Leveraging Context to Resolve Identity in Photo Albums

Mor Naaman, Hector Garcia-Molina, Andreas Paepcke, Ron B. Yeh
Stanford University
{mor, hector, paepcke, ronyeh}@cs.stanford.edu

ABSTRACT

Our system suggests likely identity labels for photographs 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.

Based on time and location, the system automatically computes event and location groupings of photos. 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. The 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.

We obtained ground-truth identity annotation for four different photo albums, and used them 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.

Categories and Subject Descriptors

H.5.1 [**Information Systems Applications**]: Information Interfaces and Presentation—*Multimedia Information Systems*

Keywords

geo-referenced digital photos, face recognition, photo collections, context

1. INTRODUCTION

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 [6] 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 [13]. 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.

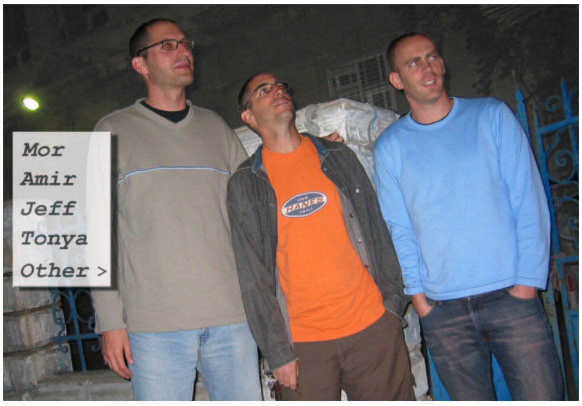

Figure 1: Possible interaction mode with the system — the user annotates the identities that appear in the photo by picking from a short list of candidates.

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 [4, 15] and recognition [18] 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 often not aligned directly into the camera. Most faces are tilted or slanted, or even partially or totally obscured (in one of our sample collections, about 70% of the faces of people in the photos were slanted, a third of those shown in profile). These facts make recognition — and even detection — an extremely difficult task. As the researchers in [2] 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 [5, 11, 14] have focused on systems that ease the manual annotation of identities in photos (see Related Work section). 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 cannot be retrieved.

Short of solving the difficult image analysis problems, we are pursuing two intermediate steps. In the first step, we look at making the manual annotations of face identities as easy as possible for the collection owner. Later, we will attempt to minimize the number of photos that need to be manually annotated in a collection, while still achieving satisfactory results for identity-based retrieval over the entire collection, including the non-annotated photos.

Our approach to reaching these two milestones is to leverage context available from photo metadata and user input. We build on our earlier work, which demonstrated statistical algorithms that successfully identify coherent 'events' and 'locations' within personal photo collections based on photo metadata. We describe here our solution to the first milestone, and we discuss how we plan to use the results for our approach to the second goal.

More specifically, in this work we focus on improving the quality of candidate name lists that are presented to the user for each photo during the annotation process. We address the problem of long-list annotation by reducing the list of candidates to a reasonable size (e.g., 5 names). Our system tries to predict which people are likely to appear in the photo, leveraging photo metadata, context (such as event and location) and previous annotations.

An example user interaction with such a candidate list is shown in Figure 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 paper is not concerned with the details of the interaction. We focus, instead, on generating good candidate lists.

We use the following terminology in this paper. The term *metadata*, when referring to photographs, commonly 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 this notion to the setting 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.[1]
- 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 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 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 in the Future Work section.

2. RELATED WORK

The research of Zhang et al. [16] and Girgensohn et al. [2] focuses on using face recognition algorithms to ease the task of annotating identities. In [16], the researches address the problem of Long-List annotation described above. Their system ranks likely candidates for a *specific* identity in each photo. The ranking is based on face similarity using a nearest-neighbor-based learning algorithm. In later work [17] the same researchers applied similar techniques to annotate multiple photos simultaneously.

The system described by Girgensohn [2] takes a different approach for annotation. 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 visual similarity. The user can quickly annotate the correct thumbnails in the set with the name of the person.

Both systems are orthogonal to our approach and can be

[1] Moreover, it is likely that the person will be wearing the same clothes, which may help recognition [16]; we are not looking at clothes or face recognition in this current work.

enhanced using the context-based techniques we use in this paper. 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 are usually 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. For future evaluation of such systems, we propose the first author's collection as a baseline and will supply the photos with geographic and time metadata, as well as identity annotation.

Non-recognition based approaches to efficient labeling of photos have been an active research field since 1999. Ease and partial-automation of the labeling task was studied in [5, 11, 14]. For example, [11] proposed a drag-and-drop approach for labeling people in photos. The latest photo browser software packages (Adobe's Photoshop Album, Apple's iPhoto, Google's Picasa[2] and others) also attempt to support efficient labeling of photos using techniques orthogonal to our.

Davis et al [1, 9] 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 labels. However, they have not yet implemented and reported on an identity-based annotation system. Our LOCALE system [7], similarly, utilizes location context to share photo labels between users, thus helping users annotate their photos by location and landmarks.

In collecting context data, the PlaceLab project [10] aims to supply ubiquitous location services to computers and other devices (mobile phone, cameras). We have shown [6] we can use location and time metadata to add additional context such as weather and daylight status to photos. For lack of space, we do not list many other projects that collect and utilize context information.

3. 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 computed and used by the system. We do not list the model for user behavior here, but rather leave that for the evaluation (Section 5).

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). While the set I may include every single person that appears in the photo collection, it may also 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.[3]

3.1 Interaction Model

The algorithms can best be described on the basis of a formal model, which we introduce in this section. A certain set of people appear in each photo $s \in S$. 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 **k**nown 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 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 in 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 [16], 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 $H_s(t_2) = \{Poly, James, Neil\}$. The

[2] www.adobe.com, www.apple.com, www.picasa.com

[3] 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.

user will be forced to enter (or select from a longer list) the name *Dylan* which is not in H_s — *a miss*.

3.2 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 underlying system, its structure and parameters. First, we introduce the raw metadata that is associated with the photos. We then explain how we derive the high level structure of the collection by using the algorithms of our PhotoCompas system [8].

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 easy to assume that the time is automatically captured by the camera; all digital cameras sold today embed a timestamp in the photo file header. In contrast, most cameras do not yet support location-stamping. However, at least two technology trends will make location capabilities widely available in the near future: the growing numbers of camera-equipped mobile phones, which are inherently location-aware, and the dropping costs and reduced battery consumption of Global Positioning System (GPS) technology.[4] In Section 5 we expand on how our test collections were location-stamped.

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

Given time and location metadata, our PhotoCompas 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 [8], we briefly describe the generated hierarchies.

A sample location hierarchy is shown at the top of Figure 2. 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 these location clusters are broken down further, when clusters are overloaded with photos. For example, in the collection represented in Figure 2, the San Francisco area cluster represented many different photos taken at different events, and were 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 2.[5] Although we use location metadata as clues for detecting events in the photo collection, it is not necessary. Time metadata is sufficient, but we have shown that knowledge of location adds accuracy to event detection.

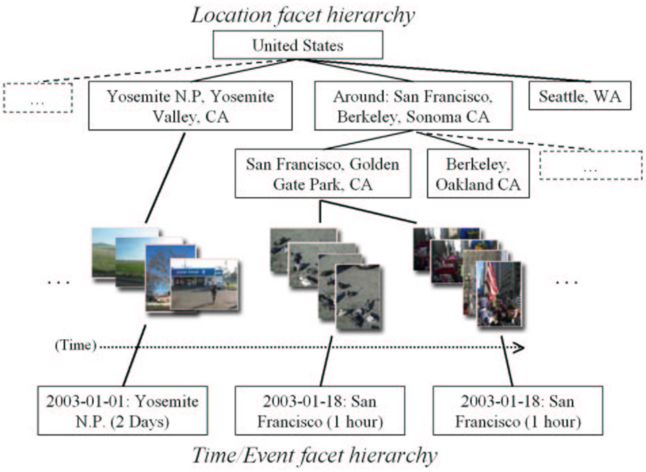
Location facet hierarchy
United States
...
Yosemite N.P. Yosemite Valley, CA
Around: San Francisco, Berkeley, Sonoma CA
Seattle, WA
San Francisco, Golden Gate Park, CA
Berkeley, Oakland CA
...
...
...
(Time)
2003-01-01: Yosemite N.P. (2 Days)
2003-01-18: San Francisco (1 hour)
2003-01-18: San Francisco (1 hour)
Time/Event facet hierarchy

Figure 2: Sample PhotoCompas structure. Parts of the location and time/event hierarchies for an actual collection of photos.

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 question 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 5).

4. GENERATING LABEL SUGGESTIONS

This section outlines our approaches to generating annotation suggestions. To use the notation from Section 3.1, 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.

[4] Even today, location-aware camera phones and GPS-enabled cameras are available. There are other ways to generate location-stamps for photos using current technology — for a summary, see [12].

[5] One can also use the notion of sub-events, and an event hierarchy (see [3]).

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.

4.1 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

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

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 $Q(s)$ 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 probability of $\frac{2}{2} = 1$, and Kimya a probability of $\frac{1}{2}$ 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 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

$$p(i,s) = \frac{\sum_{q \in Q(s)} K_q(i)}{|Q(s)|} \tag{2}$$

which produces a prior estimation of the frequency in which person i appears in the set $Q(s)$. Table 1 summarizes the basic estimators.

Table 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 1 simply assigns a prior probability 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 prior of $\frac{1}{3}$). But while we know Kimya does not appear in q_1, Nick may or may not appear in q_2 – possibly making him more likely to appear in s. We can compute $p(i,s) = \frac{\sum_{q \in Q(s)} p(q,s)}{|Q(s)|}$. Note that this formula is slightly modified from Eq. 2, in that we sum over the probabilities rather than K_q (Eq. 1). 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. Next, we introduce the *co-occurrence estimators*. Then we discuss ways to combine different estimators.

4.2 Estimating Co-occurrence: PeopleRank

The PeopleRank estimators aim to harvest the relationships between people. Such relationships naturally exist in personal photo collections due to the human nature of social interaction. 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.

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. Though 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. Generally, any co-occurrence in a semantically-connected set of photos can suggest a relation between people. 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.

The problem can now be formulated in terms of connections between people, rather than patterns in 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 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. 1). The links between all people compose a graph, and the weight represents the strength of connection between each pair.

Back to our question, if we know that Nick appears in the event, the likelihood assigned by *PeopleRank* for other people to appear in the 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 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 used as a normalizing factor). The estimator ranks all candidates based on this score, and suggests the top candidates to the user.

So far, our computation is based on a single person that 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?

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: "given that the set of people $i_1, \ldots i_n$ appear in photos from event E, which other people are likely to appear in photos from this event?" or in human social terms, "who is most connected to this *group* of people?". To answer this question using $PeopleRank$, we simply collapse all the nodes $i_1, \ldots i_n$ into one group node. We treat the group as one person, and re-compute 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 $i_1, \ldots i_n$. We can then rank all persons based on their score given the group. 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.

To summarize, at any point of the user interaction, the $PeopleRank_{event}$ estimator will generate a list of suggestions for the given photo s based on event co-occurrence with a person, or a set of people, known in event $E(s)$. The score $PeopleRank_{event}$ assigns to people already known in the event is 1, thereby "subsuming" the Event Estimator, albeit less accurately (the Event Estimator assigns people known in the event probabilities according to their frequency of appearance). Similarly, $PeopleRank_{photo}$ ranks the candidates based on photo co-occurrence with the people in K_s.

Other computations can be performed on the $PeopleRank$ graph. For example, the system can be extended to support "indirect" relationships — if Kimya appears with Nick, and Nick appears with Marvin, Kimya may be likely to appear with Marvin even if we do not have direct evidence to support this conjecture. For simplicity and lack of space we do not handle this case in this paper.

4.3 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 than 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 $T =$ 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 grade" estimators include N_{loc} (when R is small), and $PeopleRank_{photo}$. The L (location) and E (event) estimators try to strike a balance between fine and broad estimators, but even 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 (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 5.

Weighting is another way to combine estimators. In weighting, we pre-assign weights to a set of estimators. 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 or even for each photo.

5. EVALUATION

For the evaluation of the system, we obtained four different personal photo collections. On average, each collection contained 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). We ran our PhotoCompas 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 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 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 identity that appears in each. An Industrious user starts

Table 2: Statistics for the collections used in our evaluation.

Collection	A	B	C	D
Time Span	2 years	6 years	3 years	5 years
Total Number of Photos	5947	4347	766	1926
Photos Containing Named Individuals	1673	1295	550	930
Number of Named Individuals	90	94	32	78
Total Number of Annotations	2624	1941	985	2389
Avg. Num. of Photos of each individual	29	21	31	31

with an empty set of annotated photos ($K = \emptyset$, to use the terminology of Section 3). 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.

Casual Annotators annotate a certain percentage $p_{annotate}$ of the 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 select 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 starts from the same initial state, and selects a photo that was not picked before.

We omit the evaluation of the Industrious User mode from the paper for lack of space. We note that performance level and trends were similar to the casual annotator mode. As we show in Section 5.2, the performance levels off after a certain percentage of the photos were annotated. This is one of the reasons we did not note significant differences between the two modes.

An important parameter to vary when trying to model user interaction is $|I|$, the number 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_ℓ to be the subset of I_{full} containing the ℓ most important people in the set. Then we can vary ℓ 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 fixed 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.

As described in Section 3, 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. 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 executed each set of parameters between 3–5 times on each collection, reporting the average value of those runs.

Before we report the actual results, a comment regarding 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), making an interaction with a possible human user as rapid as the user can handle.

5.1 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 how the Event and Location estimators, based on our automatically computed location and event photo sets ($E(s)$ and $L(s)$), perform compared to estimators based on neighboring photos (N_{time} and N_{loc}).
- Evaluate the effect of global parameters such as length of the candidate list h, and number of interesting people $|I|$, on h-hit performance of difference 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?

5.2 Results

For all results in this section, unless otherwise noted, we used the following parameter base values. The list 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, we only 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 3 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. At the top row, we show results for the Location Estimator, based on the location clusters as detected by our PhotoCompas system. The results are shown for two different sizes of the starting set of identities, $p_{annotate} = .2$ or $.4$. For example, the second row from the bottom indicates the performance for N_{loc} with the distance parameter set to 100km. The 5-hit rate for this case is 0.83 when $p_{annotate} = .2$, and a 0.86 when more knowledge is available ($p_{annotate} = .4$). The figure shows results only for collection A (the only collection with accurate, automatic location coordinates). However,

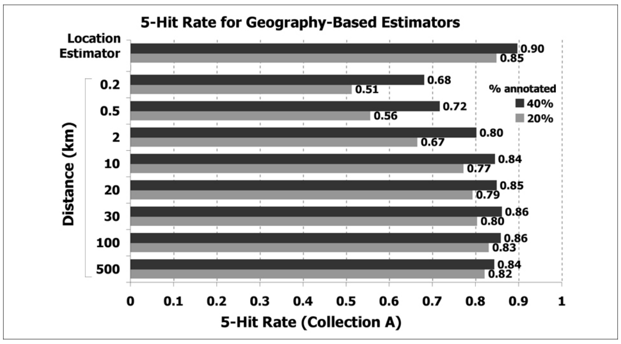

Figure 3: 5-Hit rate for different geography-based estimators — N_{loc} estimators with varying distance limits, and the cluster-based Location Estimator.

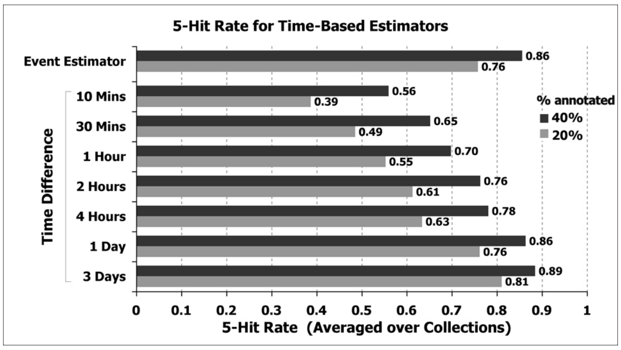

Figure 4: 5-Hit rate for different time-based estimators — N_{time} estimators with varying time limits, and the Event Estimator.

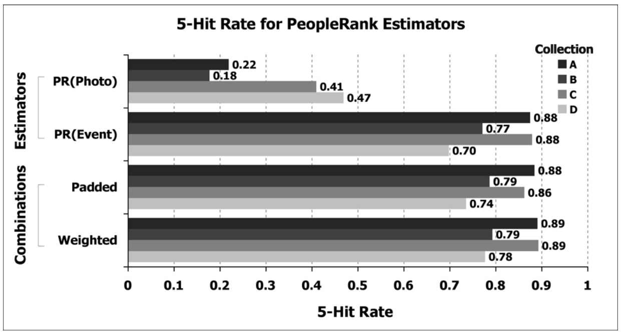

Figure 5: Performance of the PeopleRank Estimators and their combinations, for each collection.

the results for the other collections are equivalent.

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 Section 3.2 and in [8]), and therefore performs better.

In contrast, the automatically-detected events did not perform better than all time-neighbors estimators N_{time}. Figure 4 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, compared to a fixed time-neighboring estimators.

We examine the performance of the *PeopleRank* estimators and their combination in Figure 5. The figure shows the 5-hit rate for each collection and each estimator. We show results for the $PeopleRank_{photo}$ (top row) and the $PeopleRank_{event}$ (second row from the top) estimators. We also show two combinations of the estimators: $PeopleRank_{photo}$ padded with $PeopleRank_{event}$, and an equally-weighted combination of the two *PeopleRank* estimators (see Section 4.3 to recall the Padding and Weighting methods for combining estimators). For example, the 5-hit rate for collection C, using candidates from $PeopleRank_{photo}$ padded with candidates from $PeopleRank_{event}$, is 0.86. A key observation from the figure is that not all photo collections are created equal. For example, while $PeopleRank_{event}$ performs worst for 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, collection D was harder than other collections to predict who is in an event based on people known to be in the event.

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

The next few figures give a clearer picture about the overall system performance. In the figures, we show results for the basic estimators, and the two best-performing combinations. The first combination is based on weighting all estimators, using equal weights for all but the Global Estimator, which is weighted lower. The second combination is based on padding the results of the Event, Location and Global estimators, in that order. 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 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). The N_{loc} and N_{time} are tuned to 1km and 10 mins, respectively (i.e., tuned to accuracy rather than completeness). The h-hit rates are averaged over all executions and all collections for each point. For example, when $h = 3$, the Weighted estimator suggested an identity that was indeed in the photo for 81% of the new annotations.

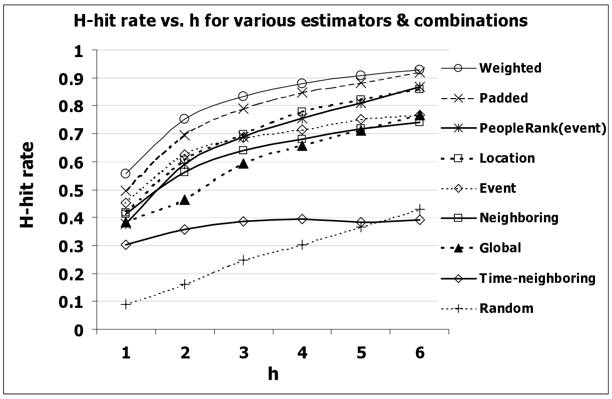

Figure 6: H-Hit rate (averaged over all collections) for various estimators vs. value of h.

We note a few observations about Figure 6, 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 only slightly worse than the Weighted estimator shown in the figure.

The basic estimators, by themselves, do not perform as well. 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. This is especially notable for N_{time} which does not improve much as h grows, such that it is comparable to the Random Estimator performance level when $h = 5$. Again, this is mostly due to false negatives.

We performed the same comparison for $p_{annotate} = 0.4$ (i.e., the initial 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 (hit rate is up by 0.1 for $h = 1$, and similarly for other h values), showing that, indeed, false negatives are its main shortfall.

Figure 7 shows the analysis of estimator performance when we vary the number of "important people" in the collection: the number of people we assume 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 a 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 and the naive Global Estimator degrade quickly as the size of I increases; the other estimators show better stability. In fact, the N_{time} estimator is the most invariant to changes in I, once again demonstrating its accuracy and resilience to false positives.

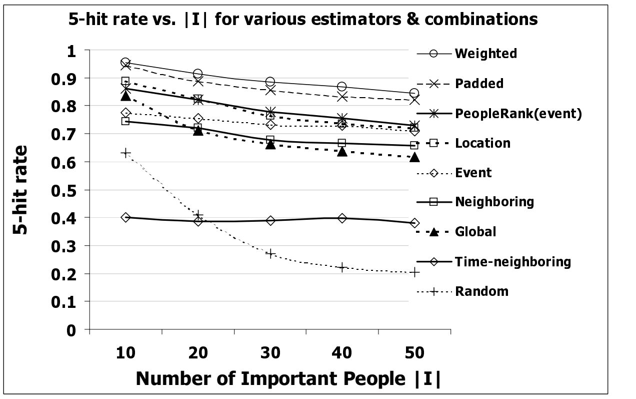

Figure 7: 5-Hit rate (averaged over all collections) for various estimators vs. size of the important people set I.

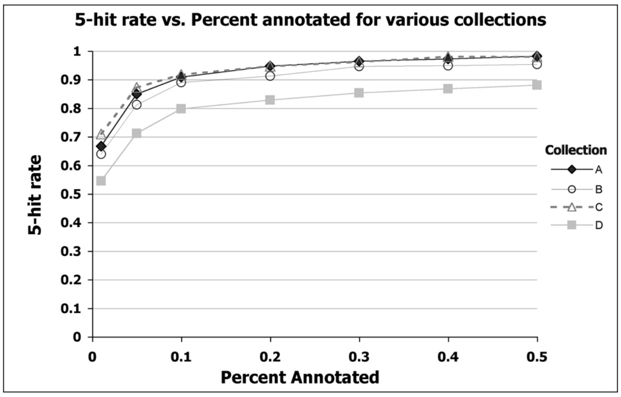

Figure 8: 5-Hit rate for the different collections, using the Weighted Estimator and varying values of $p_{annotate}$.

To illustrate the rapid improvement in the quality of the candidate list, and show the difference in performance between the different collections, Figure 8 displays the 5-hit rates of the Weighted Estimator for each of the collections in our experiment. The figure shows the effect of $p_{annotate}$, the percent of identities in photos annotated during initialization, on 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 10% of the identities are annotated, the 5-hit rate ranges from 0.74 to 0.86; 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 paper can verify this claim.

6. CONCLUSIONS AND FUTURE WORK

We have shown that our system can provide accurate identity label suggestions for users annotating their photo collections. These suggestions 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. Those patterns emerge during the annotation process.

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 little as 10–20% of the identities in the photos were previously annotated.

The short list of candidates will allow users to annotate their photo collection more rapidly, even on a small-screen device. While we were not concerned (in this paper) with the design of the interaction, one possible approach is using a Pie Menu where the user is shown a few easy-to-select options, including an additional menu option to expand the list of candidates.

Another immediate direction for future work is to test our system with a larger number of collections. This will allow us to tune the various parameters; alternatively, we can enhance the system with machine learning algorithms that will adjust the parameters for each collection. In addition, testing a system with human users will be informative in other ways: 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?

One other possible direction is adding more context-based estimators. For instance, following the example in [1], are some people more likely than others to appear in a photo if it is taken on a weekday morning, vs. a weekend afternoon?

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 our system to generate, for each photo, a short list of candidates and a prior probability for each. This data can be given to the face recognition system. Since the recognition system will have fewer candidate 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.

Lastly, transcending label suggestions, we aim at the real user need: we plan to build a system that combines context- and content-based techniques to support 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.

7. ACKNOWLEDGMENTS

We thank Emilia Anderson and QianYing Wang, who labored to annotate their photos for our experiments.

8. REFERENCES

[1] M. Davis, S. King, N. Good, and R. Sarvas. From context to content: leveraging context to infer media metadata. In *12th International Conference on Multimedia (MM2004)*, 2004.

[2] A. Girgensohn, J. Adcock, and L. Wilcox. Leveraging face recognition technology to find and organize photos. In *MIR '04: 6th ACM SIGMM international workshop on Multimedia information retrieval.*

[3] A. Graham, H. Garcia-Molina, A. Paepcke, and T. Winograd. Time as essence for photo browsing through personal digital libraries. In *Second ACM/IEEE-CS Joint Conf. on Digital Libraries*, 2002.

[4] E. Hjelmas and B. K. Low. Face detection: a survey. *Computer Vision and Image Understanding*, 83(3), 2001.

[5] A. Kuchinsky, C. Pering, M. L. Creech, D. Freeze, B. Serra, and J. Gwizdka. Fotofile: a consumer multimedia organization and retrieval system. In *Conference on Human Factors in Computing Systems CHI'99*, pages 496–503, 1999.

[6] M. Naaman, S. Harada, Q. Wang, H. Garcia-Molina, and A. Paepcke. Context data in geo-referenced digital photo collections. In *12th International Conference on Multimedia (MM2004).*

[7] M. Naaman, A. Paepcke, and H. Garcia-Molina. From where to what: Metadata sharing for digital photographs with geographic coordinates. In *10th International Conference on Cooperative Information Systems (CoopIS)*, 2003.

[8] M. Naaman, Y. J. Song, A. Paepcke, and H. Garcia-Molina. Automatic organization for digital photographs with geographic coordinates. In *Fourth ACM/IEEE-CS Joint Conf. on Digital Libraries*, 2004.

[9] R. Sarvas, E. Herrarte, A. Wilhelm, and M. Davis. Metadata creation system for mobile images. In *2nd international conference on Mobile systems, applications, and services*, 2004.

[10] B. N. Schilit, A. LaMarca, G. Borriello, W. G. Griswold, D. McDonald, E. Lazowska, A. Balachandran, J. Hong, and V. Iverson. Challenge: ubiquitous location-aware computing and the "place lab" initiative. In *WMASH '03: First ACM intl workshop on Wireless mobile applications and services on WLAN hotspots.*

[11] B. Shneiderman and H. Kang. Direct annotation: A drag-and-drop strategy for labeling photos. In *International Conference on Information Visualization*, May 2000.

[12] K. Toyama, R. Logan, and A. Roseway. Geographic location tags on digital images. In *11th International Conference on Multimedia (MM2003).*

[13] W. Wagenaar. My memory: A study of autobiographical memory over six years. *Cognitive psychology*, 18:225–252, 1986.

[14] L. Wenyin, S. Dumais, Y. Sun, H. Zhang, M. Czerwinski, and B. Field. Semi-automatic image annotation. In *8th International Conference on Human-Computer Interactions (INTERACT 2001).*

[15] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(1):34–58, 2002.

[16] L. Zhang, L. Chen, M. Li, and H. Zhang. Automated annotation of human faces in family albums. In *11th International Conference on Multimedia (MM2003).*

[17] L. Zhang, Y. Hu, M. Li, W. Ma, and H. Zhang. Efficient propagation for face annotation in family albums. In *12th International Conference on Multimedia (MM2004).*

[18] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, DEC 2003.