From Where to What: Metadata Sharing for Digital Photographs with Geographic Coordinates

Mor Naaman, Andreas Paepcke, and Hector Garcia-Molina

Stanford University

Abstract. We describe LOCALE, a system that allows cooperating information systems to share labels for photographs. Participating photographs are enhanced with a geographic location stamp – the latitude and longitude where the photograph was taken. For a photograph with no label, LOCALE can use the shared information to assign a label based on other photographs that were taken in the same area. LOCALE thus allows (i) text search over unlabeled sets of photos, and (ii) automated label suggestions for unlabeled photos. We have implemented a LOCALE prototype where users cooperate in submitting labels and locations, enhancing search quality for all users in the system. We ran an experiment to test the system in centralized and distributed settings. The results show that the system performs search tasks with surprising accuracy, even when searching for specific landmarks.

1 Introduction

Organizing digital photographs is a difficult task for many people [1, 2]. In many applications, simple text labeling of some photographs will enable much better 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. Can a cooperative information system enable a solution through sharing of existing labels so that nobody needs to do more work than they do now, yet everyone gains functionality? LOCALE is such a system.

Today's digital cameras add a considerable amount of metadata to an image file, most significantly a timestamp. The timestamp is already being used in photo browser applications. In our previous work [3] we also have shown how timestamps can be used to enhance browsing of a digital photo collection.

We believe that cameras will eventually support a "location stamp", specifying the geographic coordinates where a picture was taken. Two separate hardware advancements support this thesis: the lower cost of GPS chips, and the combination of the inherently location-aware cell phone technology[1] with digital cameras. Even today the standard EXIF header, included in most digital

[1] See http://www.fcc.gov/911/enhanced/ for FCC plan to mandate location capabilities of 50-100mts accuracy for mobile phones by 2005.

photos, supports location data. High-end cameras such as Nikon D1X have a direct interface to GPS devices. In addition, off-the-shelf software is available for merging GPS logs and digital photos to create "location-stamped" photos for any camera, without requiring a direct GPS interface.

Using location data as a pivot, we have a good basis for collaborative approaches towards photo management. For example, 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 proximity of photo origin is much easier to evaluate than current image-based proximity measures, like 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 coordinates of M's church photo will be the result of this search.

There are several potential problems to consider. First, H may have given the photo an unhelpful label (e.g., "My Son and I in California"). 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 H using a different, or shortened, name for a photographed object. Of course, H's label can just be plain wrong.

The solution we propose is LOCAtion-to-LabEl (LOCALE), a cooperative information retrieval system. The LOCALE system collects coordinates of photos and their associated labels from participating users, and responds to search queries. LOCALE applies term frequency, weighting and clustering techniques to avoid the problems mentioned above.

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 if M wishes to perform searches. 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.

Note that in both modes the photos themselves never leave their owner's machine, nor are the identities of the photographers ever needed for operation.

We have implemented LOCALE using three different strategies, in both centralized and distributed modes. The implementation strategies are described in Sect. 2. To test LOCALE, we devised an experiment to acquire geo-referenced, labeled photos from tourists visiting the Stanford campus. In Sect. 3 we present this experiment and the data we collected. The evaluation of search performance is presented in Sect. 4. We also looked at how well LOCALE can automatically assign labels to photos, and discuss some preliminary results in Sect. 5.

Two bodies of related work are image retrieval (overview in [4]) and image labeling ([5] and others) research. We expand on these and future work in Sect. 6.

2 The LOCALE System

The LOCALE (LOCAtion-to-LabEl) system consists of a centralized server with a global photo database DB_s, and users with personal photo databases DB_1, DB_2, etc. The system is illustrated in Fig. 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. Null values are permitted: in our implementation, 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).

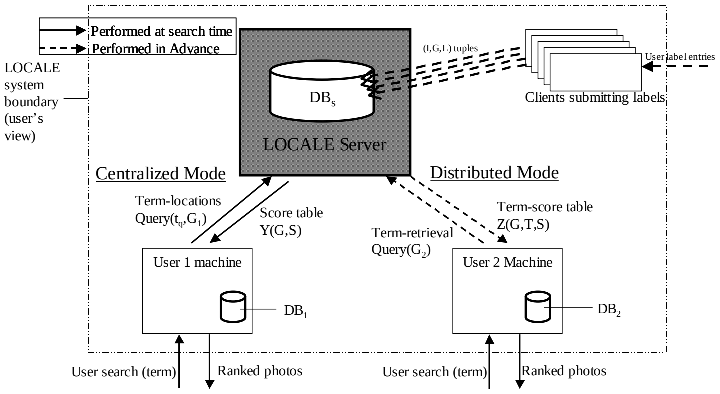


Fig. 1. Architecture of the LOCALE system

Labels can be any type of textual information attached to the photos: keywords, free-form text etc. In our system, labels are broken down into *terms* consisting of a single word or a two-word phrase.

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 is a search term t_q and, implicitly, the locations of the user's photos[2] $G_u = \Pi_G(P_u)$. The search output is an ordering of $I_u = \Pi_I(P_u)$ 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 (column L in each row of P_u is null). Therefore, we assume a different set of users that contribute labels (see Fig. 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. 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. 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.

2.1 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 Fig. 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. Recall from the introduction that we postulate digital cameras to provide the coordinates automatically. Searching users are not expected to specify coordinates manually[3]. 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 with P_1 to produce a ranking of the user's images $\Pi_I(P_1)$ based on the match to t_q.

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

[2] We will be using relational algebra operators such as Π, the attribute projection, throughout the paper.

[3] 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.

Term-Locations Query in 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 every $g \in G_1$:

$$Score(g, t_q) = \sum_{(i_s, g_s, \ell_s) \in P_s} IR(t_q, \ell_s) PROX(g, g_s)$$

The function $IR(t, l)$ computes the match between a term and a photo's label, while $PROX(g_1, g_2)$ evaluates the proximity between two photo locations. Our $PROX$ function computes the inverse of the square root of the Euclidean distance between $(g1, g2)$, but we set the value of $PROX$ to 0 if the distance between two locations is greater than a threshold. That is, we are taking into account only photos within a certain radius from g (100mts in our case). We did not use a linear distance measure since that measure assigns too much weight to close-by 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.

The IR function $IR(term, label)$, for our purposes, is a simple matching function:

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

For example, when searching for the term "tower" we gave the same score to the labels "Towers" and "The tower - what a tall tower!".

After computing the score for each g value, table Y is constructed: $Y = \Big\{(g, s) | g \in G_1; s = Score(g, t_q)\Big\}$. Table Y is the reply of the LOCALE server to the query; as mentioned above, the LOCALE system on the user's machine computes $P_1 \bowtie Y$ to produce a ranking on images instead of locations.

Term-Locations Query in Location-Clustered (LC) LOCALE. Location-Clustered LOCALE introduces some pre-processing on the location/label data. In this implementation the LOCALE server clusters the P_s table geographically using a hierarchical clustering algorithm. Then, LOCALE uses term-frequency methods to assign probable terms to each cluster in the hierarchy. The output of the pre-processing step is a clusters/terms table $CL(C, E, T, F, P)$ of clusters (C), their geographical extent (E), terms (T), the frequency of the term in pictures of this cluster (F), and the parent cluster (P). 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 cluster c_1, two tuples of the format $(c_1, e_1,$"Hoover Tower"$, f_1, p)$ and $(c_1, e_1,$"Memorial Auditorium"$, f_2, p)$ will appear in CL. Here, f_1 and f_2 are the frequencies in c_1 of "Hoover Tower" and

"Memorial Auditorium", respectively, e_1 is the geographical extent of c_1, and p is the parent cluster of c_1 (for example, p could be a cluster that includes all photos of the campus.)

During search, the ranking of User 1's photos is again computed via 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_ℓ. The cluster hierarchy is then ascended. Define c_ℓ's ancestors $ANC(c_\ell)$, and a view $TC_\ell = \sigma_{c \in \{c_\ell\} \cup ANC(c_\ell)}(CL)$. Then

$$Score(g, t_q) = \max_{(c_s, e_s, t_s, f_s, p_s) \in TC_\ell} IR'(t_q, t_s, f_s) PROX'(g, e_s)$$

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 and text/frequency match to the location g and the search term t_q respectively.

Function $PROX'$ is based on the probability of g belonging to c_s's extent, $Prob(g \in e_s)$, and (inversely) to the area of the cluster (the more broad e_s is, the less we value the match). The extent is represented by a two-dimensional Gaussian distribution. The function IR' is based on the frequency of t_q in tuples of cluster c_s in CL, but takes into account the sum of frequencies over all other terms that appear within c_s: the fewer other terms appear within c_s, the more relevant t_q is. This process is similar to the distributed-mode score computation described below (Section 2.2). The particular hierarchical clustering algorithm we used for LOCALE is agglomerative clustering (see [6]).

As in all other centralized-mode computations, the LOCALE server returns a table $Y = \Big\{(g, s) | g \in G_1; s = Score(g, t_q)\Big\}$. 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.

Term-Locations Query in 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. We compute the extents using a clustering algorithm. See Fig. 2 for the clusters corresponding to the term "fountain"; for illustration we manually marked the extents of the four main clusters in the figure. At the end of the pre-processing steps, we have a table $TC(T, C, E)$ of terms, their associated clusters, and each cluster's geographical extent described by a two-dimensional Gaussian distribution.

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 of term t_q:

$$Score(g, t_q) = \max_{(t_s, c_s, e_s) \in TC} IR''(t_q, t_s) PROX''(g, e_s)$$

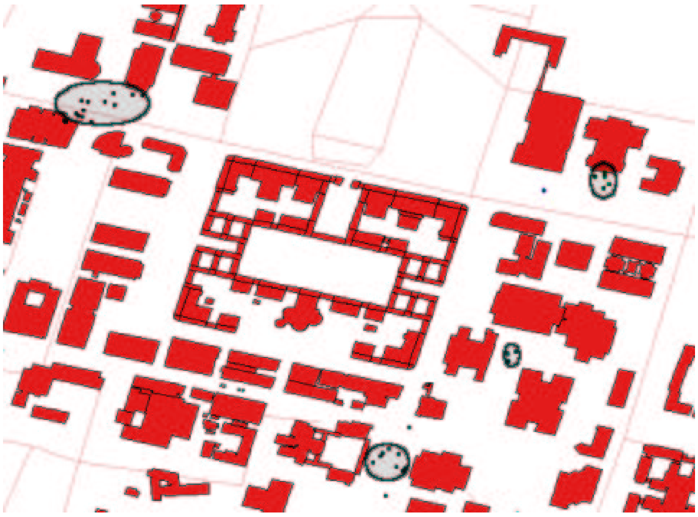


Fig. 2. Map of Stanford campus, with geographical distribution of photographs whose labels contain the term "fountain"

The function $PROX''$ is based on the probability of g belonging to the cluster extent, $Prob(g \in e_s)$. The function IR'' is the equality in singular form ($IR(t_q, t_s) = 1 \Leftrightarrow singular(t_q) = singular(t_s)$). As for the LC strategy, we use an agglomerative clustering algorithm. However, since TC required "flat" (one-level) clusters we flattened the cluster tree from the bottom up until we hit 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.

The LOCALE server then replies with $Y = \left\{(g, s) | g \in G_1; s = Score(g, t_q)\right\}$. 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. For the scope of our experiment, however, such calculation was not necessary.

2.2 Distributed Mode

In distributed mode (User 2 in Fig. 1), 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. To this end, the user's machine submits a *term-retrieval query* to the LOCALE server with the photo locations G_2. The reply from the LOCALE server consists of a table $Z(G, T, S)$ of locations, terms and scores which is then used to produce TS_2. User 2's machine retains TS_2 (it may also choose periodically to update it).

The details of this location-term score computation are different for each of the three strategies, and are listed below. In all strategies, the terms are picked and their scores are computed based on some notion of geographical closeness. Going back to our early example with users M and H, the reply Z to a query by M's machine, that includes the location of the church photo g_c, may include a tuple $(g_c, \text{“Stanford Church”}, s)$. The reply is 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, s may also incorporate other photos labeled the same way, and the reply may also include tuples for g_c with other terms.

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.

The second step, the actual search, is the same for all implementation strategies. This step is performed when the user submits a search query for term t_q. The search is done directly on table TS_2 - no other data is required. The system looks for possible matches to the search term t_q in the T column of TS_2. The lookup result, as in the centralize case, is a ranking of the photos $\Pi_I(P_2)$ 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 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 he has taken two images, i_c and i_k. The term-score table for M appears in Table 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_2 we adjust s by the total of scores for image i_c. The final score is computed as follows:

$$Score(i_c, t_q) = s \times \frac{s}{s + \sum \Pi_S(\sigma_{i=i_c, t_q \neq t}(TS_2))}$$

The same idea is extended for the case where terms consist of two words, but in this case we exclude terms that match either one of these words from the summation.

Finally, all the images in P_2 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 at LOCALE: the information is confined to the areas where the user has taken photos, and summarized as described above. In fact, in our experiments, the term-score table only kept the top 15 terms per photo. We show (Sect. 3) that even with this small amount of data (about 300 bytes per photo) we still achieved search results comparable to the centralized mode.

We now describe how the LOCALE server handles distributed mode term-retrieval queries under the different LOCALE implementation strategies.

Term-Retrieval Query in Distributed Weighted-Neighbors (DWN) LOCALE. Recall that the term-retrieval query parameter includes only the locations of the user's photos $G_2 = \Pi_G(P_2)$. A reply is a table $Z(G, T, S)$ of terms matching each location, and their matching scores. Recall also that the term-retrieval is performed in advance, before the user submits a search.

In Weighted Neighbors LOCALE, we compute Z by selecting possible terms for each location $g \in G_2$ from neighboring photos (photos in P_s taken in proximity to g). More formally, we compute a score for every term t that appears in P_s, with respect to the location g: $Score(g, t) = \sum_{(i_s, g_s, \ell_s) \in P_s} IR(t, \ell_s) PROX(g, g_s)$. The IR and $PROX$ functions are as defined in Sect. 2.1. The table $Z(G, T, S)$ is then constructed; $Z = \Big\{(g, t, s) | g \in G_2; s = Score(g, t)\Big\}$. As described above, the user's machine joins table Z with P_2 to generate the term-score table TS_2.

For example, say two photos p_j and p_k whose labels include the term "Church" appear in P_s; g_j is 10mts and g_k 15mts away from $g_c \in G_2$. Then $Score(g_c, \text{"Church"}) = IR(\text{"Church"}, \ell_j) PROX(10) + IR(\text{"Church"}, \ell_k) PROX(15)$. The reply table Z will include the tuple $(g_c, \text{"Church"}, Score(g_c, \text{"Church"}))$.

Term-Retrieval Query in Distributed Location-Clustered (DLC) and Distributed Term-Clustered (DTC) LOCALE. In both DLC and DTC strategies, the pre-processing in distributed mode is the same as in the respective centralized mode (Sect. 2.1). For both, the process of generating a term-score table for users in advance is analogous to the process performed by DWN. We therefore skip detailed discussions of these strategies.

3 The Visitor-Center 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 data set 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 statistically significant results. We therefore limited the data set to a bounded "world" – in our case, the Stanford University campus. Every day, tourists take photos on the campus[4], and we made use of these tourists to collect data for our experiment.

3.1 Experimental Setup

We provided loaner cameras and GPS devices to visitors taking the Stanford Visitor Center's campus tour. Thus, the data set 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. Some hours later, the participants were sent an email message that asked them to enter labels underneath 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 would like, but they were not *required* to label even a single photo.

[4] This fact also demonstrates that many photos are taken by different people in the same place (hence, the ability to share.)

We used software[5] to "align" the GPS track log and the corresponding tourist's photos via timestamps. We thereby created a geo-referenced collection P_u for each tourist's photos. The procedure produced geo-referenced photos with accuracy of roughly 10mts, limited by the original GPS accuracy and the track logs' time resolution.

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 geo-referenced. The primary reasons for unreferenced 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 introduced by the different use patterns with digital cameras: people take many more photos than a typical film camera owner would take in an hour. 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). Other problems may appear under any kind of setting. 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 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.

3.2 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. Fig. 3 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 position where the photo was taken. Incidentally, the third photo's position is right between the locations where the other two photos were taken.

[5] http://www.geospatialexperts.com/

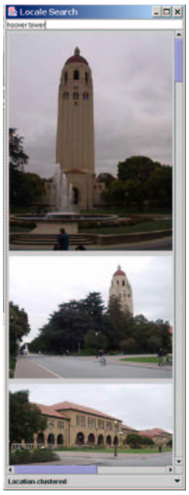


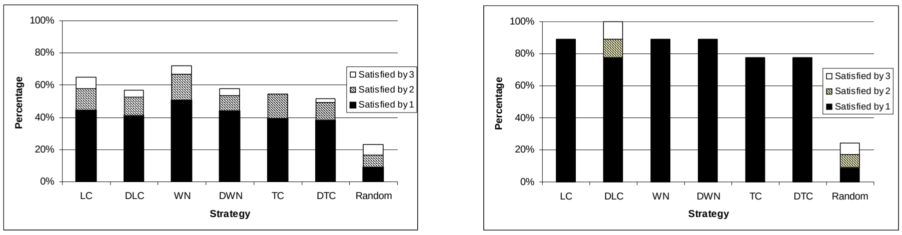
Fig. 3. LOCALE Search results for “Hoover Tower” query

We performed our evaluation for 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 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 for this collection were chosen from 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 labels that are geographically meaningful. There were 13 such collections in total. For each collection, we removed the collection’s photo metadata from the LOCALE database. 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



(a) All queries

(b) "Hoover" queries only

Fig. 4. The percentage of queries in every strategy that found a relevant photo within first three results

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."

4 Results

We first discuss the results for the individual scenario. Then we discuss the results for the global scenario.

4.1 Results for the Individual 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 Sect. 3.2, 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 lower than this limit. The results for the different strategies averaged over all collections and queries are shown in Fig. 4(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, as is included as a baseline for comparison.

A number of conclusions follow from Fig. 4(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 Sect. 2.2. 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 with the results of the global scenario.

The best performing strategies were WN, LC, and DWN. Incidentally, the percentage of queries satisfied by the first retrieved photo is around 40-50%. This percentage is quite good considering the search terms often included 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, Fig. 4(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 Fig. 4(b). In all the strategies, the first photo retrieved was relevant (a picture of the tower) in 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.

Based on the results in Fig. 4 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 query terms. We wish to examine the variability between the strategies when handling particular query terms. Figure 5 shows *recall* results for the most popular query terms: the X-axis corresponds to query terms that were used across at least three individual collections. The first number in square brackets next to each term is the number of collections we queried with this term. Remember: for each collection we picked its own query terms from its original labels. 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. "Quad" was the most popular term, appearing 80 times in photo labels.

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 x photos. As mentioned above, the number of relevant photos for each term and collection varied extensively. Thus, we could not use a fixed retrieval limit. We therefore set a different retrieval limit for each term and collection combination – the number of relevant photos. For each collection and term, we manually counted $T(t, c)$ – the total number of photos in collection c which are relevant for

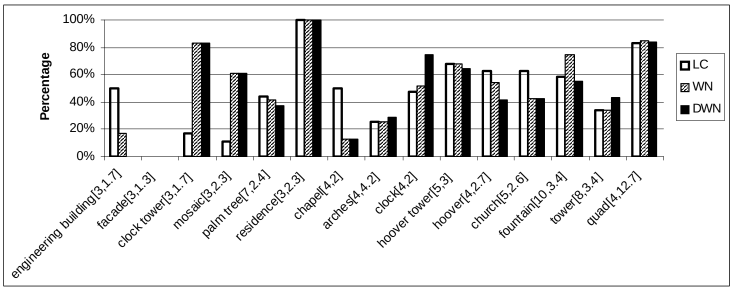


Fig. 5. Average *recall at* $T(t, c)$ for popular query terms

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 $T(t, c)^6$.

The bars in Fig. 5 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 for the fountain query in LC mode was 60%.

Generally, the performance of all three top strategies based on Fig. 5 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 Fig. 4(a). We have no intuition for why WN/DWN perform much better than LC for some terms, and much worse for others.

4.2 Results for the Global Scenario

In the global scenario, we have one collection that is the union of all unlabeled collections. As explained in Sect. 3.2, 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 6 compares the number of queries *satisfied*, the same metric used in Fig. 4(a). 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 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

[6] Note that when the retrieval is set to $T(t, c)$, the recall is equal to the precision.

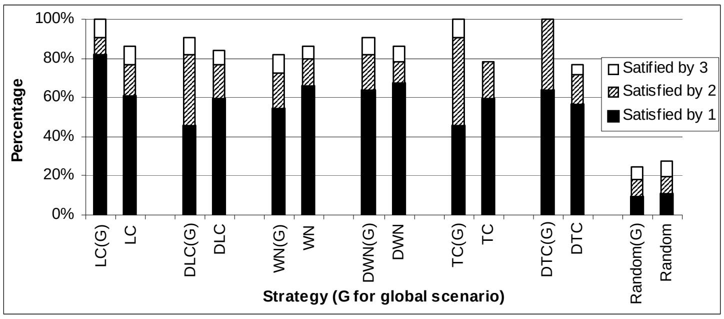


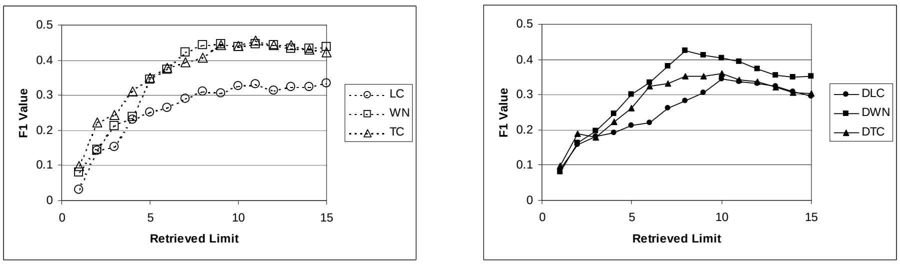
Fig. 6. The percentage of queries in every strategy that found a relevant photo in first three results, for global and individual scenarios

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 TC strategy described in Sect. 2.1. Once our system identified geographical extents that correspond to a term, we are likely to find fewer photos in 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 base for comparison. 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.

To compare the different mode/strategy combinations, we looked at the F_1 values over varying numbers of retrieved documents (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. Figs. 7(a) (strategies in centralized mode) and 7(b) (distributed mode) show the F_1 values for the 10 least frequent query terms – the terms used in the global scenario that appear the fewest number of times in the label database. For the 10 *most* frequent terms there were no considerable differences between the strategies, and hence we do not show results for them.

The X-axis in Fig. 7 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. For example, for our data, the best possible F_1 value at 1 is 0.27; at 8 is 0.85 and at 15 is 0.72. As the average number of relevant photos for these terms is 8.1, the optimal F_1 is reached at 8.



(a) Centralized mode

(b) Distributed mode

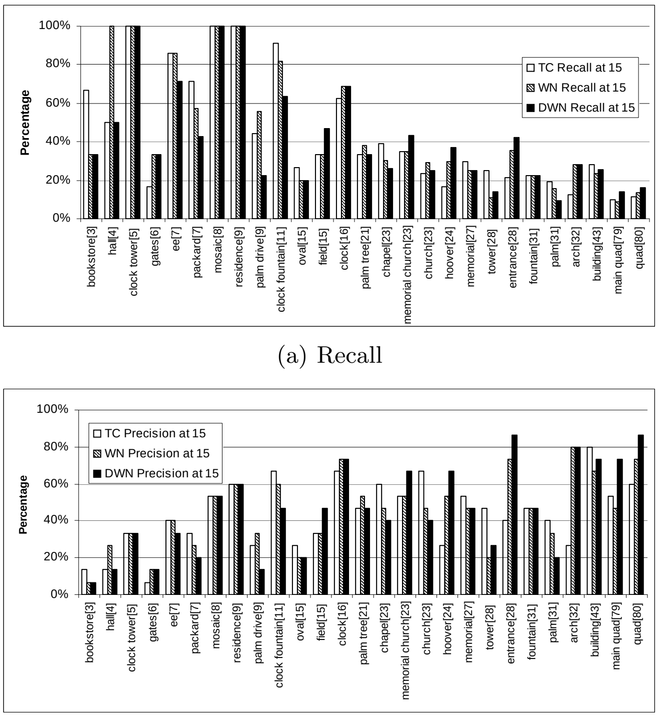
Fig. 7. Average F_1 values for least frequent query terms in different strategies, vs. retrieval limit

As seen in the figure, this is also the point where the actual F_1 peaks for most strategies.

As in the individual scenario, we see 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 Fig. 8 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 relevant photos in the collection for each query (in square brackets). In Fig. 8(a) the Y-axis is the recall. In Fig. 8(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 Fig. 8.

One possible predictor of successful vs. unsuccessful query terms is the concentration of other "interesting" landmarks around the unsuccessful terms. Two interesting outliers in Fig. 8 are "Gates" and "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 "fountain" offers a contrasting example. People are fascinated with running water, and most of the photos in the areas where fountains are found were, in fact, pictures of the fountains. The precision of fountain is expected to be higher.



(a) Recall

(b) Precision

Fig. 8. Recall and precision at 15 for each query term

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 enlarging the scope of summarization data for each photo. As we explain in Sect. 2.2, 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 then 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 holding more matching terms for each photo.

5 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 it did make these candidate terms available? 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.

Table 2. Photos and suggested terms

Actual Objects in Photo	*Suggested Terms*
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.

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 2 we show the top 5-6 suggested terms for a few of the photos in one collection. For each photo we list the objects that appear in it on the left, and the suggested terms on the right. The terms were generated by the Distributed Weighted Neighbors implementation.

These sample photos were chosen because the scores for their top suggested terms were especially high. The photos correspond to Stanford's most popular landmarks. 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. The fourth photo was taken in the Main Quad, in front of the church. 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.

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.

6 Related and Future Work

Two related fields of research are image retrieval and image labeling. Facilitating efficient labeling of photos has been a focus of research and development in recent years. Ease and automation of the labeling task was the focus of work in [7, 5, 8]. For example, [5] proposed a drag-and-drop approach for labeling people in photos. The latest photo browser software packages (Adobe's Photoshop Album, Apple's iPhoto[7] and others) also try to support efficient labeling. In Photoshop Album, labels are divided into categories (e.g., people, places). Such explicit user-entered place labels would simplify the LOCALE analysis. However, we designed the system more generally so that it could utilize any text associated with the photographs.

More relevant to our work 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 [9] for details. In contrast, our collaboration is implicit, and users do not need to share images.

In image retrieval, most of the work has been "content based" – using different technologies to extract and query by image features (see [4] for an extensive summary of research in this area). The most interesting future direction for LOCALE may be in augmenting the system with image retrieval and image-analysis tools. Feature extraction will enable better matching of labels and candidate photos. LOCALE can be augmented with systems like Blobworld [10, 11] to allow the automatic labeling of objects within images if the image occurs in a certain geographical area.

More internal to the LOCALE implementation, one possible future direction is developing a set of additional techniques to handle larger geographical areas, and higher condensation of data. For example, a LOCALE system covering the entire world should be able to assign not only local labels ("Hoover Tower") but also higher-level labels ("Stanford University"). In addition, we can think about using other data sources such as an "official" gazetteer. However, existing gazetteers (see [12] for example) are usually more reliable in identifying a city/state/country than a landmark related to a single photo. Using additional available metadata is another direction. Examples may be the direction the camera pointed when the picture was taken, or already-captured metadata such as focal point and F-stop. We also consider adding time sensitivity to LOCALE, so it can detect temporal outliers such as "graduation" appearing in an area associated with "Stanford University" during a few days in June. LOCALE should detect such anomalies, 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 - expert user being one who 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.

[7] http://www.adobe.com/, http://www.apple.com/

7 Conclusions

Our LOCALE cooperative image retrieval system addresses the problems of (a) searching and (b) labeling for global and individual photo collections.

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 be able 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 will be more significant matches than labels submitted by others. Ease of labeling will also benefit other cooperating users, as the assigned labels will be submitted and enrich the LOCALE system.

8 Acknowledgments

This work was made possible by the help of the Stanford Visitor Center. The center's staff was extremely helpful; we would especially like to thank Sean Fenton, Andrea Pazfrost and Lisa Mendelman for their efforts.

References

[1] 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.

[2] 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.

[3] 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. Available at http://dbpubs.stanford.edu/pub/2002-4.

[4] 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.

[5] 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.

[6] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[7] 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.

[8] 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), 9-13 July 2001, Tokyo, Japan*, 2001.

[9] 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.

[10] 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.

[11] 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.

[12] 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.