PhotoWhere: Annotation of Digital Photographs based on GPS and Web Data

by

Dan Relihan
drelihan@mit.edu
Interactive Experience Group
MIT Media Laboratory

Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
6.199 Spring 2004 - Advanced Undergraduate Project
Advisor: Pattie Maes
Abstract

This paper describes a process to annotate digital photographs based on GPS and Web data. A GPS receiver “tags” a photograph with its corresponding GPS coordinates. These GPS coordinates are used to locate web pages that correspond to this location. These web sites are programmatically examined to determine likely annotations. Some important factors include the number of times certain terms appear in the web pages weighted by the manner in which they appear (e.g. bold text) and Cartesian distance from the photograph’s GPS coordinates. Using the techniques discussed in this paper, PhotoWhere is able to produce very accurate annotations, with a minimal number of false positives.
Table of Contents

1. Introduction ......................................................... 4

2. Modules ................................................................. 4
   2.1 GPS to Web Database ........................................... 5
       2.1.1 MetaCarta Database ..................................... 6

3. Annotator ............................................................. 6
   3.1 Annotation Server .............................................. 6
   3.2 Indexing Web Pages ............................................ 6
   3.3 Filters ............................................................. 8
   3.4 Scorer ............................................................ 8
       3.4.1 TermExtractor ............................................. 8
       3.4.2 TermScorer ................................................ 9
           3.4.2.1 Weighting Terms Based on Field ................ 9
           3.4.2.2 Overcoming the Page-Hog Problem ............ 10
           3.4.2.1 Scaling by Distance ............................. 10

4. Demonstration ....................................................... 10
   4.1 Results .......................................................... 12

5. Future Work ........................................................ 13
   5.1 Consumer Applications ...................................... 14
       5.1.1 Automatically Organized Photograph Albums ...... 14
       5.1.2 “I was there!” Maps ................................... 15
   5.2 Other Additions ................................................ 15

6. Acknowledgments .................................................. 15

7. References .......................................................... 15
1. Introduction

One problem with all digital cameras is that they tend to name pictures in a cryptic fashion (i.e. DSC0001.jpg). To help users keep more accurate records of their digital photographs, I have developed a system, entitled PhotoWhere, which annotates digital photographs automatically using data from a GPS receiver and the World Wide Web. Using a GPS receiver, which tags each picture with specific GPS coordinates, the system queries a database that indexes web pages by GPS coordinates. Then, given a set of web pages associated with these coordinates, web mining techniques can be employed to best guess what these websites are about, or even better, what they have in common. From this information, an accurate annotation can be deduced. Please refer to Figure 1 below for an illustration of the project. Although all aspects of this system are discussed, this paper focuses on the annotation stage of this system.

Figure 1 - Diagram of the proposed project, PhotoWhere

2. Modules

This project consists of several different modules that must work together to produce a desirable final outcome. Each module is detailed below:
**GPS Receiver** – The GPS receiver used in this project is a Bluetooth-enabled device made by Socket Communications [1]. This device was chosen for its ability to communicate on a Bluetooth network.

**Camera-Enabled Phone** – Since there was some customized programming to be done in this module, a camera-enabled digital phone, the Nokia 3650, was chosen. This phone not only allows Visual C++ applications to be written for it, but also is capable of sending and receiving information over a Bluetooth and WWW network.

**Annotator** – This module is responsible for taking in a digital photograph and its corresponding GPS coordinates and outputting a list of possible annotations for that particular photograph. The Annotator is a piece of software that resides on a separate server machine. The software is primarily written in JAVA. This area of the project is the most complex and involved piece of the system and it is the focus of this paper.

The Camera-enabled phone takes a digital photograph and then immediately asks the GPS receiver for coordinates. This photograph and coordinate are then sent to the Annotator. The Annotator processes the information and returns a list of possible annotations for the photograph.

### 2.1 GPS to Web Database

PhotoWhere is designed to work with any database that indexes web pages by GPS coordinates. The GPS to Web Database that I used is very simple and consists of only three columns. Figure 2 illustrates this basic design with example data.

<table>
<thead>
<tr>
<th>Latitude</th>
<th>Longitude</th>
<th>Web Page Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>42.42.34</td>
<td>71.5.38</td>
<td>/webpages/location1</td>
</tr>
<tr>
<td>43.8.2</td>
<td>70.55.37</td>
<td>/webpages/location2</td>
</tr>
<tr>
<td>42.21.30</td>
<td>71.3.37</td>
<td>/webpages/location3</td>
</tr>
<tr>
<td>42.20.48</td>
<td>71.5.56</td>
<td>/webpages/location4</td>
</tr>
<tr>
<td>42.21.57</td>
<td>71.3.40</td>
<td>/webpages/location5</td>
</tr>
</tbody>
</table>

**Figure 2 - Example Data from the GPS to Web Database used in PhotoWhere**

When the system acquires a GPS coordinate, it will query this database for the closest N number of matches within a reasonable radius, typically set to 100 meters. The database instructs the program where the physical web page(s) that are related to the GPS Coordinate are stored. This Web Page Location can be a single file, a directory, or a URL address.
2.1.1 MetaCarta

An example of an existing database that indexes the web via GPS coordinates is MetaCarta. MetaCarta, currently a start-up company in Cambridge, MA, focuses on bridging the gap between Geographic Information Systems (GIS) and text search. MetaCarta’s geographic text search tool, which displays web pages relating to keywords around a specified location, is not only a prime example of a GPS to Web database, but also the source of information for my test database. More information about MetaCarta can be found on the Web at http://www.metacarta.com. A few other geographical search engines do exist, such as Google Local (http://local.google.com/). Typically search engines such as Google Local are less complete and only feature web pages of businesses.

3. Annotator

As mentioned above, the Annotator is a software program that resides on its own separate server machine. Its primary function is to listen for requests to annotate pictures and respond to these requests with an appropriate list of annotations.

3.1 Annotation Server

The server is a separate machine to which mobile devices can connect in order to request photographic annotations. The server is implemented in JAVA and creates a SocketServer, which will accept connections made to a specified port. PhotoWhere uses port number 9595. Since this version of PhotoWhere classifies as a “proof of concept” model, no authentication or security is implemented.

3.2 Indexing Web Pages

Significantly more information exists within an HTML document than within a text document. By using HTML markup tags as clues, PhotoWhere is able to determine with better accuracy what a web page is actually about. The system first parses web documents into groups of terms organized by HTML markup tags. This is the indexing stage of the process. For this stage, PhotoWhere depends on open source code from the Jakarta Lucene[2] project to index the terms, along with an open source HTML parser to parse each web page.

The Indexer works one page at a time. First, it selects a page and parse elements within certain HTML tags into different fields; counting how many times each term appears within each field. For example, in a single document, the term ‘baseball’ could be indexed into three different fields if it appears within three different HTML markup tags. Figure 3.1 depicts a list of the different fields and the corresponding HTML tags that represent that field.
Figure 3.1 - List of different Fields and the corresponding HTML markup tags

All text not within a tag or within a tag that does not appear in the list above is stored in the 'general' field. Once this is complete, the index consists of a list of terms for each field and the number of times each term appears within that field. An example of this list is shown in Figure 3.2 below.

<table>
<thead>
<tr>
<th>Field</th>
<th>Corresponding HTML Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Links</td>
<td><code>&lt;a&gt;...&lt;/a&gt;</code></td>
</tr>
<tr>
<td>Meta</td>
<td><code>&lt;meta keywords=“…”&gt;</code> <code>&lt;meta description=“…”&gt;</code></td>
</tr>
<tr>
<td>imageText</td>
<td><code>&lt;img ALT=“…”&gt;</code></td>
</tr>
<tr>
<td>Decorated</td>
<td><code>&lt;b&gt;...&lt;/b&gt;</code> <code>&lt;i&gt;...&lt;/i&gt;</code> <code>&lt;strong&gt;...&lt;/strong&gt;</code></td>
</tr>
<tr>
<td>Big</td>
<td><code>&lt;font size=7,6, or 5&gt;...&lt;/font&gt;</code> <code>&lt;h1&gt;...&lt;/h1&gt;</code> <code>&lt;h2&gt;...&lt;/h2&gt;</code></td>
</tr>
<tr>
<td>Medium</td>
<td><code>&lt;font size=4&gt;...&lt;/font&gt;</code> <code>&lt;h3&gt;...&lt;/h3&gt;</code></td>
</tr>
<tr>
<td>Small</td>
<td><code>&lt;font size=2 or 1&gt;...&lt;/font&gt;</code> <code>&lt;h2&gt;...&lt;/h1&gt;</code> <code>&lt;h1&gt;...&lt;/h1&gt;</code></td>
</tr>
<tr>
<td>Title</td>
<td><code>&lt;title&gt;...&lt;/title&gt;</code></td>
</tr>
<tr>
<td>imgNames</td>
<td><code>&lt;img src=/path/to/image/…&gt;</code></td>
</tr>
<tr>
<td>General</td>
<td>Anything else</td>
</tr>
</tbody>
</table>

Figure 3.2 - List of terms after indexing stage. Note that identical terms are listed separately if they occur within different fields.

As multiple pages are added to the index, more terms are added to and incremented (if the term already appears) within this list. The system also keeps track of which document each term originally came from, which will be important in weighting terms. According to this
system, the indexing is complete; however, in order for this information to be of any further use, this information must be combined. A single list, which combines terms from different fields, but with the identical text, would be ideal. The system accomplishes this in the Scoring stage, which is discussed in the next section.

### 3.3 Filters

There is likely to be a number of terms occurring with a web page that under extremely rare circumstances would actually relate to the context or successful annotation of a web page. Therefore, filters must be introduced to remove undesirable terms, such as prepositions and modifiers. Such terms are filtered out with the WebStopAnalyzer class, which instructs the Indexer to immediately throw out terms that appear in an array of terms. The list used in the current PhotoWhere system is shown below:

```java
public static final String[] = "a", "all", "also", "and", "an", "are", "as", "at", "be", "but",
"by", "for", "from", "http", "if", "in", "into", "is", "it", "no", "not",
"of", "on", "or", "s", "such",
"i", "that", "the", "their", "then", "there", "these",
"they", "this", "to", "too", "was", "were", "will", "with",
"while", "www", "you", "b", "c", "d", "e", "f", "g", "h", "o", "1", 
"2", "3", "4", "5", "6", "7", "8", "9", "10", "617", "02215",
"02139", "your", "one", "their", "my", "his", "her", "me", "i",
"&quot", "charset", "new";
```

### 3.4 Scorer

The purpose of the Scorer is to ascertain which term or terms best describe the set of web pages that the Index represents. First, the Scorer must condense the Index into a single list, which is handled by a TermExtractor. Next, the terms must be examined and scored appropriately, which is handled by the TermScorer.

#### 3.4.1 TermExtractor

The TermExtractor is responsible for extracting terms from the Index and then combining these terms into one list. First, the TermExtractor reads the index term by term and create a list similar to the one in the Index. Now, the TermExtractor must order the list alphabetically by term, rather than by field. Since identical terms are now next to each other in the list, the system now just has to check the next element in the list and continue combining the terms until the next term is no longer identical to the previous term. These identical terms are combined into ExtractedTerms, which keeps track of how many times a term appears in each field.
3.4.2 TermScorer

The TermScorer is responsible for weighting all the terms so that the system can identify the N most-likely annotations. This section describes the criteria for weighting each term.

3.4.2.1 Weighting Terms Based on Field

Depending on what markup tag a term is associate with relates directly to how heavily it should be weighted. The weighting scale is centered at 1.0 for general text. Terms that come from ALT tags of images are weighted the highest, as the system is interested in annotating images. On the other hand, small text, such as terms appearing in smaller print than the rest of the page, is weighted much less than general text. A listing of all the scalar weights is shown below in Figure 3.3.

These values were mostly determined experimentally. Initial estimates were made and experimental trial and error fine-tuned the scalar values to those listed below.

<table>
<thead>
<tr>
<th>Field</th>
<th>Scalar Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Links</td>
<td>1.5</td>
</tr>
<tr>
<td>Meta</td>
<td>2.5</td>
</tr>
<tr>
<td>imageText</td>
<td>5.0</td>
</tr>
<tr>
<td>Decorated</td>
<td>1.5</td>
</tr>
<tr>
<td>Big</td>
<td>2.0</td>
</tr>
<tr>
<td>Medium</td>
<td>1.5</td>
</tr>
<tr>
<td>Small</td>
<td>.5</td>
</tr>
<tr>
<td>Title</td>
<td>3.0</td>
</tr>
<tr>
<td>imgNames</td>
<td>2.0</td>
</tr>
<tr>
<td>General</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Figure 3.3 - Field weights.

The TermScorer iterates through each ExtractedTerm in the TermExtractor. For each term, the TermScorer scales the number of occurrences in each field by the proper scalar value and sums each scaled number from each field together to ascertain a total weight.

3.4.2.2 Overcoming the Page-Hog Problem

The “Page Hog” problem occurs when one term, which is present a large number of times in a very small portion of the web pages, overshadows all other terms. For example, in early testing of the system, I noticed that when I ran the system on web pages around MIT, it would return with a professor’s last name as the top annotation, which was an undesirable
outcome. One of the web pages that the system indexed was the professor’s personal home page, which listed his name next to each one of his many publications, in bold text. Therefore, his last name was the term that was “hogging” the annotation. To overcome this undesirable effect, the system scales the final weight of the term by the fraction of the documents that it appears in. For example, if the term only appears in one out of ten documents, then its final weight is scaled by 1/10. For terms that appear in all or almost all the documents, the score is barely effected.

3.4.2.3 Scaling by Distance

Another important factor in the scoring of terms in a document set is distance. Generally stated, a web page closer in distance to a photograph is more likely to better represent what the photograph is depicting than a web page further away. Therefore, PhotoWhere linearly reduces the weight of a term as the geographical distance to that web page increases. The system uses the following formula to scale the terms from each document:

$$\alpha = \frac{1}{\alpha} \text{ for } 20 \text{ meters } < \alpha$$  \hspace{1cm} (Equation 1)

where \( \alpha \) represents Cartesian distance from the web page where the term came from to the photograph.

This relationship holds for all distances further than 20 meters away from the photograph. Terms from web pages within 20 meters are determined to be within the line of sight (LOS) distance. Moreover, the system is postulating that if someone was taking the photograph, that person would be able to see the location represented by that web page. For locations within the LOS, the system scales all terms from these web pages by a factor of \( \frac{2}{\alpha_{MINIMUM}} \) (or twice the scaling of the closest document outside the LOS distance).

4. Demonstration

This section describes the final working model demonstration that was completed. In addition to constructing the system shown in Figure 1, I built a stationary version that runs off a desktop system, in order to be able to demonstrate and test the system without having to move around town. In order to provide the best test cases and demonstration, the initial PhotoWhere system is modeled by a GUI client, implemented with Swing [3], which emulates a generic mobile device and a central server. A screen shot from the client demo is shown in Figure 4.1.
This demo has a set of GPS Coordinates programmed into it, which a user can select, see a picture of that location, and then request an annotation of the photograph. When the user enters the request, a message is sent to a central server with the corresponding GPS coordinates. The demo queries the GPS to Web database for locations near those GPS coordinates. These web pages are indexed and the terms scored in the manner described in previous sections. The top ten most heavily weighted terms are returned to the GUI client and displayed to the user, along with their appropriate weights.

The screen shot of the client demo depicted above in Figure 4.1 displays a picture of the FleetCenter in Boston, MA. The top annotations of this photograph include “boston”, “celtics”, “bruins”, and “fleetcenter”, which can be considered quality annotations based on the fact that the Celtics and the Bruins are the professional sports teams that play in the FleetCenter. One may also notice some other annotations associated with this photograph, including “var” and “document.write”, which do not seem to have any relationship with the location. These undesirables annotations most likely originate from Javascript buried within one the corresponding HTML pages. This version of PhotoWhere does not filter out Javascript or any other embedded code within an HTML document. Thus, an addition that could so would improve the system.

A screen shot of what the system looks like on the mobile device used in this project, the Nokia 3650, is shown in Figure 4.2. The annotations could be stored on the phone together with the corresponding picture, downloaded and stored on a central computer, or both (depending on the specific application). This initial version of PhotoWhere does not worry about storing past results, only displaying initial results to the user.
4.1 Results

The demonstration consisted of annotations of seven different locations around Boston and one location on the University of New Hampshire’s campus. The top ten annotations for each location are shown in the chart below.

| Location                  | MIT  
<table>
<thead>
<tr>
<th>77 Mass Ave. Cambridge, MA</th>
<th>University of New Hampshire Garrison Ave. Durham, NH</th>
<th>Fenway Park Yawkey Way Boston, MA</th>
<th>Fleet Center Boston, MA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotations (listed in order of weights)</td>
<td>mit massachusetts ma cambridge room institute 77 technology office ave</td>
<td>unh news 2001 bureau nh 2000 03824 durham garrison credit</td>
<td>League site red sox field baseball boston york team Park</td>
</tr>
<tr>
<td>Location</td>
<td>Annotations (listed in order of weights)</td>
<td>Annotations (listed in order of weights)</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------</td>
<td>------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>MIT 20 Ames St. Media Lab</td>
<td>media mit cambridge 20 ames lab next laboratory user learning</td>
<td>cambridge cambridgeside 100 lechmere england home sports museum cambridge.zami.com Ma</td>
<td></td>
</tr>
<tr>
<td>Cambridge, MA</td>
<td>Boston library services university information center public students ma Staff</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cambridge Side Galleria</td>
<td>Citgo sign Kenmore Sq. Boston, MA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cambridge, MA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.3 – Result from seven trial experiments

As you can see, some of these annotations, such as that of the Fleet Center, are very good. On the other hand, some annotations, such as that of the Citgo sign, are unclear. This Citgo sign annotations list “boston” as the top choice, and that would be a correct annotation, albeit it is very general. The other terms in that annotation, such as “library”, “university”, and “students” may seem out of place, until one remembers that Boston University is located around the area. So, in this case, the system provides a sensible annotation for the location, but not necessarily the photograph. Measures can be taken to improve a photograph’s annotation and are described in the next section.

5. Future Work

After working on this project and completing the initial demonstration, I feel that I have only scraped the tip of the iceberg. There is a huge portion of this field that is left undiscovered and could make for very interesting and useful research. With the ever-growing world wide web and the upcoming improvements in GPS technology, many more wireless devices will be GPS-enabled and for a cheaper price. There is no doubt that the popularity and omnipresence of the World Wide Web has increased the amount of information available by magnitudes upon magnitudes. Now, the rise of GPS-enabled devices provides even more specific information when cross-referenced with the Web. Moreover, if more and more users share information on the web that is tagged with GPS data, the environment in which we lived could even become a virtual web. This section of my paper is written to stimulate an interest in this field and encourage further research by suggesting a few consumer applications that could stem from this technology. Please take note that my advanced project did not accomplish these following tasks.
5.1 Consumer Applications

As stated above, this section will discuss three possible consumer applications that the application of this technology could achieve.

5.1.1 Automatically Organized Photograph Albums

The system’s top annotation will not always directly relate to the particular picture taken. However, the top ten most heavily weighted terms, as a set, will, with high probability, represent the location accurately. Therefore, if a picture is stored in an electronic photo album that keeps records of the terms associated with specific pictures, then a text search can be easily constructed. For example, imagine that a high school student is trying to decide which college he or she would like to attend and visits five different schools. At each school, the student takes many digital pictures, which can be uploaded and stored in this special digital photograph album. If the student, still undecided, wants to take another look, the student can enter text, such as “dorm at MIT”, in which case pictures with the annotations of “dorm” and “mit” would be returned. Moreover, the user could simply search for “dorm”, and pictures of dorms from all schools would be returned. This is a simple example and it can be argued that the student can just look through the fifty pictures he or she took and examine all the different dorms. However, one can imagine a similar type of situation involving thousands of pictures all with cryptic (DSC0001.jpg) titles.

5.1.2 “I was there!” Maps

I have a map of the world that I hang in my room that has a thumbtack stuck in each place that I have been to in my life. I enjoy traveling and enjoy the memories of a trip that come into my head after glancing at a particular thumbtack in that map. Imagine if I had taken a GPS-enabled camera with me to each place that I have traveled to. When I download pictures from a particular trip, a system could place a digital “thumbtack” and display a map depicting all the places that I have traveled. In addition, I would be able to select a location, view pictures from the trip, and even browse web pages associated with those pictures. The application described in the above section could be integrated into this system to provide text search capability as well. This data could be directly linked to my personal home page, where I could share pictures and experiences with many others via the Web. Some of this functionality is provided in Erik Blankinship’s Imagemaps at the MIT Media Laboratory [4].

The above two applications were merely examples of how this technology could be applied. I am sure that there are many more areas that could use a system similar to PhotoWhere that have not been mentioned in this paper. I hope that this paper outlined the basic strategy for annotating photographs and web pages based on GPS data and that the paper will inspire interest in further development and research in this area. The Web is a very powerful source of information, and if it is cross-referenced with another powerful resource, such as GPS data, more accurate and detailed information would become available and accessible to everyone.
5.2 Other Additions

In addition to the above consumer applications, a compass attached on the mobile device to determine which way the photographer was facing would could also greatly enhance annotations. If a compass was added, then any web page associated with a location behind the photographer would be severely scaled down. A user could also annotate a specific location (e.g. “Grammie’s House”) and all future photographs taken at that location would include that annotation. Also, if the camera was able to report the distance to the object being photographed, then the system could decide which web pages are closer to that object and weight the different pages accordingly.

6. Acknowledgements

I would like to thank the MIT Media Laboratory for allowing me to conduct this research and providing me with the tools to do so. Specifically, I would like to thank my advisors Pattie Maes and Brad Lassey for their support and advice throughout this project. I would also like to thank MetaCarta (http://www.metacarta.com) for access to their geographical search text tool.

7. References


