.g., “Grandma’s birthday”). Another example is adding information about the people in the photos. These are indeed standard

features in today’s photo browsers.

- The size of thumbnails was an issue for many participants, in both interfaces. A full implementation of the browsers should allow the users to scale the thumbnails to the preferred size.
- The text-based search mechanisms were too limiting in PhC. Subjects requested that keyword search should be made smarter. For example, if a majority of photos had been shot in Yosemite National Park, but one or two were taken in Groveland, a small town near, but outside the Park, then the PhC algorithm would produce the label “Yosemite National Park” for all photos from that area. The term “Groveland” would therefore not occur in the system’s label corpus. A search for this term would thus fail. A more intelligent engine would test whether the given location name was within one of the clusters the algorithm had identified, and would return a more helpful answer. In this case, entering of the terms can be aided by auto-completion based on the locations that appear in each user’s collection. Such auto-completion is of course also possible for the WWMX interface.

Finally, and expectedly, many participants noted that some combination of the two applications would be beneficial. As one participant put it, “These are two metaphors I need at various times.”

## 4.5 Fixed vs. Ad Hoc Hierarchies

In Chapter 3 we hypothesized that fixed location and time hierarchies may not perform as well as flexible, ad hoc hierarchies like the ones created by PhotoCompass. For example, a system that organizes locations into a country/state/city hierarchy may become too “bushy” with too many leaf elements (cities visited by the user). We tried to check our hypothesis using a small-scale experiment.

The experiment was executed along the same lines as the comparison between the map and non-map interface described above. We compared the pre-defined hierarchy

approach with our personalized, automatically created hierarchy, via tasks similar to the ones described earlier in this chapter.

Unfortunately, the small number of participants in this study (7 participants) does not allow for conclusive judgement. We note that even when pictures occurred in relatively few different cities, the performance of users browsing the fixed hierarchy did not exceed the performance when browsing the personalized (ad hoc) hierarchy. We expect that our method will outperform the fixed hierarchy for richer collections, where more cities were visited. More specifically, we found no significant difference in average search time and average mouse click for searching photos with the two layouts. In addition, participants found these two layouts similar in most subjective scales such as level of confusion, ease of use, satisfying interaction and so forth.

The only subjective scale that exhibited statistical significance is the usefulness of time information. Participants found that time information is more useful for hierarchical layout than personalized layout. The reason may be two shortfalls of the event representation in our automatic hierarchy. First, events are listed directly under the year in which they occur; another level of “months” might have simplified browsing as mentioned above. Second, as noted above, in the experiment implementation version, events that span multiple days were not split into sub-events, which made it hard to find a specific photo from a long-duration event.

## 4.6 Conclusions and Future Work

We showed in a controlled experiment that, against intuitive expectation, a textual-menu-oriented browser can enable as good a user access performance as a map-based interface to a time- and location-stamped personal collection of photographs. Subjects performed search and browse tasks just as quickly and completely with the textual approach as with the map alternative.

To compensate for the deficiency in location-based browsing, our browser was aided by an enhanced time-based support in form of events. However, participants were also able to efficiently navigate the text-based location hierarchy.

The result of our study is important for guiding designs that cannot rely on maps.

A prominent example is the access of collections on small devices. The limited screen size on those platforms severely handicaps the use of maps as a primary interface element. While our study focused on photo collections, it may also be relevant for other types of collections, especially when the location distribution is inherently “clustered.”

In any case, the visual appeal of the map-based approach was not lost on our subjects. Many suggested a combination of the two facilities. Such a combination is indeed natural. A challenge will be to accomplish the fusion such that drawbacks of the map are compensated for. The drawbacks include inefficient screen real-estate usage and the confusion that portions of the population experience when viewing maps. Once that fusion has been prototyped, this study will serve as a baseline for measuring its success.

## Chapter 5

# Enhancing the Context Metadata

In this chapter we extend our PhotoCompass system described in previous chapters. Location and time metadata are considered to be *primary context types* [20] that can be used as indices to other sources of context. Indeed, in addition to utilizing the time and location metadata to automatically organize a photo collection, here the system employs location and time as a key to harvest additional, derived context about each photo from various sources.

In the first parts of this thesis, through this chapter, the location and time context in which photos are taken is treated as *passive context* [15]. In other words, we collect context parameters, and make them persistent for the user. Later, the user can retrieve the context metadata, or use the context information for searching the collection.<sup>1</sup> We extend this framework to the newly generated derived context categories. Once these new context metadata are generated, we integrate them into the browser's interface. The users can then view the metadata associated with each photo, or use the metadata for search.

For example, we obtain weather information about each photo. The time and place where the photo was taken allows us to retrieve archival data from weather stations that are local to the photo's exposure location. Similarly, given time and location we automatically obtain the time of sunrise and sunset where the photo was

---

<sup>1</sup>In contrast, an application can be based on *active context*, where the context actively determines the behavior of the application.

<b>Time</b> <a href="#">2002</a> (398) <a href="#">2003</a> (3306)	<b>Time of Day</b> <a href="#">Afternoon (12pm-5pm)</a> (1573) <a href="#">Late night (12am-3am)</a> (28) <a href="#">Early morning (3am-6am)</a> (22) <a href="#">Morning (6am-12pm)</a> (923) <a href="#">Evening (5pm-8pm)</a> (650) <a href="#">Night (8pm-12am)</a> (508)
<b>Location</b> <a href="#">Cambodia</a> (151) <a href="#">Italy</a> (146) <a href="#">France</a> (167) <a href="#">Sri Lanka</a> (512) <a href="#">Hungary</a> (176) <a href="#">Thailand</a> (60) <a href="#">Israel</a> (670) <a href="#">United states</a> (1822)	<b>Weather Status</b> <a href="#">Clear</a> (944) <a href="#">Mist</a> (61) <a href="#">Fog</a> (2) <a href="#">Mostly cloudy</a> (373) <a href="#">Haze</a> (135) <a href="#">Overcast</a> (110) <a href="#">Heavy rain</a> (6) <a href="#">Partly cloudy</a> (590) <a href="#">Light rain</a> (237) <a href="#">Patches of fog</a> (1) <a href="#">Light rain showers</a> (3) <a href="#">more...</a> <a href="#">Light snow showers</a> (3)
<b>Elevation</b> <a href="#">-2000--1001</a> (36) <a href="#">10000-10999</a> (85) <a href="#">-1000--1</a> (327) <a href="#">11000-11999</a> (59) <a href="#">0-999</a> (2425) <a href="#">12000-12999</a> (37) <a href="#">1000-1999</a> (151) <a href="#">13000-13999</a> (40) <a href="#">2000-2999</a> (53) <a href="#">14000-14999</a> (33) <a href="#">3000-3999</a> (43) <a href="#">more...</a> <a href="#">4000-4999</a> (57)	<b>Temperature</b> <a href="#">20-40</a> (87) <a href="#">80-100</a> (239) <a href="#">40-60</a> (972) <a href="#">Unknown</a> (1646) <a href="#">60-80</a> (760)
<b>Season</b> <a href="#">Autumn (sep 21st-dec 20th)</a> (1007) <a href="#">Summer (june 21st-sep 20th)</a> (1059) <a href="#">Spring (march 21st-june 20th)</a> (953) <a href="#">Winter (dec 21st-march 20th)</a> (685)	<b>Time Zone</b> <a href="#">-5</a> (8) <a href="#">2</a> (670) <a href="#">-7</a> (180) <a href="#">5</a> (512) <a href="#">-8</a> (1634) <a href="#">7</a> (211) <a href="#">1</a> (489)
<b>Light Status</b> <a href="#">Dawn</a> (47) <a href="#">Dusk</a> (495) <a href="#">Day</a> (2367) <a href="#">Night</a> (789)	

Figure 5.1: The metadata categories generated by our system, as shown in the interface opening screen.

taken. In Section 5.1 we describe the metadata we thus assemble for each photograph. PhotoCompas integrates this contextual information in its user interface, as shown in Figure 5.1. The figure displays the opening screen of the interface interaction; the details of the interaction are listed in Section 5.2.

The metadata produced can be helpful in the retrieval of photos from a collection that spans many years and thousands of photos. Without trying to analyze the contents of images, our system annotates each photo with the relevant context data. Then, the user can search for a photo by the context category. For instance, a user may remember that a certain photograph was taken during a rainstorm. She could filter the collection to show only the “rainstorm” photos, thus limiting the number of photos she needs to browse in order to find the wanted image.

While this contextual metadata can serve well as memory cues, it can sometimes also imply the content of the image. For example, by clicking on the *Dusk* entry in the *Light Status* section of Figure 5.1, a photographer requests to see only photos that were taken when dusk had fallen at the exposure location. Figure 5.2 shows the result of this action on one of our test collections, further restricted to show only photos from Sri Lanka.

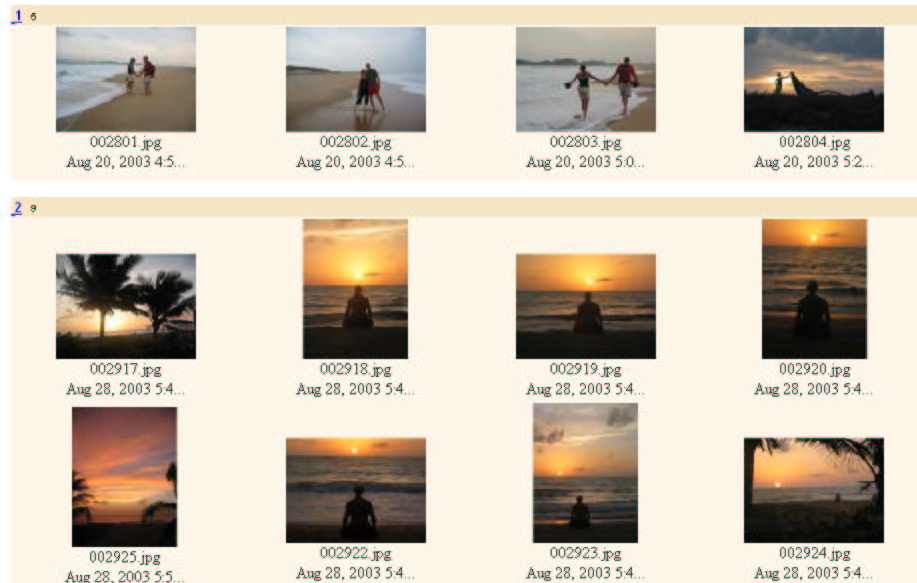


Figure 5.2: A subset of the "Sri Lanka dusk photos" from the thesis author's collection, detected using contextual metadata.

Our focus here is on (i) measuring users' belief regarding how effective our particular set of contextual metadata is for photo retrieval, (ii) observing which of these metadata users actually take advantage of when searching through our interface, and (iii) exploring what other contextual metadata would be profitable to capture in the future.

To this end we gathered and analyzed several data sets by means of a user study and a separate survey. In the user study we had nine subjects find photographs from their own collection by interacting with our metadata-enriched PhotoCompass browser. We recorded their paths through the interface and analyzed the results. Also in the user study we asked each participant to rate 24 potential categories of contextual information by how useful they believed the additional information would be for retrieval of the same photos they searched for during the study.

In the survey, which involved a larger number of participants (35), we had participants describe some of their own photos from memory. We also asked them to rate the same 24 metadata categories in terms of how well they were remembered for each photo. We then compared the collected data and conclude which of the 24 categories

promises to be effective for photo collections.

The practical results of this work are recommendations on which metadata sources would be particularly profitable to create or tap into, in order to make photo collections more easily accessible.

The methods we describe here may extend beyond photo collections to assist in the management and retrieval from, for example, an all-encompassing personal data store (e.g., [23, 29]). Alternatively, some of the metadata may be useful for enhancement of, and retrieval from, global repositories such as a set of geo- and time-referenced news reports.

The rest of this chapter is organized as follows. In Section 5.1, we describe the contextual metadata that we currently extract from various available sources, given time and location information for each photo in the collection. Section 5.2 discusses the use of these metadata in the user interface. Section 5.3 describes the user study we devised to assess the usefulness of our added metadata, and presents the results of the study. In Section 5.4 we describe the methods and results of the survey, and how they relate to the user study's results. We conclude in Section 5.5.

## 5.1 Metadata Categories

This section describes the various context metadata categories that can be derived using the time and location stamps embedded in the photographs. We show here how we generated each metadata category. In Section 5.2 we describe how we use these metadata in the user interface. The metadata categories are listed below, where for each we supply:

- An explanation of the category, and how it contributes to personal photo collections.
- Possible ways to generate the context in this category using current or future technologies.
- A description of the method by which the category is currently generated by our system.



For completeness, we include the “location” and “event” categories, already described in Chapter 3. We omit the discussion of how these two categories are generated in our system, as it was already discussed there at length.

## Location

The location recorded for each photo refers to the location of the *camera* at the moment of capture. Another interesting location is the location of the *subject* of the photo, which we assume is similar to the camera location.

The location is represented by latitude and longitude. This  $(latitude, longitude)$  pair must be translated into a useful location representation in order to supply context that can be understood by humans. A map is one option [90] to provide such context. In Chapter 4 we have shown that browsing a textual location hierarchy (where the location context is given using textual names) can augment map-based location browsing, or even replace the map (for example, when screen real-estate is limited).

Textual location names can be generated using an off-the-shelf geographical dataset that enables querying with a  $(latitude, longitude)$  pair to get the containing country, province/state, county, city (if any) and park (if any), as explained in Chapter 3. Additional location-name context can be supplied by gazetteers such as Alexandria [38], which can be queried for nearby geographic landmarks to each photo. In Chapter 7 we present a way to get additional location-based textual context that may pertain to the location of the subject of the photo, and even to the actual landmark that appears in the image.

## Event

In personal photo collections there is a strong perceived context of events (see Chapter 4, as well as [17, 25, 75], and more). People often think of their photos in terms of events, and usually associate each photo with a certain event: a wedding; a vacation; a birthday party etc. The “event” can therefore be considered useful context.

Photo management systems today cannot detect the actual event context, e.g.,

“grandma’s birthday party.” However, the technology is mature enough to be able to automatically group photos that were taken during the same event [17, 34], especially when location data is available, as we showed in Chapter 3. In a user interface, users will be able to add their own caption for the event (e.g., “Grandma’s birthday”). This latter step can also be done automatically, for example, by correlating detected events in the photo collection with events from the user’s calendar; we do not implement such a scheme at this time. If the event is public (e.g., a wedding, a baseball game, or a music concert), then sharing data between users may help in assigning a meaningful name to the event. The assignment can be done automatically or semi-automatically, based on systems like the one we describe in Chapter 7, or the system described in [77]. A different system that can support this event-name function is an event gazetteer [2].

In our current implementation, the textual context that represents each event and is presented to the user is derived from the time and location information in each photo. An event is annotated with a place name (based on the location of the photos in the event) and time, e.g., “San Francisco, July 7<sup>th</sup> 2004” (see Chapter 3).

## Time of Day

Knowledge about the time of day a photo was shot can be used when searching for an image in a collection. Given an exact, dependable local time, we can utilize it to help the user search for photos by “time of day.” Naturally, users are not likely to remember the exact hour each photo was taken. The query/search mechanism needs to support range queries or group the time of day into categories (e.g., “morning” or “late night”).

In most cases, however, users set their camera clock once, usually when they operate the camera for the first time. The clock is used to timestamp all the photos taken with the camera. Indeed, this one-time setting is sufficient to sort all the pictures by time, or to associate pictures with the (approximate) month and date in which they were taken. The pitfall appears when the user travels to a different time zone. Most of the users do not bother re-setting their camera’s clock, resulting in

timestamps that inaccurately portray the time the picture was taken. For example, a photo taken in Israel in the morning, using a camera set to Pacific Standard Time, would appear to have been taken at night because of the 10-hour difference.

Even if people do re-set the time whenever they move to a different time zone, the time zone information is currently not retained for most cameras (today's digital cameras simply embed the timestamp, without a time zone). A situation can occur where the sorting of photos can be flawed: a picture that was taken in Japan, and another taken 12 hours later in San Francisco, will appear out of order since the local time in San Francisco is 19 hours behind Japan.

In the future, location-aware cameras will likely automatically detect the local time, as do cellular phones today. In the meantime, the time zone problem can be solved given the location information where each photo was taken, and the original time zone according to which the camera clock was set. This information is sufficient to compute the local time for each photo. For the computation we use a geographical dataset containing the world's time zones. The dataset can be queried by the photo coordinates, returning the correct time zone for each photo. The next step is simply calculating the offset between the camera time zone and the actual time zone at the given location. The system then applies the difference to the photo time to get a local time. A caveat of this scheme, as mentioned above, is that time zone information is not available with current cameras. Until camera settings include the time zone, the user will have to enter the time zone manually (one entry dialog will fix all the photos generated with a single camera) for this adjustment to be performed.

## Light Status

People's perception of the time is sometimes not derived from the clock time, but rather from the daylight status. For example, people may recall a certain picture was taken when it was dark outside; around sunset; before sunrise; etc.

Given the local time and location for each photo, a system can find how many minutes away from the sunset and sunrise each picture was taken. One way to get this information is to implement an algorithm, and compute the sunset and sunrise times

for each photo based on the location and date when the photo was taken. Another way, which we use in our system, is to query the US Naval Observatory Web service<sup>2</sup> to get sunset and sunrise data for the photos. The service returns a complete set of daily sunset and sunrise times for a  $(year, latitude, longitude)$  query. When querying, we round the latitude and longitude to integer values, and cache the replies from the server. Each reply contains data for a full year, and can be re-used across different photos in the same latitude and longitude. To give an idea about the number of necessary queries, only 42 network requests were required for a personal collection of 3706 photos spanning 13 months and 8 countries.

As people are not expected to remember exactly how much time before or after sunset or sunrise their pictures were taken, we group photos into *day*, *dusk*, *night* and *dawn* categories. In our current implementation, the dusk grouping includes all photos taken within one hour before or after sunset; the night photos include all photos taken one hour after sunset to one hour before sunrise, and so on. This arbitrary grouping may be a somewhat inaccurate. For example, northern locations have longer sunsets. We are considering ways to overcome this problem.

## Weather Status and Temperature

Often, people can filter photos using weather information: they recall a certain event occurred on a stormy night (even if the event took place indoors), another event on a clear day, and so forth. In addition, people may remember the temperature at the time the picture was taken (“it was freezing!”).

One important dichotomy, when discussing the weather, is using outdoors conditions versus conditions where the camera is situated (possibly indoors). When opting for outdoors conditions, a practical consideration is whether the system is using general weather information (that pertains to a large area such as a city) or actual conditions at the precise time and place where the camera was situated. Another dichotomy is the measured versus adjusted (e.g., windchill) temperature.

Given location and time, historical or current weather information can be retrieved

---

<sup>2</sup><http://aa.usno.navy.mil/>

from a weather service. Getting the data from the weather service implies measured, outdoors, general-area temperatures and conditions. In the future, general weather data will likely be accessible from a cellular service: the cellular provider may supply a weather service that can be used to mark each photo with the current weather, based on the general location of the camera at that time. Alternatively, weather-related data could be extracted from sensors installed on the camera (e.g., temperature sensors). This latter method may not accurately portray the temperature outdoors if the camera happens to be indoors, or in other words, this method supplies adjusted temperature in the precise time and location.

In our system, we use the Weather Underground<sup>3</sup> web service to get weather information. Thus, the weather in our system is based on outdoors, general-area measurements. In the Weather Underground service, historic data can be queried by a (*zipcode, date*) pair or a (*weather station, date*) pair for weather outside the United States. Our geographic dataset allows us to translate any (*latitude, longitude*) pair into a zip code or weather station. The results of a query to the server can be cached and used for all photos taken on the same day and in the same area, reducing the required number of queries. Again, the number of required queries is small in comparison to the number of photos. For 1823 US photos, for example, spanning 13 months and 70 different zip codes, only 139 queries to the Web service were required when caching the results.

The weather data we get for each day is a hourly (approximately) report of the weather conditions (e.g., “rainy”, “clear”) and temperature. Since we currently retrieve general-area weather, rather than precise location data, we annotate each photo with all weather conditions that appear between two hours before and after the photo (a single photo can be associated both with “rainy” and “cloudy”, for example). This way we compensate for the inaccuracy inherent in the data, while adding to our data a small amount of noise (i.e., weather conditions that happen to be irrelevant). Similarly, the temperature is computed as the average of temperatures measured in the hours around the photo time. We used actual measured temperatures; another option (as listed above) is using perceived temperatures, e.g., factoring in the windchill

---

<sup>3</sup><http://www.wunderground.com/>

information from the weather service.

## Additional Categories

Other metadata categories that we derive from the location coordinates and time stamps of the photographs are:

- Elevation — available either from the GPS data or from a digital elevation model (given the latitude and longitude where the photo was taken, and assuming the photographer was close to the earth’s surface at the time).
- Season (autumn, winter, spring, summer) — by the date in which the photo was taken (of course, this will differ between hemispheres).
- The time zone (offset from GMT) is also shown as a separate category in addition to using it to compute local time.

Many other categories could be produced for browsing photos using current image processing technology, although we have not, so far, integrated such methods into our work. For example, the number of people that appear in the photo may be feasible to detect with sufficient accuracy, at least as a lower bound (detection technology today can be performed at high precision for well-aligned faces [32, 39]); whether the photo was taken indoor or outdoor [61]; prominent colors or other content data [91], etc. In this work we emphasize contextual metadata that pertains to location and time only, but we included many additional categories in our survey as described in Section 5.4.

## 5.2 The Application Interface

While our work here is to deploy the contextual information explicitly in a user interface, it must be noted that the context can also be used implicitly, e.g., for determining similarity between photos, ranking search results, summarizing a collection and so forth. For example, a similarity score of two photos can be boosted if both photos were taken in the rain. To give another example, if the system needs to choose  $n$

representative photos from a larger subset (“summarization”), the system could bias towards choosing photos from different times of day (e.g., some daylight photos, some sunset photos, some night photos).

Focusing on the interface, we describe the prototype interface we used to initially test the contextual metadata we produced. Before we get to that, we describe the requirements for an “ideal” interface, given this contextual information.

An interface that will enable effective usage of context must be:

- Non-intrusive. The context selection/filtering mechanism should not utilize an excessive amount of screen space, nor should it create clutter that requires additional mental effort.
- Simple and clear. For example, weather conditions should be represented by very few general groups (e.g., sunny, cloudy, rainy, snowy, windy) and not by dozens of possible weather conditions (showers, light rain, thunderstorms and so forth).
- Able to accept range specifications. For example, being able to specify “it was cold — between 0 and 25 degrees” for temperature; specify a calendar range; a time-of-day range (“sometime between 6am and 8am”).
- Allow exclusion of groups of photos easily. E.g., “any time but night”; “it wasn’t raining.”
- Be flexible enough to simultaneously apply multiple filters (“it was a cold, rainy morning”) and simple boolean queries. According to Kang and Shneiderman [44], users can handle simple boolean queries that include an intra-domain ‘or’s and inter-domain ‘and’s: “San Francisco or Palo Alto, in the rain.”

As explained in Chapter 4, we used the Flamenco HTML-based toolkit [97] to create the interface for our PhotoCompass. In particular, we extended it here to enable browsing of the contextual metadata. The Flamenco system does not meet all the requirements laid out above. For instance, Flamenco does not allow selection of multiple categories from the same facet (e.g., “rainy” *or* “cloudy”). In addition,

freely specifying a range of values, for example, for the time of day, is not possible. However, Flamenco allowed us to prototype a browsing interface quickly so that we can proceed to build a complete user interface once we have the certainty that the contextual information is indeed useful, and have more information pertaining to how people use, or would like to use, this context.

In each step of the interaction with the Flamenco interface, the user can select from a number of values under each category. After a user clicks on a value, the interface shows only the photos that match that value, as well as other category values that were chosen earlier. For example, “United States” and “Sri Lanka” appear under the Location category in Figure 5.1. Once the user clicks on “Sri Lanka”, and then “Dusk” in the Light Status category, the collection is constrained to show only photos that satisfy “Dusk” *and* “Sri Lanka”, as shown in Figure 5.2. At each point of the interaction, further refinement is possible. Refinement can be done using other categories (for example, clicking on “rain” under the weather category) or more detailed groupings of the chosen category (e.g., clicking on a location within Sri Lanka to show only photos from this location).

As the Flamenco interaction did not allow for freely choosing a range of values in each facet, we created pre-defined ranges as browsing categories when needed, usually when the values that the category takes are continuous. For example, under the Local Time category, we grouped photos by the major parts of the day — Morning (6am-12pm); Afternoon (12pm-5pm); Evening (5pm-8pm); Night (8pm-12am); Late Night (12am-3am); Early Morning (3am-6am).<sup>4</sup> Similarly, we grouped the temperature and the elevation values into pre-defined ranges as shown in Figure 5.1.

### 5.3 User Study

We set out to test the use of the contextual metadata on real-life, large personal collections of geo-referenced digital photos. Our goals were to discover:

- Whether participants are able to use the contextual information to browse for

---

<sup>4</sup>This breakdown may be a matter of cultural preference; people in Europe, for example, may consider the night to last at least until 2am.



photos.

- Which of the context metadata that can be derived using location and time are useful.
- Whether the results correspond to the outcome of the survey. In other words, do the categories people used correspond to categories people remember well. This is discussed in Section 5.4.

We first describe the user study setup, and then the methods and results.

### 5.3.1 Statistics and Setup

Recruiting suitable users for the experiment was again a difficult task, as we had also experienced in previous experiments (Chapter 4). See Section 4.2.1 for an outline of the recruiting problems, and how we were able to partially overcome them. For this experiment, in addition to requiring that the subjects' collections have valid photo timestamps, and requiring the subjects to location-stamp their photos manually (as described in Appendix A), the participants also had to have an idea regarding the camera's time zone settings so that we could adjust the time properly to local time as described in Section 5.1.

At the end of the subject search process, we were able to recruit nine subjects for our user study. The most important criterion for subject selection was that each subject was required to have a sizeable collection of digital photographs that spans at least one year. The participants' ages ranged from 18 to 32, with the highest representation in the 20s. Six subjects were male, and three were female. Seven of the photo collections exceeded 1,000 photographs, and the average collection size was 1,876 images. The average time span of the collections was 22 months. On average, each collection had photos from four countries, and 82% of the photos were taken in the United States.

### 5.3.2 Method and Results

The study was executed using the subjects' own personal collection of photographs. Each participant completed one task. In this task, we asked the participants at the beginning of the session to mentally recall three of their photographs. Then, we asked them to navigate to each of these photos in turn, applying constraints by manipulating the PhotoCompas user interface, as described in Section 5.2. For the user study, we removed the location context (country, place names as described in Section 5.1) from the interface. We also removed the event breakdown. The reason for the removals is that we felt the direct location/time context information would dominate over the others, especially as the collections span only a few years (i.e., not many photos from each location). Participants were thus faced with six available contextual categories: temperature, weather status, elevation, time of day, light status and season.

We logged the order in which subjects clicked on the different metadata categories in every trial. We focused on the first few clicks in each trial. Beyond that, the data was too spotty (often, the choice of category to click on was very limited after the first 2–3 clicks).

In addition, we used a questionnaire, parts of which were handed to the participants after they completed each search task, and another part upon completion of their session.

We first report on the actual category 'clicks' the participants applied in the different trials, looking for the different photos. Then we discuss the questionnaire.

The click count is summarized in Figure 5.3, showing how many times each category was clicked in the first or second click of each trial. By far, most user trials (12 out of a total of 27 trials) started by clicking on the season corresponding to the photograph they were looking for during that trial. The next most useful categories were light status and time of day, which we suspect were used interchangeably by the participants. Each of these categories was used within the first two clicks in 10 out of 27 trials. The weather conditions category proved useful as well, as it was clicked within the first two clicks 9 out of 27 times. More notably, participants preferred the weather conditions category to the temperature category; the latter was only used twice within the first two clicks.

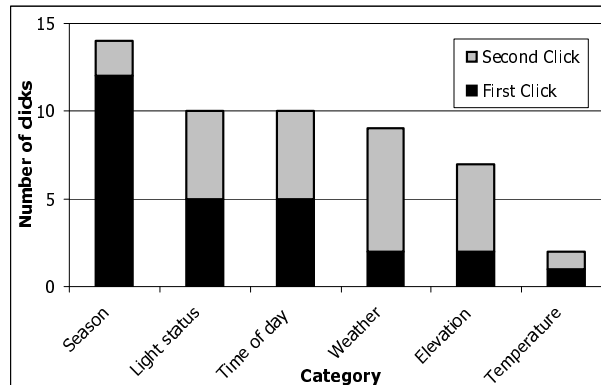


Figure 5.3: Usage of the different metadata categories within the first two clicks in each trial.

In the first part of the questionnaire, we asked the participants after each trial to rank the categories based on the importance of each available metadata facet for finding the specific photo in that specific trial. This part was intended to back the observations on click order, as well as correspond to the independent recall part of the survey. The results were consistent with the order of the clicks. Indeed, participants attested that time of day, season and light status (in this order) were effectively used much more frequently than weather, temperature and elevation.

In the second part of the questionnaire, the participants were asked to freely specify other contextual metadata categories that would have helped them look for that trial’s photo. Prominent answers to this open-ended question were number (or presence) of people in the photo (five participants listed this category, for 13 different photos); indoor/ outdoor classification (5, 7); type of activity (2, 2); and prominent colors (2, 2). This part, and its results, correspond to the independent recall part of the survey (see Section 5.4.2).

At the end of all trials, each participant ranked a group of 24 contextual metadata categories on how useful they would have been in retrieving each photo in the trials. This section of the questionnaire was administered at the end of the process, after all trials were executed, so that it would not bias the subjects when asked after each task, as described above, to freely specify other categories that would have been helpful. We used a 10 point Likert scale. For example, subjects were asked to rank how useful

the data regarding number of people in the image would have been when looking for the specific image.

This part of the questionnaire contrasts with the assisted recall rating part of the survey, where we asked the participants how well they *remember* the specific properties of the photo, instead of rating the *usefulness* of the data as we do here (see Section 5.4). We leave the detailed comparison to Section 5.4, but note that in this part of the questionnaire, users ranked the metadata that were available in the interface in a consistent manner to the measured click order and the first part of the questionnaire.

Finally, the questionnaire had an open ended question regarding the perceived overall usefulness of the contextual metadata that participants were working with in the user study. While five out of nine participants suggested at least some usefulness when browsing their current photo collection, all except one hypothesized that the context will be useful when a collection grows to span dozens of years.

## 5.4 Survey

Independent of the user study that we described in the previous section, we administered a survey to 35 participants. This survey allowed us to gather information about the usefulness of various contextual metadata from a broader number of people than was practical in the context of the user study. None of the experiment's subjects were also in the set of participants we surveyed. In particular, we sought in the survey to understand which metadata categories are *well remembered* about the photographs, rather than for being *useful in finding* a photo in a collection, as had been the focus in the user study and its questionnaire.

The survey consisted of two parts that respondents worked on in order. In the first part of the survey, the *independent recall*, we asked the participants to mentally recall any one to three of their personal photographs. Once they had a photo in mind, the survey had them describe the photo in prose “*in as much detail as possible, and with as much context about the surrounding environment and conditions as they could remember.*” These expositions usually amounted to about half a page of handwritten

text. Through this method we harvested a total of 83 descriptions from the 35 participants.

The open-ended nature of the survey’s photo descriptions provided an opportunity to discover the range of metadata that is at work in mental recall, independent of the facilities we provided in our computer implementation. To this end we coded all the prose descriptions, extracting which kind of metadata about the respective photo was being used in, or exposed by, each sentence of the descriptions. We then ranked these metadata categories by the number of times that respondents made use of them in their descriptions.

In the second half of the survey, the *assisted recall*, we asked participants to rate how well they remembered different categories of contextual metadata. These were the same categories that we had asked about in the user study questionnaire and included categories like “time of day”, “indoor/outdoor” and “other people at the same event (but not in the photo).”

### 5.4.1 Method and Results

We administered the survey in informal settings; participants’ collections were not nearby. Nor had we prompted participants to interact with their collections ahead of time. No computer interaction was involved: the responses were put down on paper. The survey also did not limit participants to digital photographs, but rather, encouraged them to recall *any* photo from their collection. Of the 83 photos, 39 were described by female respondents, 42 by men (and two unknown). One respondent was below 18 years of age. Out of our 35 respondents, fifteen were in the age range of 18-25, eight were 26-35, two were 36-45, and nine were 46-55 years of age.

We begin with the results of the assisted recall section of the survey, where we asked participants to rate how well they remembered different cues for each photo.

Figure 5.4 shows a radar graph that summarizes how subjects rated the 24 metadata categories. The radial axis corresponds to the Likert scale ratings. The categories are arranged around the circle. The further out a point is situated from the center of the circle the more highly our respondents rated the corresponding contextual



metadata category.

The displayed categories are listed below in counter-clockwise order, from **Outdindr** (on the right side) and on, following the curve.

- **Outindr** refers to whether the photo was taken outdoor or indoor.
- **Identpeopl** is the identity of the people who are captured in the photograph.
- **Location** is the location where the photo was taken
- **Event** — the event context in which the photo was taken.
- **Numpeopl** refers to how many persons are captured in the photograph.
- **Season** refers to the time of year the photo was taken.
- **Year** — the year in which the photo was taken.
- **Wherestor** — short for ‘where the photo is stored.’ Example values are “a shoebox”; “my laptop’s drive.”
- **Camera** — which camera the photographer had used.
- **Daylight** is the day/night/sunrise/sunset categorization as described in Section 5.1.
- **Timeofda** — the time of day as described in Section 5.1.
- **Otherpeo** refers to other people who were present at the time the photo was taken, but were not captured in the photo.
- **Weather** is the weather conditions at the time the photo was taken, as discussed in Section 5.1.
- **Colors** — prominent colors that appear in the visual image.
- **Month** — the exact month in which the photo was taken.
- **Clothes** is the clothes worn by people in the image.

- **Temptr** refers to the outside temperature at the time the photo was taken, as described in Section 5.1.
- **Flash** is whether or not a flash was used for the photo.
- **Yourmood**, of course, refers to the photographer’s mood at the time he snapped the picture.
- **Camorien** is the camera orientation (portrait or landscape).
- **Textinph** refers to text that is visible in the photo, such as a road sign, or the entrance sign to a park.
- **Elevation** is the elevation at the location in which the photo was taken.
- **Othercset** is ‘other camera settings,’ or camera settings that are not accounted for separately in the survey (other than whether the photographer used a flash, and whether the camera was held in portrait or landscape mode — these latter two categories were important enough to merit a separate category). The other camera settings included, for example, whether the photographer had deployed a zoom lens, whether a timer function was used, and observations about lighting conditions.
- **Date** refers to the exact calendar date on which the photo was taken.

We had collected the categories from prior pilot interviews. Indeed, the categories in Figure 5.4 cover all but two of the metadata categories we observed in the independent recall part of the survey; see Section 5.4.2.

For an overview, we show in Figure 5.4 results for four subgroupings of respondents. Figure 5.4(a) compares memory cue ratings by respondents in different age groups; Figure 5.4(b) demonstrates differences between genders. We first discuss general observations about the full set of survey responses, and then discuss the variance in the responses between the different groups.

We compared the differences among cue ratings for the consolidated mean over all groups in the graph. The results roughly partition the cues into groups: *outdoor/indoor* stands in a group of its own. It is statistically higher ranked than any



of the other cues. A second tier is formed by the cues *number of people*, *identity of people*, *location* and *event*. These cues are ranked higher than the next cues in a statistically significant manner, with the exception of the *where stored* cue (all significant differences reported are  $p < 0.05$  unless otherwise noted). Moving further counter-clockwise around the graph, *season* through *weather* form a roughly equivalent group. While *mood* is remembered better than the following cues, importance of cues beyond *mood* drops off quickly.

The low values for *date* for all groups confirms the findings of Wagenaar [93]. That study of recalling events also explored the relative importance of different memory cues and concluded that the most important cues are ‘who,’ ‘where,’ and ‘when,’ in that order. However, notice that finer-grained distinctions need to be made than a simple ‘when.’ The *time of day* ranks significantly higher in our data than the more general *date*, and is also a significantly stronger cue than the *month*. In these results at least, the time of day draws even with the memory of which year a photo was taken. This finding points to the necessity of ensuring accurate capture of the *local time* when a photo is shot, as described in Section 5.1.

After *indoor/outdoor*, the four cues that involve people and location are rated highest on average by the respondents. This result again confirms the findings of Wagenaar. The result also reinforces the opportunity that the geo-referencing of photographs presents for the automation of organizing photo collections. Also, our results clearly call for the integration into photo browsers of the category referring to the number of people in a photo. Face detection technology is not perfect, but current systems can be tuned to a very low rate of false positives — and maybe automatically categorizing photos by the “lower bound” of the number of people in each. At the same time, our data ranks the indoor/outdoor distinction as important enough to warrant research investment and/or implementation in photo browser applications.

Looking at the graph as a whole we see that most findings are consistent across age groups and gender. However, notice that in most cases the oldest group of respondents ranked memory cues lower than their cohorts, even if relative tendencies match. This difference between the age groups is most noticeable for the *year* ( $p < 0.01$ ) and *clothes* (not significant) cues. The difference in the *year* ranking might be explained

by the fact that older respondents have described older photos, thereby losing this cue as a greatly distinguishing feature. In contrast, notice that there is no significant drop in how well the *time of day* is remembered by older people, supporting our earlier claim regarding the importance of this type of time context.

### 5.4.2 Independent Recall Descriptions

We analyzed the textual descriptions of photographs, given by participants in the independent recall part of the survey, as described on page 93. Our goals were to see which are the prominent categories when participants freely recall photographs, and whether the findings match the assisted recall findings. The latter question is discussed in Section 5.4.3.

An interesting question that we were not able to answer with this survey, is whether bias is introduced by the fact that the prose description of the photo is written for a reader. Possibly, the person describing the photo listed facts and context information they thought would be interesting for a reader, and not necessarily all the facts about the photos. In the survey form, however, we attempted to phrase the request for the photo description very generally, so that the description will be as complete as possible (the text of the survey question appears on page 93).

As we hinted in the beginning of this chapter, it is sometimes the case that context categories overlap with the content or ‘subject’ of the image. Our earlier example refers to the daylight status; photos taken around sunset (context) tend to be “sunset” photos (content). Even more significant is the category representing the identity of people in the image, which always explicitly refers to the content or even the subject of the photos. It is not clear what effect the content vs. context dichotomy has on the photo descriptions. For example, some content information may be left out from photo descriptions as our respondents found it redundant (e.g., “the sky color was blue”). On the other hand, other content categories may have been described only because they appear in the image, and would not have been listed as important otherwise. We believe that the order in which categories appear will give us supporting data as to which categories people recall, and think is important.

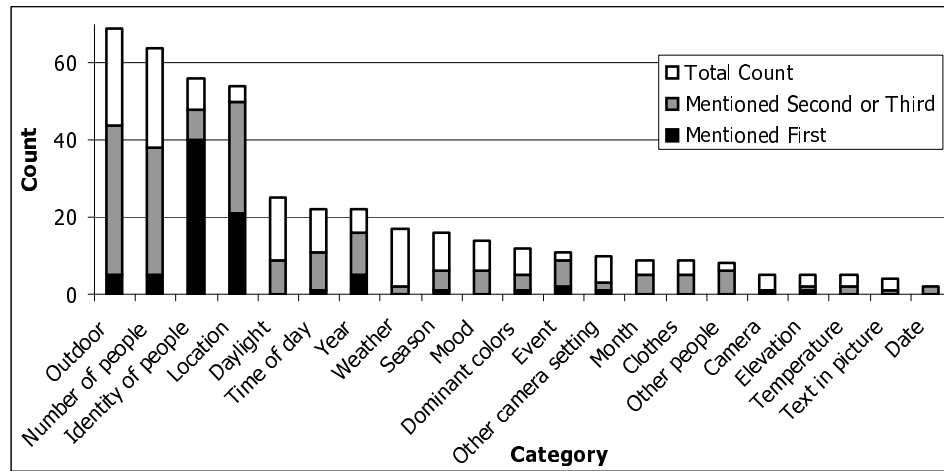


Figure 5.5: Number of times each metadata category was revealed in photo descriptions.

Figure 5.5 shows the number of times each category was revealed in the photo descriptions. The bar for each category is divided to show how many times the category was listed first in a description, how many times it appeared within the first three categories, and how many times in total the category was coded from a description. For instance, the figure shows that the most popular categories that were revealed by the descriptions were indoor/outdoor (69 out of 83 photo descriptions touched on this category), the number and identity of people (64 and 56 descriptions, respectively) and the location (54). We also show the *order* in which the categories were mentioned. For example, the identity of people in the photo was revealed first by the description in 40 of the 56 descriptions that touched on this category. In 48 photo descriptions, the identity of people was featured within the first three categories. We can see that the top categories were also, by far, the most frequently mentioned within the first three categories for each photo. Overwhelmingly, the identity of people in the photo was the first category to be mentioned (40 out of 83 photos), suggesting the importance of identity when managing personal photo collections.

The daylight, time of day, and weather, some of the categories we automatically include in our browser prototype, played an important role in the descriptions (25, 22 and 17 mentions, respectively). The weather descriptions were sometimes quite

specific: “really nice, sunny, blue sky, not much of a wind, and it was warm, but I wouldn’t say it was hot.” Sometimes the descriptions of the weather were somewhat less precise: “the sky is clear and blue,” “a winter storm.” Another category we used, season, was mentioned 16 times — almost as much as the year (22) and a lot more than the month (9) when the photo was taken. Descriptions that fall into the event category appeared 11 times.

Similarly, emotions that are associated with photos populated the photo descriptions (14 appearances). For example, one respondent described the image of cows tightly packed and readied for slaughter as “unbelievably terrible.” On the other end of the spectrum an excited respondent described the photo of her very young son with “he was standing!” Clearly, the emotional state of a photographer is inaccessible to the camera. Nevertheless, this evidence of strong emotional associations with photographs does suggest features for photo browsers such as easy methods of associating emotions with photos in browsers (for example, a browser might allow a user to quickly drag icons that denote emotions onto an image to facilitate subsequent retrieval by this facet).

The detailed analysis of descriptions did, however, suggest one metadata facet that *is* accessible to image analytic methods today: color. The descriptions contained numerous mentions of this facet: “bluish-green ocean,” “mostly black... also subtle shade of golden yellow,” “a nice green spot [of grass where a photo’s subject was standing].” or “whitish-pink [the shot of someone’s belly].” Integration of color-based querying systems [5] in photo browsers may be useful.

Finally, we looked at whether the independent recall resulted in cue categories that did not appear in the set of categories we assembled for the assisted recall part. Our category set indeed covered the majority of explicitly mentioned cues. The major category that we had not included in our set, and that was often evoked in the photo descriptions were *objects* (27 appearances). Examples of objects that appeared in the text were “a camping van,” “stuffed animals,” “a mansion,” and “the world’s largest lobster.” Clearly, this category is very difficult to detect automatically with today’s technology. Another category, which we denote *activities* (such as hiking, camping, cycling and so forth) was mentioned in 7 photo descriptions.

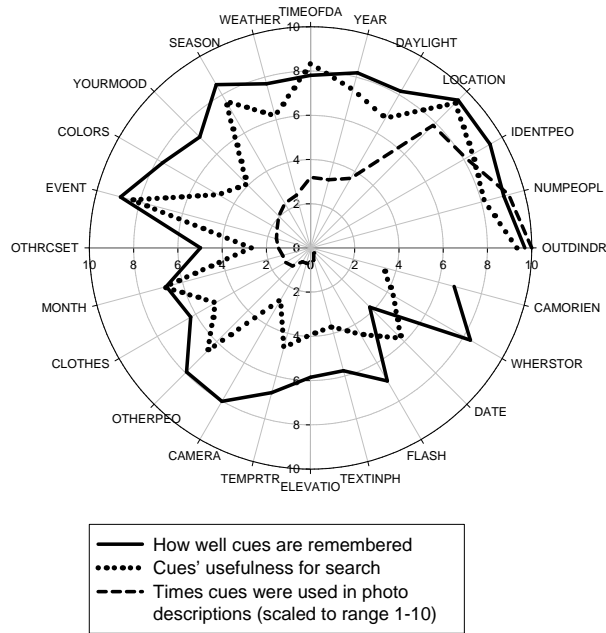


Figure 5.6: Comparing perceived importance of search cues, recall cues, and cues used in photo descriptions.

### 5.4.3 Recalled Cues vs. Useful Cues

As described above, in the user study's questionnaire we presented the subjects with the same categories of metadata as the assisted recall part of the survey (and that are shown in Figures 5.4 and 5.5). In the study's context subjects rated the cues by how *useful* they had been for finding the photo they had chosen to search for on the computer. In the context of the survey, participants had evaluated the cues for how well they *remembered* them for a photo. Is there a difference between how the cues rank for (i) remembering and mentally visualizing a photo, (ii) usefulness for computer-assisted search, and (iii) describing photos in the open ended independent recall part of the survey? Figure 5.6 provides answers to these questions.

The solid line shows how survey participants ranked the 24 cues for how well they are remembered for each photograph. This line is the aggregate of Figure 5.4, although note that the order of categories is not the same. The dotted line plots the rankings that user study participants assigned to the cues (how useful each cue

was in searching). The dashed line, finally, traces the number of times that survey subjects revealed the respective cue in the photo descriptions of their survey form. These latter values are scaled to match the 0–10 dimension of the graph; the graph categories are sorted in descending order based on the number of description mentions. The quickly diminishing appearance of this line arises since some cues were never used by survey participants when describing their photographs. For example, look at the *camera* category at the bottom left side of the figure. The ‘camera used’ was well remembered (solid line), but not perceived as useful (dotted line), and was mentioned very rarely in the textual descriptions (dashed line).

As Figure 5.4 shows, outdoor/indoor, location, and the identity and number of subjects in a photo are clearly top choices both for usefulness in searching and remembering; accordingly, they are the top categories that were mentioned in the descriptions. The day/night status when the photo was taken is close behind.

A few noteworthy differences between subjects’ evaluation of cue importance for searching vs. remembering are *where stored*, *camera* and *colors*. According to the figure, these cues are much better remembered than they are useful for search (also, they were mentioned very few times in the open descriptions). A simple explanation for the difference in the case of the former two cues, is that these cues do not filter much. For example, most photos were taken by one camera, so the originating camera is well remembered but not very useful. On the other hand, a possible explanation to why the *colors* cue is perceived as un-useful for search, is that the users could not imagine the right user interface or language to specify a color query.

Conversely, there is only one category that participants in the study rated significantly higher (as useful for search) than the survey’s participants (as well-remembered). That category is *date*. As discussed above, indeed, people do not remember dates or exact time periods well. However, in the case where users do remember the photo’s date, it is very easy to help them find the photo, as all photos are time-stamped and thus can be easily presented and retrieved using a calendar-based metaphor.

## 5.5 Conclusions and Future Work

The user study and survey we conducted suggest that contextual cues are useful for browsing and managing personal collections of photos.

In particular, location cues prove to be as important as cues involving people in the images. Context involving location and people was rated high in usefulness for retrieval, and subjects claimed they also remembered such context cues well. In addition, these categories were often listed independently by participants describing their photographs.

Location- and time-derived context such as weather, local time, daylight status and season, proved to be useful as well. Participants interacted with these categories comfortably in a user interface, used them in photo descriptions, and found them useful for retrieval on top of being well-remembered. More specifically, local time and daylight status seem to be stronger cues than the absolute date or exact time of the photo; as collections span many years, the former cues may even be more important for retrieval than the actual year when the photo was taken.

This study will serve as a guide towards our, and hopefully others', future effort in automatically integrating contextual metadata into photo browsers. Our ranking of recall values of cues can help decide where research efforts should most profitably be invested: we need to move along the curves of Figure 5.6 in order to make photo collections increasingly accessible.

## Chapter 6

# Resolving Identity in Photo Collections

In personal photo albums, no organizational category is more important than the identity of people in the photo. The need to retrieve photos by identity of people is very common; such retrieval is also cognitively highly effective. Our previous study (Chapter 5) showed that identity of people is one of the most important cues for users, and also one of the best-remembered features when people are recalling photos from their collection. Our study verified results from W.H. Wagenaar [93]. In his study of autobiographical memory, Wagenaar has shown that identity of people participating at some past event is well remembered when recalling the specific event.

As a consequence, an important goal of photo organizing systems is to support retrieval by identity of people that appear in photos. Such a system would allow users to query for photos containing specific person(s). An ideal system would accurately retrieve all the photos where the queried person appears, and only these photos. At the same time, such system will ideally require the collection owner to invest minimal effort in annotating or organizing the collection prior to the query.

However, current technology is far from this ideal scenario. Image analysis techniques for face detection [39, 96] and recognition [100] are far from supporting reliable person-based retrieval, even when the set of people that appear in the photos is extremely limited. An obstacle that often arises in family albums is that faces are not





Figure 6.1: Possible interaction mode with the system — the user annotates the identities that appear in the photo by picking from a short list of candidates.

directly aligned with the camera. Most faces are tilted or slanted, or even partially or totally obscured (in one of our sample collections, about 27% of the faces were significantly tilted up or down; roughly 70% of the faces of people in the photos were slanted, a third of those were shown in profile). These facts make recognition — and even detection — an extremely difficult task. As the researchers in [32] report, “The face detector used in our photo application is highly accurate for photos with faces *where both eyes and the nose are visible.*”

Since 1999, research efforts [49, 80, 94] have focused on systems that ease the manual annotation of identities in photos (see details in Chapter 2, Related Work). These systems, adopted since then in commercial products, suffer from two major shortfalls.

- Long-List annotation. To annotate a photo (or group of photos) the user is required to choose from the list of all names in his collection, which makes the process tedious. Many users resort to annotation of at most a few faces.
- Limited retrieval. As retrieval depends on annotation, The users must manually annotate their entire collection with the identity of people in the photos. Photos that are not annotated manually cannot be retrieved.

Short of solving the difficult image analysis problems, our system can contribute to alleviate both shortfalls. Our approach is to leverage context available from photo metadata and user input. Our system tries to predict which people are likely to appear in the photo, leveraging photo metadata, context (such as event and location, which our algorithms of Chapter 3 can identify automatically) and previous annotations.

The system’s predictions are expressed in terms of a likelihood score for each person to appear in each photo that is not yet fully annotated. This score can be used to generate a small set of candidates that are likely to appear in each photo. The candidate set can be passed to a face recognition module, hopefully improving face recognition results by limiting the number of candidates. Alternatively, the likelihood score can be used when retrieving non-annotated images from the collection, whether or not we integrate image recognition capabilities. In this work, we use the likelihood scores in the framework of user interaction. We improve the quality of candidate name lists that are presented to the user for each photo during the annotation process by reducing the long list of candidates to a reasonable size (e.g., 5 names). This reduction allows for more rapid user input.

An example user interaction with such a candidate list is shown in Figure 6.1. In the figure, the user is trying to annotate the photograph. Our system computes and displays a list of likely candidates. The user can choose a candidate from the list, or enter a name that is not on the list. This work is not concerned with the *details* of the interaction. For example, users may want to annotate a number of photos with a single name at the same time. Our system can support interfaces with such features, but we focus here on the prediction algorithms and therefore keep the interaction details rudimentary.

We use the following terminology in this chapter. The term *metadata*, in the domain of photographs, refers to all the information about the photograph that is not reflected in the actual visual image. It is convenient to make a distinction between two types of metadata: user entered metadata (or *annotation*), and automatically captured metadata (which we will simply call ‘metadata’). Annotation may include, for example, the identities of people in the photo, a textual caption, or a user-entered identifier of the location. Metadata, for example, may include the timestamp when

the photo was taken, or even the location coordinates where the photo was taken.

Our idea is simple: in a personal photo collection, people do not appear with uniform frequency. For example, there is a correlation between appearance of different people. There are also patterns with which people appear at certain times and locations. We harvest the emerging patterns to generate a progressively improving list of candidate identities for each photo that is about to be annotated.

We use the following intuitive guidelines:

- Popularity. Some people appear more often than others.
- Co-occurrence. People that appear in the same photos may be associated with each other, and have a higher likelihood of appearing together in other photos. We expand the association notion to the context of an “event”, a set of photos taken at the same time, with similar context (for example a party, or a trip). People that appear together in the same events are likely to appear together in other events, even if they are never captured in the same picture.
- Temporal re-occurrence. Within a specific event, there tend to be multiple photos of the same person.<sup>1</sup>
- Spatial re-occurrence. People that appear in a certain location have an elevated likelihood of appearing again in that same location, even during different events.

Utilizing these intuitions requires various levels of metadata. Much of this metadata can be acquired automatically. If no such metadata is available (e.g., for scanned photos), we can only utilize the popularity and co-occurrence in photos, both of which emerge from partial user annotation. To consider co-occurrence in events or temporal re-occurrence, we require the photos to have a timestamp (so that the system can automatically compute likely events), or the user to annotate events. Spatial re-occurrence requires, naturally, location data which can be captured automatically by the camera, or annotated by the user.

Note that we are not using any recognition technology or any other type of image analysis at this point. This allows us to demonstrate the usefulness of our system as

---

<sup>1</sup>Moreover, it is likely that the person will be wearing the same clothes, which may help recognition [98]; we are not looking at clothes or face recognition in this current work.

a context-based annotation tool, independent of recognition techniques. In addition, our technique is robust with respect to issues that commonly arise in systems that are based on image analysis: faces that are tilted, slanted, and partially or even completely obscured. Having said that, in the future we will attempt to combine our system with a good recognition system, so we can enjoy the best of both worlds. We discuss a few possible directions for future work in Section 6.6.

The rest of this chapter, before we get to the conclusions, is organized as follows. In Section 6.2 we lay the foundation for our algorithm by describing our formal model for photos, identities and annotation. We also describe the model and formulation of the system and its parameters. Section 6.3 describes the core algorithm: how the system generates label suggestions, or assigns a score for each identity to appear in each photo based on context and past annotation. In Section 6.4 we set up the evaluation, and outline a model that helps us emulate the process in which users label their photos. Section 6.5 reports on the results of running the evaluation on our test collections. We start, though, with a survey of related work.

## 6.1 Related Work

The research of Zhang et al. [98] and Girgensohn et al. [32] focuses on using face recognition algorithms to ease the task of annotating identities in photo collections. In [98], the researchers address the problem of Long-List annotation described above. Their system ranks likely candidates for a *specific* identity in each photo. The system presents the candidates to the user as she is trying to annotate that identity, in a similar fashion to the interaction style described in this chapter. The ranking in [98] is based on face similarity, using a nearest-neighbor-based learning algorithm. In later work [99] the same researchers applied similar image analysis techniques to annotate multiple photos simultaneously.

The system described by Girgensohn [32] takes a different approach for the annotation process. In this system, after several instances of a person are annotated by the user, they are presented with face-only thumbnails that are likely to be of the same person. The thumbnails are chosen based on nearest-neighbor visual similarity.

The user can quickly annotate the correct thumbnails in the set with the name of the person.

The work of Kuchinsky et al. [49] predates both projects mentioned above. In their paper, the authors describe a semi-automated face recognition system. The application suggests a likely identity for un-annotated faces in a photo when the photo is viewed. The annotation is done one photo at a time, and the suggested list is comprised of one name only. The system uses recognition techniques only, and does not utilize context information. The authors do not report on results and accuracy of the annotation suggestions in their system.

All of these systems are orthogonal to our approach and can be enhanced using the context-based techniques we use in this work. Their evaluation results, though, are hard to compare with ours. One reason is that both algorithms, unlike our system, depend on face detection. As a result, the faces they attempt to annotate must be clear and well-aligned. In contrast, our system does not depend on detection or appearance of the face in the photo. Moreover, it is hard to compare systems while using different datasets.

A system that *does* utilize context was proposed by Davis et al. [18, 77]. As we report in Chapter 2, the researchers utilize spatial and temporal context to help annotation of photographs taken with camera-equipped mobile phones. While their approach to identity labeling is similar to ours, they have not yet implemented and reported on an identity-based annotation system.

Without analyzing content *or* context, many projects studied efficient labeling of photos, including annotation of identities that appear in them. This topic has been an active research field since 1999 [49, 80, 95, 94]; refer to Chapter 2 for details.

In the slightly different domain of news photographs, Berg et al. [10] attempted to match names appearing in photo captions to faces that appear in the actual image. Their approach requires integrating uncertain information about the context with face recognition and detection techniques. While the context used in our work is derived from location and time, in their case the context is derived from the free-text caption.

## 6.2 Model

In this section we formally describe the model for photos, identities and annotation, which is the portion of the system that is exposed to users. We also describe concepts such as events and locations, which are not necessarily exposed to the users, but are computed and used by the system. We do not list the model for user behavior here, but rather leave that for the evaluation (Section 6.4).

The basic constructs of our model are the set of photos,  $S$ , and the set of people that appear in the photos,  $I$  — for *Identities*, represented by person names.<sup>2</sup> While the set  $I$  may include every single person that appears in the photo collection, it may instead be defined as the set of identities the users are interested in, e.g., only their family and friends. We try various options for  $I$  in our evaluation. We should note that as the model assumes users will not annotate identities not in  $I$ , the set of photos  $S$  we consider is simply the set of photos that contain at least one identity from the set  $I$ .

### 6.2.1 Interaction Model

The user interaction can best be described on the basis of a formal model, which we introduce in this section. Each photo  $s \in S$  contains a certain set of people. This set of identities, represented by  $I_s \subseteq I$ , is the ground-truth list of identities that appear in the photo.

The process of annotation can be seen as entering into the system the ground-truth knowledge about the identities in each photo. More formally, we define  $K_s \subseteq I_s$  as the set of people in photo  $s$  that are known to the system. An annotation step occurs when the user enters the knowledge about one identity in  $s$  — adds  $i \in I_s$  to  $K_s$ .

Since the interaction between the user and the system occurs in discrete steps, it makes sense to talk about  $K_s(t)$  — the set of identities in photo  $s$  that is known to the system at time  $t$ . Nevertheless, for simplicity of exposition, when it is clear from

---

<sup>2</sup>In previous chapters we used  $p$  to refer to a photograph; in this chapter, to avoid confusion between photos, persons, and probabilities, we use  $s$  and  $i$  for photos and identities, respectively.

context we just use  $K_s$  to represent the system knowledge at a given time.

The full set of annotations already entered by the user is represented by the set (of sets)  $K = K(t) = \{K_s(t), \forall s \in S\}$  where zero or more identities  $i \in K_s \subseteq I_s$  are known at time  $t$  for each photo  $s$ .

We now describe the model for the annotation interaction between the user and the system. The annotation process takes place in steps, during which a single identity in a single photo is annotated. At each step, the system considers photo  $s$  for which  $K_s \subset I_s$ , and tries to help the user annotate one identity from  $I_s - K_s$ . This is how the system advances from time  $t$  to time  $t + 1$ :

1. The system suggests a *short* list  $H_s(t)$  of  $h$  possible identities in  $s$  to the user. The suggested list  $H_s$  is time-dependent — it is generated based on knowledge in  $K(t)$  and therefore subject to change due to any additions to  $K$ . Of course,  $H_s(t) \cap K_s(t) = \emptyset$  as there is no sense for the system to suggest names that are already known to appear in the photo. The system's goal is that  $H_s(t) \cap I_s \neq \emptyset$  ( $H_s$  correctly listed an identity that appears in the photo.) We call the case when the two sets overlap a *hit* (or *h-hit* following the notation of [98], corresponding to the candidate list length  $h$ ). If there is no overlap, we call it a *miss*.
2. As feedback, the user annotates the identity of one person  $i \in I_s - K_s$ . Note that the suggested annotation is not for a specific person in the photo. The user can reveal *any* identity in  $I_s$  to the system at each iteration, whether we had a hit or a miss. In case of a hit,  $i$  is selected from  $H_s(t) \cap I_s$ . In case of a miss,  $i$  is picked from all the un-annotated identities  $I_s - K_s$ .
3. The knowledge about  $i$  is added to  $K_s$ . The system advances to time  $t + 1$ , and the new knowledge can be used when the system generates the next  $H_s(t + 1)$  in trying to identify another identity in the same photo  $s$ , or when annotating a new photo.

For example, say photo  $s$  has two people in it,  $I_s = \{Dylan, Alex\}$ , neither of which has been identified in an annotation ( $K_s = \emptyset$ ). The photo is then picked for annotation at time  $t_1$ . The system suggests a list of candidates  $H_s(t_1)$  that might

appear in this photo, for example  $\{Neil, Marvin, Alex\}$ . The user acknowledges that *Alex* is indeed in the photo ( $Alex \in I_s$ ) — a *hit*. The system adds *Alex* to  $K_s$ . When photo  $s$  is picked for annotation again at a later time  $t_2$ , the system may suggest a new list  $H_s(t_2)$ , based on the fact that *Alex* is known to appear in photo  $s$ , and other knowledge that may have been accumulated in  $K$  between  $t_1$  and  $t_2$ . Say  $H_s(t_2) = \{Polly, James, Neil\}$ . This new  $H_s$  does not include the name *Dylan* — a *miss*.

### 6.2.2 System and Parameter Model

In the previous section, we listed the parameters and definitions pertaining to the annotation process and its output. In this section we define the model for the underlying system, its structure and parameters. First, we introduce the raw metadata that is associated with the photos. We then briefly remind the reader the characteristics of the high-level structure of the collection, as derived by the algorithms of our PhotoCompass system (Chapter 3).

Like previous chapters, the metadata we assume to be associated with each photo  $s$  includes time and location:

- $t(s)$  = The time when the photo was taken.
- $g(s)$  = The geographic location coordinates where the photo was taken.

It is important to note that large portions of the results in this chapter do not *require* location metadata, and are valid even when location data is unavailable. We investigate this issue in Section 6.5.

Given time and location metadata, our PhotoCompass system can *automatically* organize a photo collection into two hierarchies: a hierarchy of time-based events, and a location hierarchy. While the details of how this organization is achieved are found in Chapter 3, we briefly scan here the relevant details of the generated hierarchies.

A sample location hierarchy is shown at the top of Figure 4.3 on page 61. The location hierarchy's highest level is the country level. Below the country level, we refrain from using administrative divisions such as states or provinces. Instead, the next level of the hierarchy represents location clusters that are unique to the specific



collection, such as the “Seattle” cluster in the figure. Sometimes, when clusters are overloaded with photos, these location clusters are broken down further. For example, in the collection represented in Figure 4.3, the San Francisco area cluster represented many different photos taken at different events, and was therefore split into finer locations.

Events can be thought of as consecutive photos that were taken in the same context, e.g., a party or a trip. PhotoCompas automatically detects events in photo collections. The event hierarchy is, in this implementation, flat: the system creates a list of consecutive events as shown at the bottom of Figure 4.3.<sup>3</sup> Although we use location metadata as a clue for detecting events in the photo collection, it is not absolutely required to use location for this purpose. Time metadata is sufficient ([17, 27, 71] and more) but we have shown in Chapter 3 that knowledge of location adds accuracy to event detection.

Every photo in the collection belongs to exactly one location leaf node, and one event. Therefore, we can define the following notation for location- and event-based sets of photographs:

- $L(s)$  = The set of photos belonging to the location leaf node that contains photo  $s$ .
- $E(s)$  = The set of photos taken at the event that contains photo  $s$ .

In addition to the generated hierarchy based on time and location, we explore the notion of “neighboring” photos, both in time and in space. In other words:

- $N_{loc}(s)$  = The set of photos taken within some fixed physical distance  $R$  from photo  $s$ . The set is defined as:  $N_{loc}(s) = \{q \mid distance(g(s), g(q)) < R\}$ .
- $N_{time}(s)$  = The set of photos taken within  $T$  seconds from photo  $s$ .  $N_{time}(s) = \{q \mid |t(s) - t(q)| < T\}$ .

For annotation purposes, one of the questions we try to answer is whether the notion of neighboring photos can supplement and enhance, or maybe even replace, the more high-level event and location hierarchies. This is one of the issues we studied in our evaluation (Section 6.4).

---

<sup>3</sup>One can also use the notion of sub-events, and an event hierarchy (see [34]).

## 6.3 Generating Label Suggestions

This section outlines our approaches to generating annotation suggestions. Using the notation from Section 6.2.1, this section explains how the system generates the candidate list  $H_s$  for a photo  $s$ .

We remind the reader, once more, that no face detection or face recognition is used by our system. Instead, the candidate list is generated using clues from already-annotated identities (the set  $K$ ), and the photos' metadata.

We employ various estimators, each applied to some dimension of the data, and each generating a ranked candidate list. To rank the candidates, an estimator assigns to each person in  $K$  a prior probability with which that person is likely to appear in  $s$ . The top  $h$  candidates ranked by the estimator become that estimator's list of candidates  $H_s$ .

In the next subsection we outline a few basic estimators that we implemented in our system. Later, we expand on another type of estimator we used, the PeopleRank. At the end of this section, we discuss some ways of combining results from different estimators.

### 6.3.1 Basic Estimators

We introduce the idea behind the basic estimators with a specific example. The Event Estimator generates a ranked candidate list for photo  $s$  based on identities that already appeared in  $E(s)$ , the set of photos from the event that contains photo  $s$ . The prior probability that is assigned to each person  $i$  by the Event Estimator is simply the percentage of appearances of  $i$  within all photos  $q \in E(s)$ . More precisely,  $p(i, s) = \frac{\sum_{q \in E(s)} K_q(i)}{|E(s)|}$ , where

$$K_q(i) = \begin{cases} 1 & \text{if } i \in K_q \\ 0 & \text{otherwise} \end{cases} \quad (6.1)$$

For example, say three identities have been annotated in a total of two photos  $q_1, q_2 \in E(s)$ . Further assume  $K_{q_1} = \{Kimya, Dylan\}$  and  $K_{q_2} = \{Dylan\}$ . Note

that in practice (and in this example) we do not consider photos in  $E(s)$  that are not yet annotated, as those photos will simply change the  $p$  values for all  $i$  by a constant factor, which will not effect the ranking. The Event Estimator will then give Dylan a likelihood score of  $\frac{2}{2} = 1$ , and Kimya a likelihood score of  $\frac{1}{2} = 0.5$  to appear in picture  $s$ .

We now generalize this computation, which is the basis for all basic estimators. When generating a candidate list for some photo  $s$ , each estimator considers some *semantically connected* set of photos  $Q(s)$ , and their annotated identities. The prior probability for each person  $i$  to appear in  $s$  is computed by the formula

$$p(i, s) = \frac{\sum_{q \in Q(s)} K_q(i)}{|Q(s)|} \quad (6.2)$$

which produces a prior estimation of the frequency in which person  $i$  appears in the set  $Q(s)$ . Table 6.1 summarizes the basic estimators and the semantic group that represents each, in respect to photo  $s$ .

Table 6.1: The basic estimators and the set of photos each estimator considers when ranking candidates to appear in photo  $s$ .

Estimator	$Q(s) = ?$
Event	$E(s)$
Location	$L(s)$
Neighboring	$N_{loc}(s)$
Time-neighboring	$N_{time}(s)$
Global	$S$ (all photos)

To give another example, the Global Estimator (bottom line of Table 6.1) simply assigns a likelihood score for each person according to the frequency in which the person has appeared in the entire photo collection.

If reliable face detection were available, we could make the computation somewhat more accurate. For example, assume that for some photo  $q_1$  the system knows that the user annotated all the people in the photo, say  $K_{q_1} = I_{q_1} = \{Nick, Dylan\}$ . For another photo  $q_2$  the system knows  $K_{q_2} = \{Kimya\}$ , but assume the system also knows there is some other, still not annotated, person in that photo. If our set of

photos to consider is  $Q(s) = \{q_1, q_2, s\}$ , is Nick more likely to appear in  $s$  than Kimya? According to our model, Nick is not more likely (both have a score of  $\frac{1}{3}$ ). But while we know Kimya does not appear in  $q_1$ , Nick may or may not appear in  $q_2$  – making him more likely to appear in photo  $s$ . We can take this new factor into account by computing the score using  $p(i, s) = \frac{\sum_{q \in Q(s)} p(q, s)}{|Q(s)|}$ . Note that this formula is slightly modified from Eq. 6.2, in that we sum over the probabilities rather than  $K_q$ . More importantly, this formula is recursive but can be either simplified or computed iteratively until a fixed point is reached. However, the problem with this approach is that the system needs to know in advance the *number* of people that appear in each photo in the set, an additional requirement that is not necessarily available without reliable face detection.

These basic estimators address the temporal re-occurrence, spatial re-occurrence, and popularity guidelines mentioned above. Using the estimator framework will allow simple extensibility of the system to support other types of context, or other sources of information, as  $Q(s)$  can be any semantic grouping of photos. For example, a new estimator can be added that is based on the time of day and weekday in which photos were taken. If a photo  $s$  was taken on a weekend afternoon, a different set of people is likely to be present in the  $s$  than in a photo  $s'$  taken on a weekday morning. The set  $Q(s)$  in this case will be all the photos taken on that weekday and time of day, and the computation will be done similarly to the other estimators listed above.

Other possible estimators, which are not yet considered in our work, are listed below. For each estimator we mention the semantic group of photos  $Q(s)$  that the estimator is based on.

- Recurrence Estimator:  $Q(s)$  is the set of photos taken on or around the same date in different years (e.g., a birthday). Alternatively,  $Q(s)$  can represent weekly or monthly recurrence.
- Time of Day Estimator: are some people more likely than others (as suggested by [18]) to appear in a photo if it is taken on a weekday morning, vs. a weekend afternoon?
- Tag-based Estimator:  $Q(s)$  represents all the photos tagged by the user with the

same (non-identity) tag (in a photo browser application). For example, photos that are tagged “favorites” — are they likely to be photos of certain individuals (maybe pictures of the kids)? The assumption is that photos containing the same individual may be tagged similarly.

- Email Activity Estimator:  $Q(s)$  represents all the photos sent to the same email addressee as  $s$  (or shared with the same group of people, or discussed via Instant Messaging with the same person, or sent to a mobile phone, etc.). Similar to our *PeopleRank* estimators (see Section 6.3.2), this estimator presumes that certain social interactions are captured by the habits of communication with photos. For example, if I share a photo with my family, the identities in the photo are likely to be family members as well.
- Text-based Estimator:  $Q(s)$  is all the photos whose descriptions (be it in a caption, an email message, a voice-recorded annotation, a phone text-message etc.) include similar words to those appearing in  $s$ . For example, imagine  $s$  contains the word ‘kids’ in its description; other photos whose caption contained the word ‘kids’ were photos of Nick or Robyn. Thus,  $s$  is likely to be a photo of Nick or Robyn as well.

In addition to these estimators, we can use different sources for likelihood scores and incorporate them into our Estimator framework, despite the fact that they were generated in a different fashion. For example,

- Face recognition. The score for each person is the match of the face in the current photo to the already-known faces of the same person. While earlier we hinted at a different way to combine our methods with face recognition, this alternative method is also possible.
- Voice recognition. The score for each person is the match between the person’s name and the voice recognition transcript of a voice-recorded photo tag. For example, the tag may include a word that sounds like ‘Kimya’.

We describe later how all these estimators and score-generators can be combined

into a single score, which is the final step for our ranking. But first, we introduce the *co-occurrence estimators*.

### 6.3.2 Estimating Co-occurrence: PeopleRank

Our PeopleRank estimators aim to harvest the relationships between people. Such relationships naturally exist in personal photo collections due to the human nature of interaction within social networks. In our system the relationships emerge from the annotation of photos. For example, the system may learn that Kimya often appears in photos where Nick appears, or that Dylan often appears together with both of them.

To illustrate, Figure 6.2 shows the connections between people that appear in the collection of the author of this thesis (which is also our *A* test collection in Section 6.4). The visualization was generated using the Prefuse open source visualization package.<sup>4</sup> The nodes in the graph represent identities of people in the photos (as they appear in the ground-truth annotation of the collection). The edges connect each pair of nodes (people) that appeared together in at least one automatically-detected event. The darker node (“Sharon”, at the top-left region of the figure) represents the person currently in focus; all the nodes that are connected to it directly are also colored differently (slightly darker than the rest). That highlighted node represents the cousin of this work’s author. She is connected directly only to her side of the family tree, and including one close family friend. In general, the patterns that emerge from Figure 6.2 represent the different social circles of this thesis’ author quite accurately. This incidental exploration affirms our intuition that social patterns can be leveraged to support identity annotation.

Removed from Figure 6.2 are the nodes of two most popular people in the collection, i.e., the author himself and his girlfriend. They were removed to make the visualization clearer, since the two nodes that represent these people were connected to almost all the other nodes in the graph. This fact illustrates the type of social network that is captured by the annotations in a photo collection: it is *an approximate*

---

<sup>4</sup><http://prefuse.sourceforge.net/>

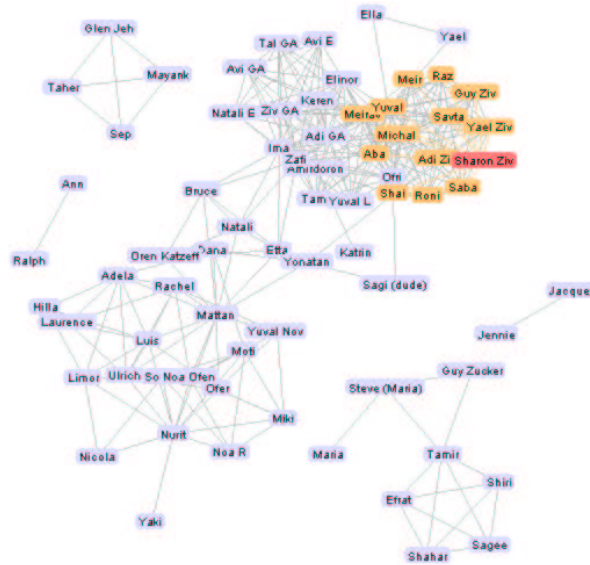


Figure 6.2: Relations between people in test collection  $A$ .

*view on an ego-centric network with alter connections.* In plain words, the social network generated by the annotation is derived from the point of view of a single person (or maybe a family), hence “ego-centric.” In addition to the links from the centric node to others, we can derive information about relationship amongst others (hence “alter connections”). However, all our information is derived from photo annotation, and may therefore be incomplete (i.e., some connections between alters that exist in reality, are not represented) or wrong (i.e., some connections are deduced from annotation that do not exist in the true social network). Hence the use of “approximate view” in the network description. For more details about social networks and on line social networks see, for example, [28].

We define two ways in which people can be related: if they appear together in some event ( $PeopleRank_{event}$ ), or if they appear together in a single photo ( $PeopleRank_{photo}$ ). More formally,  $i$  and  $j$  are related if for some event  $E$  we have  $i, j \in K_E$ , where  $K_E$  is the set of identities known in the set of photos  $E$ . Similarly, if  $i, j \in K_s$  for some photo  $s$ , they may be related. Generally, any co-occurrence in a semantically connected set of photos can suggest a relation between people. While we relate people just by the concepts of co-occurrence in events or in single photos, other “connections”

are possible, for example, if both  $i$  and  $j$  appear in the collection during the same day. The set of photos taken in some location can also be seen as semantically connected. However, for the purpose of evaluating relationships between people, this connector may not be a good predictor: does the fact that two people appeared in photos from the same location, possibly during different events, suggest they are related?

Again, we only use the “event” and “single-photo” sets in this work. For the rest of this section we mostly use the event-based relationship. The single-photo case is analogous.

Modifying slightly the way we previously looked at estimators, the problem can now be formulated in terms of connections between people, rather than patterns of appearances of a single person. We can ask, “given that person  $i$  appears in photos from event  $E$ , which *other* people are likely to appear in photos from this event?” (or in the single-photo case, “given that person  $i$  appears in photo  $s$ , who is most likely to also appear in  $s$ ?”)

To answer this question we use the *PeopleRank* estimator. We re-formulate the question in terms of links between people. The link between two people  $j_1, j_2$  has a weight that represents the total number of events (or photos, in the analogous case) where both people appear together:  $W(j_1, j_2) = \sum_{\forall E} K_E(j_1, j_2)$  (function  $K_E(j_1, j_2)$  is defined similarly to Eq. 6.1, i.e., takes the value 1 if  $j_1$  and  $j_2$  both appeared in  $E$  and 0 otherwise). The links between all people compose a graph, and the weights  $W$  on links represent the strength of connection between each pair.

Unlike the famous *PageRank* [67] algorithm, which assigns a global, static score to each node in a link-based graph (and only occasionally re-computes the graph and scores), *PeopleRank* assigns an ad-hoc, context-based score to each node. In other words, the *PageRank* score is computed on demand, and both the graph structure and the score might change every time a new fact is added to the knowledge base  $K$ .

To give an example of *PeopleRank* computation, if we know that Nick appears in an event, the likelihood assigned by *PeopleRank* for other people to appear in the same event is relative to the weight of their link to Nick. If Dylan appears in 4 events together with Nick, and Kimya had 8 co-occurrences with Nick, Kimya is twice as likely based on *PeopleRank* to appear in an event, given that Nick appears, than



Dylan.

More formally, if  $j_1$  is known to appear in a given event, we assign a score for  $j_2$  using the formula  $PeopleRank(j_2|j_1) = \frac{W(j_1, j_2)}{\sum_{i \in I} W(j_1, i)}$  (the denominator is the sum of all weights of people connected to person  $j_1$ , and is used as a normalizing factor). The estimator ranks all candidates based on this score.

So far, our computation is based on a single person who is known to appear in the event. Often, though, more than one person is known. How do we compute the score for  $j$  if we know that both  $i_1$  and  $i_2$  appear in the event? One option is to generate scores based on the individual relationship: compute  $PeopleRank(j|i_1)$  and  $PeopleRank(j|i_2)$  for each person  $j$ . However, we are then reduced to the problem of how to combine and compare the different scores, raising the question of support and confidence, as defined by Agrawal et al. [1]. Naturally, the more events a pair of people appear in together, the stronger their “relationship” is; there is more support. But the more regularly they appear together, as opposed to appearing alone, again the stronger their relationship is (confidence is higher). Weighing the importance of support vs. confidence is not straightforward.

Another possible way to handle the case where more than a single person is known, is to count and compute the appearances of each person  $j$  when both people  $i_1$  and  $i_2$  appear together. For example, how often did Kimya appear at events where Nick and Mark both appeared? Often, though, the knowledge base of available annotation is not broad enough to accurately estimate such 3-way correlations.

Instead of having to compute 3-way (or  $n$ -way) correlations, we leverage the social factor and generalize the problem to be naturally supported by the *PeopleRank* approach: we rephrase the question “given that the set of people  $i_1, \dots, i_n$  appear in photos from event  $E$ , which other people are likely to appear in photos from this event?” in human social terms: “who is most connected to this *group* of people?” As *PeopleRank* is re-computed for each given context, we can modify the graph to answer this question. The system simply collapses all the known nodes  $i_1, \dots, i_n$  into a single group node. The group is now treated as one person, and the system re-computes the links to all other nodes in the graph. The link weight from the group node to  $j$  is the number of events in which  $j$  appeared with at least one of the people

in  $i_1, \dots, i_n$ . We can then rank all persons based on their strength of connection to the group, treating the new group node as we looked before at an individual node. For example, Kimya may have appeared 10 times with either Nick or Alex (or both), and Dylan appeared 8 times with either one of them. Kimya is more likely than Dylan to appear in an event that Nick and Alex both appear in.

Another possible approach to computing multi-people correlation is to use the concept of communities or subgroups in a social network. Communities can be computed from the graph structure using various methods (see for example [36]). We have not tried such approach in our current work.

To summarize, at any point during the process, the *PeopleRank<sub>event</sub>* estimator will generate a list of suggestions for a given photo  $s$  based on event co-occurrence with a person, or a set of people, known in event  $E(s)$ . The score *PeopleRank<sub>event</sub>* assigns to people who are already known in the event is 1, thereby “subsuming” the Event Estimator, albeit less accurately (the Event Estimator assigns people who are known in the event a score according to their frequency of appearance). The equivalent *PeopleRank<sub>photo</sub>* ranks the candidates based on photo co-occurrence with the people in  $K_s$ . Of course, *PeopleRank<sub>photo</sub>* will never be asked to rank people *already known* to be in the photo.

Other computations can be performed on the *PeopleRank* graph. As mentioned above, our network represents some approximate ego-centric view on the collection owner’s social network. Maybe some computation can help us expose relations that do not immediately emerge from the graph connections? For example, the system can be extended to support “indirect” relationships — if Kimya appears with Nick, and Nick appears with Marvin, is Kimya also likely to appear with Marvin even if we do not have direct evidence to support this conjecture? We implemented this multi-step computation using *PeopleRank*, but haven’t noticed any improvement in the labeling results. A possible explanation is that indeed, evidence of meaningful direct relationships between people does emerge from little annotation; if a relationship can only be deduced from a 2-hop connection, maybe it is not as strong, and therefore, does not improve our algorithm’s result. The results of this computation and other computation of our social network graph may be dependent upon social culture and

picture taking habits of different individual and across different cultures. See [36] for more methods and computations on social networks that may be applicable to our graph.

To summarize, the *PeopleRank* estimators generate a score similar to the basic estimators listed in Section 6.3.1. Even the computation is done similarly: in the terminology of Section 6.3.1, *PeopleRank* is considering a certain semantic set of photos ( $Q(S)$  is all photos containing the individuals who appear in the photo we wish to label or the event containing this photo), and assigning scores to new candidates according to their frequency in the set.

In any case, the *PeopleRank* estimators can be incorporated, together with other estimators, to generate a final list of candidates to appear in a certain photo, as we describe next.

### 6.3.3 Combining Estimators

Ideally, we would like to combine the estimators on a person-by-person basis. Given all the evidence, we would like to have a single number that evaluates the likelihood that person  $i$  is in photo  $s$ . While using machine learning techniques to learn and classify features may be a possible direction, a few simpler techniques to combine estimators, or combine candidate lists generated by different estimators, yield good results.

Some estimators might be more accurate but not as comprehensive as other estimators. For example, the  $N_{time}$  estimator, when used with a small time span  $T$  (say, 1 minute) may predict well the identity of people if repeated photos of the same individuals are taken. On the other hand, if a new person is in the photo, or if no other photos were taken within one minute from the given photo, the estimator will fail. In other words, false negatives are more likely than false positives. On the other hand, a different  $N_{time}$  estimator with  $T = \text{One Day}$ , may produce plenty of candidates, but their ranking would not be optimized to photos taken in the last few minutes, and therefore more prone to false positives. Other “fine grain” estimators include  $N_{loc}$  (when  $R$  is small), and *PeopleRank<sub>photo</sub>*. The  $L$  (location) and  $E$  (event) estimators

try to strike a balance between fine and broad estimations, but combining them with other estimators may perform better than  $L$  or  $E$  alone.

*Padding* is one way to combine fine and broad estimators. The fine estimators will generally offer very few, yet very accurate, candidates. When generating  $H_s$ , the system will choose the first candidates amongst the top-ranked candidates by a fine estimator, and the rest (padding the list until  $h$  candidates are found) from broader estimators. Of course, if a candidate was suggested by one estimator, there is no need for later estimators to add it to the list. More than two estimators can be combined this way. For example, estimator  $Pad_1$  selects candidates for picture  $s$  by padding, using the following estimators (in this order):  $N_{time}(s)$  ( $T = 10$  Mins),  $E(s)$ ,  $N_{loc}(s)$  ( $R = 1$  Km),  $L(s)$ . We tried various such combinations, as reported in Section 6.4.

*Weighting* is another way to combine estimators. In weighting, we assign a weight to each estimator. When assigning a “probability” (or score) for person  $i$  to appear in photo  $s$ , we simply compute the weighted sum of the score that the person receives from each of the estimators we consider. For example, estimator  $Weighting_1$  combines scores from the  $PeopleRank_{photo}$ ,  $N_{loc}$  ( $R=1km$ ),  $N_{time}$  ( $T=10$  mins), Event, Location, and Global estimators. For now we assigned weights for each estimator heuristically; in the future the weights can possibly be learned separately for each collection.

## 6.4 Evaluation Methods

For the evaluation of the system, we obtained four different personal photo collections with an average of 3246 photos. Each photo in the collections had time and location metadata associated with it. The location metadata in most collections was added manually, by dragging photos onto a map (for Collection  $A$ , the location metadata was captured as the photos were taken, using a GPS device) — see Appendix A for details. We ran our PhotoCompass application on all the collections to create the location and event hierarchies.

For evaluation purposes, we had the owner of each collection annotate the ground truth of identities of people in each photo, so we could compare and verify the names

Table 6.2: Statistics for the collections used in our evaluation.

<i>Collection</i>	A	B	C	D
<i>Time Span</i>	2 years	6 years	3 years	5 years
<i>Total Number of Photos</i>	5947	4347	766	1926
<i>Number of Photos Containing Named Individuals</i>	1673	1295	550	930
<i>Number of Named Individuals</i>	90	94	32	78
<i>Total Number of Annotated Identities</i>	2624	1941	985	2389
<i>Average Number of Named Individuals Per Photo</i>	1.6	1.5	1.8	2.6
<i>Average Number of Photos of Each Named Individual</i>	29	21	31	31
<i>Average Number of Photos of 5 Most Popular Individuals</i>	241	209	153	191

that were suggested by the system. One or more persons appeared in 31–78% of the photos in each collection. *Named individuals* — people known to the collection owner — appeared in 28–72% of the photos in the different collections. Some statistics about the collections are shown in Table 6.2.

In the evaluation, we emulated the process of users annotating their photos. Having obtained the ground truth in advance, we do not require an interaction with human subjects. Rather, we “hide” the ground truth from the algorithm; the process of users annotating photos is simulated by revealing identities to the system. In other words, we have a “virtual user” who adds annotations to the system. We considered two modes for the virtual user:

**Industrious Users** annotate all the photos with every “interesting identity” (see below) that appears in each. An industrious user starts with an empty set of annotated photos ( $K = \emptyset$ , to use the terminology of Section 6.2). At each step, an industrious user picks the next photo  $s$  in time order, and annotates the identities of all people that appear in  $s$ , helped by the system’s suggestions. At the end of the process, the collection is fully annotated.

Table 6.3: Summary of the virtual user modes.

<i>User Mode</i>	Industrious Users	Casual Annotators
<i>Initializing <math>K</math></i>	Empty set	Random selection from $I_s, s = 1 \dots  S $
<i>Add exposed identity <math>i</math> to <math>K</math>?</i>	Yes	No ( $K$ fixed)
<i>Picking next photo <math>s</math></i>	Time ordered	Any order (as $K$ is fixed)

**Casual Annotators** annotate a certain percentage  $p_{annotate}$  of the interesting identities appearing in photos in their collection. The purpose of this mode is to evaluate how good the system’s suggestions are when only a fixed percentage of the identities are annotated. Thus, in the beginning of the process, the system initializes itself by randomly selecting photos and retrieving from the ground truth some identities that appear in those photos, such that after the initialization we have  $\sum_{s \in S} |K_s| = p_{annotate} * \sum_{s \in S} |I_s|$ . After initialization, the casual annotator selects one random photo to fully annotate. In our experiment, the “user” goes through *all* photos that were not fully annotated in the initialization step, so that we get accurate statistics. Each repetition, though, starts from the same initial state, and selects a photo that was not picked before — until all photos have been picked.

The interaction modes for the two virtual user modes are summarized in Table 6.3.

An important parameter to vary when trying to model user interaction is  $|I|$ , the size of the set of different people that are of interest to be annotated in the photo collection. For example, some users are mostly interested in annotating photos of their close family and friends, while others annotate every person that appears in their collection. As  $|I|$  grows, we expect any label-suggesting algorithm to decrease in accuracy, as there are more candidates to choose from.

To evaluate the impact of  $|I|$ , we create several synthetic scenarios with different sets  $I$ , based on the full set of ground-truth annotations from the user. Let the full set of people that appear in ground-truth labels be  $I_{full}$ . We define  $I_\ell$  to be the subset of

$I_{full}$  containing the  $\ell$  most important people in the set. Then we can vary  $\ell$  and study the effect on performance. Since we do not have an importance rank for the people appearing in our test collections, we estimate importance by measuring popularity: the number of times a person appears in the ground-truth annotation.

Each evaluation run is based on one of the estimators, or a combination of estimators. In each evaluation step at time  $t$ , the system uses the estimator to generate  $H_s(t)$ , a list of  $k$  possible annotations for identities in photo  $s$ . The photo is picked such that at least one identity  $i \in I_s$  is not in  $K_s$ . The photos are picked in time order: for the industrious user, it is part of the model; for the casual annotator, since (after initialization) the system “forgets” the new annotations after each photo is fully annotated, the order does not matter.

As described in Section 6.2,  $H_s$  is then evaluated as a hit or a miss, by revealing the ground truth annotation of one person  $i \in I_s - K_s$  to the system (we remind the reader again that the evaluation is automated, given the ground-truth annotation of all collections). The exposed  $i$  is selected from  $H_s \cap I_s$  in case of a hit, or from  $I_s - K_s$  if  $I_s \cap H_s = \emptyset$  (a miss). The evaluation results are then added to our result statistics.

Since the initialization of  $K$  and the selection  $i \in I_s - K_s$  at each step is done at random, we execute the application with each set of parameters multiple times, reporting the average value of those runs. When reporting results for a single collection we execute the evaluation at least 5 times. When averaging over all collections, we execute the evaluation at least 3 times on each collection.

Before we report the actual results, a comment regarding system run-time performance is necessary. While we do not have exact execution times, our system demonstrates no human-perceived delay in generating the candidate lists (the entire emulated annotation of *all* photos is executed in less than a few seconds), certainly far from a performance bottleneck for user interaction or image-recognition algorithms.

### 6.4.1 Evaluation Goals

Our evaluation goals range from fine-tuning estimator parameters, to comparing estimators, to evaluating effects of system parameters on performance. In more detail,

we:

- Compare the performance of the Event and Location estimators ( $E(s)$  and  $L(s)$ ), that are based on our automatically computed location and event photo sets, to the performance of the estimators based on neighboring photos ( $N_{time}$  and  $N_{loc}$ ).
- Study various options for the PeopleRank estimators.
- Evaluate the effect of global parameters such as length of the candidate list  $h$ , and number of interesting people  $|I|$ , on the performance of different estimators and estimator combinations.
- Compare the performance of different estimators and combinations thereof. Does one of the estimators or combinations perform significantly better than others?
- Verify that our system is useful even when there is relatively little annotation effort by the users.

## 6.5 Results

We first examine the relative performance of the different estimators and strategies using the casual annotator mode, as listed above in Section 6.4. Then, we report on further experiments we executed using the industrious user mode.

### 6.5.1 Casual Annotator Mode

For all results in this section, unless otherwise noted, we used the following parameter base values. The set of candidates  $H_s$  is limited to 5 identities. The set  $I$  for each collection is the 20 most popular people appearing in the collection. As mentioned above, here we use the casual annotator mode, with size of the initial  $K$  set to 20% of the identities in the photos ( $p_{annotate} = 0.2$ ).



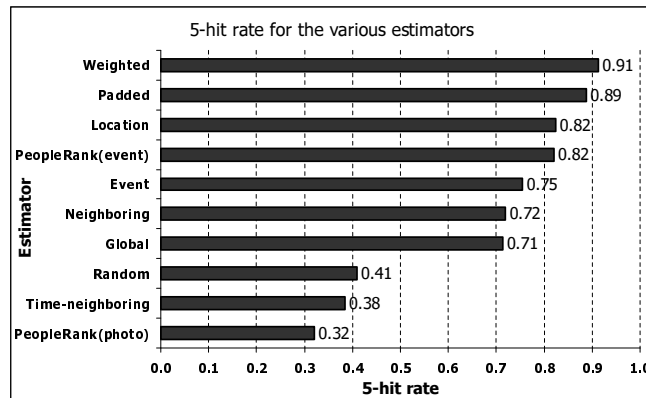


Figure 6.3: 5-Hit rate for the different estimators, averaged over all collections.

Indeed, these basic settings are reflected in Figure 6.3. The figure summarizes the main results: performance of the various estimators and combinations, averaged over multiple runs over each of the collection. In the figure, we show results for all the basic estimators, and the two best-performing combinations. The first combination (marked “weighted” in the figure) is based on weighting all estimators, using equal weights for all but the Global Estimator, which is weighted lower. The second combination (“Padded” in Figure 6.3) is based on padding the results of the Event, Location and Global estimators, in that order. The  $N_{loc}$  (“Neighboring”) and  $N_{time}$  (“Time-neighboring”) are tuned to 1km and 10 minutes, respectively (i.e., tuned to accuracy rather than completeness). For illustration purposes, we use the Random Estimator, which simply ranks people *already known* in  $K$  in a random fashion for each photo  $s$ . Figure 6.3 displays the estimators from top to bottom based on their performance.

We note a few observations about Figure 6.3, from the best-performing estimator down. We see that the two combined estimators perform better than any other estimator. The Weighted estimator performs best, slightly better than the Padded estimator. Other padded estimators also performed well, but are not shown in the figure. Also, we tried a Weighted estimator that assigned no weight to *PeopleRank* – it performed slightly worse than the Weighted estimator shown in the figure.

Figure 6.3 shows that the basic estimators, by themselves, do not perform as well.

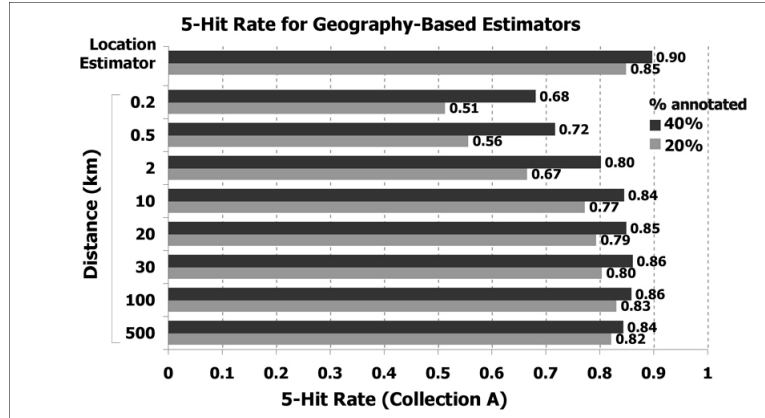


Figure 6.4: 5-Hit rate for different geography-based estimators —  $N_{loc}$  estimators with varying distance limits, and the cluster-based Location Estimator.

However, the Location, Event and the  $PeopleRank_{event}$  estimators perform better than the naive Global Estimator that ranks people by their overall frequency. The more specific basic estimators, Time-Neighboring ( $N_{time}$ ) and Neighboring ( $N_{loc}$ ) suffer from false negatives (not enough candidates), since, as explained above, their tuning parameters were set very low. Next, we explore the settings of the Time-Neighboring ( $N_{time}$ ) and Neighboring ( $N_{loc}$ ) estimators, and compare their performance to the Location and Event estimators.

Figure 6.4 shows a comparison between the different geography-based estimators. On the X-axis we plot the 5-hit rate (hit rate when  $H_s$  is limited to five names). We show results for a number of Neighboring ( $N_{loc}$ ) estimators, with various values for the maximum distance allowed,  $R$ . On the top row, we show results for the Location Estimator, based on the location clusters as detected by our PhotoCompass system. The results are shown for two different sizes of the starting set of identities,  $p_{annotate} = .2$  and  $p_{annotate} = .4$ . For example, the bottom row indicates the performance for  $N_{loc}$  with the distance parameter set to 500km. The 5-hit rate for this case is 0.82 when  $p_{annotate} = .2$ , and a 0.84 when more knowledge is available ( $p_{annotate} = .4$ ). The figure shows results only for collection  $A$  (the only collection with accurate, automatically gathered, location coordinates). However, result trends for the other collections were similar.

We can see in the figure that the Location Estimator is consistently better than estimators based on neighboring photos, regardless of the radius that is being considered. When the radius is too small, not enough candidates are available for the  $N_{loc}$  estimators. When the radius is too large, too many candidates add noise to these estimators. Conversely, the Location Estimator offers a semantically coherent set of photos that in some way “belong together” (see the details of how the clusters are created in Chapter 3), and therefore performs better.

In contrast, the Event Estimator, based automatically detected events, did not perform better than all time-neighbors estimators  $N_{time}$ . Figure 6.5 shows the 5-hit rate for various time-based estimators, averaged over all collections. In the top row, we see the results of the Event Estimator, while the other rows show the time-neighboring estimators  $N_{time}$ , varying the maximum-allowed time difference  $T$ . We can see that performance improves as we increase the time span. In fact, even when we used a time span of 20 days for  $N_{time}$ , the performance stayed at the same level. To summarize, the Event Estimator did not perform better than the long time-neighboring estimators. The reason may be that our system is aggressive in splitting events, preferring to err on the side of over-segmentation, while people that appear in photos may linger around for a longer time (for example, a visit from Kimya that lasts a few days and contains a few photographed events). However, we suspect that the event concept will be more robust to changes in  $h$ , the size of the candidate list, compared to the day-range time-neighboring estimators.

We examine the performance of the *PeopleRank* estimators and their combination in Figure 6.6. The figure shows the 5-hit rate for each collection and each estimator. We show results for the *PeopleRank*<sub>photo</sub> (top row) and the *PeopleRank*<sub>event</sub> (second row from the top) estimators. We also show two combinations of the estimators: *PeopleRank*<sub>photo</sub> padded with *PeopleRank*<sub>event</sub>, and an equally-weighted combination of these two *PeopleRank* estimators (see Section 6.3.3 to recall the Padding and Weighting methods for combining estimators). For example, the 5-hit rate for collection  $C$ , using candidates from *PeopleRank*<sub>photo</sub> padded with candidates from *PeopleRank*<sub>event</sub>, is 0.86. A key observation from the figure is that not all photo collections are created equal. For example, while *PeopleRank*<sub>event</sub> performs worst for

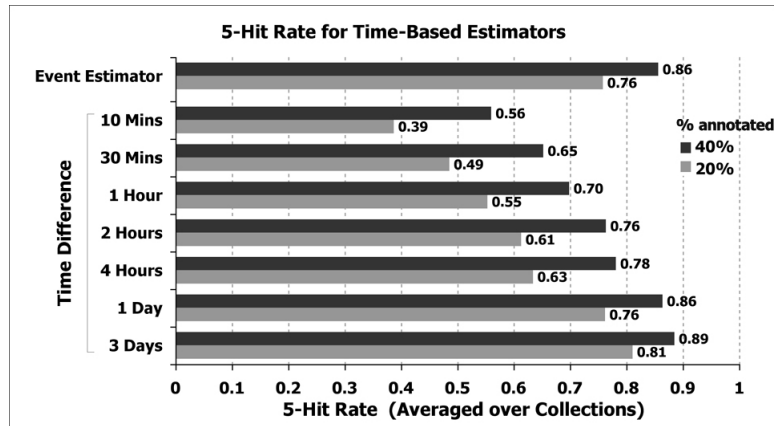


Figure 6.5: 5-Hit rate for different time-based estimators —  $N_{time}$  estimators with varying time limits, and the Event Estimator.

collection  $D$ , the same collection enjoys the best  $PeopleRank_{photo}$  performance. In other words, collection  $D$  was easier than others to predict who is in a photo based on known people in the photo. At the same time, for collection  $D$  it was harder than other collections to predict who is in an event based on people known to be in the event. We hypothesize that the unique characteristic of collection  $D$  explains this discrepancy: part of the better success rate for  $PeopleRank_{photo}$  is the fact that, as shown in Table 6.2, collection  $D$  had the most named individuals on average in each photo, making label suggestions more likely to match.

Also note that both combinations of the  $PeopleRank$  estimators yielded performance similar to  $PeopleRank_{event}$  alone, improving mainly for collection  $D$ . As we show next other estimator combinations involving the  $PeopleRank$  estimators perform even better.

The next few figures give a clearer picture of the overall system performance and its dependence on basic parameter values.

Figure 6.7 shows the  $h$ -hit rate for the basic estimators and various combinations as we vary the value of  $h$  (number of candidates in the list  $H_s$ ). For example, when  $h = 3$ , the Weighted estimator suggested an identity that was indeed in the photo for 81% of the new annotations. Recall that Figure 6.3 on page 130 presented the results for the different estimators that are equivalent to the point where  $h = 5$  in

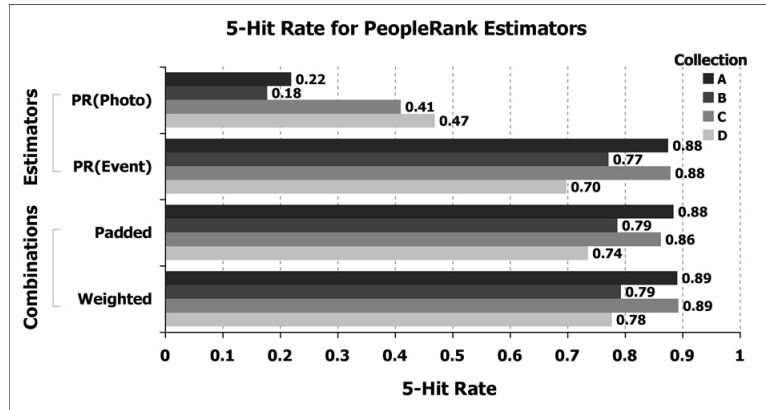


Figure 6.6: Performance of the PeopleRank Estimators and their combinations, for each collection.

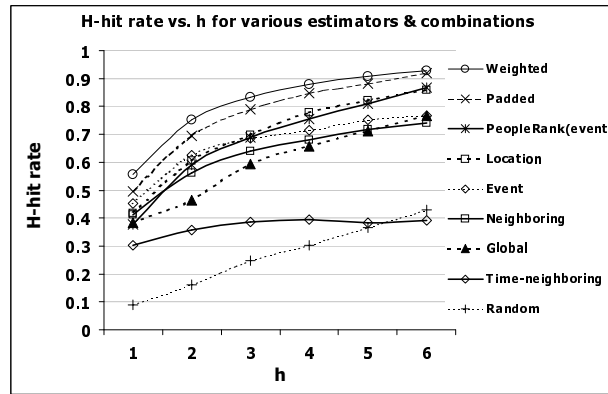


Figure 6.7: h-Hit rate (averaged over all collections) for various estimators vs. value of  $h$ .

Figure 6.7. The parameters and estimator combinations are also equivalent to those of Figure 6.3. In particular, we note again that the  $N_{loc}$  is tuned to 1km and  $N_{time}$  to 10 minutes, both rather limited values that aim at precision rather than recall.

We also remind the reader that the h-hit rates are averaged over all executions and all collections for each point.

As expected, all estimators and combinations improve as the size of the candidate list grows. It is also worth noting that the relative performance of the estimators remains mostly consistent across various values of  $h$ , except for a few noticeable trends. Especially notable is the performance of  $N_{time}$ , which does not improve much

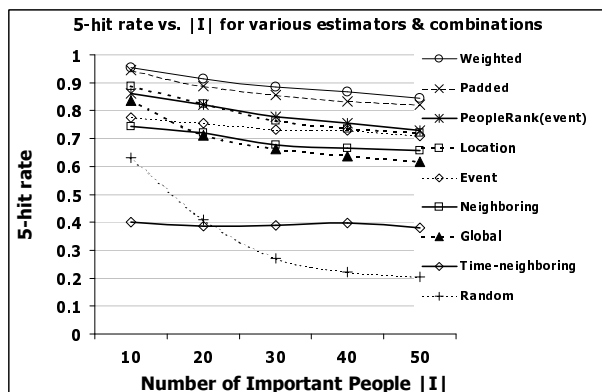


Figure 6.8: 5-Hit rate (averaged over all collections) for various estimators vs. size of the important people set  $I$ .

as  $h$  grows, such that it is comparable to the Random Estimator performance level when  $h = 5$ . As mentioned above, the constant performance is mostly due to false negatives. We verified this hypothesis by padding the list provided by  $N_{time}$  with candidates from the Event Estimator: the hit rate was better than the hit rate of each estimator on its own.

We performed the same comparison for  $p_{annotate} = 0.4$  (i.e., when the initially annotated corpus is 40% of all identities). The relative performance of the different estimators is roughly the same, as all estimators are now performing slightly better given the additional knowledge. However, the  $N_{time}$  estimator is the only one doing *significantly* better (for example, the hit rate when  $h = 1$  is up from 0.3 to 0.4, and similarly for other  $h$  values), showing that, indeed, false negatives are its main shortfall.

Figure 6.8 shows the analysis of estimator performance when we vary the number of “important people” in the collection: the number of different people that users would like to label, or  $|I|$  (as a reminder, for all other figures we had  $|I| = 20$ ). Naturally, the more candidates there are (larger set  $I$ ), the tougher it is for the system to correctly guess who appears in a given photo. For example, the Weighted Estimator has an average 5-hit rate of 0.94 when the users are only interested in 10 people, and 0.84 when 50 people are of interest for annotation. Indeed, the performance of all estimators drops as the size of  $I$  increases. However, only the Random Estimator

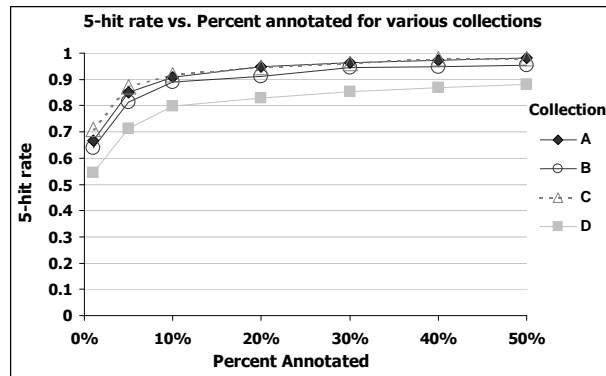


Figure 6.9: 5-Hit rate for the different collections, using the Weighted Estimator and varying values of  $p_{annotate}$ .

and the naive Global Estimator degrade quickly as the size of  $I$  increases; the other estimators show better stability. On the other hand, the  $N_{time}$  estimator is the most invariant to changes in  $I$ , once again demonstrating its accuracy and resilience to false positives.

Figure 6.9 displays the 5-hit rates of the Weighted Estimator for each of the collections in our experiment. The figure illustrates the rapid improvement in the quality of the candidate list as more annotation is available, as well as shows the difference in performance between the different collections. More specifically, the figure shows the effect of  $p_{annotate}$ , the percent of identities in photos annotated during initialization, on annotation suggestions performance for each collection. For example, when half of the identities in the photos are annotated, the Weighted Estimator performs at a 5-hit rate of 0.98 for collections  $A$  and  $C$ . Even when only 5% of the identities are annotated, the 5-hit rate ranges from 0.71 to 0.87; when as little as 1% of the identities in photos are annotated, the hit rate reaches 0.54 to 0.71, and rises rapidly. Moreover, performance for all collections exhibits similar trends, suggesting that our system may perform well for different types of users. Of course, only a broader investigation which is beyond the scope of this work can verify this claim.

Until now we have used a specific weighting scheme when combining estimators. In Figure 6.10 we investigate the performance of different weighting strategies. The five strategies we examine are:

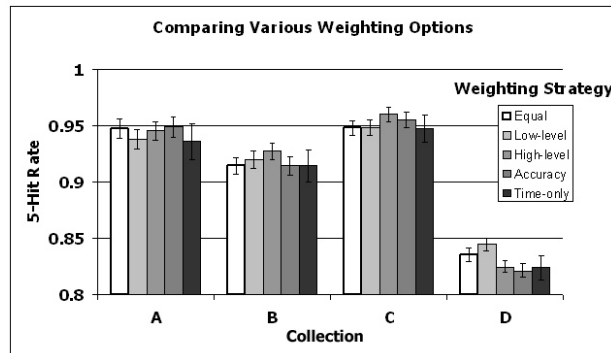


Figure 6.10: Comparing different weighting strategies. Results are averaged over five runs for each collection and strategy. Confidence intervals are shown for 95% confidence.

**Equal** : All estimators are assigned equal weights, except the Global Estimator, weighted lower. This is the weighting strategy we report on elsewhere in this work.

**Low-level** : Low-level estimators ( $N_{loc}$ ,  $N_{time}$ ,  $PeopleRank_{photo}$ ) are weighted more heavily than broad estimators (Event, Location, Global,  $PeopleRank_{event}$ ).

**High-level** : The high-level, broad estimators weighted more heavily than low-level estimators.

**Accuracy** : The weight of each estimator is relative to the 5-hit rate for that estimator, as shown in Figure 6.7. The actual weights were quite similar to the weights assigned in High-level weighting.

**Time-only** : The weighted estimator is restricted to include only estimators which do not require location information (Global, Event,  $N_{time}$  and the  $PeopleRank$  estimators). This will give us an idea regarding how useful our system is when location metadata is not available.

The average 5-hit rate for each weighting strategy, and for each collection, is shown in Figure 6.10. The results are grouped by collection, so we can compare the performance of the different weighting strategy for each collection. For example, for collection C, the “equal” weighting strategy resulted in an average 5-hit rate of 0.945. Note that



in order to visually compare the different option more easily, the Y-axis ranges from 0.8 to 1.

We can see from Figure 6.10 that there is no significant difference in the performance of different weighting strategies. Within each collection, there are very few cases where one strategy is significantly better than another (e.g., the low-level weighting vs. high-level weighting for collection *D*). Those differences are not consistent across the different collections (e.g., the low-level weighting and high-level weighting relative performance for collection *C* is opposite of that of collection *D*). A possible future exploration is an adaptive strategy that can learn the appropriate weights, as the user annotates the photos. However, it is not clear that an adaptive strategy will be very beneficial, as performance of the different weighting strategies does not vary significantly.

An additional observation from Figure 6.10 is that the time-only weighting, which does not use the estimators that are based on location metadata, is not significantly worse than all other weighting strategies. However, the results for time-only weighting vary more widely, and on average perform consistently worse (but again, not significantly) than all other strategies. To conclude, while using location metadata is beneficial, our system can prove very useful even when such metadata is not available.

### 6.5.2 Industrious User Mode

In studying the industrious user mode we did not repeat all the experiments as performed in the casual user mode. We did, however, verify that performance trends are indeed similar. A hint to this equivalence can already be found in Figure 6.9, where we show how performance is affected by the number of annotated photos. The more annotated identities available, the better candidate sets are suggested by the system, but the improvement is rapid and “levels out” quickly (only moderate gains after 30% of the photos are annotated).

It is important to remember that the interaction mode for the industrious user is sequential: the user annotates photos in time sequence (for the casual annotator the initial set of annotated identities is picked randomly). When a certain portion of the

identities are annotated, in the industrious user scenario, this means all the identities that appear in the collection up to the current photo are known to the system. How does this fact effect performance? In particular, we wish to find out:

- What are the failure patterns in the photo sequence? We hypothesize that failures often occur in the first photos of each event, before the system learns the identities of some people in the event.
- Do the patterns persist throughout the collection? For example, when annotating the first event in the collection, the system can mostly utilize the Event Estimator: there is no knowledge of past locations, and no information about co-occurrence of people in the collection. On the other hand, for later events, the Location Estimator can be utilized to generate suggestions even if no person is known in the event yet.

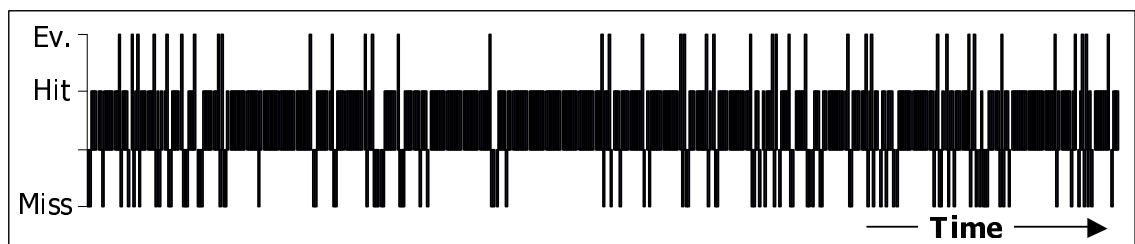
For a more concrete example, say Robyn lives in New York City. The first time Robyn appears in the collection, the system does not know about him. Thus, the system must have a “miss” when suggesting identity candidates for the first time Robyn appears. After the identities in that first event were annotated, a later event may occur in NYC. The Location Estimator may kick in to suggest Robyn even if he has not yet been annotated in the new event.

This type of “learning” occurs in the casual annotator mode as well. However, in the industrious user mode, as the annotation process occurs in time sequence, it is easier to present the system behavior as annotations are added.

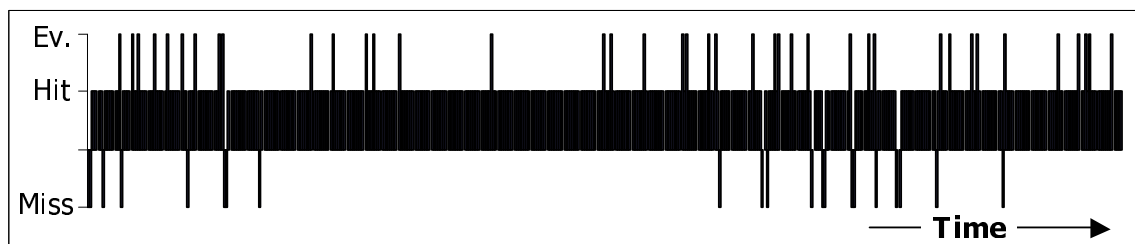
Figure 6.11 examines the system’s performance in time sequence. The X-Axis represents  $t$ , the advancing system time. Recall from Section 6.2 that the system advances from time  $t$  to time  $t + 1$  when a single identity is annotated in a single photo, and that in the industrious mode, photos are annotated in the order they were taken. The figure<sup>5</sup> shows, for  $t = 1, 2, \dots$ , whether the the system’s label suggestions at time  $t$  where a hit or a miss ( $H_s$  contained an identity that appears in the photo, or not). A hit is represented by a thin vertical bar of height 1 (rising above the X-Axis). A miss is represented by a similar vertical bar of height -1 (dropping below

---

<sup>5</sup>This visualization was inspired by Edward Tufte’s SparkLines.



(a) Event Estimator



(b) Padded Event, Location and Global Estimators

Figure 6.11: Time sequence analysis of the system's identity suggestion hit/miss result for individual identities. Results are shown for each of the first 500 annotations in the photos of collection *A*.

the X-Axis). We also mark the beginning point of each new event in the collection, as automatically detected by the system, with a vertical bar of height 2. For example, in Figure 6.11(a), the system's first two suggestions (left-most bars along the X-axis) culminated in a miss (negative bar); the first hit was achieved for the third identity. The first event ends after 15 identities were annotated. Notice that to save space, the figure is condensed so that adjacent bars fuse together.

The different parts of Figure 6.11 show results for the first 500 identities annotated in collection *A* of our test collections described earlier. The top figure, Figure 6.11(a) shows results for the Event Estimator. Not surprisingly, at the start of each new event the estimator fails to suggest a correct identity (a miss). In fact, since the Event Estimator only assigns non-zero scores to people who are already known to appear in the same event, it is guaranteed to generate a miss each time a new identity appears *in each event*. Also note that these misses are the bulk of the mistakes made by the Event Estimator: when it learns the identities of people in a given event, the Event Estimator performs quite well.

Figure 6.11(b) demonstrates that padding the list of candidates generated by the Event Estimators with the top identities according to the Location Estimator, and then with Global Estimator candidates, improves the performance. In fact, since padding only adds new candidates, there is no way in which it could detract from performance. Comparing Figures 6.11(a) and 6.11(b), then, reveals that in many cases the system can guess identities in the first photos of each event, based on location or global popularity. In some cases, as Figure 6.11(b) shows, even those misses by the Event Estimator that occur in the midst of an event, become hits when the candidate list is padded with location-based and globally-popular candidates.

The Weighted Estimator, unlike the Padded Estimator, might introduce new misses. Since the estimator assigns different weights to the various basic estimators, it is possible that candidates ranked high by the Event Estimator will be pushed out of the top list of candidates in the final ranking. However, in the settings of Figure 6.11, the Weighted Estimator had exactly the same results for the first 500 identities as the Event/Location/Global Padded Estimator, and therefore is not shown in the figure. The differences between the Padded and Weighted estimators emerge when

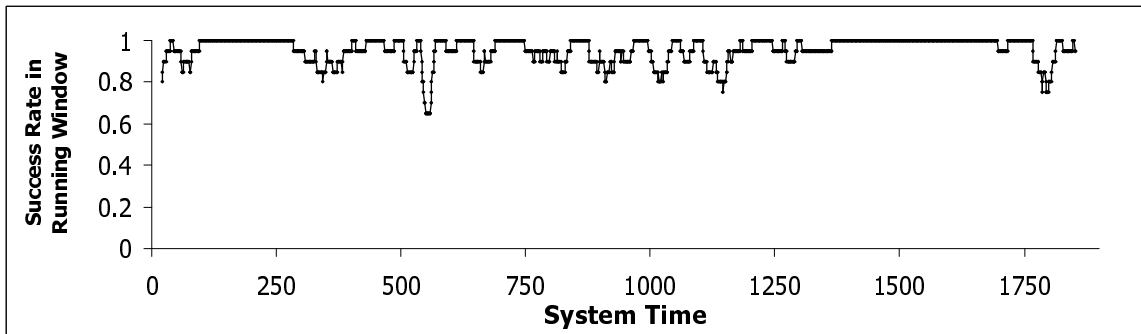


Figure 6.12: Running window average 5-hit rate over the last 20 identity suggestions (labeling steps) for the *A* collection.

more information is available (i.e., further down the time line). The Weighted Estimators, given additional information, perform better overall, although in some cases the Padded Estimator had a hit where the Weighted Estimator missed.

A more aggregated look at the time-sequenced performance of the system is shown in Figure 6.12. The figure shows a running average over a window of the last 20 identity suggestions by the system. As in Figure 6.11, according to the industrious user model, the annotations are done in time-sequence based on the time each photo was taken. Notice that the line starts after 20 identities were annotated. At that point, the average success rate for those first identities was 0.8 (16 label suggestions that resulted in a hit). This running average allows us to visualize trends in the process more easily. For example, the figure demonstrates that roughly speaking, the performance is rather constant. Very few 20-identities blocks drop beneath the 80% hit rate.

The most prominent “dip” in Figure 6.12 occurs around the 559th identity, where the running average hits a low of 65%. The photos represented by this dip were taken in a family event with a lot of participants.

In contrast, the flat plateau of 100% success that spans over the 1500th identity represents photos taken on a two long travel-oriented trips where the number of photographed persons of interest was low, thus easing the system task of suggesting possible identities.

## 6.6 Conclusions and Future Work

We have shown that our system can provide accurate identity-label suggestion sets for non-annotated photos in a collection. These suggestion sets are based on temporal, spatial and social context: we leverage patterns of spatial and temporal re-occurrence of people in the photo album, as well as co-occurrence of different people in the album. These patterns emerge during the annotation process.

Our use scenario for the identity suggestion sets was to present the sets to users as they are annotating their photo collection. The short list of candidates will allow users to annotate their photo collections more rapidly, even on a small-screen device.

Our results show that in most cases, when a user tries to annotate an identity in one photo, our system correctly suggested at least one person that appeared in the photo, even when limiting our suggestion list to as few as five identities. The success rates for our top methods were 80–90% or higher, even when as few as 10% of the identities in the photos were previously annotated.

A possible direction for future work is to test our system with a larger number of collections. This will allow us to tune the various parameters. In addition, testing a system with human users will be informative: are users likely to do more annotation when they are helped by the system? Or are they quickly frustrated when the system does not provide a correct list?

In addition, a possible future direction is to enhance the system with machine learning algorithms that will adjust the system parameters dynamically for each collection.

One other possible future direction is adding more context-based estimators. We list a few possible estimators in Section 6.3.1, for instance, an estimator based on the time of day or the day of the week in which a photo was taken. Our framework allows for easy integration of such new estimators.

Most importantly, we would like to combine our context-based approach with face detection and face recognition algorithms, hopefully creating a system that performs better than each technique by itself. A possible direction is to use the candidate sets and likelihood scores generated by our system for each photo as an input to a

face recognition system. Since the recognition system will have fewer candidates to consider, we expect its accuracy to improve. The system can then assign a final score based on the analysis of the visual features and the prior context-based probability.

We envision a system that combines context- and content-based techniques, together with specialized UI techniques, to support quick and powerful labeling environment. Transcending label suggestions, such a system should aim to support at the real user need: accurate retrieval of photos by identity. The retrieval will include photos that were not yet annotated, while still minimizing the effort users need to invest in annotation.

## Chapter 7

# From Where to What: Sharing Metadata

Our system as described in Chapter 6 uses location and other context information, coupled with user annotation, to effectively guess who appears in each photo. In this chapter, we introduce sharing information between users based on the location where photos were taken. The location-based sharing, coupled with free-text user labeling of photos, allows us to effectively guess the landmarks that appear in each photo.

In many applications, simple text labeling of some photographs will enable much improved results when searching or browsing a collection. However, many people do not label more than a few of their photos, or do not invest the effort of labeling their photos at all. The system we describe in this chapter enables a solution through sharing of existing labels, so that nobody needs to do more work than they do now, yet everyone gains functionality.

Using location data as a pivot we can enable the sharing of information about photos. By comparing where photos were taken, we can associate photos from a set of labeled photographs with unlabeled photos from another set. We then associate the corresponding labels with the unlabeled photographs. Physical origin proximity of geo-referenced photos is much easier to evaluate than current image-based proximity measures, such as visual similarity, which are still computationally expensive and inaccurate.



We now illustrate the general idea using a simple example. Meet H, an avid photographer. H has taken a photo of Stanford University’s Memorial Church. H labeled the photo “Stanford Church” using some desktop software tool such as a photo browser. The label and the coordinates of the photos are submitted to an online repository that H agreed to participate in. Another photographer, M, takes a picture of the church from the same location a day later. Now, M does not have to label the photo: M queries the online repository by the coordinates of M’s photo and receives, in reply, the label submitted by H.

Another scenario is for users to perform a *term search* over their own unlabeled collection – without having explicitly associated labels with any of their photos. For example, M submits a “Stanford Church” query to the system. The system finds H’s matching label, notes the location where H’s church photo was taken, and then searches M’s photos for ones taken near the location of H’s church photo. The ID or the coordinates of M’s church photo will be the result of this search.

The underlying assumption in our work is that many photos are taken by different people in the same place. This assumption enables sharing, as it is very likely that some photographer X has already taken a photo in the location where M’s photo was just taken. More importantly, this “picture spot” assumption<sup>1</sup> is imperative since a critical mass of submissions in some location is needed for such a system to work for that spot. We illustrate below some problems in the our simple scenarios that demonstrate why this critical mass is required, but first, we note an important consideration regarding the location metadata.

The nature of “location” as it is used throughout the thesis corresponds to the camera location. As we explain in more detail in Appendix A, the location metadata associated with photos in this work is in fact the location of the camera at the time the picture was taken. For the scheme proposed in this chapter, however, one may suggest that the location of the *photographed object* is also relevant. Indeed, this location dichotomy is discussed in more detail in Section 7.5. In practice, since our work builds on the “picture spot” assumption, we use the camera location: following

---

<sup>1</sup>We of course refer to the *Kodak Picture Spot* signs that were sprinkled in the past around popular tourist destinations, instructing tourists to take a photo from some specific spot.

the same line of reasoning, people are likely to take photos of the *same objects and landmarks* from the same location.

Indeed, there are several potential problems to consider in our simple example above. First, H may have given the photo an unhelpful label (e.g., “my son and I at Stanford”, which is still somewhat informative; but a label such as “lovely bird” is not very likely to be of use to M or other users). More confusingly, M may have taken a photo near Stanford’s Memorial Church but pointed the camera at an entirely different subject (the Stanford campus offers nice views in many directions). Another potential problem is that H may be using a different, or shortened, name for a photographed object. Of course, H’s label can just be plain wrong. To make matters more complicated, H may be worried about privacy and information leakage.

The solution we propose is LOCATION-to-LABEL (LOCALE<sup>2</sup>), a system for implicitly sharing label information based on information retrieval techniques. The LOCALE system collects coordinates of photos and their associated labels from participating users. Patterns of photos taken at the same locations by different users are likely to emerge from the data. LOCALE then utilizes these patterns by applying term frequency, weighting and clustering techniques to overcome some of the problems mentioned above. LOCALE supports search over unlabeled or partially-labeled collections, based on this data and its analysis.

To test LOCALE, we devised an experiment to acquire geo-referenced, labeled photos from tourists visiting the Stanford campus. Once acquired, we used the dataset to perform various search and labeling tasks with LOCALE. Despite being small in geographic scale, and restricted to tourist-type situations, the experiment demonstrates that:

- Our “picture spot” assumption above has merit, as shown by trends of where and when people took photos during the experiment.
- At least in the experiment scenario, LOCALE is useful in capitalizing on this labeled set of photos to enhance collections of unlabeled photos, as described above.

---

<sup>2</sup>Also stands for Location-Oriented Computer-Aided Labeling Environment.

Under the hood, there are distributed and centralized modes for search in LOCALE. In centralized mode, the LOCALE server stores the database of photo metadata (photo locations and labels) and handles all the computation, including the process of searching M's photos. Thus, the server has to know the location of all the photos in M's collection for M to be able to perform a search. In distributed mode, a summarization of LOCALE data is cached on M's machine. After the information is cached, the term search and ranking of the photos can be performed over the LOCALE cache on M's machine without contacting the server.

While not tuned for this task in our current implementation, LOCALE can also be used for making label suggestions for unlabeled photos in a collection. In fact, the cached summarization mentioned above for the distributed mode comprises likely labels for each photograph. These candidates can be exposed to the user to choose from, in order to ease the user's labeling task.<sup>3</sup> However, in this work we focus on the search task, providing benefits to all users, even if they do not wish to invest time in labeling photos in their collection.

Privacy poses a major concern in systems like LOCALE. Note that LOCALE does not require the photos themselves to ever leave their owner's machine. Nor are the identities of the photographers ever needed for operation. Nevertheless, implications of privacy are two-fold. First, the system should not expose any private information that emerges from labels given by specific users to others. The aggregation techniques used by LOCALE should indeed prevent that from happening, but stricter guarantees need to be in place. Second, users should be made aware that their labels are to be used in a system that exposes some information to the public. Will the labeling behavior change as a consequence? Research on private vs. public annotation of books [63] has shown that they might.

Related work to the LOCALE system includes work on labeling and annotation in photo collections [46, 49, 80, 94, 95], and work on collective labeling of photo repositories (such as flickr.com, or the work of [50]). The area is surveyed in more detail in Chapter 2.

---

<sup>3</sup>Alternatively, these terms can be used to remind the user "what is shown in this photo", without labeling it.

Much like our work, the research of Davis et al. on Mobile Media Metadata (MMM) [18, 77] utilizes spatial and temporal context to help with the annotation of photographs that are taken with camera-equipped mobile phones. One of the categories of captions proposed in their work is location. The way location annotations are suggested to the users in MMM is similar to the ideas we lay out in this chapter, or more specifically in Section 7.4. One notable difference is derived from the enabling technology: in LOCALE, we assume GPS-grade accuracy for the photos in the system, while MMM is based on the coarser location capabilities of mobile phones, reducing the system's possible precision relative to ours. Another difference is that MMM already supports suggested captions for events, in addition to location-based captions.

We have implemented LOCALE using three different strategies, in both centralized and distributed modes. The implementation strategies are described in Section 7.1. In Section 7.2 we present the experiment we devised to test LOCALE and the data collected in the experiment. The evaluation of LOCALE performance given the experiment data is presented in Section 7.3. We also looked at how well LOCALE can automatically assign labels to photos, and discuss some preliminary results in Section 7.4.

## 7.1 The LOCALE System

The LOCALE system consists of a centralized server  $S$  with a global photo database  $DB_S$ , and users with personal photo databases  $DB_1, DB_2$ , etc. The system is illustrated in Figure 7.1. The main table in these photo databases is the photos table  $P(I, G, L)$  where the columns are image (I), geographic location coordinates (G), and Label (L). In each user  $u$ 's  $DB_u$ , table  $P_u$  consists of tuples for  $u$ 's own photos. The server's table  $P_S$  consists of tuples submitted by cooperating users. In our implementation, null values are permitted. In fact, the I values in  $P_S$  are always null (the server never requires the submission of *photographs* – users submit  $(null, g, \ell)$  tuples). Also, some user databases could lack labels for some, or all, of their photos (L values may be null).

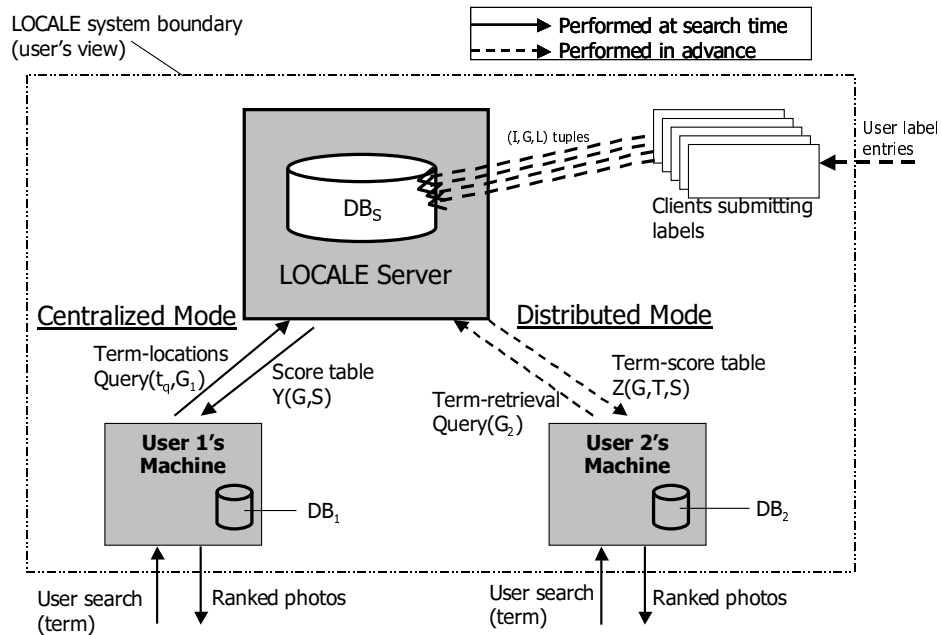


Figure 7.1: Architecture of the LOCALE system.

Labels can be any type of textual information attached to the photos: keywords, free-form text etc. The idea is that LOCALE can leverage different sources of text about the image. Those can range from a caption assigned to the photo in a photo browser application, to a text of an email message that carries the photo as attachment. Other possible sources are cell phone SMS (“Short Message Service”) messages sent with photos taken by cell phone cameras, or even SMS messages without an associated image.<sup>4</sup>

In our system, labels are broken down into *terms* consisting of a single word or a two-word phrase. LOCALE creates an index based on one- or two-word terms extracted from the label text. As a consequence, in this work we limit the user search to one or two words. However, information retrieval techniques supply methods for extending the queries to arbitrary length.

As we discuss in Appendix A, we assume location accuracy in the scale of under 30 meters. In fact, in practice the accuracy of photos acquired in our experiment

<sup>4</sup>As our system does not require submission of photos.

exceeded that accuracy. The location data is represented in the LOCALE database by (latitude,longitude) pairs.

The purpose of LOCALE is to enable a user  $u$  to perform term searches over  $u$ 's collection  $P_u$ , even if those photos are not labeled. The user input to the search mechanism is a search term  $t_q$  and, implicitly, the locations of the user's photos<sup>5</sup>  $G_u = \Pi_G(P_u)$ . The search output is an ordering of the user's images  $I_u = \Pi_I(P_u)$  based on relevance to the search term. The search is performed using only the data in  $P_S$ . For simplicity, we only consider the case where users performing search, like  $u$ , have not labeled any photos (the value of attribute  $L$  in each row of  $P_u$  is null). Therefore, we assume a different set of users that contribute labels (see the top right corner of Figure 7.1). Optimally, a search will be able to integrate the user's own labels with the LOCALE search based on other users' labels.

User search is handled differently depending on whether LOCALE is in a *distributed* or *centralized* mode. For each mode, we list three implementation strategies (Weighted Neighbors, Location-Clustered and Term-Clustered). For each mode and strategy we explain the way data is stored and pre-processed and the way queries are handled.

The three implementation strategies reflect different possible approaches to the problem of associating photo labels and photo locations. Roughly speaking,

- Weighted Neighbors is an ad-hoc strategy, performing a local computation based on locations and labels of photos taken near a new photo in question.
- Location Clustered attempts to organize the dataset into a location hierarchy. LOCALE will associate the relevant text terms with each node in the hierarchy. In other words, this strategy maps locations to text terms. Conceptually, queries to LOCALE will be resolved by first associating the queried location with a location node.
- Term Clustered is looking at the geographic distribution of the terms in the

---

<sup>5</sup>We will be using relational algebra operators such as  $\Pi$ , the attribute projection, throughout this chapter.

dataset. In other words, this strategy maps from text terms to locations. Conceptually, queries to LOCALE will be resolved by first associating the queried term with a textual term from the dataset, then evaluating the location match.

In the following subsection we describe the search process in centralized mode. In the next subsection we describe the distributed mode and note the differences from centralized LOCALE.

### 7.1.1 Centralized Mode

In centralized mode the LOCALE server is contacted at search time and performs most of the search-time computation. User 1 demonstrates the process in Figure 7.1. The user search query is translated to a *term-locations* query with parameters  $t_q, G_1 = \Pi_G(P_1)$ : the search term and the set of coordinates of the user’s photos.<sup>6</sup> The LOCALE server ranks  $G_1$  with respect to  $t_q$ , using the information in  $P_S$ . We implemented this ranking using three different strategies; the details of each are below. At the end of the ranking step, the LOCALE server replies with a table  $Y(G, S)$  of geographic locations and the score of their match to term  $t_q$ . The user’s machine then executes a simple natural join between  $Y$  and  $P_1$  to produce a ranking of the user’s images  $\Pi_I(P_1 \bowtie Y)$  based on the match to  $t_q$ .

We now show how the term-locations queries are handled in each implementation strategy.

#### Weighted-Neighbors (WN) LOCALE.

The process of ranking locations based on their match to the search term is done by finding, for each location, nearby photos in  $P_S$  whose labels include the search term. This can be done efficiently if indices for the location and the terms exist for  $P_S$ . The score for the match between each location and the search term  $t_q$  is computed for

---

<sup>6</sup>In practice, the coordinates of  $u$ ’s photos may already have been stored in the LOCALE server’s photo table  $P_S$  ahead of time. In this case the users will identify themselves to LOCALE at query time using a unique ID.

every  $g \in G_1$ :

$$Score(g, t_q) = \sum_{p \in P_S} IR(t_q, \ell_p) PROX(g, g_p)$$

The function  $IR(t, \ell)$  computes the match between a term and a photo's label. In our case, we compute the match between the search term  $t_q$  and every photo's  $p \in P_S$  label  $\ell_p$ . The  $PROX(g_1, g_2)$  function evaluates the proximity between two photo locations. In our case, we evaluate the distance between the current location  $g$ , and every photo's  $p \in P_S$  location  $g_p$ .

Our  $PROX$  function computes the inverse of the square root of the Euclidean distance between  $(g_1, g_2)$ . However, we cap the function as its extremes:

$$PROX(g_1, g_2) = \begin{cases} \frac{1}{\sqrt{r_{min}}} & Distance(g_1, g_2) < r_{min} \\ \frac{1}{\sqrt{Distance(g_1, g_2)}} & r_{max} > Distance(g_1, g_2) > r_{min} \\ 0 & Distance(g_1, g_2) > r_{max} \end{cases}$$

In words, we set the value of  $PROX$  to 0 if the distance between two locations is greater than a threshold (hence “Weighted Neighbors.” That is, we are taking into account only photos within a certain radius from  $g$  (100 meters may be a reasonable such threshold). We did not use a linear distance measure since that measure assigns too much weight to nearby photos. We also capped the value for  $PROX$ , assigning equal values to all photos within a minimal distance. This cap avoids disproportionate weight bias induced by very close pictures. For example, a photo taken 20cm away should not be weighted much higher than a photo taken 1m away. In addition, the minimal distance is required given the equipment's accuracy limitations. Current GPS devices are accurate to about 5 meters; any distance measurement that is smaller than 5 meters is meaningless.

The IR function  $IR(t, \ell)$ , for our purposes, is a simple matching function:

$$IR(t, \ell) = \begin{cases} 1 & \text{if term } t, \text{ in singular or plural, is in label } \ell \\ 0 & \text{otherwise} \end{cases}$$

For example, when searching for the term “tower” we gave the same score to the



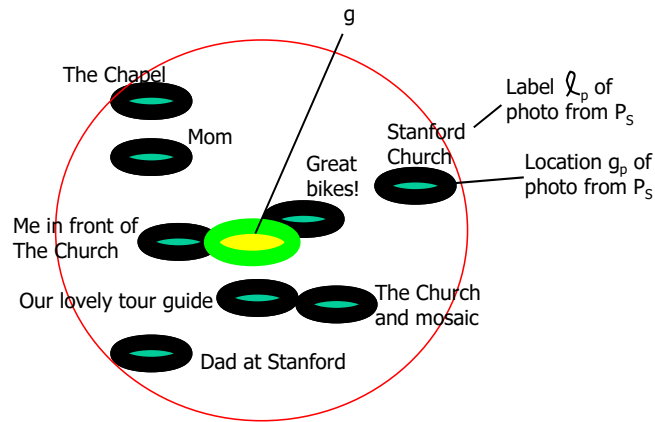


Figure 7.2: Sample Weighted Neighbors computation scenario.

labels “Towers” and “The tower - what a tall tower!”

Figure 7.2 shows a simple example. In the figure we show the basis for computation of  $Score(g, \text{“Church”})$  for a single location  $g$ . While theoretically we consider all the photos in  $P_S$  when assigning a score for  $g$ , in practice, as  $PROX$  assigns score 0 to photos beyond a certain radius from  $g$ , the system only has to consider photos in that range. These photos are shown in Figure 7.2 as black ellipses, together with their labels as they appear in  $P_S$ . Location  $g$  is shown as a bright-colored ellipse in the center. Three photos in the figure have the term “church” in their label. These photos will contribute to  $Score(g, \text{“Church”})$ . The contribution will be scaled by the distance of each photo from  $g$ , as reflected by the  $PROX$  function. For example, the photo labeled “Me in front of the church” will contribute more to the score than the “Stanford Church” photo.

After computing the score for each  $g \in G_1$  value (all locations of photos in the user’s collection), table  $Y$  is constructed:  $Y = \{(g, s) | g \in G_1; s = Score(g, t_q)\}$ . Table  $Y$  is the reply of the LOCALE server to the query; as mentioned above, this table ranks the geographic locations of the user’s photos in terms of their match to the search term. The LOCALE system on the user’s machine computes  $P_1 \bowtie Y$  to produce a ranking on images instead of locations.

**Location-Clustered (LC) LOCALE.**

Under the Location-Clustered LOCALE strategy we introduce some pre-processing on the location/label data. In this strategy, the LOCALE server clusters the  $P_S$  table geographically using a hierarchical clustering algorithm. Then, LOCALE uses term-frequency methods to rank terms according to how relevant they are to each cluster in the hierarchy. The output of the pre-processing step is a clusters/terms table  $CT(\underline{C}, \underline{T}, E, F, P)$ . Each tuple in the table is composed of a cluster (C), a term (T), the cluster's geographical extent (E), the frequency of the term in pictures of this cluster (F), and the parent cluster (P). The keys of the table (underlined) are C and T: there is one tuple for each cluster and term.<sup>7</sup> For example, many pictures are taken in front of Stanford's Hoover Tower; but at the same location one can turn around and take a photo of Memorial Auditorium. Assuming all these photos are geo-clustered together in geographic cluster  $c_1$ , two tuples of the format  $(c_1, \text{"Hoover Tower"}, e_1, f_1, p)$  and  $(c_1, \text{"Memorial Auditorium"}, e_1, f_2, p)$  will appear in  $CT$ . Here,  $f_1$  and  $f_2$  are the frequencies in  $c_1$  of "Hoover Tower" and "Memorial Auditorium", respectively. The geographic extent of  $c_1$  is represented by  $e_1$ , and  $p$  is the parent cluster of  $c_1$  (for example,  $p$  could be a cluster that includes all photos taken around campus).

During search, the ranking of User 1's photos is again spawned by a term-locations query with parameters  $t_q, G_1$  to the LOCALE server. For each  $g \in G_1$  the LOCALE server assigns  $g$  to the closest leaf cluster  $c_\ell$ . The cluster hierarchy is then ascended. Define  $c_\ell$ 's ancestors to be  $ANC(c_\ell)$ , and a view  $CT_\ell$  which corresponds to all the ancestors of cluster  $c_\ell$  in the hierarchy, including  $c_\ell$  itself:  $CT_\ell = \sigma_{c \in \{c_\ell\} \cup ANC(c_\ell)}(CT)$ . Then the match of location  $g$  to the search term  $t_q$  is computed by

$$Score(g, t_q) = \max_{ct \in CT_\ell} IR'(t_q, t_{ct}, f_{ct}) PROX'(g, e_{ct})$$

In other words, the score for the search term and the current geographic location  $g$  is taken from the cluster in the hierarchy that maximizes the geographical match to

---

<sup>7</sup>Notice that the table  $CT$  is not normalized, as the cluster's identity determines the extent and parent attributes.

$g$  and the text/frequency match to the search term  $t_q$  at the same time.

Function  $PROX'$  is based on the probability of  $g$  belonging to  $ct$ 's extent,  $Prob(g \in e_{ct})$ , and (inversely) to the area of the cluster (the more broad  $e_{ct}$  is, the less we value the match). The extent is represented by a two-dimensional Gaussian distribution, and the probability is simply the probability point  $g$  is part of the given Gaussian model. The hierarchical clustering algorithm we used in LOCALE to create the cluster hierarchy is agglomerative clustering (see [41]). Other options for hierarchical clustering are possible as well. For example, one can employ a simple grid hierarchy instead of clusters.

The function  $IR'$  is based on the frequency of  $t_q$  in tuples of cluster  $ct$  in  $CT_\ell$ , but takes into account the sum of frequencies over all other terms that appear within  $ct$ : the fewer other terms appear within  $ct$ , the more relevant  $t_q$  is.

For instance, Figure 7.3 shows a hierarchy of clusters. The current photo's location  $g$  is shown in a lighter color. The locations of photos in  $P_S$  are shown in black. The extents of clusters in the hierarchy are marked by a line. We can see that  $g$  belongs to leaf cluster  $c_1$ , which is the descendent of  $c_2$  and  $c_3$  in the hierarchy. Say the photos marked by an 'X' are photos whose label include the term "church". The system now assigns  $Score(g, \text{"church"})$ . While  $c_1$  is a very good geographic match to  $g$ , the relative frequency of the term "church" in  $c_1$  is low (only 2 of 8 photos in the cluster). Cluster  $c_2$ , while a less significant geographic match, has higher relative frequency for "church". Therefore, the score for location  $g$  and the term "church" will probably be derived from cluster  $c_2$ , as  $c_3$  is both too broad geographically as well as sparsely populated with the term "church".

Now imagine the system trying to assign a score to another user location  $g'$  with respect to the query term "church". Let's further assume  $g'$  belongs to leaf cluster  $c_4$ . It would seem, in that case, that the score for  $(g', \text{"church"})$  will be derived from  $c_4$ , and will exceed the score given to  $g$ .

As in all other centralized-mode computations, the LOCALE server returns a table  $Y = \{(g, s) | g \in G_1; s = Score(g, t_q)\}$ . This is the reply of the LOCALE server to the query; the LOCALE system on the user's machine computes  $P_1 \bowtie Y$  to produce a ranking over images pertaining to their match to the search term.

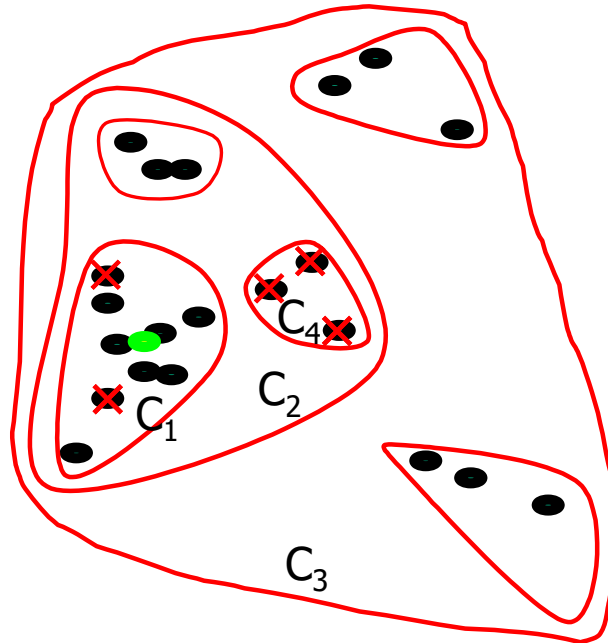


Figure 7.3: Sample Location-Clustered computation scenario.

### Term-Clustered (TC) LOCALE.

Under the Term-Clustered LOCALE strategy, the server pre-processes the label/location database to compute the geographical extent, or extents, of every term (one- or two-word phrase) that appears in the labels. For example, the algorithm may determine that the term “Hoover Tower” corresponds to two areas: one adjacent to the tower, and the other at a good viewpoint some 500 meters away from where many photographs of Hoover were taken (the “picture spots” discussed above and in more detail in Section 7.5. See Figure 7.4 for the clusters corresponding to the term “fountain” in our experiment, presented on a map of Stanford campus; for illustration purposes, we manually marked the extents of the four main clusters in the figure.

At the end of the pre-processing step, we have a table  $TC(\underline{T}, \underline{C}, E)$  of terms, clusters, and the clusters’ geographical extent described by a two-dimensional Gaussian distribution. As we may have a few clusters for each term, both attributes together are a key in this table.

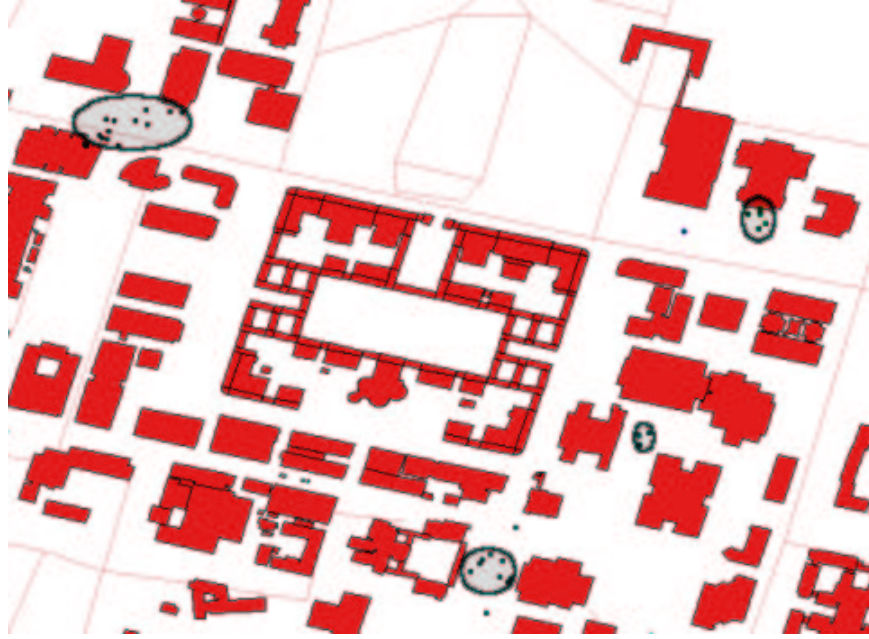


Figure 7.4: Map of Stanford campus, with geographical distribution of photographs whose labels contain the term “fountain”.

As usual in centralized mode, the user search is translated to a term-locations query with parameters  $t_q, G_1$  and sent to the LOCALE server. For each user photo location  $g \in G_1$ , the server assigns a score according to the geographical match between  $g$  and the clusters associated with term  $t_q$ :

$$Score(g, t_q) = \max_{tc \in TC} IR''(t_q, t_{tc}) PROX''(g, e_{tc})$$

The function  $PROX''$  is based on the probability of  $g$  belonging to cluster  $tc$ 's extent,  $Prob(g \in e_{tc})$ . As for the LC strategy, we use an agglomerative clustering algorithm. However, since TC required “flat” (one-level) clusters we flattened the cluster tree bottom up, merging adjacent nodes until we reach sibling clusters that are further than 50 meters away from each other. This provided sufficient results since we found that in the case of terms (like “fountain”) that have a number of extents, the extents were distinctly remote from each other. One would expect that in any dataset, terms that are not uniformly distributed (i.e., geographically meaningful)

will indeed have a natural clustering tendency.

The function  $IR''$  is the equality in singular form ( $IR(t_q, t_{tc}) = 1 \Leftrightarrow singular(t_q) = singular(t_{tc})$ ). In other words, we only consider clusters of terms that match our search term  $t_q$  in plural or singular.

The LOCALE server then replies with  $Y = \{(g, s) | g \in G_1; s = Score(g, t_q)\}$ .  $Y$  is joined with  $P_1$  to produce a ranking of  $u_1$ 's images.

In the process described above, we generate the clusters and extents for every term, even some terms that are not meaningful geographically. For example, the words “mom”, “bicycle”, “student” appeared in the labels but are not associated, of course, with a specific location. Indeed, we expected these terms to be randomly distributed around campus. We studied mechanisms to identify such high “entropy” terms, in order to flag those terms as *geographical stop-words* and skip pre-processing for them. We did not find an accurate enough mechanism, mostly, we suspect, because of the limited scope of our experiment.

Similarly, we could extend the IR functions to capture the notion of Inverse Document Frequency (IDF), by calculating the number of regions in the map where each term appears. Terms that appear frequently in one area, but less frequently in others are likely to be more relevant to the former area. For the scope of our experiment, such calculation was not necessary, but we suspect that the calculation will be required when the dataset grows to cover larger areas.

### 7.1.2 Distributed Mode

So far we have described the different strategies' execution in centralized mode, where the LOCALE server is contacted at query time and performs all the computation. This section presents a modification of the strategies to execute in distributed mode, where some information is retrieved from the LOCALE server in advance, and stored in the client.

User 2 in Figure 7.1 serves as illustration for the distributed mode interaction. As mentioned above, the LOCALE computation is executed in two steps. In the first step, performed in advance, the LOCALE server is used in conjunction with the

location data in User 2’s collection  $G_2 = \Pi_G(P_2)$  to create a new *term-score table*  $TS_2(I, T, S)$  of User 2’s images (I), possible matching terms (T), and the score (S) of the match between the image and the term. In effect, the first step results in a list of possible terms that match each of the photos in  $P_2$ . To this end, the user’s machine submits a *term-retrieval query* to the LOCALE server with the photo locations  $G_2$ . The location-term score is then computed – the computation method is different for each of the three strategies. The reply from the LOCALE server to the term-retrieval query consists of a table  $Z(G, T, S)$  of locations, terms and scores.

Going back to our early example with users M and H, the reply  $Z$  to a query by M’s machine, which includes the location of the church photo  $g_c$ , may include a tuple  $(g_c, \text{“Stanford Church”}, s)$ . The reply is partially based on the label submitted by H earlier, where the score  $s$  is based on the distance between H’s and M’s photos. However, the score  $s$  is likely to also incorporate other photos labeled “church” that were taken nearby. In addition, the reply will also have tuples representing other terms, for example,  $(g_c, \text{“Quad”}, s')$ .

At the end of this first (advance) step,  $Z$  is joined (on attribute G) with User 2’s photo table  $P_2$  to generate  $TS_2(I, T, S)$ . Notice that the geographic information can now be discarded. Also notice that the LOCALE server need not be contacted further after we constructed the  $TS_2$  table. User 2’s machine retains  $TS_2$ , and it may also choose periodically to update it.

In the system described in this work, we use the  $TS$  table to enable search for photos on the user machine. However, other ways to utilize  $TS$  are available. For example, the system can suggest to the user possible labels for each photo. The label suggestions are based on the terms as ranked by the  $TS$  table. The users, then, will have a way to choose labels from a drop-down interface (or some input method that may support more rapid input), without having to type one in — if indeed a matching label exists in the table for each of the photos. We can thus obliterate the uncertainty in which label actually matches the given photograph in a single collection.

We discuss label suggestions in Section 7.4, but did not focus on enabling label suggestions in our application. Instead, the second step occurs when the user *actively searches* for photos by a textual search term, alleviating the need for selecting a

specific label from a list. This search step is performed in the same manner for all implementation strategies in distributed mode. The search is performed when the user submits a query for term  $t_q$ . The search is done directly on table  $TS$  - no other data is required. The system looks for possible matches to the search term  $t_q$  in the  $T$  column of  $TS$ . The lookup result, as in the centralized case, is a ranking of the photos  $\Pi_I(P)$  based on an adjusted score of each image  $i$  with respect to the search term  $t_q$ . The adjustment is based on the “evidence” in favor of the search term, in contrast to evidence against it for each photo.

Table 7.1: Sample term-score table  $TS_M$  for User M

$I$	$T$	$S$
$i_c$	Church	30
$i_c$	Quad	15
$i_k$	Church	30
$i_k$	Quad	200

For example, going back to user M – suppose M has taken two images,  $i_c$  and  $i_k$ . The term-score table for M appears in Table 7.1. In this example, there are two tuples in  $TS_M$  for the “Church” photo  $i_c$ . The initial score of 30 for  $i_c$  and term “Church” does not tell the entire story. Obviously, it is more likely that  $i_c$  is a picture of the church than  $i_k$  (which is probably a picture of the Quad). For this reason we use a correction factor, the ratio of the score to the total score of terms suggested for this photo. In this case, the final score for photo  $i_c$  and the term “Church” will be  $30 \times \frac{30}{30+15}$ .

Formally, if a tuple  $(i_c, t_q, s)$  appears in  $TS$  we adjust  $s$  by the total of scores for image  $i_c$ . The final score is computed as follows:

$$Score(i_c, t_q) = s \times \frac{s}{\sum \Pi_S(\sigma_{i=i_c}(TS))}$$

Finally, all the images in  $P$  are ranked according to their computed match score with  $t_q$ , and returned to the user.

As usual in distributed problems, there is a tradeoff between the accuracy of



distributed processing and the amount of data stored on users' machines. The good news is that the user's machine does not have to hold all the information available on LOCALE: the information is confined to the areas where the user has taken photos, and summarized as described above by keeping only the top-scoring terms for each photo. In fact, in our experiments, the term-score table only kept the top 15 terms per photo. We show (Section 7.3) that with the top 15 terms we achieve search results comparable to the centralized mode. Allowing more terms did not improve recall, as terms that were not ranked within the first 15 were, as we hoped, and at least in the context of our experiment, not relevant to the photo.

We now describe how the LOCALE server handles distributed mode term-retrieval queries (i.e., generates a list of possible terms) under the different LOCALE implementation strategies.

### **Distributed Weighted-Neighbors (DWN) LOCALE.**

Recall that the parameter of the term-retrieval query to the LOCALE server includes only the locations of the user's photos  $G = \Pi_G(P)$ . A reply is a table  $Z(G, T, S)$  of terms matching each location, and their matching scores.

In the centralized implementation of Weighted Neighbors LOCALE, as shown on page 152 (in Section 7.1.1), we compute a score for each location  $g \in G$  with respect to a *specific* term. In Distributed Weighted Neighbors, we compute a score for each location and *each term* that appears in its vicinity, resulting in table  $Z$ . Recall that this step is performed in advance, before the user submits a search.

To illustrate, refer back to Figure 7.2 on page 154. The tuples that correspond to location  $g$  in table  $Z$  include scores for *all* the terms that appear in the vicinity of  $g$ . For example,  $g$  will have a score for the term "church", computed the same way as in Section 7.1.1. In addition,  $g$  will be associated with a score for the other terms, like "Stanford", "mom", "bike" etc. The reply table  $Z$  will include all these tuples, and the scores for each.

More formally, we compute  $Z$  by selecting possible terms from neighboring photos (photos in  $P_S$  taken in proximity to  $g$ ) for each location  $g \in G$ . More formally, we compute a score for every term  $t$  that appears in  $P_S$ , with respect to the location

$g$ :  $Score(g, t) = \sum_{p \in P_S} IR(t, \ell_p) PROX(g, g_p)$ . The  $IR$  and  $PROX$  functions are as defined in Section 7.1.1. The table  $Z(G, T, S)$  is then constructed;  $Z = \{(g, t, s) | g \in G; s = Score(g, t)\}$ . As described above, the user's machine joins table  $Z$  with  $P$  to generate the term-score table  $TS$ .

### **Distributed Location-Clustered (DLC) LOCALE**

In *centralized* Location-Clustered mode, as shown on page 155 in Section 7.1.1, the system generates a location cluster hierarchy. Roughly speaking, the system assigns each location  $g$  in the user's photo collection to a cluster (recall that the clusters are computed by the system in advance). Then the cluster hierarchy is ascended, and the match of the location to the search term is derived from the cluster that maximizes the location match to the cluster, and the cluster match to the search term.

In distributed mode, as explained above, this is no query term when contacting the LOCALE server. Instead, LOCALE needs to generate the table  $Z$  of locations, terms and scores. Instead of generating a score for location  $g$  with respect to a query term, LOCALE generates a score for  $g$  and every term that appears in the labels of photos in the location cluster  $g$  is closest to, and in labels from that cluster's ancestors. The scores are generated as described on page 155, where the analogous centralized mode was described.

Finally, the table  $Z(G, T, S)$  is constructed and send back to the client. The users can then perform searches on their local machine, as described above.

### **Distributed Term-Clustered (DTC) LOCALE.**

The distributed version of the Term-Clustered strategy is very similar to the centralized implementation. Recall that the centralized Term-Clustered strategy creates a mapping of terms to their geographic extents. When a query arrives, each of the photos in the client's collection is ranked based on the fit of the photo's location to the term's geographic extents.

In distributed mode, the system simply generates a match for each location and each possible query word. In other words, for each location  $g$  of the user's photos, and

for each possible word, the system goes through the process described in Section 7.1.1 on page 157. The system retains, for each location, the terms that ranked highest according to this scheme, thereby creating the table  $Z$ . The table is joined with the table of the user's photos to create the table  $TS$  as described above.

In reality, the system does not need to iterate on every single term for each location  $g$ . A simple location-based index can map from each area to the terms that appear in that area, therefore reducing redundant computation.

## 7.2 Experiment

We ran an experiment to see if the LOCALE system is effective in terms of executing the following user task: “find among my unlabeled pictures the ones that best match the term  $t_q$ .” In particular, we sought to determine which implementation strategy offers the best result. We also wished to determine whether the results of the distributed search are comparable to the centralized search. Finally, the experiment would help us tune the system's parameters.

For this experiment we required a dataset of labeled, geo-referenced photos. Moreover, we needed a high concentration of such photos in a single geographic area; otherwise it would not be possible to obtain sufficient amount of aggregate data for any single area. We therefore limited the dataset to a bounded “world” – in our case, the Stanford University campus. Every day, tourists take photos on the campus, and we made use of these tourist visits to collect data for our experiment. The underlying “picture spot” assumption that many photos are taken by different people in the same place is certainly supported by the tourist photo-taking patterns on Stanford campus.

### 7.2.1 Experimental Setup

We provided loaner cameras and GPS devices to visitors taking the Stanford Visitor Center's campus tour. The tour, and therefore our dataset, was limited to one part of campus (albeit the most photographed one). We asked for volunteers among the

groups that were taking the tour. The volunteers were instructed to take photos at their leisure, as if the loaner were their own camera. The GPS devices continuously tracked and logged their carrier's location. After the tour we collected the cameras and GPS units for the participating visitors. Some hours later, the participants were sent an email message that asked them to enter labels for their photos on a web page we had prepared for this purpose and that we promised to host for them. Most of the participants completed this task a few hours to one week after the end of the tour, much in the fashion of people labeling their own photos upon return from a trip. The participants were instructed to label their photos for their own use: the labels would be used as captions for their online photos and on a photo CD that we sent them in return for their effort. The hosting of photos and the photo CD served as incentives for people to participate in our experiment and to label their photos. We requested that participants label as many of their photos as they liked, but they were not *required* to label even a single photo.

We used software to “align” the GPS time-stamped track log and the corresponding photo timestamps, thus finding for each photo the location of the GPS device, and therefore the camera, at the time of the photo (see Appendix A). We thereby created a geo-referenced collection  $P_u$  for each tourist's photos. The procedure produced geo-referenced photos with accuracy of roughly 10 meters.

We lent cameras to 52 visitors who took an average of 20 photos each. A total of 37 of the participants visited our web site to submit labels. We collected 761 labeled photos, 460 of them were successfully geo-referenced. The primary reasons for un-referenced photos were bad GPS reception (e.g., inside buildings, underpasses) and incorrect handling of the equipment (holding the GPS unit out of clear view of the sky). For those labeled photos that were not referenced due to incorrect handling, we manually added location stamps: our knowledge of campus allowed us to determine where each photo was taken. At the end, we had 672 labeled, geo-referenced photos.

The label/location data was prone to problems, some specific to our experiment and some more general. Specific to our experiment are visitors who clearly labeled the photos for no other reason than pleasing us (“Building 1”, “Building 2”, “Building 3”). One set of labels was clearly produced this way and was removed from our data.

Another problem is derived from the new use patterns of digital cameras: people take many more photos than a typical film camera owner would take. In our experiment, this effect sometimes reduces the accuracy of labeling (in real life this may not be the case since people may not label “uninteresting” photos, while in our experiment such labels were sometimes assigned). Another specific problem is related to our idea of assigning lower weight to terms that appear in many different locations. To use an example, the term “mom” did appear in specific locations within our limited experiment, and was not uniformly distributed as we expected. The reason was that photos of “mom” were often taken in the most popular locations of campus, thus highly concentrated in few locations. However, we suspect that in a global database of geo-referenced locations, we will be able to detect that the term “mom” does tend to appear uniformly across the globe, and will assign it a lower weight.

Other problems may appear under any kind of settings. First, since we did not restrict the labels in any way, some of the labels (“Our tour leader: a fine young woman”) do not contribute any relevant geographical information. Second, tourists everywhere tend to be less knowledgeable about landmarks and their names than locals may be. Thus, in many cases the labels were not accurate or just plain wrong. We retained this data since such inaccuracies reflect the realities of collective labeling and were thus pertinent to the experiment. Indeed, the results show that our system was able to overcome these issues.

## 7.2.2 Experiment Procedure

We performed keyword searches over various users’ collections using our LOCALE database of 672 labeled, geo-referenced photos of Stanford’s campus. A human referee decided on the relevance of the retrieved images. Strict relevance measures were applied: a result image was deemed to match a search term if and only if an object described by the search term clearly appears in the image. Figure 7.5 shows the three top-ranked results for the query “Hoover Tower” on one of the collections. For the purpose of our experiment, the top two photos were determined relevant to “Hoover Tower”; the third is not relevant even though the tower would be visible from the



Figure 7.5: LOCALE Search results for “Hoover Tower” query.

position where the photo was taken.

Incidentally, the location where the third-ranked photograph in Figure 7.5 was taken is precisely between the locations where the other two photos were taken. This fact shows that there are other factors coming into play in LOCALE other than the distance to the photo’s subject. In particular, at the location where the third photo was taken, the trees (visible in the second photo) obscure the tower. For this reason, the location of the third photo was not associated with as many photos that had the term “Hoover” in their label, and the third photo’s score for “Hoover” match was low. The dichotomy of camera location and object location is discussed in more detail in Section 7.5.

We performed our evaluation for two different “scenarios”: a *global* and an *individual* scenario, as described below. For each scenario, we require:

- A collection of geo-referenced yet unlabeled photographs on which we can perform search.
- A set of search terms we can test on this collection.

For the global scenario, the collection we used was the set of photographs taken by those visitors who never accessed our web site to label their photographs. We had a total of 253 such photographs. This collection emulates a multi-user pool of photographs such as an image database. The search terms we used to test this collection were chosen from the pool of terms that appeared in all the *labeled* photographs we collected in the experiment, with two conditions: a) The term appears at least four times in the LOCALE database and b) The term is meaningful in *some* geographical manner. For example, we did not include terms like “car” or “student”, but *did* include “fountain” and “mosaic”. We also excluded search terms that match all the photos in our collection like “Stanford” or “campus”. We retained a total of 27 qualifying terms.

For the individual scenario, we picked user collections that *were* labeled, removed their labels, and used the labels as a source for search terms. Each collection comprised pictures taken by *one* visitor. Search on these user collections better emulates search on a personal collection of photos than the global scenario. The collections we picked for the individual scenario had to have a reasonable number of photos ( $> 25$ ) and some labels that are geographically meaningful. There were 13 such collections in total. For each collection in turn, we removed the collection’s labels from the LOCALE database  $P_S$  before performing the search tasks. The search terms for each collection were picked so that they a) appear in the user’s own (removed) labels and b) are meaningful geographically as described above. In picking only terms that appear in the user’s labels we are able to simulate a “personal search”: we search for terms as the user thought about them – for example, someone may want to locate their photo of the “chapel” while most people labeled their photo of the same building “church”. An average of 8.7 search terms per collection were picked. A sample of the query terms picked for one collection is: “Hoover, tower, engineering building, fountain, clock fountain, palm, Quad, arches, chapel, mosaic, residence.”

## 7.3 Results

We first discuss the results for the individual scenario. Then we discuss the results for the global scenario.

### 7.3.1 Individual Collection Scenario

As a first step we examine which strategy performs best for the individual scenario, and compare the distributed and centralized implementations. We looked at how many of the queries were *satisfied* – returned at least one relevant photo (a photo matching the search term according to a human referee). We executed the queries as described in Section 7.2.2, while limiting the number of photos retrieved to one, two and three photographs. Often, the actual number of photos with a score greater than 0 was smaller than the set limit. The results for the different strategies, averaged over all collections and queries, are shown in Figure 7.6(a). On the X-axis we identify the strategy (by acronyms - WN for Weighted Neighbors, DWN for Distributed Weighted Neighbors and so forth). The Y-axis shows the percentage of queries that returned at least one relevant photo within one, two and three retrieved photos. For example, WN produced a relevant photograph within the first three photos retrieved in 72% of the queries. The “random” strategy reflects the expected values when the results are completely random, and is included as a baseline for comparison.

A number of conclusions follow from Figure 7.6(a). We can see that all the strategies performed better in centralized mode than they did in the corresponding distributed mode (strategy name starts with ‘D’). For example, when the retrieval limit was set to 3, WN satisfied 72% of the queries while DWN satisfied only 58%. Part of this difference can be attributed to the summarization done in the distributed modes, as described in Section 7.1.2. Less popular terms may not score high enough to make it into the summary of a relevant photo, and therefore the photo will not be retrieved. We try to address this issue in more detail when discussing the results of the global scenario.

The best performing strategies were WN, LC, and DWN. The percentage of queries satisfied by the *first* retrieved photo is around 40-50%. This percentage, at first



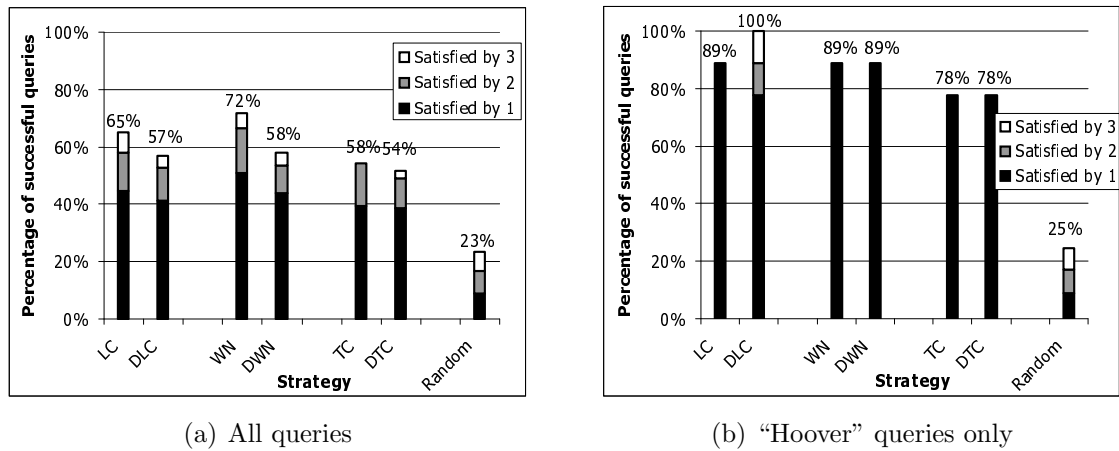


Figure 7.6: The percentage of individual scenario queries that returned a relevant photo within first three results, for each strategy.

glance, appears low, but remember that the search terms were often low-frequency terms (e.g., “red fountain”). Compare these numbers to the baseline results of the “random” strategy where the probability that the first retrieved image matches the query is less than 9%. By the third retrieved image, 50-70% of the queries were satisfied (at least one relevant photo was retrieved).

Performance is improved significantly when concentrating on more common terms. To illustrate, Figure 7.6(b) shows the same metric limited to queries for the popular terms “Hoover” and “Hoover Tower”. Of our 13 collections, our term selection procedure generated “Hoover” or “Hoover Tower” in 9 instances. We used these 9 collections to produce Figure 7.6(b). In all the strategies, the first photo retrieved was relevant (i.e., a picture of the tower) for 78% of the queries or more. By the third image retrieved, at least one image of Hoover Tower was found in 78–100% of the queries (in other words, the query was successful for 7–9 of the collections, depending on the strategy).

Based on the results in Figure 7.6 we decided to concentrate on Weighted Neighbors, Distributed Weighted Neighbors, and Location Clustered strategies for the rest of this discussion of the individual scenario.

Instead of the aggregate results presented so far, we now drill down to the level of

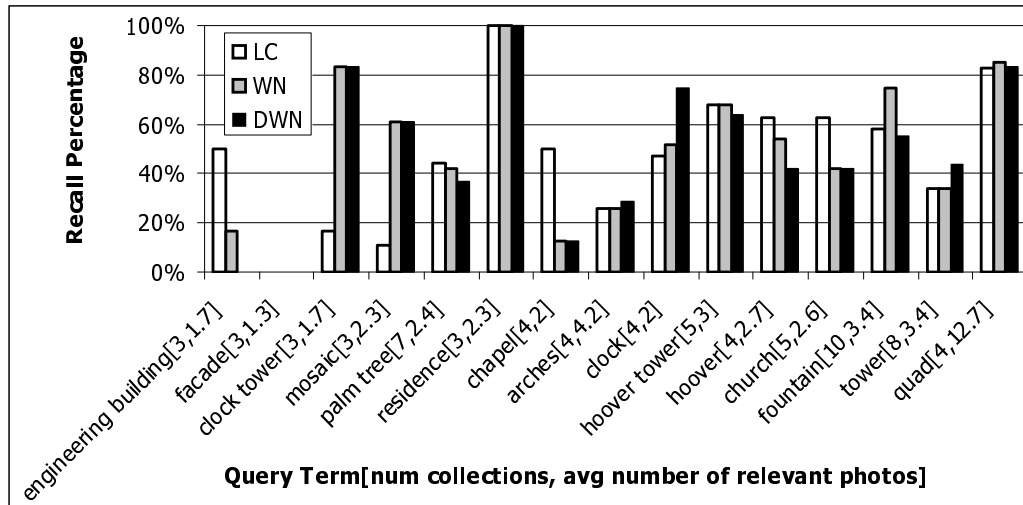


Figure 7.7: Average *recall* at  $T(t, c)$  for popular query terms.

query terms. We wish to examine the variability between the strategies when handling particular query terms. Figure 7.7 shows *recall* results for the most popular query terms: the X-axis lists query terms that were used across at least three individual collections (remember: for each collection we picked its own query terms from its original labels). The first number in square brackets next to each term is the number of collections we queried with this term. For example, the query term “fountain” was picked for search in 10 out of the 13 collections. The second number within the brackets is the average number of relevant photos in these collections. For example, the “fountain collections” have on average 3.4 photos of a fountain. The terms are presented from left to right in order of rising popularity in the label database.

Recall is usually measured at some pre-defined number of retrieved results: how many of the relevant photos were retrieved when the retrieval is limited to  $n$  photos, and computed as  $\frac{\text{Relevant Retrieved}}{\text{Total Relevant}}$ . As mentioned above, the number of relevant photos for each term and collection varied extensively. Thus, using a fixed retrieval limit for all terms would have produced varied and inconsistent results, which would be hard to evaluate. We therefore set a different retrieval limit for each term and collection combination: the number of relevant photos for that term in the collection.

For each collection and term, we manually counted  $T(t, c)$ , the total number of photos in collection  $c$  that are relevant for term  $t$ . Then we submitted the query  $t$  over collection  $c$ , while limiting the number of retrieved photos to  $T(t, c)$ . Finally, the recall was computed by dividing the number of relevant photos retrieved by the  $T(t, c)$  value.<sup>8</sup>

The bars in Figure 7.7 group the recall results by term, simply by averaging the computed recall over all collections queried with this term. For example, the average recall at  $T(t, c)$  for the “fountain” query in LC mode was 60%.

Generally, the performance of all three top strategies based on Figure 7.7 is comparable. The average recall at  $T(t, c)$  is usually between 25-75%, and on average higher than 45%. The DWN strategy performs almost as well as the centralized WN; in a few cases it even outperforms the centralized implementation. This fact reaffirms our thesis that the unpopular terms are responsible for the lower performance of the distributed strategies in Figure 7.6(a).

We have no intuition for why WN/DWN perform much better than LC for some terms, and much worse for others. Possibly, a larger experimental dataset could assist in bringing out more prominently the differences between the strategies.

### 7.3.2 Global Collection Scenario

In the global scenario, we have one collection that is the union of all unlabeled collections. As explained in Section 7.2.2, we had 253 photographs in this collection. We start by comparing the results to the individual scenario. We then compare the different strategies and modes in more depth, and pick three strategy/mode combinations for extended evaluation.

How different is retrieval in the global scenario? Figure 7.8 compares the percentage of queries *satisfied*, the same metric used in Figure 7.6. However, we limited the queries to terms appearing both in the global and in the individual scenario, and we show the stacked results side-by-side for each strategy. The bars corresponding to the

---

<sup>8</sup>Note that when the retrieval is set to  $T(t, c)$ , the recall is equal to the precision, which is computed as  $\frac{\text{Relevant Retrieved}}{\text{Total Retrieved}}$ , since we set Total Retrieved = Total Relevant. Figure 7.7 therefore represents precision, as well as recall values for each query term.

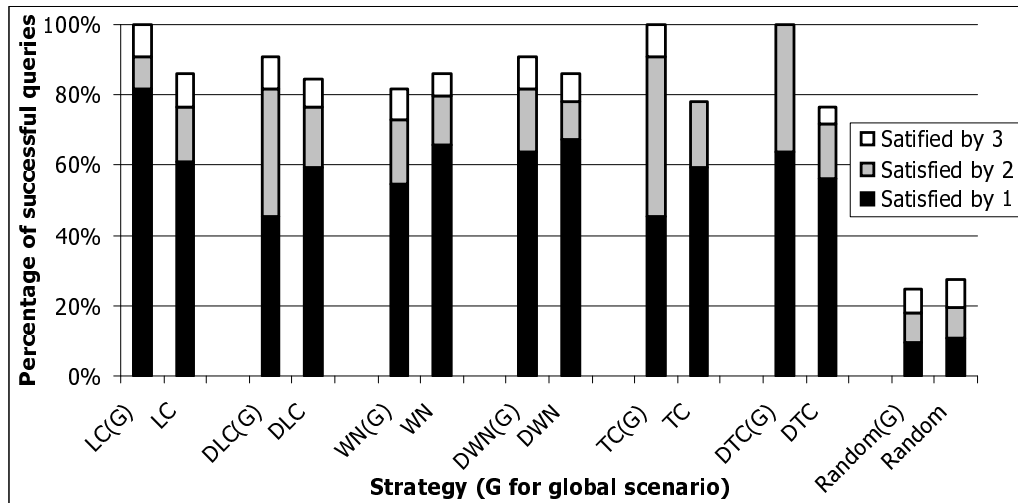


Figure 7.8: The percentage of queries in each strategy that found a relevant photo in first three results, for global (G) and individual scenarios.

global scenario are noted with (G). Again, random retrieval is shown as a baseline.

Interestingly, for most strategies, the first result in the individual scenario was relevant more often than in the global scenario; but by the third result, more queries found a match in the global scenario. The reason for this phenomenon, we believe, is “cluttering” in the global scenario. The individual photos in each collection tend not to be as close to each other as in the global scenario. Take for example the Term-Clustered (TC) strategy described in Section 7.1.1. Once our system identified geographical extents that correspond to a term, we are likely to find fewer photos from that area in an individual collection than we may find in a global collection. Therefore, the first result is more precise for the individual collection. However, if a match is not found in the first result, there are more match prospects (candidate photos from the same area) in a global collection.

The previous discussion was limited to a subset of the query terms and a small number of retrieved results, in order to have a basis for comparison between the results in the different scenarios. To investigate in more depth how the strategies perform in global scenario, we expand on this evaluation. Since our global collection is large, we can now use standard IR measures such as recall, precision, and  $F_1$ .

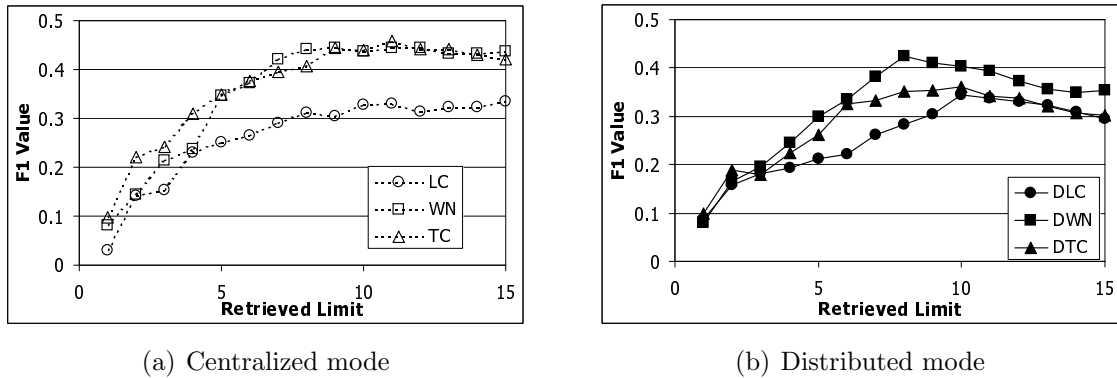


Figure 7.9: Average  $F_1$  values for least frequent query terms in different strategies, vs. retrieval limit.

To compare the different mode/strategy combinations, we looked at the  $F_1$  values over varying numbers of retrieved photos (1 to 15), averaged over all queries. The  $F_1$  measure combines precision and recall into a single metric that represents the value of the results to the user. The measure is computed using the formula  $F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$ .

The search results for the 10 *most* frequent terms demonstrated no significant differences in  $F_1$  value among the strategies, and hence we do not show results for them. Instead, Figures 7.9(a) (strategies in centralized mode) and 7.9(b) (distributed mode) show the  $F_1$  values of search for the 10 least frequent query terms – the terms used in the global scenario that appear the least number of times in the label database. Recall from Section 7.2.2 that the terms we used appeared at least 4 times in the labels of photos from the experiment.

The X-axis in Figure 7.9 corresponds to the photo retrieval limit. The Y-axis shows the  $F_1$  value for each strategy. Although values for  $F_1$  range from 0 to 1, the maximum possible  $F_1$  value at each point is not 1, but is dependent on the maximum possible recall/precision at that point. The optimal  $F_1$  values are not shown in the figure in order to keep the scale of the Y-axis, making the differences between the strategies more apparent. Sample values of the optimal  $F_1$ , for the figure's data, are 0.27 at retrieval limit of 1; 0.85 at 8 and 0.72 at 15. The average number of relevant photos for the query terms represented in Figure 7.9 is 8.1. As a consequence, the

optimal  $F_1$  is reached at the retrieval limit of 8 photos — beyond that, the optimal recall cannot improve (all relevant photos must have been retrieved), and therefore precision suffers as well. As seen in the figure, the retrieval limit of 8 is also the point where the actual  $F_1$  peaks for most strategies.

As in the individual scenario, we can see by comparing Figures 7.9(a) and 7.9(b) that all strategies perform better in centralized mode than in distributed mode. The LC and DLC strategies perform the worst while WN and DWN perform the best. For further evaluation, we picked the WN, DWN and TC strategies. As a reminder, WN and DWN were also the choice strategies in the individual scenario (together with LC).

Now that we have limited the discussion to three strategies, we wish to understand the variability between strategies and between queries. We drill down again and list the results by query terms. In Figure 7.10 we plot the recall and precision for each search term in WN, DWN and TC LOCALE. The retrieval limit is set to 15 photos. The search terms are displayed in order of popularity, as determined by the number of ground-truth relevant photos in the collection for each query (in square brackets). In Figure 7.10(a) the Y-axis shows the recall. In Figure 7.10(b), the Y-axis shows the precision. We can see how the more popular term's recall is lower (mainly because there are a lot more than 15 photos which are relevant) and precision is generally higher. While the results for WN and TC are comparable, it seems that recall for the distributed mode (DWN) is higher than the other strategies for the popular terms, yet slightly lower for the unpopular ones. Before we address this, we make a general observation about Figure 7.10.

One possible predictor of successful vs. unsuccessful query terms is the concentration of other “interesting” landmarks around the unsuccessful terms. Two interesting outliers in Figure 7.10 are “Gates” and “clock fountain”. There are a number of attractions around the Gates building. Thus, in a global collection it may happen that only, suppose, 15% of the photos taken from a Gates viewpoint are indeed pictures of the building; “Gates” retrieval precision is expected to be less than 15%. The term “clock fountain” offers a contrasting example. People seem to be fascinated with running water, and a lot of the photos in the areas where fountains are found were,

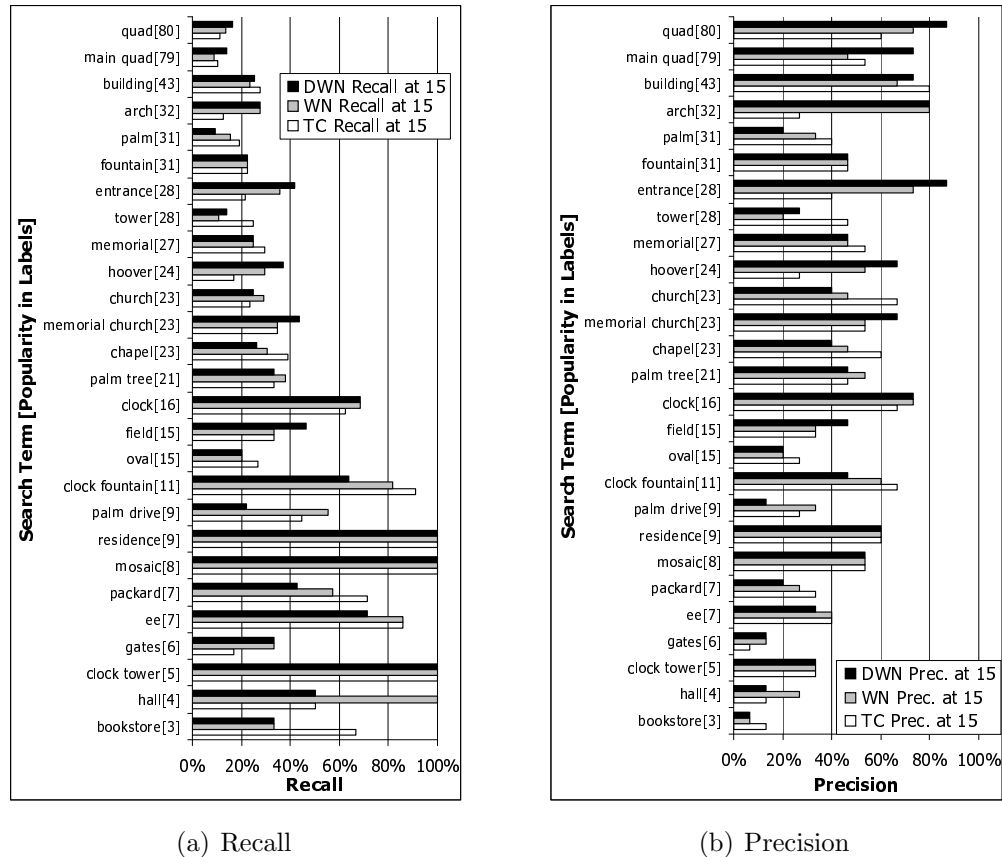


Figure 7.10: Recall and precision at 15 for each query term.

in fact, pictures of the fountains.

As we already hinted above, the problem of “cluttered attractions” was not as acute in the individual collection evaluations. The reason is that in an individual collection there are very few pictures taken at every location. Back to the Gates example, a single person may take one picture of Gates building and one of the other attractions around it; now, when looking for “Gates”, the precision should not fall under 50% (compared to the 15% in the global collection).

Going back to the less-popular terms, can we improve on the lower recall of DWN for them? A possible remedy is to enlarge the scope of summarization data for each photo. As we explain in Section 7.1.2, in distributed mode we only keep the 15 top matching terms for each photo. We tried the same less-popular term queries when

25 terms are allowed in the term-score table for each photo. Two of the queries, for “clock fountain” and “hall”, retrieved three and one (respectively) more relevant photographs than they did before, while the precision for all the queries did not change (i.e., no negative effect was noted on user query satisfaction). In summary, there seem to be marginal benefits in retrieving and storing more matching terms for each photo.

## 7.4 Automatically Assigning Captions to Photos

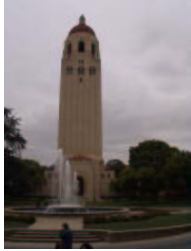



For LOCALE’s distributed mode, we used term-retrieval queries that collect potentially matching terms for each photo in the user’s collection. The system never exposes these suggestions to the users, as these terms are used during search only. But what if the system did make these candidate terms available? As we previously hinted, the system could suggest these terms as a location caption for photos. For example, we could automatically display the top-matching term for every photo as its caption when the user is browsing his collection. Alternatively, our UI design could enable the user to choose the appropriate term from the suggested term list. These UI considerations are similar to ones discussed in Chapter 6, where the system made suggestions for possible identities that appear in the photo.

Our algorithms were not tuned for this task, but we wanted to examine the prospects of this idea on a sample collection of photos. In Table 7.2 we show the top 5-6 suggested terms for a few of the photos in one of the individual collections. For each photo in the table, we list the objects that appear in it in the middle column, and the suggested terms on the right column. The terms were generated by the Distributed Weighted Neighbors strategy. These sample photos were chosen because the scores for their top suggested terms were especially high. Indeed, the photos correspond to Stanford’s most popular landmarks.

For example, the first photo depicts Stanford’s Hoover Tower, and the fountain in front of it. The suggested labels listed match the actual objects quite well. The word ‘tour’ appears as well, possibly since all our participants were taking the Stanford campus tour that starts from that location, and therefore the term appeared on



Table 7.2: Sample photos and terms suggested by LOCALE.

<i>Photo</i>	<i>Actual Objects in Photo</i>	<i>Suggested Terms</i>
	Hoover Tower, Fountain	Tower, Hoover, Hoover Tower, Tour, Fountain.
	The Oval	Quad, Oval, Main, Main Quad, Field, The Oval.
	Clock Fountain	Building, Fountain, Clock, Stanford, Gates.
	Main Quad	Church, Quad, Chapel, Stanford, Memorial, Memorial Church.

many of the photos taken at that point.

The second photo was taken in front of The Oval (an oval-shaped field), but Stanford's Main Quad is right across from it in the other direction, demonstrating the problem created by the lack of direction information. Even in this case, the Oval is the second-ranked option in the list of term. The fourth photo reflects the same problem of direction; the photo was taken in the Main Quad, in front of the Stanford church. Going back to the third photo, the photo shows the clock fountain that is in front of the Packard building and across from the Gates building.

Suggested terms for other photos in the collection were not as accurate; but the

terms were also scored lower, which demonstrates LOCALE’s appropriately low confidence in the match. LOCALE can use a threshold for label suggestions; only terms that score higher than that threshold will be suggested, even if no suggestions will be available for some of the photos. Alternatively, if LOCALE is used in a hierarchical fashion, and there is no high-confidence landmark to be suggested to the user, a label that reflects a more general area description could be suggested.

Another possible and similar use for these suggested terms, is using them as a “what’s in this photo” feature: the user may have forgotten, or possibly never recognized, the objects captured in the image. After connecting to the LOCALE system, the given terms can supply new information to the users that they wouldn’t have received otherwise.

Further work is needed to tune the algorithms that suggest terms to users. The thresholds are extremely sensitive, and concerns of exposing private information are greater than in the search scenario. However, we do believe that a reliable system that supports this feature can be implemented.

## 7.5 Capture Location and Object Location

The technology we build on in this thesis generates camera location metadata. In other words, we assume the *capture location* is known for every photo. Another type of possible location metadata is the *object location*: the coordinates of the object that appears in the photo.

In certain situations, the capture location is a very close approximation of the object location. For example, the capture and object locations are almost identical when taking a closeup of an individual. The two location types are still approximately the same, yet slightly less accurate, when taking a photo of a feature such as a building or a waterfall. On the other hand, in some situations, the capture and object locations may be miles apart. For an extreme example, consider California’s Mount Lassen and Mount Shasta. One can capture a photo of Shasta from the top of Lassen, dozens of miles away. Perhaps a more common example is found in tall urban landmarks. For example, Stanford’s Hoover Tower is visible for many miles around campus. It is

interesting to note that if the capture location is very close to the tower (e.g., right next to the tower, or actually on top of it), the photo is *least* likely to be one of the tower. To give a final example, many photos of Manhattan are taken from across the Hudson river, in New Jersey, such that the capture location is in a different state than the object location.

LOCALE translates the capture locations of photos into the objects and landmarks that may appear in them, without requiring the object location to be known. For example, if many photos of Mount Shasta are taken from the top of Mount Lassen, LOCALE would suggest the label Mount Shasta for photos taken in that location. The “picture spot” assumption made earlier claimed that people take the same photos from the same locations (and label them similarly). In other words, LOCALE is not sensitive to the distance of location of the pictured object; if the photo was taken at a location where many photos of the same object were taken, it is likely to be a photo of that object.

If the enabling technology for generating object location metadata were available, the accuracy of systems like LOCALE could nevertheless improve. For this technology, the camera needs to be able to detect not only the location where it is located, but also the direction it is pointed to, the tilt, and the distance to the photographed object. If all cameras could produce exact object location, the location uncertainty in our system could be eliminated. Without location uncertainty, the results can potentially improve as the problem will be restricted to term/label uncertainty.

Even more significantly, given object location metadata, the LOCALE task may be vastly simplified due to the existence of landmark databases. These databases and gazetteers, such as the Alexandria gazetteer [38], often include many important named landmarks and their location. Given the object location of a photo, a query to such a gazetteer would likely be able to identify the landmark in the photo.

As long as object location metadata is not available, are the object location datasets (or “landmark datasets”) useful? LOCALE creates a “capture location” dataset, because of the capture location enabling technology. However, it is possible for LOCALE to utilize a landmark dataset. For example, LOCALE can give a score boost to terms that appear in the landmark dataset in the vicinity of the capture

location, possibly enabling term suggestions even when the number of submissions from the capture location is relatively small. However, because of the issues with capture versus object location discussed above, a landmark dataset alone, without a capture location dataset, is not likely to be useful.

## 7.6 Conclusions and Future Work

Our LOCALE prototype system addresses the problems of (a) searching and (b) labeling for global and individual photo collections, utilizing location-based implicit sharing of labels between participating users.

LOCALE shows promising results for keyword search over personal collections of photographs. Even in our limited experiment, the system was able to retrieve and identify landmarks and geographical features with surprising accuracy. On the other hand, the geographic scope of the experiment was small, and the results have to be verified when broader-coverage collections are available.

In addition, our system proved quite useful in a global scenario, providing support for image search on a multi-user database of photos. However, it seems that for such scenarios the system needs to be augmented by other techniques to improve precision. Again, we would like to evaluate the system across broader geographical coverage.

We have also shown satisfying preliminary results for assigning location-related captions to photos, either automatically or semi-automatically with some human assistance (i.e., the user can choose a caption from a few top-scoring candidate terms). Helping users assign labels may assist future search: when users perform searches, their own labels or terms they acknowledged by picking from the suggestion list will be more significant matches than labels submitted by others. Ease of labeling will also benefit other users, as the data regarding the verified labels will be submitted and enrich the LOCALE system.

The most interesting future direction for LOCALE may be in augmenting the system with image retrieval and image-analysis tools (see [91] for an extensive summary of research in this area). Feature extraction will enable better matching of labels and candidate photos. LOCALE can be augmented with systems like Blobworld [13, 6]

to allow the automatic labeling of objects within images if the image occurs in a certain geographical area. It may be possible to use simple content-based techniques to disambiguate the direction the camera was pointed at. For example, the Oval photo shown in Table 7.2 is very different than any photos of the Quad taken from the same location. Given that information, the suggested label “Quad” for this image could be easily pruned.

More internal to the LOCALE implementation, one possible future direction is to develop a set of additional techniques to handle larger geographical areas, and higher condensation of data. For example, a LOCALE system that cover the entire world should be able to assign not only local labels (“Hoover Tower”) but also higher-level labels (“Stanford University” or even “The Bay Area”). In addition, we can think about using other data sources such as an “official” gazetteer. However, existing gazetteers (see [38] for example) are usually more reliable in identifying a city/state/country than a landmark related to a single photo.

Using additional camera-supplied metadata is another source for future investigation. Examples may be a physical device that could supply the direction the camera was pointing when the picture was taken. Other useful metadata could be the already-captured metadata such as focal point and F-stop, hinting at the distance of the photographed object from the location of the camera.

We can certainly consider adding time sensitivity to LOCALE, so it could be used to search for, and label events. The system could detect temporal outliers such as “graduation” that appear in an area associated with “Stanford University” during a few days in June. Once LOCALE detects such anomalies, the system could remove those labels from the time-neutral dataset, but at the same time suggest these labels for photos taken at the time of the event.

Another type of context-sensitivity in LOCALE can be automatic detection of expert users that had contributed many photos of the same geographical area at different times. The expert user’s labels may be better trusted based on the assumption that the user knows the mentioned area well. Finally, getting text associated with photos from other sources (web pages, newspaper articles etc.) may be possible as geo-referenced photos become abundant.

# Chapter 8

## Conclusions

Kimya returns home after a long vacation in Europe. Her photos are already accessible at her home machine: the camera detected wireless internet spots during her journey and uploaded all the photos to the server. Kimya is saddened by the end of her travels. After a short rest, she sits next to her desktop computer to reminisce. First, she asks to see the photos of her friends, Joanna and Mia, which she took during the trip. Kimya then asks for the photos taken in the rain: Joanna was soaking wet after the storm, yet they had a great time. The most memorable photo, which Kimya asks for next, is the one taken in front of Paris' Louvre. Kimya queries for that image, and then asks for a photo taken in the same place during her previous trip to Europe. "We really have aged", she thinks.

Given current technology trends and the research reported on in this thesis, the hypothetical story in the previous paragraph is not far fetched. In the three years since work on this project started, location-enabled cameras have turned from a slightly-risky projection into an off-the-shelf commodity: programmable phones with GPS-grade accuracy are available in many countries. While current applications and systems are not yet set to leverage the available information and metadata generated by these appliances, such applications will be in place as more and more location-capable units are sold.

Our research has shown that using location information, coupled with timestamps

and other metadata about photos, systems can be developed that alleviate the “semantic gap” that is inherent in personal photo collections. Even if image signal analysis and content-based tools such as face detection, face recognition and object detection do not advance very rapidly, photo management applications could do much more for users than they do now.

In the first part of this thesis we have shown how our system can automatically organize a collection of photos into meaningful location and event hierarchies, based on the time and location metadata. These hierarchies, created specifically based on the properties of each collection, enable better semantic interaction with the photos. In addition, our system augments the browsing interface with context information from other sources, like weather and light status, that is derived from location and time. This additional metadata promises to prove useful in collections that span many years and multiple different events.

In the second part of this work, we have discussed how system can benefit from a minimal amount of manual annotation by users. We discuss ways to ease identity annotation and retrieval in photo collections. Our system utilizes patterns of re-appearance and co-appearance that emerge from user annotation to try and guess which person is likely to appear in a new, un-annotated photograph.

Finally, we have introduced sharing of information between users, based on the location (and possibly time) in which photos are taken. We attach tentative labels to new photos based on labels of other photos that were taken in the same location. The system thus allows automatic identification of landmarks that appear in un-labeled photos.

Our system can be enhanced with content-based tools. As the content-based methods improve, and more context becomes available, we will get closer to the “perfect system” vision: a photo solution that requires no user input, yet answers all the user’s requirements and supports her information needs.

Taking a broad perspective, two interesting future directions that transcend personal photo management emerge from the work in this thesis.

Firstly, photographs have been presented here as a window into human memory:

photographs sample the human memory, facilitate it and enhance it; sometimes photographs have the power to alter our recollection of events and even people. Which of the techniques developed in this thesis can be extended to serve as a general human memory assistance tool? Can technology help us manage not only our photographs, but also our meetings and encounters, conversations, actions or even thoughts? For example, imagine that some device, maybe a mobile phone, recorded every location we ever visited. How can this information be organized, queried and used in an appropriate manner? A large number of research projects (e.g., [29]) have been dealing with similar questions. We hope that this thesis has made a contribution in this field, and we will look to extend it to broader questions of human memory.

The other future direction pushes this work in an entirely different track. Most of the work in this thesis focused on consequences and benefits of geo-referenced photography for personal photo collections. Even when introducing a shared, universal repository (in Chapter 7) we only used it to generate added value in the context of a personal collection of photos. What are the global consequences of a worldwide repository of textual labels, describing photos taken at every single location on the globe? What are the benefits and potentials of a repository holding the *actual photos* taken around the world? As some repositories of this type begin to appear, these questions should be investigated in more detail. In her book *On Photography*, Susan Sontag wrote that “everything [in the world] exists to end in a photograph.” What if all the *photographs* exist to end in our database?



# Appendix A

## Generating Geo-referenced Photographs

In this appendix, we briefly describe the manner in which geo-referenced photos were produced for this work by the author. Then we describe other possible ways to generate geo-referenced photos using current and emerging technologies.

During this thesis, I have used a separate GPS (Global Positioning System) device and digital camera to generate geo-referenced photos. Modern GPS devices are palm-sized and easy to carry. I have been using the Garmin eTrex Vista, which is 4.4 x 2.0 x 1.2 (Height x Width x Depth) inches in size, and weighs 5.3 ounces: the equivalent of an average-sized mobile phone.

The eTrex, similarly to other off-the-shelf GPS devices, logs its location regularly unto a log file. Each log line includes the time, location and elevation where the device is located. The file can later be downloaded to a computer with a serial port connection. The eTrex Vista holds up to 10000 log points, the equivalent of 83 full hours of logging with 30-second gaps between one log point to the next.

Once the log file as well as the photos from the camera are stored on a computer, their timestamps can be compared, and photos can be associated with log points, assuming the camera and GPS clocks are synchronized. Of course, the GPS is assumed to be in the same place as the camera when the photos are taken. To illustrate, say one of the photos was taken on September 1<sup>st</sup>, 2003, at 11:34:14am. The system looks

up the two GPS log entries in the log file that bound the photo timestamp. Indeed, imagine there is a log line written 9 seconds earlier at 11:34:05am (*Log1*), and another 21 seconds later at 11:34:35am (*Log2*). The system can use the location data from the closest log point in time (*Log1* in this case). Alternatively, it could interpolate to derive the photo location using both points:  $latitude = latitude(Log1) \times \frac{21}{30} + latitude(Log2) \times \frac{9}{30}$  (and similarly for the longitude) assuming, of course, the GPS carrier had advanced in a straight line and at constant speed between the two log points.

The location accuracy for photos generated in this way should ideally be 30 meters or better for a walking scenario (i.e., photos are not taken from a car, plane, or some other moving vehicle). The accuracy of the GPS device is 15 meters or better; average walking speed is less than 1 meter a second. In a 30-second span, then, the camera could be at most 15 meters away from both recorded log points. Total accuracy should not drop under 30 meters = 15 (GPS accuracy) + 15 (Log resolution error).

Unfortunately, this scheme often fails. GPS signals can only be received when the device is within clear view of the sky. Not only indoor reception is impossible; the signal is often unavailable under tree cover, and even in cities where buildings obscure part of the sky. Even in open areas, if the GPS device is carried in the wrong manner (for example, the antenna is partially covered), the signal will not be received.

For the case of indoor photos, a satisfactory solution would be to use the last recorded log point, under the assumption that the device had stopped receiving when the carrier stepped into a building, and that the indoor location is of small scale. However, using current tools it is not possible to distinguish between this scenario and a situation where the user carrying a GPS device lost reception outdoors, while moving, due to one of the reasons described earlier. In such a case, it will not be appropriate to use the last recorded log point.

I used the software tool *GPS-Photo Link*<sup>1</sup> to associate the GPS log with the timestamped photos. GPL allows its user to control for the offset between the GPS time and the camera time so the devices do not have to be synchronized in advance. GPL also allows one to specify a maximum permitted time gap — if no log points were

---

<sup>1</sup><http://geospatalexperts.com/>

recorded within this time gap from a certain photo, the photo will not be associated with a location from the log. Usually, I set the permitted gap according to the activity depicted in the photos currently being referenced: for an indoor event, a few hours; for an outdoor trip, a few minutes. Another feature of GPL allows to manually set the coordinates for photos that were not automatically referenced for any reason, assuming the photos location is known to the user.

Other software tools exist that can associate GPS logs and timestamped photographs, including the Location Stamper from WWMX [90] (see below).

In our experiment, we often required several personal geo-referenced photo collections (see Chapters 3, 4, 5 and 6). While GPS-based photo collections were available for our small-scale experiments in Chapter 3 and our controlled experiment of Chapter 7, we could not find enough such collections for larger-scope experiments. Instead, we asked participants in our experiments to retrospectively mark their photos with location information.

The tool that participants used to retrospectively mark photos with location data was the Location Stamper, made available by the authors of the World Wide Media Exchange<sup>2</sup> [90]. Figure A.1 shows a sample screen shot of the Location Stamper. At the right-hand side, the user's photos are shown, grouped by the date each photo was taken. The users can select one or more photos and drag them onto the map at the center pane, thus "marking" the photos as taken at the location where they were dropped.

Using the Location Stamper, of course, means that the location data for the collection is at best only as good as the user can remember it. Of course, many users did not remember the exact location for each of their photos, and dragged photos instead to an approximate location. In addition, marking thousands of photos with exact location is a cumbersome task, even if the locations are well remembered. In many cases, the users grouped photos together and dragged them to the rough area where most of them were taken. Data from the Location Stamper was therefore not very accurate. Such problems did not affect our experiments greatly, as accurate location information was not required for many of them. Methodological issues with

---

<sup>2</sup>at <http://wmx.org>

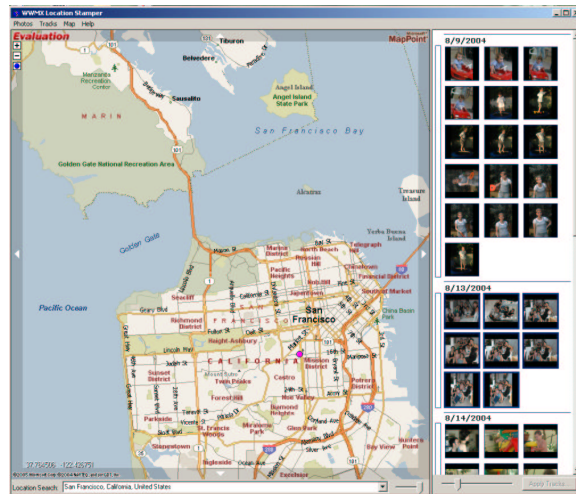


Figure A.1: Location Stamper screen shot.

using the Location Stamper emerged mostly in Chapter 4; refer to the discussion there for more details.

It seems natural to embed the GPS capabilities into digital cameras. Back in 1999, Smith et al. [83] tried manually wiring a GPS device (and a digital compass) into the Kodak DC260 to produce geo-referenced photos. Today, “GPS-ready” cameras are available off-the-shelf, such as the Ricoh Pro G3, which supports a GPS device through an expansion slot.

While GPS technology appears to begin supporting location-based camera services, the bulk of location-oriented photos taken today are captured on mobile camera phones. Mobile phones (i.e., cellular phones) are inherently location aware: the device connects to a local cell for operation, and the ID of the cell is exposed to the phone. The cell ID, at best, guarantees location accuracy to the order of several kilometers. However, the location capabilities of mobile phones are improving rapidly, and location accuracy on the order of 50–100 meters is already available with some devices. In the US, enhanced emergency (“911”) regulations will require all mobile phones to support location accuracy of 50–100 meters within the next few years.<sup>3</sup>

<sup>3</sup>See <http://www.fcc.gov/911/enhanced/>

Camera phones are already widely available, and as their camera-related capabilities are enhanced (Samsung recently announced the SCH-V770 phone equipped with a 7-megapixel camera), the phones threaten to replace the pocket camera as personal photo-taking device of choice. Of course, cameras will always be available that surpass the photographic capabilities of phones. However, those cameras could utilize a wireless connection to query a nearby cellular phone for their location. Such wireless connectivity (e.g., Bluetooth) may already be present in cameras for easy transmission of photos to a computer. Alternatively, basic cellular technology could be embedded in cameras strictly to support the location services.

Other location technologies are emerging. Recently, a new technology was introduced that uses signals from TV towers in addition to GPS information to get accurate location information indoors as well as in dense urban environments, but this technology is proprietary and was not available at the time of this work on consumer devices.

Most notably, Intel's Place Lab project [52, 78] attempts to "allow commodity hardware clients like notebooks, PDAs and cell phones to locate themselves by listening for radio beacons such as 802.11 access points, GSM cell phone towers, and fixed Bluetooth devices that already exist in large numbers around us in the environment." Place Lab already supplies 20–30 meters accuracy with almost-perfect coverage (indoors and outdoors) in locales such as the greater Seattle area in Washington state. Functionality and accuracy in other areas depends on the existence of databases of network access points, and on the density of access points in that area. Camera-based devices enhanced with network capabilities will be able to utilize Place Lab technology to discover their location.

# Bibliography

- [1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the International Conference on Management of Data*, pages 207–216. ACM Press, 1993.
- [2] Robert B. Allen. A query interface for an event gazetteer. In *Proceedings of the Fourth ACM/IEEE-CS Joint Conference on Digital Libraries*, 2004.
- [3] Einat Amitay, Nadav Har’El, Ron Sivan, and Aya Soffer. Web-a-where: geotagging web content. In *SIGIR ’04: Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 273–280. ACM Press, 2004.
- [4] Avi Arampatzis, Marc van Kreveld, Iris Reinbacher, Paul Clough, Hideo Joho, Mark Sanderson, Christopher B. Jones, Subodh Vaid, Marc Benkert, and Alexander Wolff. Web-based delineation of imprecise regions. In *Proceedings of the Workshop on Geographic Information Retrieval*, 2004.
- [5] J. Ashley, M. Flickner, J. Hafner, D. Lee, W. Niblack, and D. Petkovic. The query by image content (QBIC) system. In *Proceedings of the International Conference on Management of Data (SIGMOD)*. ACM Press, 1995.
- [6] Kobus Barnard and David .A. Forsyth. Learning the semantics of words and pictures. In *Proceedings of the IEEE International Conference on Computer Vision*, July 2001.

- [7] Benjamin B. Bederson. Photomesa: a zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 71–80. ACM Press, 2001.
- [8] Benjamin B. Bederson, Ben Shneiderman, and Martin Wattenberg. Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions on Graphics*, 21(4):833–854, 2002.
- [9] Doug Beeferman, Adam Berger, and John D. Lafferty. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210, 1999.
- [10] Tamara L. Berg, Alexander C. Berg, Jaety Edwards, Michael Maire, Ryan White, Yee-Whye Teh, Erik Learned-Miller, and D.A. Forsyth. Names and faces in the news. In *CVPR 2004: Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2004.
- [11] Vannevar Bush. As we may think. *The Atlantic Monthly*, July 1945.
- [12] Orkut Buyukkokten, Hector Garcia-Molina, and Andreas Paepcke. Accordion summarization for end-game browsing on pdas and cellular phones. In *Proceedings of the Conference on Human Factors in Computing Systems CHI'01*, 2000.
- [13] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. Blobworld: A system for region-based image indexing and retrieval. In *Proceedings of the Third International Conference on Visual Information Systems*, June 1999.
- [14] Duncan Cavens, Stephen Sheppard, and Michael Meitner. Image database extension to arcview: How to find the photograph you want. In *Proceedings of ESRI Users Conference*, 2001.
- [15] Guanling Chen and David Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dartmouth College, 2000.

- [16] Tammara T.A. Combs and Benjamin B. Bederson. Does zooming improve image browsing? In *Proceedings of the Fourth ACM International Conference on Digital Libraries*, 1999.
- [17] Matthew Cooper, Jonathan Foote, Andreas Girgensohn, and Lynn Wilcox. Temporal event clustering for digital photo collections. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 364–373. ACM Press, 2003.
- [18] Marc Davis, Simon King, Nathan Good, and Risto Sarvas. From context to content: leveraging context to infer media metadata. In *Proceedings of the 12th International Conference on Multimedia (MM2004)*, pages 188–195. ACM Press, 2004.
- [19] Ann S. Devlin. *Mind and maze : spatial cognition and environmental behavior*. Praeger, 2001.
- [20] Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. In *Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of the 2000 Conference on Human Factors in Computing Systems (CHI 2000)*, April 2000.
- [21] Junyan Ding, Luis Gravano, and Narayanan Shivakumar. Computing geographical scopes of web resources. In *Proceedings of the Twenty-sixth International Conference on Very Large Databases*, pages 545–556. Morgan Kaufmann Publishers Inc., 2000.
- [22] Steven M. Drucker, Curtis Wong, Asta Roseway, Steven Glenner, and Steven De Mar. Mediabrowser: reclaiming the shoebox. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 433–436, New York, NY, USA, 2004. ACM Press.
- [23] Susan Dumais, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. Stuff i've seen: a system for personal information retrieval



- and re-use. In *Proceedings of the Twenty-Sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 72–79. ACM Press, 2003.
- [24] P. Duygulu, Kobus Barnard, J. F. G. de Freitas, and David A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the 7th European Conference on Computer Vision (ECCV '02)*, pages 97–112. Springer-Verlag, 2002.
- [25] David Frohlich, Allan Kuchinsky, Celine Pering, Abbe Don, and Steven Ariss. Requirements for photoware. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, 2002.
- [26] Ullas Gargi. Managing and searching personal photo collections. Technical Report HPL-2002-67, HP Laboratories, March 2002.
- [27] Ullas Gargi. Consumer media capture: Time-based analysis and event clustering. Technical Report HPL-2003-165, HP Laboratories, August 2003.
- [28] Laura Garton, Caroline Haythornthwaite, and Barry Wellman. Studying online social networks. *Journal of Computer-Mediated Communication*, 3(1), June 1997.
- [29] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. Mylifebits: fulfilling the memex vision. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 235–238. ACM Press, 2002.
- [30] Aristides Gionis and Heikki Mannila. Finding recurrent sources in sequences. In *Proceedings of the seventh annual international conference on Computational molecular biology*, pages 123–130. ACM Press, 2003.
- [31] Andreas Girgensohn, John Adcock, Matthew Cooper, Jonathan Foote, and Lynn Wilcox. Simplifying the management of large photo collections. In *INTERACT '03: Ninth IFIP TC13 International Conference on Human-Computer Interaction*, pages 196–203. IOS Press, September 2003.

- [32] Andreas Girgensohn, John Adcock, and Lynn Wilcox. Leveraging face recognition technology to find and organize photos. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 99–106. ACM Press, 2004.
- [33] Google inc. <http://www.google.com>.
- [34] Adrian Graham, Hector Garcia-Molina, Andreas Paepcke, and Terry Winograd. Time as essence for photo browsing through personal digital libraries. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries*, 2002.
- [35] Karen D. Grant, Adrian Graham, Tom Nguyen, Andreas Paepcke, and Terry Winograd. Beyond the shoe box: Foundations for flexibly organizing photographs on a computer. Technical Report 2002-45, Stanford University, 2002.
- [36] Robert A. Hanneman and Mark Riddle. *Introduction to Social Network Methods: Online Textbook*. 2000. available at <http://faculty.ucr.edu/hanneman/nettext/>.
- [37] Susumu Harada, Mor Naaman, Yee Jiun Song, QianYing Wang, and Andreas Paepcke. Lost in memories: Interacting with photo collections on PDAs. In *Proceedings of the Fourth ACM/IEEE-CS Joint Conference on Digital Libraries*, 2004.
- [38] Linda L. Hill, James Frew, and Qi Zheng. Geographic names - the implementation of a gazetteer in a georeferenced digital library. *CNRI D-Lib Magazine*, January 1999.
- [39] E. Hjelmås and B. K. Low. Face detection: a survey. *Computer Vision and Image Understanding*, 83(3):236 – 74, SEP 2001.
- [40] Ben Shneiderman Jack Kustanowitz. Meaningful presentations of photo libraries: Rationale and applications of bi-level radial quantum layouts. In *Proceedings of the Fifth ACM/IEEE-CS Joint Conference on Digital Libraries*, 2005.

- [41] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [42] Christopher B. Jones, R. Purves, A. Ruas, M. Sanderson, M. Sester, M. van Kreveld, and R. Weibel. Spatial information retrieval and geographical ontologies an overview of the spirit project. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 387–388. ACM Press, 2002.
- [43] Volker Jung. Metaviz: Visual interaction with geospatial digital libraries. Technical Report TR-99-017, International Computer Science Institute, 1999.
- [44] Hyunmo Kang and Ben Shneiderman. Visualization methods for personal photo collections: Browsing and searching in the photofinder. In *IEEE International Conference on Multimedia and Expo*, 2000.
- [45] Hyunmo Kang and Ben Shneiderman. Mediafinder: an interface for dynamic personal media management with semantic regions. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 764–765, New York, NY, USA, 2003. ACM Press.
- [46] Hyunmo Kang and Ben Shneiderman. Exploring personal media: A spatial interface supporting user-defined semantic regions. Technical Report ISR 2005-51, University of Maryland, 2005.
- [47] Amir Khella and Benjamin B. Bederson. Pocket PhotoMesa: a zoomable image browser for PDAs. In *MUM '04: Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, pages 19–24, New York, NY, USA, 2004. ACM Press.
- [48] Menno-Jan Kraak. Integrating multimedia in geographical information systems. *IEEE MultiMedia*, 3(2):59–65, 1996.
- [49] Allan Kuchinsky, Celine Pering, Michael L. Creech, Dennis Freeze, Bill Serra, and Jacek Gwizdka. Fotofile: a consumer multimedia organization and retrieval

- system. In *Proceedings of the Conference on Human Factors in Computing Systems CHI'99*, pages 496–503, 1999.
- [50] Bill Kules, Hyunmo Kang, Catherine Plaisant, Anne Rose, and Ben Shneiderman. Immediate usability: Kiosk design principles from the CHI 2001 photo library. Technical Report CS-TR-4293, University of Maryland, 2003.
- [51] Pei-Jeng Kuo, Terumasa Aoki, and Hiroshi Yasuda. Building personal digital photograph libraries: An approach with ontology-based mpeg-7 dozen dimensional digital content architecture. In *CGI '04: Proceedings of the Computer Graphics International (CGI'04)*, pages 482–489, Washington, DC, USA, 2004. IEEE Computer Society.
- [52] Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Tim Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello, and Bill Schilit. Place lab: Device positioning using radio beacons in the wild. In *Pervasive 2005: Third International Conference on Pervasive Computing*, pages 116–133. Springer-Verlag, 2005.
- [53] Ray R. Larson and Patricia Frontiera. Geographic information retrieval (gir) ranking methods for digital libraries. In *Proceedings of the Fourth ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 415–415. ACM Press, 2004.
- [54] Yvan Leclerc, Martin Reddy, Lee Iverson, and Michael Eriksen. The geoweb - a new paradigm for finding data on the web. In *International Cartographic Conference (ICC2001)*, 2001.
- [55] H. Lieberman and Hugo Liu. Adaptive linking between text and photos using common sense reasoning. In *Adaptive Hypermedia and Adaptive Web-Based Systems. Second International Conference, AH 2002. Proceedings, 29-31 May 2002, Malaga, Spain*, pages 2 – 11. Springer-Verlag, 2002, 2002.
- [56] Henry Lieberman, Elizabeth Rozenweig, and Push Singh. Aria: An agent for annotating and retrieving images. *Computer*, 34(7):57–62, 2001.

- [57] Joo-Hwee Lim, Philippe Mulhem, and Qi Tian. Home photo content modeling for personalized event-based retrieval. *IEEE Multimedia*, 10(4):28–37, 2003.
- [58] Paul Longley, Michael Goodchild, David Maguire, and David Rhind. *Geographic Information Systems and Science*. John Wiley & Sons, 2001.
- [59] A. Loui and A. E. Savakis. Automatic image event segmentation and quality screening for albuming applications. In *IEEE International Conference on Multimedia and Expo*, 2000.
- [60] Alexander C. Loui and Mark D. Wood. A software system for automatic albuming of consumer pictures. In *MULTIMEDIA '99: Proceedings of the seventh ACM international conference on Multimedia (Part 2)*, pages 159–162, New York, NY, USA, 1999. ACM Press.
- [61] J. Luo and A. Savakis. Indoor vs. outdoor classification of consumer photographs. In *Proceedings of the international conference on Image Processing (ICIP 2001)*, 2001.
- [62] Kevin Lynch. *The Image of the City*. MIT Press, Cambridge, MA, USA, 1960.
- [63] Catherine C. Marshall and A.J. Bernheim Brush. Exploring the relationship between personal and public annotations. In *Proceedings of the Fourth ACM/IEEE-CS Joint Conference on Digital Libraries*, 2004.
- [64] Timothy J. Mills, David Pye, David Sinclair, and Kenneth R. Wood. Shoebox: A digital photo management system. Technical Report 2000.10, AT&T Laboratories Cambridge, 2000.
- [65] Baback Moghaddam, Qi Tian, Neal Lesh, Chia Shen, and Thomas S. Huang. Visualization and user-modeling for browsing personal photo libraries. *International Journal of Computer Vision*, 56(1-2):109–130, 2004.
- [66] Virginia E. Ogle and Michael Stonebraker. Chabot: Retrieval from a relational database of images. *Computer*, 28(9):40–48, 1995.

- [67] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Computer Science Department, Stanford University, 1998.
- [68] Lev Pevzner and Marti Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36, 2002.
- [69] A. Pigeau and M. Gelgon. Organizing a personal image collection with statistical model-based ICL clustering on spatio-temporal camera phone meta-data. *Journal of Visual Communication and Image Representation*, 15(3):425–445, September 2004.
- [70] John C. Platt. Autoalbum: Clustering digital photographs using probabilistic model merging. In *CBAIVL '00: Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, page 96, Washington, DC, USA, 2000. IEEE Computer Society.
- [71] John C. Platt, Mary Czerwinski, and Brent A. Field. Phototoc: Automatic clustering for browsing personal photographs. Technical Report MSR-TR-2002-17, Microsoft Research, February 2002.
- [72] Daniel C. Robbins, Edward Cutrell, Raman Sarin, and Eric Horvitz. Zonezoom: map navigation for smartphones with recursive view segmentation. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 231–234, New York, NY, USA, 2004. ACM Press.
- [73] Kerry Rodden. How do people organise their photographs? In *21st Annual BCS-IRSG Colloquium on IR*, 1999. Available at <http://www.rodnen.org/kerry/irsg.pdf>.
- [74] Kerry Rodden, Wojciech Basalaj, David Sinclair, and Kenneth Wood. Does organisation by similarity assist image browsing? In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 190–197. ACM Press, 2001.

- [75] Kerry Rodden and Kenneth R. Wood. How do people manage their digital photographs? In *Proceedings of the conference on Human factors in computing systems*, pages 409–416. ACM Press, 2003.
- [76] Manojit Sarkar, Scott S. Snibbe, Oren J. Tversky, and Steven P. Reiss. Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 81–91, New York, NY, USA, 1993. ACM Press.
- [77] Risto Sarvas, Erick Herrarte, Anita Wilhelm, and Marc Davis. Metadata creation system for mobile images. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 36–48. ACM Press, 2004.
- [78] Bill N. Schilit, Anthony LaMarca, Gaetano Borriello, William G. Griswold, David McDonald, Edward Lazowska, Anand Balachandran, Jason Hong, and Vaughn Iverson. Challenge: ubiquitous location-aware computing and the “place lab” initiative. In *WMASH '03: Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, pages 29–35. ACM Press, 2003.
- [79] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.
- [80] Ben Shneiderman and Hyunmo Kang. Direct annotation: A drag-and-drop strategy for labeling photos. In *Proceedings of the International Conference on Information Visualization*, May 2000.
- [81] Mario J. Silva, Bruno Martins, Marcirio Chaves, and Nuno Cardoso. Adding geographic scope to web resources. In *Proceedings of the Workshop on Geographic Information Retrieval*, 2004.
- [82] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.

- [83] Brian K. Smith, Erik Blankinship, III Alfred Ashford, Michael Baker, and Timothy Hirzel. Inquiry with imagery: historical archive retrieval with digital cameras. In *MULTIMEDIA '99: Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 405–408, New York, NY, USA, 1999. ACM Press.
- [84] Terence R. Smith. A digital library for geographically referenced materials. *Computer*, 29(5):54 – 60, MAY 1996.
- [85] Michael Southworth and Susan Southworth. *Maps : a visual survey and design guide*. Little, Brown, 1982.
- [86] Diomidis Spinellis. Position-annotated photographs: A geotemporal web. *IEEE Pervasive Computing*, 2(2):72–79, 2003.
- [87] Amanda Stent and Alexander Loui. Using event segmentation to improve indexing of consumer photographs. In *24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 59–65. ACM Press, 2001.
- [88] Yanfeng Sun, Hongjiang Zhang, Lei Zhang, and Mingjing Li. Myphotos: a system for home photo management and processing. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 81–82. ACM Press, 2002.
- [89] Dario Teixeira and Yassine Faihe. In-home access to multimedia content. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 49–56, New York, NY, USA, 2002. ACM Press.
- [90] Kentaro Toyama, Ron Logan, and Asta Roseway. Geographic location tags on digital images. In *Proceedings of the 11th International Conference on Multimedia (MM2003)*, pages 156–166. ACM Press, 2003.
- [91] Remco C. Veltkamp and Mirela Tanase. Content-based image retrieval systems: A survey. Technical Report TR UU-CS-2000-34 (revised version), Department of Computing Science, Utrecht University, October 2002.



- [92] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the Conference on Human Factors in Computing Systems CHI'04*, pages 319–326, New York, NY, USA, 2004. ACM Press.
- [93] W.A. Wagenaar. My memory: A study of autobiographical memory over six years. *Cognitive psychology*, 18:225–252, 1986.
- [94] Liu Wenyin, Susan Dumais, Yanfeng Sun, HongJiang Zhang, Mary Czerwinski, and Brent Field. Semi-automatic image annotation. In *8th International Conference on Human-Computer Interactions (INTERACT 2001)*, July 2001.
- [95] Liu Wenyin, Yanfeng Sun, and Hongjiang Zhang. Mialbum - a system for home photo managemet using the semi-automatic image annotation approach. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 479–480, New York, NY, USA, 2000. ACM Press.
- [96] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(1):34–58, 2002.
- [97] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *Proceedings of the conference on Human factors in computing systems*, pages 401–408. ACM Press, 2003.
- [98] Lei Zhang, Longbin Chen, Mingjing Li, and Hongjiang Zhang. Automated annotation of human faces in family albums. In *Proceedings of the 11th International Conference on Multimedia (MM2003)*, pages 355–358. ACM Press, 2003.
- [99] Lei Zhang, Yuxiao Hu, Mingjing Li, Weiyang Ma, and Hongjiang Zhang. Efficient propagation for face annotation in family albums. In *Proceedings of the 12th International Conference on Multimedia (MM2004)*, pages 716–723. ACM Press, 2004.

- [100] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, DEC 2003.
- [101] Bin Zhu, Marshall Ramsey, Hsinchun Chen, Hauck Rosie V, Tobun D. Ng, and Bruce Schatz. Create a large-scale digital library for geo-referenced information. In *DL '99: Proceedings of the fourth ACM conference on Digital libraries*, pages 260–261, New York, NY, USA, 1999. ACM Press.