Assigning Textual Names to Sets of Geographic Coordinates

Mor Naaman, Yee Jiun Song, Andreas Paepcke, Hector Garcia-Molina

Stanford University

Abstract

NameSet is a system that translates a set of geographic coordinates into a textual name based on the geographic regions where the coordinates occur. One possible application of NameSet is to concisely present the geographical scope of a set of geo-referenced observations to a human user. Another application is to generate text to depict a set of coordinates that appear on a web site – text that could later be used for information retrieval applications. NameSet's computation is based on a simple algorithm, using off-the-shelf and web-based data sources. The system was proven effective in an application that automatically organizes and names sets of geo-referenced digital photographs.

Key words: geo-referenced data, named locations, information retrieval, gazetteers, photographs

1 Introduction

In many situations, it is necessary to describe a set of geographic coordinates using textual place names that are familiar to humans. For example, a certain web page may contain a list of geo-referenced observations, or a set of geo-referenced digital photographs. Representing this set using a textual human-readable name is useful for many applications including making the set available for text-based retrieval, as well as display and navigation of the coordinate set. [1] In the web context, research had tried to infer the geographic scope of web pages (Amitay, Har'El, Sivan, & Soffer, 2004; Ding, Gravano, & Shivakumar, 2000; Silva, Martins, Chaves, & Cardoso, 2004) — see Related Work Section. These techniques rely, among other parameters, on geographic

[1] Expressing geographical information in voice for the visually impaired is also a favorable outcome of this representation.

names that appear in the text, and are not useful when only a list of coordinates is available. We focus and report here on the problem of textual display and navigation of coordinate sets. However, the techniques can be applied towards text-based retrieval as well.

The problem can be simply stated as follows: *Given a set of diverse geographic coordinates, find a textual name that describes them best.* The use of 'diverse' proposes that the coordinate set may not be trivial (e.g., not all the coordinates occur within a single city). However, it is assumed that the set is somewhat coherent — it is not the case that some coordinates are in Switzerland while other coordinates in the same set are in New Zealand. A name for a coordinate set may include names of features such as cities or parks inside which coordinates in the set occur. The name may also include names of nearby features. A sample such name is "Long Beach (56kms S of Los Angeles)".

While it is clear how naming a set of coordinates is beneficial for text-based retrieval, it may not be clear why naming is required for presentation to users, and for collection navigation, as using a map to display the set of coordinates might be sufficient. However, maps are inefficient in their use of screen real estate. For example, the map-based overview of two sets of coordinates, one from the San Francisco area and one from Paris, would occupy much of the screen with (visual) geographic information that is not pertinent to the data. Moreover, when a new set is added that is very close to Paris, the scale of the screen, which must include San Francisco, will not allow differentiation of the two Paris area sets.

Research of map-based interfaces to digital libraries and geo-referenced material (Cavens, Sheppard, & Meitner, 2001; Leclerc, Reddy, Iverson, & Eriksen, 2001; Smith, 1996; Zhu et al., 1999) has focused on browsers for the desktop environment. Yet the map problems intensify when the user operates on a small-screen device as maps are not well suited for this environment. While work has been done on summarizing maps for screen constraints (Sarkar, Snibbe, Tversky, & Reiss, 1993), most applications of maps on small screen devices, like (Robbins, Cutrell, Sarin, & Horvitz, 2004), focus on navigating a restricted area and context (e.g., driving directions).

In contrast to map-based interfaces, textual browsers may be more capable of screen estate parsimony, offering more efficient presentation for global, unrestricted collections. In a textual browser, it only takes two short lines of text to represent the San Francisco and Paris sets mentioned above.

Maps also pose a problem when handling devices with limited input mechanisms (such as cell phone inputs or voice activation, for example). Such mechanisms are not be well suited to map-based manipulations. Finally, a number of people are uncomfortable with maps and prefer other types of browsing.

The *NameSet* system described in this paper can provide an alternative to a map-based representation of sets of coordinates. In fact, the system was implemented and tested successfully using a sample application — representing sets of geo-referenced digital photographs.

In the next section we briefly mention some related work. In Section 3 we describe a sample application of the NameSet system. Section 4 describes the algorithms and implementation details of NameSet. An evaluation of the system is found in Section 5.

2 Related Work

The related work described in this section includes efforts to determine the geographic scope of web pages; on-line gazetteers and geographic information sources; work on matching and naming geographic regions; and work in the GIS community on browsing location coordinates in digital libraries using maps. In addition, we mention some relevant research on collections of photos and photo labeling.

A number of research efforts have been trying to utilize and enhance the web using geographic concepts. The Web-a-Where system (Amitay et al., 2004) disambiguates geographic terms that appear on a web page, and tries to determine the geographic focus of that page. Other projects (Ding et al., 2000; Silva et al., 2004) try to use the textual content as well as the geographic distribution of hyperlinks to a web resource to asses the resource's geographic scope.

Several online databases are available that provide valuable sources of geographic data. In our work, we used an off-the-shelf geographic dataset as a source of named location entities, but other sources of relevant information are available. The Alexandria Project is a distributed digital library for materials that are referenced in geographic terms ("gazetteer") (Hill, Frew, & Zheng, 1999; Smith, 1996). The Alexandria database provides a rich resource for named geographic entities. Other publicly available gazetteers include the Geographic Names Information System, the Columbia Gazetteer of the World, and the US Gazetteer operated by the US census bureau. [2] In addition, Microsoft's MapPoint Web Service [3] provides a programmable interface that supports geographic queries for businesses and other points of interests.

The Geo-SPIRIT project (Jones et al., 2002) is looking at ways to use the

[2] http://geonames.usgs.gov/, http://www.columbiagazetteer.org/, http://www.census.gov/cgi-bin/gazetteer/

[3] http://mappoint.msn.com/

web to describe imprecise regions (such as "Northern California" or "The Midwest") that could prove useful in naming a set of coordinates (Arampatzis et al., 2004). In (Larson & Frontiera, 2004), Larson and Frontiera proposed and evaluated area-matching algorithms. Such work could provide the matching needed between areas defined by our sets of coordinates to precise or imprecise geographic regions that overlap the coordinate sets. It may be possible to use these techniques to better inform our system about the suitability of generated names.

A number of systems have been developed that deal with presenting geo-referenced data, especially within the Geographic Information Systems (GIS) community (Cavens et al., 2001; Leclerc et al., 2001; Smith, 1996; Zhu et al., 1999), but nearly all of them rely solely on a map based interface to represent the location data. In (Cavens et al., 2001) the application displays geo-referenced photos as points on a zoomable map interface, but the user is unable to see any of the actual photos until a specific point is selected. For examples of work on summarizing maps and displaying maps on small-screen devices, as mentioned above, see (Robbins et al., 2004; Sarkar et al., 1993).

The main application domain we report on in this paper is geo-referenced digital photographs. Associating GPS coordinates with digital photographs is a fairly recent development. To the best of our knowledge, ours is the first attempt to automatically name sets of photographs utilizing their geographic location. However, there have been many other projects which attempt to exploit the geographic information of digital photographs for different purposes. Most notably, Toyama et al built a database that indexes photographs using time and location coordinates (Toyama, Logan, & Roseway, 2003). The Toyama work explored methods for acquiring GPS coordinates for photographs, and exploiting the metadata in a graphical user interface for browsing. The mappr.com project is another well-known effort to utilize a map for browsing global collections of geo-referenced photos.

In (Naaman, Paepcke, & Garcia-Molina, 2003) we explored a different version of the problem defined in this paper. In that earlier work we built on an existing set of coordinates, each already associated with some *free-text* caption. Given that set and its captions, our system (called *LOCALE*) proposes a good geographically-meaningful name for the set. More interestingly, LOCALE can propose a name for a new un-labeled coordinate (or coordinate set) that occurred in the same geographic area, based on the original set's coordinates and captions. Sarvas et al studied a similar problem in (Sarvas, Herrarte, Wilhelm, & Davis, 2004).

While our work focuses on providing useful textual names for collections of photographs, Liberman and Liu have done work that attempts the reverse. In (Lieberman & Liu, 2002), they use natural language parsing techniques

to suggest relevant photos as a user types a story to describe a series of photographs. Of course, other work that attempts to automatically produce metadata for digital photographs has been based on image analysis techniques. Many research projects attempt to utilize image analysis to propose annotations for photographs, like (Kuchinsky et al., 1999; Duygulu, Barnard, Freitas, & Forsyth, 2002). See (Veltkamp & Tanase, 2002) for an updated summary of content-based image retrieval techniques.

3 Sample Application

In (Naaman, Song, Paepcke, & Garcia-Molina, 2004) we used location information to automatically organize collections of geo-referenced digital photographs. Our system, PhotoCompas, automatically groups the photos into sets that represent different events and locations where photos were taken. The grouping is executed using the time and location metadata associated with each photo. Once this step is completed, PhotoCompas needs a way to present the results in a user interface, without the benefit of a map. The second processing step is therefore to assign textual names to the nodes in the location and event hierarchies. The names for the nodes are generated by NameSet, based on the coordinates of the photos belonging to these nodes.

Figure 1 shows a subset of a sample location and event grouping created by the PhotoCompas algorithm. The location hierarchy is shown at the top of the figure. At the highest level, PhotoCompas simply groups the coordinates by country. The second level of the hierarchy is generated based on the specifics of the collection. Again, PhotoCompas generates this hierarchy based on the coordinate values only, independently from the information about containing features. In this sample case, we can see in Figure 1 that PhotoCompas created sets that correspond to the Yosemite area, Seattle area and the San Francisco area. The San Francisco node is split into more detailed nodes, because PhotoCompas decided that this set of coordinates is large enough to be split again. In reality, the decision is more nuanced but exceeds the scope of this paper; for details refer to (Naaman, Song, et al., 2004).

When the hierarchy is ready, the NameSet module kicks in to assign names to nodes (i.e., coordinate sets) in this hierarchy. NameSet considers the set of coordinates in each node, but also considers the position of the node in the hierarchy: leaf nodes (e.g. "Seattle", "Berkeley" in Figure 1) will be named in a slightly different manner than inner nodes (e.g., "Around: San Francisco..."), as we show in Section 4.

At the bottom of Figure 1, we show a sample of a different dimension of grouping coordinates in the collection. This grouping is based on "events" —

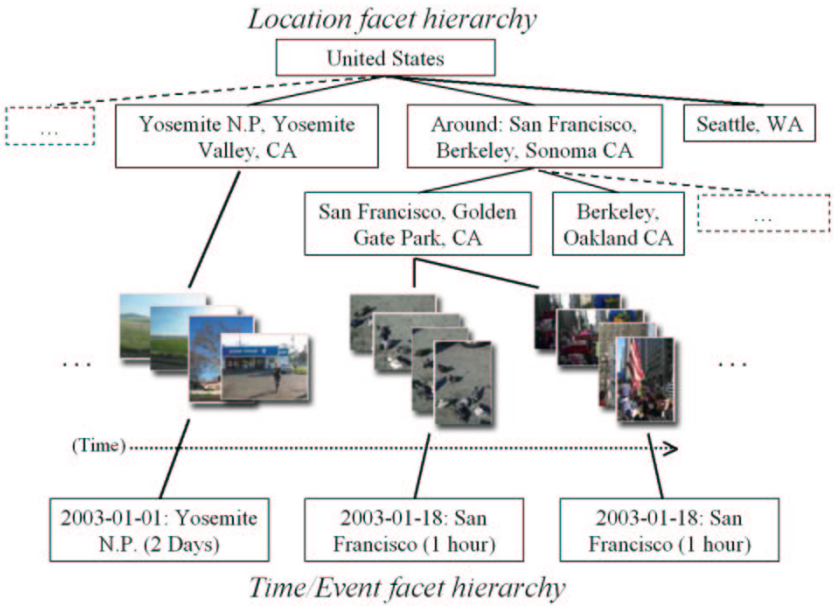

Fig. 1. Sample PhotoCompas structure. Parts of the location hierarchy (top) and the time/event hierarchy (bottom) for an actual collection of photos, including names as generated by our algorithm.

roughly speaking, sets of coordinates that represent photos taken at the same place *and time*. As events represent coherent locations, we can use NameSet to name them as well.

To summarize, our system creates three different types of coordinate sets that are guaranteed to be relatively coherent (and thus relevant to the problem we address in this paper): leaf location nodes, inner location nodes, and event nodes. NameSet's goal is to have the textual representation of those sets "make sense."

4 The NameSet System

The naming process in the NameSet system is composed of three distinct processing modules, indicated by the rectangles in Figure 2 (the data in the figure is represented by ellipses). In the first processing step, the system finds the containing features (such as cities, parks, or states) for each coordinate in the set to be named. In parallel, the system looks for good reference points such as nearby big cities, even if none of the coordinates in the set appear to be inside these cities. Finally, NameSet decides which of the containing features or nearby reference points to use when picking the final name for the given coordinate set. The name can include more than one feature of each type: "Sonoma, Boyes Hot Springs (98kms N of San Francisco)" is one example of a name created by NameSet. Another such name may simply be "Stanford".

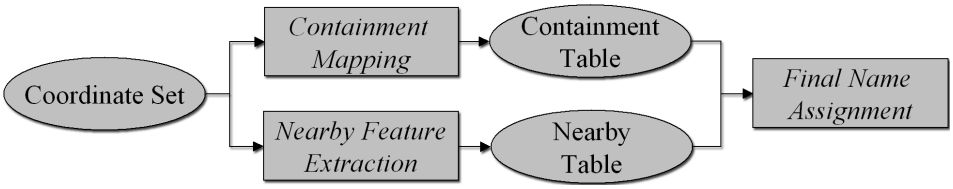

Fig. 2. Processing diagram of the NameSet system

For concreteness we refer in the description of NameSet to our sample application, geo-referenced digital photographs. We use the terms 'photos' and 'coordinates' interchangeably. Similarly, we use 'node' as a synonym for 'coordinate set'.

4.1 Generating a Containment Table

This initial step of generating a containment table involves finding the containing features for each coordinate. For each latitude/longitude pair, we find the country, state, city and/or park that contain it. This is done using an off-the-shelf geographic dataset of administrative regions (from Environmental Systems Research Institute (ESRI), 1995). For example, a particular coordinate may be inside the United States (country), California (state), San Francisco (city), and Golden Gate Park. Another coordinate may be in Washington (state) and Seattle (city), but not in any park. Regretfully, below the country level, our dataset included only US administrative areas. Thus, we have only tested our naming procedure on US coordinates.

The dataset we use is based on accurate polygon-based representations for the different administrative features (country, state, city etc.). The data is slightly dated (some of the data goes back to the mid-1990s), but still largely relevant.[4] In addition, the polygon representations can never be perfectly accurate. Based on informal observations, we estimate the error of the representations in our particular dataset to be around a few dozens to hundreds of meters, certainly within reason for this application.

The coverage of some administrative features in our dataset is exhaustive and exclusive. For example, inside the US, each coordinate is contained by exactly one state and one county. Naturally, some features do not exhaustively cover the entire map. For instance, unfortunately, parks do not cover the entire space of the US. Similarly, a coordinate is not guaranteed to fall inside a city. The park/city data is not exclusive: a certain location can be contained in both a city and a park.

If no polygon (or administrative feature) of a certain type matches the queried

[4] Up-to-date datasets are available if needed, of course, for a price.

Fig. 3. Sample set of photos (circle markers, possibly overlapping) and a subset of the containing features of type "cities" and "parks."

coordinate pair, we allow the system to look for relaxed matches by artificially expanding the borders of the polygons of that type. For example, if our dataset suggests that the (latitude, longitude) pair does not fall inside any park, but is 200m away from Yosemite National Park, the algorithm would mark the coordinate as contained in Yosemite park. While the error allowed is relatively strict (limited to less than 1km) when looking for a containing park, county or city, we relax the distance requirement significantly when trying to find containing *countries*, making sure we capture coordinates that are a quite a few kilometers off shore.

At the end of the coordinate resolution process we have the containing features for every coordinate in the set. The system then computes the frequency at which each administrative area name occurs in the set of coordinates, building a term-frequency table.

For example, we may have a set of 700 photos, 400 of which were taken in the city (and county, as they are the same) of San Francisco. Out of the San Francisco photos, 50 were taken in Golden Gate Park. The other 300 photos in the set were taken in various locations in Marin county. Figure 3 shows an example for such a coordinate set. Table 1 shows a possible term-frequency count for the coordinate set depicted in Figure 3. The figure demonstrates cases where relaxation is required in a city query: there are two coordinates in the set, seen just off the northern-most tip of San Francisco, that seem to fall just outside the city limits (in fact, they are on the Golden Gate bridge). NameSet will treat these coordinates as if they are contained in San Francisco.

Not all types of administrative areas are equally recognizable to users. For example, national parks may be more famous than county parks; similarly, city names are more likely to be recognized than names of national forests. For this reason, the system weighs each type of administrative area differently,

Table 1
Possible term-frequency table for a sample set of coordinates.

Area Name	Type	Count
San Francisco	City	400
Marin	County	300
Golden Gate NRA	Nat'l Rec. Area	120
Muir Woods	Nat'l Park	80
Golden Gate Park	County Park	50

with national parks weighed more heavily than cities, and cities weighed more heavily than other parks such as state parks or national forests. The different weights are listed in Table 2 and allow us to give more importance to names that are more likely to be recognizable to users. The weights were assigned empirically, and we show in Section 5 that they perform well. Notice that for some of the administrative types the assigned weight is 0, implying that features of this type are ignored. Indeed, we found that counties and national forests are rarely recognizable by name to users, unless the users live in that specific county.

In the future we would like to be able to more adaptively select weights based on learning and user feedback. Alternatively, instead of assigning weights to *types* of locations, the system could rate each individual named place separately: San Francisco will be rated differently than Palo Alto, although they are both cities. The rating could be done by using a measure of "importance", such as population density, or a Google count (the number of hits returned by Google when searching for the location name). We use the latter method when looking for nearby reference points, as described below.

NameSet applies the weights (of Table 2) to the relevant counts (e.g., Table 1). For example, the adjusted score/count for Muir Woods National Park is $80 \times 5 = 400$, while Golden Gate National Recreation Area scores $120 \times 2 = 240$, now ranking lower than Muir Woods.

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

4.2 Generating a Nearby-Cities Table

In parallel to finding containing features, as shown in Figure 2, we look for cities that *neighbor* the set of coordinates to be named. These nearby cities can serve as reference for the given coordinate set, in case the containment

Table 2
Types of administrative areas and the weights assigned by the system to instances of each.

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

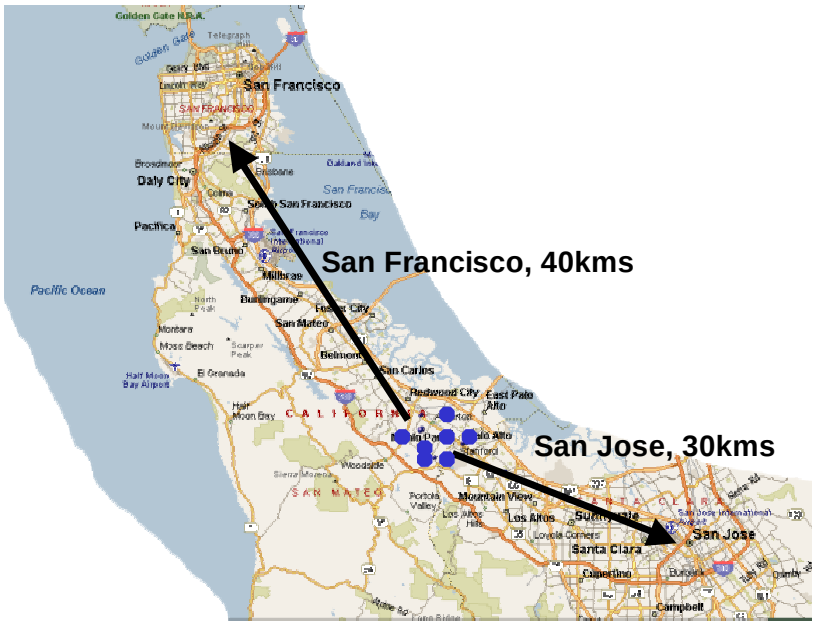

Fig. 4. A set of coordinates and two nearby cities that may serve as reference points for the set.

features of coordinates in the set are not sufficient to represent the set well. This problem can emerge, for example, if coordinates in the set do not fall within any city or park boundaries, or occur sparsely in some area, without any critical mass of coordinates inside any of the named locations. See for example Figure 4, where a small set of coordinates appears sparsely scattered within a number of cities around Stanford University. By locating cities that are close to the coordinates in this set and computing the distance from the center of the set to the city, the system is able to produce textual names for such nodes, such as "40kms south of San Francisco".

To compute the center of a coordinate set, we considered two methods. The

first is simply to compute and use the weighted average of the coordinates in the set. This method is satisfactory only if the set of photos is relatively coherent; e.g., represents a single cluster of points (perhaps like the one represented in Figure 4). However, in many cases the set of coordinates represents a number of different, yet proximate, locations. For example, Figure 3 shows a set of photos that occur around a few key locations in the San Francisco area. A weighted average of the coordinates will result in a center location that does not represent well any of the locations included in the set.

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

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

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

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

If the search for nearby cities results in more than one city, the system tries rank them. The ranking attempts to strike a balance between a city being relevant and known to the user, and relevant (in terms of distance) to the set of coordinates. To do that, we use the notion of "gravity": a combination of

Algorithm 1 Computing the "biased center" of coordinate set S

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

population size, the city's "Google count", and (inversely) the square root of the city's distance from the center of the set. The "Google count" of a city is the number of results that are returned by Google [5] when the name of the city (together with the state) is used as a search term. We use this heuristic as a measure of how well known a city is, and thus how useful it would be as a reference point.

To give an example of the application of gravity in our system, refer to the set of coordinates near the Stanford campus, as depicted in Figure 4. The center of the set is computed as described above to be somewhere around Stanford campus. A query to our dataset reveals two major nearby cities: San Jose and San Francisco. The nearby reference points can be either "40kms South of San Francisco", or "30kms North of San Jose". The population of these cities is comparable: 945,000 residents in San Jose, vs. 800,000 in San Francisco. The distance of both cities from the set of coordinates is also comparable. However, the Google count for San Francisco (search for 'San Francisco, California' results in 44,800,000 hits) is much higher than San Jose's (search for 'San Jose, California' results in 13,300,000 hits). San Francisco therefore is ranked higher despite being further away: $Score(\text{San Francisco}) = \frac{800*44.8}{\sqrt{40}} = 5666 > Score(\text{San Jose}) = \frac{945*13.3}{\sqrt{30}} = 2294$.

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

The computation described in this section, as shown in Figure 2, creates a *nearby-cities table*, again with terms and their scores.

The final step, as depicted in Figure 2, involves picking 1–3 terms from the nearby-cities and containment tables to appear in the final name of each set of coordinates. For example, a possible name can include the two top terms from the containment table, and the top nearby city: "Stanford, Butano State Park, 40kms South of San Francisco, CA". Our method of picking the terms that compose the final name varies according to the semantics of the set of coordinates we are trying to name, as we explain in Section 4.4. Before we do that, we report on our attempts at utilizing a web-based gazetteer to generate more candidate names for a coordinate set.

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

4.3 Using Bounding-Box Gazetteers

We have experimented with using existing web-based gazetteers as a source of additional name candidates for a coordinate set. These gazetteers offer comprehensive lists of named geographical features of various kinds, as opposed to our dataset that offers only features of certain administrative types. On the other hand, the area representation used by web-based gazetteers is often simplified, and is not as accurate as the polygon-based representation of our dataset.

While it seems like web-based gazetteers would have been useful for this task, we could not overcome some inherent limitations in using them for our purposes.

Specifically, we chose to work with the Alexandria Digital Library's gazetteer (Hill et al., 1999; Smith, 1996), possibly the most encompassing and well-known among web-based gazetteers.

We saw two possible ways to utilize Alexandria for our problem. Firstly, we could use it to map from coordinates to a containing feature, corresponding to our first step described in Section 4.1. However, Alexandria represents geographic features by a rectangular bounding box, which is not accurate enough for our needs. For example, querying with a coordinate in San Francisco returns Canada as one of the containing areas. Our aforementioned dataset is less encompassing but uses much more accurate polygon representations of geographic features.

Secondly, we could use Alexandria to map *from* an area (bounding rectangle representing a set of coordinates we wish to name) to a list of contained features like cities, parks and even rivers, waterfalls and mountains. Then we could use this list to find prominent features that could be candidate names for the given set of coordinates. Unfortunately, we found it hard to formulate a query that will supply us with a good enough list of contained features — the list was either too noisy, containing highly specialized features, or too sparse. When too noisy, we did not find a good way to distinguish features in the returned set that would possibly be useful names to the system's users.

We did try to use the Google count metric to distinguish more well-known features from others, but results were not satisfactory in this case. In particular, when the features are not cities, it is not clear which search query should be used to estimate the popularity of a certain feature, as similar names often appear in many different contexts. For example, "The Presidio": should the search be performed on query term as listed (The Presidio is a name of a park and residential area in San Francisco among other references), or should "San Francisco" or even "California" be appended to the query term? This

ambiguity prevented us from using the full power of the Alexandria gazetteer to generate an extensive list of related features.

We could limit the retrieval from Alexandria to a set of well-defined administrative regions, as we did with our own dataset. However, it was preferable to use our dataset, since it offered a more accurate representation of the features than a bounding box.

Despite these difficulties, we are still hoping to find a way to use the vast information space of the Alexandria gazetteer, or other online gazetteers, in future work.

4.4 Approaches for Assigning Final Names

Picking the final terms for the textual description of the coordinate set from the containment and nearby-cities tables, is done in different ways depending on the semantics and purpose of the name. Recall that in our sample application, NameSet aims to augment the automatically generated organization of photo collections. In this context, we have three different ways to pick a final name, depending on whether we are naming an event from the event hierarchy, a leaf node in the location hierarchy, or an inner node in the location hierarchy (sample event names are shown at the bottom of Figure 1, and samples of the latter two are shown at the top of Figure 1). The discussion in the rest of this section, therefore, is geared towards the semantics of our application. Other applications of NameSet could have different approaches for assigning the final name for a coordinate set.

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

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

Other considerations may also be relevant. For example, the names assigned to different nodes should not be too similar, both in the sense of containing references to the same locations (which can occur in some data, depending on the way location nodes are computed), and in the sense of containing similar-looking words that do not represent the same location but might hinder the user's visual scan capabilities.

Naming a leaf node in the location hierarchy is the hardest and most impor-

tant of the three different node types in our PhotoCompas system. For the PhotoCompas application, these leaf node names must be assigned the most accurate description since there is no lower level of the hierarchy where more location details are exposed. On the other hand, the leaf location nodes may still represent a heterogenous set of photos that were shot in a diverse set of administrative regions.

The rules we use for naming a leaf node are:

(1) Use the top term from the containment table, if one exists.
(2) Concatenate the second term from the containment table only if it has a significant score in this set (e.g., if this term appeared only twice, we do not want to use it).
(3) If the number of coordinates in this set is low (suggesting that the user is not very familiar with this area), or if the scores for the top two terms are low (suggesting no one of them will serve as a good reference for the user), concatenate the top term from the nearby cities table to the name.

At the end of this process, we may have anything from 1 to 3 terms that are combined into a textual geographic description for this node.

Next let us consider the inner nodes in the location hierarchy. The top nodes represent a country, and are easy to caption. For the other inner nodes, such as the "Around San Francisco" node in Figure 1, we pick the single top-scoring term from the containment table of each of the node's descendants, and take the top three terms in this list. For example, a set of photos taken in the San Francisco bay area may be named "San Francisco, Berkeley, Sonoma" as the corresponding node has 5 descendants, and these three chosen names are ranked highest out of the 5 candidates. The hope is that the three names represent the general area of the coordinate set represented by this node; for a more accurate description of the area, one can turn to the lower level nodes. However, if this node is the only one in this specific state, and does not occur predominately in one city or park, it will be assigned the state's name ("California").

Finally, when assigning a textual geographic caption to an event (a set of photos taken at the same place, on a single occasion), we assume that the user has visited relatively few places over the duration of the event. Furthermore, we assume the user will get some orientation, and more geographical context and browsing power, from the location hierarchy. For event names, then, we pick only the top term in the containment table. If one does not exist, we choose the top term from the nearby cities table. In any case, we augment the name with the date and time span of the event, e.g. "Boston, Dec 31st 2003 (3 Hours)".

Note that strategies listed in this section are geared towards a naming a node

Table 3
Details of the collections used in our experiments

Collection	*Number of Photos*	*Location Nodes*
A	2580	16
B	1192	9
C	1823	17

in a personal collection of photos. In other words, the assumptions is that the person for whom the labels are generated in somewhat knowledgeable about the locations of the coordinates in the set. A different strategy may be applied when naming a globally-accessed set of coordinates, where the familiarity of people with the locations in the set is not guaranteed. For example, it may be appropriate to bias more heavily towards well-known features, rather than features that contain more of the coordinates.

5 Evaluation

Since the context of the NameSet application was collections of geo-referenced digital photographs, we evaluated the results using three real-life collections of photos. The number of collections is low due to the current scarcity of large geo-referenced consumer photo collections. We expect more collections to be available in the future, but as of now, we were only able to find three subjects with a large enough collection of such photos (each spanning thousands of photos, and at least one year of photo-taking). Our results are then, by necessity, case studies rather than a statistically significant analysis. However, these collections supplied dozens of distinct sets of coordinates for the algorithm to name; in this sense, the evaluation was quite broad. In addition, Section 5.1 reports on an additional, indirect evaluation of NameSet.

Some key properties of the test collections are listed in Table 3, including the number of photos in each, and the number of location nodes (or coordinate sets) that NameSet was executed and tested on.

For illustrative purposes, Figure 5 shows the names created by NameSet for all the nodes in the location hierarchy of the *C* collection. Notice that, as described in Section 4.4, the number of features mentioned in a single node's name ranges from one to three. Some of the names include a nearby additional reference point (e.g., "6 kms NW of San Jose" for the Mountain View node), while others do not (e.g., the Stanford node, as NameSet decided that Stanford alone was sufficient). Some of the reference points are further away than others, for example the "Colorado" node where the coordinate set represents a remote area in the Rocky Mountains.

-San Francisco, Berkeley, Sonoma, CA	876
•**Berkeley; Oakland**	188
•**Glen Ellen; Eldridge** (97 kms N of San Francisco)	22
•**Petaluma** (91 kms NW of San Francisco)	3
•**San Francisco; Golden Gate N.R.A**	637
•**Sonoma; Boyes Hot Springs** (83 kms N of San Francisco)	26
-Stanford, Mountain View, Monterey, CA	284
•**Monterey** (93 kms S of San Jose)	12
•**Mountain View** (6 kms NW of San Jose)	29
•**Stanford**	243
-Colorado (350 kms W of Denver)	180
-Long Beach (56 kms S of Los Angeles, CA)	90
-Philadelphia, PA	8
-Seattle, WA	39
-Sequoia N.P. (244 kms E of Fresno, CA)	133
-South lake Tahoe; Bear Valley, CA	96
-Yosemite N.P.; Yosemite Valley, CA	116

Fig. 5. The nodes in the location hierarchy of Test Collection C (1822 photos), names assigned to the nodes by NameSet, and the number of photos in each node.

We evaluated NameSet's performance through interviews with the owners of the test collections. We concentrated on names for the nodes in the location hierarchy and did not study the success rate of the geographic names produced for events. Our evaluation goals were to verify that the produced textual names are:

- Useful to the subjects, in that a) the name includes terms that are familiar to the subjects and help them understand which geographic area is covered by the node, b) the subjects are able to tell the node apart from other nodes, based on the name and c) the subjects can tell which pictures in the collection belong a specific node based on its name.
- Similar to the names that the subjects themselves would have suggested for these nodes.

For each collection, and each node, we performed several tests. In the first test, we showed the subjects a *candidate terms set* for each node. The candidate terms set is the union of terms that appeared in the *containment table* and *nearby cities table* as defined in Section 4. For example, if a certain node contains photos from Redwood National Park, the city of Eugene (Oregon) and Crescent City (California), then our list included all those city, park and state names, plus the appropriate counties and other nearby big cities. The average length of the candidate terms sets for the different nodes was 19 place names; it was generally shorter for leaf nodes as they contain fewer coordinates. Figure 6 shows a set of candidate location terms generated for the "Yosemite" leaf nodes in collection C.

For each node, we asked the owner of the collection to pick up to three place names from the candidate set, that represent this node best (in fact, they only

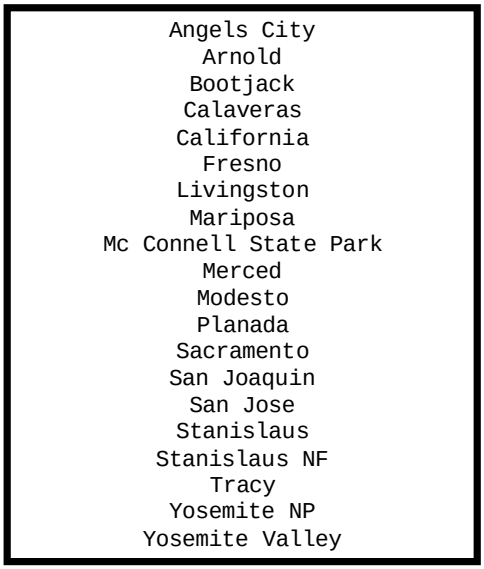

Fig. 6. Candidate term set with all possible location names for the Yosemite node.

picked 1.84 place names on average; NameSet used an average of 2.3 place names for each node). As an aid, we showed the subjects maps displaying the coordinates of photos in each node, similar to the map in Figure 3, but without the outline of the containing features that appears in the figure. We also offered to show them the actual photos contained in each node — this latter aid proved unnecessary as subjects usually had a very good idea what those pictures were.

For 79% of the nodes, NameSet and the subjects picked at least one place name in common. In other words, for 79% of the coordinate sets there is a good chance that the name chosen by the system will be not only recognizable, but also intuitive to the user as a representation for the given set. Furthermore, our next test shows that even for most of the other coordinate sets, where the user and the system did not pick a name in common, the subjects still found the given name useful.

In the second test, we asked the subjects to comment on the usefulness of each node's name. We also asked them to comment on how accurate the system name is in describing the contents of the coordinate set represented by the node. We found that the majority of the automatically-produced names were useful (as defined above). Out of the total of 42 nodes, subjects were contented with all but three node names. In one of the unsuccessful cases, our algorithm grouped together all photos from three different US East Coast cities; but the name only represented one of them. In another case, the user felt the node's name was not encompassing enough for the area it represented. Finally, one node name was simply unknown to the user. The node in question represented a relatively small coordinate set (i.e., the user did not take many photos in the area, and thus was relatively unfamiliar with it). There were a few more

instances where the names were not accurate, yet still useful. For example, a set of coordinates from an area the user thought of as "Mount Whitney" were named "Sequoia Nat'l Park", a fact that the user found somewhat bewildering yet acceptable (the park is adjacent to the national forest that contains Mt. Whitney).

Other comments from users were: a) Three of the node names included one park or city name that was unknown to the subject; b) The name of one of the nodes was not representative enough of a small subset of its photos. In general, all subjects expressed strong satisfaction with the usefulness of the names.

Finally, we noted that in some cases the users entered names that are currently hard to derive from traditional location-based datasets. For example, names like "The San Francisco Bay Area" or "Northern England" are well understood by humans, yet hard to represent or define in a computer-based system. Systems like (Arampatzis et al., 2004), or our LOCALE system (Naaman et al., 2003) may be able to help with the generation of such names in the future.

5.1 Incidental Experiment

In a separate experiment (Naaman, Harada, Wang, & Paepcke, 2004), we evaluated the PhotoCompas photo browsing system using photo collections that were manually associated with location coordinates: for the experiment, 13 participating subjects drag-and-dropped their photos onto a digital map to indicate to the system where each photo was taken. In effect, the users were retrospectively geo-referencing their collection. However, the "referencing" was not always as accurate as originally geo-referenced photos: subjects did not always remember exactly were each photo was taken, dragging them instead to an approximate location. In addition, the subjects often dragged a group of photos from the same area together, onto one location, losing some detail in the process.

As a result, it did not seem appropriate to evaluate NameSet directly on these test collections. We therefore did not go through the procedure described above of comparing user-selected terms with our system's suggestions.

Instead, our evaluation of NameSet performance for these collections is derived from the collection owners' interaction with the PhotoCompas system, since the system was utilizing the names automatically generated by NameSet. In other words, we measured and observed how users navigated a collection organized by PhotoCompas, with names assigned to nodes by NameSet. Because NameSet is critical for navigation, the fact that users performed navigation tasks well suggests that NameSet was effective.

Indeed, the names generated by NameSet were found by the collection owners to be relevant and useful for browsing. As we report in (Naaman, Harada, et al., 2004), the subjects' performance using PhotoCompas to browse their photos was comparable to that of a map-based photo browsing system. The browsing hierarchy created by PhotoCompas, annotated with names generated by NameSet, allowed an equivalent level of interaction as a powerful map-based system in terms of browse/search time, number of mouse clicks, and user satisfaction.

6 Conclusions

We have shown that our system, NameSet, can automatically generate meaningful and useful textual names for sets of coordinates in the context of a geo-referenced personal photo collection. NameSet could also "translate" other types of coordinates into textual names, like, for example, a list of geo-referenced observations that appear on a web site. In the future we plan to expand this work to other domains, using different semantics.

In our work, we investigated how useful NameSet is for browsing and navigation tasks: the created names are used in the context of a user interface. We also suggested that the names can be used in the settings of an information retrieval system, where a page can be indexed by the textual names instead of the meaningless (in this context) latitude/longitude values.

Whatever the setting in which our system is used, one important underlying assumption is that the set of coordinates to annotate is biased. In other words, the set is meaningful to a person, or a group of people, and is not just a collection of random points. Also, the set is assumed not to be created by some arbitrary grouping of coordinates based on proximity, but is semantically coherent in some way or another.

In particular, systems like WWMX (Toyama et al., 2003) and Mappr.com (for example) may want to consider how to apply our system to replace their map-based representation of geo-referenced material. As their collections are global and public, a meaningful grouping of photos based on geographic areas is not easy to generate. For example, if the system coverage of the San Francisco Bay Area is heavy, and photos are spread throughout the region, where does the system draw the line between the San Francisco area photos and the Palo Alto photos? A system like the ones mentioned may be better off using a simple location hierarchy to textually browse geo-referenced items (e.g., the containing country → state → city/park for the US), making the naming task trivial for most coordinates. However, as the city/park coverage is never exhaustive, it is possible that in this scheme some coordinates could only

be identified by the containing state — a serious hurdle for finding these coordinates by browsing the menu or by search for a place name. In such a case, some of our techniques can be utilized, like finding the best reference point for the "orphaned" coordinates.

7 Acknowledgments

We thank Chris Jones and Ross Purves who organized the Workshop on Geographic Information Retrieval at SIGIR 2004, where this work was originally presented. We also thank Meredith Williams at Stanford for providing crucial help and guidance in the world of Geographic Information Systems. Finally, we thank the anonymous reviewers for their instructive and helpful comments.

References

Amitay, E., Har'El, N., Sivan, R., & Soffer, A. (2004). Web-a-where: geotagging web content. In *Sigir '04: Proceedings of the 27th annual international conference on research and development in information retrieval* (pp. 273–280). ACM Press.

Arampatzis, A., Kreveld, M. van, Reinbacher, I., Clough, P., Joho, H., Sanderson, M., et al. (2004). Web-based delineation of imprecise regions. In *Proceedings of the workshop on geographic information retrieval.*

Cavens, D., Sheppard, S., & Meitner, M. (2001). Image database extension to arcview: How to find the photograph you want. In *Proceedings of esri users conference.*

Ding, J., Gravano, L., & Shivakumar, N. (2000). Computing geographical scopes of web resources. In *Proceedings of the twenty-sixth international conference on very large databases* (pp. 545–556). Morgan Kaufmann Publishers Inc.

Duygulu, P., Barnard, K., Freitas, J. F. G. de, & Forsyth, D. A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the 7th european conference on computer vision (eccv '02)* (pp. 97–112). Springer-Verlag.

Hill, L. L., Frew, J., & Zheng, Q. (1999, January). Geographic names - the implementation of a gazetteer in a georeferenced digital library. *CNRI D-Lib Magazine.*

Jones, C. B., Purves, R., Ruas, A., Sanderson, M., Sester, M., van Kreveld, M., et al. (2002). Spatial information retrieval and geographical ontologies an overview of the spirit project. In *Proceedings of the 25th*

annual international acm sigir conference on research and development in information retrieval (pp. 387–388). ACM Press.

Kuchinsky, A., Pering, C., Creech, M. L., Freeze, D., Serra, B., & Gwizdka, J. (1999). Fotofile: a consumer multimedia organization and retrieval system. In *Proceedings of the conference on human factors in computing systems chi'99* (pp. 496–503).

Larson, R. R., & Frontiera, P. (2004). Spatial ranking methods for geographic information retrieval (GIR) in digital libraries. In *Proceedings of the 8th european conference on research and advanced technology for digital libraries (ECDL 2005).*

Leclerc, Y., Reddy, M., Iverson, L., & Eriksen, M. (2001). The geoweb - a new paradigm for finding data on the web. In *International cartographic conference (ICC2001).*

Lieberman, H., & Liu, H. (2002). Adaptive linking between text and photos using common sense reasoning. In *Adaptive hypermedia and adaptive web-based systems. second international conference, ah 2002. proceedings, 29-31 may 2002, malaga, spain* (pp. 2 – 11). Springer-Verlag, 2002.

Naaman, M., Harada, S., Wang, Q., & Paepcke, A. (2004, April). *Adventures in space and time: Browsing personal collections of geo-referenced digital photographs* (Tech. Rep.). Stanford University. (Submitted for Publication)

Naaman, M., Paepcke, A., & Garcia-Molina, H. (2003). From where to what: Metadata sharing for digital photographs with geographic coordinates. In *10th international conference on cooperative information systems (coopis).*

Naaman, M., Song, Y. J., Paepcke, A., & Garcia-Molina, H. (2004). Automatic organization for digital photographs with geographic coordinates. In *Proceedings of the fourth acm/ieee-cs joint conference on digital libraries.*

Robbins, D. C., Cutrell, E., Sarin, R., & Horvitz, E. (2004). Zonezoom: map navigation for smartphones with recursive view segmentation. In *Avi '04: Proceedings of the working conference on advanced visual interfaces* (pp. 231–234). New York, NY, USA: ACM Press.

Sarkar, M., Snibbe, S. S., Tversky, O. J., & Reiss, S. P. (1993). Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *Uist '93: Proceedings of the 6th annual acm symposium on user interface software and technology* (pp. 81–91). New York, NY, USA: ACM Press.

Sarvas, R., Herrarte, E., Wilhelm, A., & Davis, M. (2004). Metadata creation system for mobile images. In *Proceedings of the 2nd international conference on mobile systems, applications, and services* (pp. 36–48). ACM Press.

Silva, M. J., Martins, B., Chaves, M., & Cardoso, N. (2004). Adding geographic scope to web resources. In *Proceedings of the workshop on geographic information retrieval.*

Smith, T. R. (1996, MAY). A digital library for geographically referenced materials. *Computer*, *29*(5), 54 – 60.

Toyama, K., Logan, R., & Roseway, A. (2003). Geographic location tags on digital images. In *Proceedings of the eleventh acm international conference on multimedia* (pp. 156–166). ACM Press.

Veltkamp, R. C., & Tanase, M. (2002, October). *Content-based image retrieval systems: A survey* (Tech. Rep. No. TR UU-CS-2000-34 (revised version)). Department of Computing Science, Utrecht University.

Zhu, B., Ramsey, M., Chen, H., Rosie V, H., Ng, T. D., & Schatz, B. (1999). Create a large-scale digital library for geo-referenced information. In *Dl '99: Proceedings of the fourth acm conference on digital libraries* (pp. 260–261). New York, NY, USA: ACM Press.