Published in: 3rd International Workshop on Web Dynamics, 2004, New York.

Geographic Information Retrieval

Alexander Markowetz [*] Thomas Brinkhoff [†] Bernhard Seeger [‡]

Abstract

Due to the abundance of information, querying the World Wide Web has become increasingly difficult. Geographic properties of Internet resources can be directly used for making Internet search more personalized and for increasing the quality of results. However, little work has been done in this area so far and the general direction of research and development has been uncertain. In this paper, essential questions in this field are addressed: First, we outline a three-stage architecture for an efficient and effective mapping of Internet resources to geographic locations. Geospatial search engines are one important application of this mapping. Such search engines fundamentally differ from their traditional counterparts, particularly in respect to selecting and ranking search results. Finally, we propose geospatial analyses using localized web crawls. Such analyses allow the support of new types of queries as well as the reduction of cost compared to conventional data-collection techniques. The paper concludes with an overview on challenging research questions.

1 Introduction

The state of the Internet is characterized by an abundance of information. Two-term queries posted by search-engine users lead to extreme iceberg queries. There are two major strategies to circumvent the burial of interesting pages under a large quantity of less interesting results. The first approach is to let the user browse dynamically through search results. He trims parameters and therefore quickly drills into the iceberg. The second strategy is personalized search. On a very abstract level, it could be described as the addition of query attributes inferred from personal information of the author and/or of his peers. This allows reducing and reordering the result set in order to match the user's interests. From all personal properties, the user's geographic position is most relevant for a search engine. Using this position allows building *geospatial search engines*, a new and important field of research.

Geospatial search engines are based on two underlying assumptions. First, almost every web page has a local context: *Where* was this information created? *Which locations* does this information apply to? *Where* does the targeted audience reside? We will later show how to infer this context. Second, geographically close information is the most interesting. This idea is best captured by the proverb: *"All business is local"*. Of course, this is not limited to business. From all pizza restaurants in the world, the closest is the most interesting. From all hotels, those from my next holiday destination are the most interesting. From all yoga courses, those within a driving distance from my house are the most interesting. Given the importance of the "local" factor, geospatial search engines could very well turn out to be the killer application for location-based services.

[*] Fachbereich Mathematik und Informatik, Philipps Universität Marburg, markow@Mathematik.Uni-Marburg.de

[†] Institute for Applied Photogrammetry and Geoinformatics (IAPG), FH Oldenburg/Ostfriesland/Wilhelmshaven, Thomas.Brinkhoff@fh-oldenburg.de

[‡] Fachbereich Mathematik und Informatik, Philipps Universität Marburg, seeger@Mathematik.Uni-Marburg.de

Despite the obvious advantages of geospatial search engines, the interest of commercial search companies is only slowly increasing. There have been some first early prototypes in 2003. From the scientific community, the topic has received little attention. There is some work on localizing servers, but almost none on content.

The first issue in this paper will be how to infer geographic locations for Internet resources. If these are not already stored explicitly as metadata, there are still numerous ways of deducing them from various aspects of page content and link structure. We present a three-stage architecture that allows combining different techniques and producing a satisfying mapping from web pages to geographic locations. Once this is achieved, groundwork has been laid for two major applications: a location-aware search engine targeting individual users and geospatial analysis for corporate users.

A geospatial search engine will allow users to specify a location in addition to the keywords they are searching for. The search engine will then return results that are not only valid to these keywords, but are also located near the specified location. It will also need to incorporate the first major strategy and allow for dynamic trimming between distance and topic importance. The search engine therefore does not only require new interfaces but also efficient implementations that allow for this flexibility.

The second application will be the *geospatial analysis* of Internet resources. The World Wide Web proves predestined for any sort of social analysis, because it mirrors society to an exceptional degree. This observation does not only hold for the explicit information stored as page content. It also applies to its less obvious properties, such as relationships between pages. By taking this implicit information into account, we can derive insight regarding social systems, such as business or science.

Locality is a key factor in all aspects of human interaction. Enhancing the Web by geospatial properties will take web analysis to a new level that will allow a new class of queries.

The rest of the paper is organized as follows. In Section 2, we demonstrate how to map web pages to geographic locations. In the next two sections, we outline geospatial search engines and geospatial analysis. Related work that has not already been discussed in the applying sections will be treated in Section 5. Finally, we provide conclusions and an outlook on a broad field of future work.

2 Computing Geospatial Properties of Internet Resources

In this section, we introduce two essential *geospatial properties* of web pages: The *location* of a page will later be used to compute its distance to the position a user is searching for. The *locality* of a page allows distinguishing between pages that are globally important and those that are only of local interest.

2.1 Geographic Locations of Internet Resources

Computing the geographic locations of a web page is not an easy task and like most things on the Internet "best effort". There is a multitude of mapping techniques, however, none of which work very well by themselves. Therefore, we propose a three-stage architecture. In the first stage, a broad range of techniques is used, each of them assigning a set of locations to a page. In a second stage, we fuse the different sets of locations. In the final stage, we consider link structures and user behavior to validate and refine the mapping. These techniques produce high quality results, but require the initial mapping from the first two stages. Note that multiple locations may be associated to a single page, e.g., a page of a retailer might refer to the locations of multiple outlets.

2.1.1 Initial Mappings

A whole range of techniques can be applied to assign initial locations to web resources. For a broad overview, we refer to [MCu01].

One of the most basic, yet powerful approaches simply processes the `admin-c` section of the `whois` *entry* of a URL. In most cases, this section directly points to the company or individual who registered that domain. For most companies, this corresponds to exactly the location, for which that information is relevant to. Our evaluations have demonstrated the very high relevance of the `admin-c` section. The evaluation of other parts of the `whois` entry often fails because they are concerned with the location of web servers. However, most small companies or individuals do not host their own server, but may co-host at a server farm, hundreds of miles away from their home.

Many authors propose adding *geospatial meta information* to web pages, denoting that the content of this page is relevant to a certain location. The location may be described by using the proposals of the Dublin Core Metadata Initiative [DCMI00] or according to the ISO/TC 211 standard 19115. The use of *geospatial tags*, however, is quite problematic. As long as no search engine relies on geospatial tags, there is no need for administrators to implement them and vice versa. Even worse, webmasters may not be trusted. They may maliciously include tags for regions, for which their site is of no relevance. For this reason, no commercial search engine takes `<meta>` HTML tags into account. So geospatial tags can serve as a mere hint of the location of a web resource.

Another range of techniques requires parsing URLs as well as entire web pages for *extracting names of geographic features* like cities and landmarks, which can be mapped to locations. There are several problematic issues regarding the use of parsing techniques that prohibit their exclusive use. First of all, parsing is quite expensive and might not be applicable to large amounts of web pages. Second, homonyms and synonyms cause tremendous problems. For example, wide-spread names such as "Springfield" are impossible to map. Analogically, *geospatial codes* like zip or dialing codes can be extracted. However, the same problems hold for such an approach. Therefore, several of such hints need to be combined for an acceptable guess.

2.1.2 Fusion and Integration of Multiple Mappings

The previous discussion demonstrated that multiple sets of locations might be assigned to a single web page. In the following, we outline how to integrate the results in such a way that a unique set of locations is computed for each page. First, we detect and remove outliers. The general assumption is that outliers are produced by faulty data such as misleading geographic tags.

Since spatial data is generally imprecise due to different underlying resolutions, we require a second step to condense clusters of locations that refer to the same place. We try to identify these clusters and represent each by a single location. In case of point locations only, there are two common representative locations for a cluster: the centroid and medoid. For areas, the common intersection might be taken into account.

2.1.3 Further Refinement

The final stage of our architecture validates and refines the previous mapping. The following techniques increase the quality of the results, but would not work without the initial mapping from the first two stages.

As a simple, yet again very powerful approach, we propose using the *web's link structure*. If a cluster of pages from NY points to a web site, which is so far assumed to be in LA without any links from that area, we might conclude that the site is more relevant to NY than LA. Finding such clusters and detecting outliers is a task, for which data mining techniques [KH00] need to be adapted.

Additionally, *locations of users accessing a web resource* can be used for verifying its location. In the near future, the widespread use of mobile web clients can be expected, which know their position from GPS, Galileo or their mobile phone. Then, it is reasonable to assume a strong relation between the location of web resources and their users. This relation can be evaluated by analyzing corresponding clickstreams.

2.2 Locality of Internet Resources

The introduction of *locality* enables us to distinguish between sites that are globally important on the subject and those that are not so significant on a global level but are of highest local importance. On the one hand, there are web sites that have high global importance, but are locally rather irrelevant. Examples can be found among web sites of magazines, mail-order stores, etc. On the other hand, there are web sites that have high local importance, but are outperformed on the overall subject by a multitude of other web sites. Examples can be found among sites of local stores and institutions.

The idea of locality is equally important in the context of geospatial search engines as well as geospatial analysis. Let us consider the web sites of a pizza restaurant and an international magazine for Pizza lovers, both based in Marburg. The magazine for pizza lovers will have thousands of links from all over. It is globally important. The local pizza parlor might be referenced by only twenty or thirty links, but all from within Marburg. It is locally important. When searching for *pizza* in *Marburg*, it is really the locally important site that is desired, not the global one. Equally, in geospatial analysis, one might want to distinguish between sites with a global audience and those that target a more local one.

Computing and storing locality does not come for free. Therefore, the granularity in which it is computed and the way it is stored highly depend on the application. One needs to take into account, how flexible the modelling of locality has to be, how long its computation is allowed to take, how much storage is required and how long retrieval will take. Depending on these factors, one will have to select the appropriate level of detail and flexibility.

The highest level can be achieved by storing the precise distribution of links as a function of their distance. For efficient storing, it might be smoothed and compressed. For many applications however, this will prove to be an overkill. The average lengths of links with other sites can serve as a measure of locality, which is much easier to compute, store and retrieve later. Also, one could simply count the links coming from within a distance of ϵ. Taking the total number of links into account, one could compute the relative locality. Together with the variance of link lengths, this could serve as an appropriate measure for locality. In particular, in the context of geospatial analysis, one might want to distinguish between *inbound locality* and *outbound locality*, taking only in- or outgoing links into account.

3 Search Criteria of Geospatial Search Engines

Geospatial search engines will be the first comemrcialy available geospatial web applications. First prototypes are already available [NL03, Ov03, Go03]. In addition to the search terms, Geospatial search engines require a specification of the location a user is interested in. The simplest solution for specifying such a *search area* is a text field for defining a place or an address. In the case of mobile clients, the current position can automatically be passed to the search engine. The search engine will return those results first that are not only relevant to the search terms, but also within close distance to the specified location.

Localized queries are poorly supported by traditional search engines for various reasons.
There is no support for continuous space. When searching for *"Marburg AND Cycling"*, the user will typically receive pages for all interesting cycling activities in Marburg, but some of the real interesting results just outside the city boundaries will be missing.
The available granularity is often to coarse. Searching for a *pizza*, a web site from the same city might not be "close enough", if the city is L.A.
The name of the search area is a poor indication. A web resource might not contain the name of the location exactly as the user spelled it. Synonyms might be used. In consequence, many interesting pages will fail to show up in the results.

The order, in which search results are presented by geospatial search engines, differs fundamentally from the ranking of their traditional counterpart. The ordering does not only depend on one criterion (the *relevance of subject*) but also on a second (the *geographic proximity*). The balance between these two criteria is crucial for delivering useful results. Depending on the search terms, one criterion could be of a higher importance than the other. For example, when looking for a restaurant, its proximity is of much higher importance than when looking for a car dealership. Depending on the first batch of results delivered to the user, he or she might even want to re-adjust the balance. In the following, we compare different solutions that allow an adjustment of the balance between the two search criteria.

3.1 A Post-Processing Solution

A simple solution is to use a traditional search engine that allows specifying the search terms by some keywords k. The search engine offers r results, from which n results are retrieved and reordered according to their proximity to the search area l. From all parameters, n is extremely important for a useful result. If n is chosen too small, only very important pages on the keyword search are retrieved. It may happen that all of them are located far away from l, and therefore proved useless in a geographic context. If on the other hand, n is chosen too large, there may be a multitude of pages that are located very close to l, but are only remotely interesting in the context of k. The interesting web sites will be buried under these useless results. A relevant page may show up so late, that a user gets tired of searching through the results and aborts the search too early.

Modifying n allows changing the balance between the two search criteria. By making n smaller, the overall importance on the subject becomes more important. By making n larger, geographic proximity shifts into the center of focus. Setting n to a fixed number is useless, since for some k, the search engine will return hundreds of results, while for another, it may return millions. Setting n to a fixed percentage of r seems a better approach. Still, the percentage of locally interesting sites very much depends on the topic, so a fixed percentage that will work for searches regarding mountain bikes might not work for searches regarding computers or knitting patterns.

Ideally, we want the user to change the balance between the two criteria dynamically, while he is browsing the results. Using any standard search engine, results are presented in chunks of ten or twenty. If none of the results from the first batch look interesting, the user finds a button at the bottom of the page that will show the *next* batch. This is the point, at which the user is allowed to change his preferences. Say, the user has just browsed through a batch of results. By the time he reached the end, there are four possibilities: `Done`: In the case one of the results proved of sufficient quality, we consider the search done. `More`: If the user thinks he is on the right track, but somehow the results just seen were not what he wanted, he can continue browsing through the results, using the same balance. `More Important` If the results seem only remotely important in the context of k, the user might want to trade geographic proximity for importance. `Closer`: If the results just seen were important on the subject, but too far away, the user could chose to consider results that are not as important on the subject, but closer to l.

Geographic search engines will be judged by how efficient they support this dynamic balancing. How many intermediate results have to be materialized, before the first batch is returned? How many can be re-used, if the user re-adjusts the balance between the two factors?

The simple approach described in the beginning of this subsection does not perform well under any of these questions. It necessarily materializes all results, before returning the first batch to the user. The algorithm does not allow any re-adjustment between importance and proximity. If any such re-adjustment should take place, the entire query has to be recomputed. Therefore, none of the already materialized results can be reused. This method is not suitable for production use.

3.2 Zones

So far, we have not given much thought to the properties of *distance*. Intuitively, we assumed it being smooth and strictly monotonous. In our everyday lives however, our perception of distance is quite different. We do not care if the nearest supermarket is 6.8 or 7.2 km from our home. In fact, we probably do not even know. Instead, we tend to think in terms of: Can I walk there or do I need to take the car? Do I have to cross an international border? Therefore, we end up conceptualizing distances in zones, such as: In walking distance of l. A short or medium drive. Travels within the same political entity as l.

Applying this observation to web sites, we have developed a second technique that is much more flexible than the first, yet as simple. It offers significant and meaningful re-adjustments while browsing results and does not require any re-computation after a re-adjustment. In this approach, sorting and browsing are two entirely independent steps. In the first step, we sort the important pages, or any significant subset, into fixed categories such as presented above. We name the zones z_0 through z_{max}, the first being the innermost, and $\forall_{0 \leq i < max} z_i \subset z_{i+1}$. Within these categories, we order pages entirely regarding their relevance for k. This is the same order that any search engine would have imposed. Navigation would be similar to the previous approach.

The major drawback of this method is that still all results have to be materialized, before sorting them into their zones. Therefore, it might not be suitable for production use, even though it proves so flexible.

3.3 Adaptive Weight Adjustment

The third method makes use of special indices, such as [PTF+03] which allow skyband queries. They allow for a maximum of flexibility but require pre-computation and maintenance of the indices. However, they could prove crucial for a timely execution of a query.

The search engine's indices, which will need to be adapted in order to perform such queries efficiently, are beyond the scope if this paper.

4 Geospatial Analyses Using Web Crawls

In this section, we consider complex spatial queries over sets of web pages. In the previous section, we were mainly concerned in computing a location-dependent order on web pages. From the user's point of view, this order is important, but he/she is really interested in the content of these pages. In this section, we will show how to retrieve information implicitly contained in the structure of the Internet. Relationships like incoming links and geospatial locations are only implicitly available and expensive to compute online for complex queries. For this reason, we create so-called *web crawls* in advance, by traversing the Internet and recording all visited pages and links. Similar to [RGM03], we store this information in an object-relational web warehouse [KH00]. We extend this approach in the sense that we also store the geospatial location of pages in this repository. This is the prerequisite to support spatial analyses on web pages.

Transferring the web crawl into a warehouse-like repository has been described by [RGM03]. Pages and links are stored in an object-relational schema. Typical properties of web pages frequently used in queries such as word count or page rank are stored as attributes. The addition of geospatial properties generated from the methods of Section 3 is rather straightforward. Every page will receive a set of *locations* to which it refers to. Additionally, a link will receive an attribute for the distance between the corresponding pages. This property suffices to measure the corresponding pages' *locality*, which can be computed by aggregation among incoming and outgoing links. The following list illustrates a few examples on problems whose solution we might support:

In which regions is BMW more popular than Audi?

Draw a map of Germany, paint web sites regarding BMW in red and those regarding Audi in green.

This question could be answered by traditional means, such as employing data sets captured by techniques like door-to-door surveys. These are however extremely expensive and take weeks, while our approach might return first hints within a few minutes.

Which BMW dealers outperform their local Audi competitor?

Detect all BMW dealers that are within distance of 3 km to an Audi dealer and that outperform this competitor in the number of local incoming links.

This type of question is almost always impossible to be answered correctly, because the required information as the business volume of a local competitor will never be available. Our techniques might still help to provide some useful hints.

Which German collection of BMW-related links target a global audience?

Find all web pages regarding BMW with more than 250 outgoing links that have more than 1000 incoming links from outside Germany.

This question regarding the internet itself can only be answered by examining its properties.

These three queries reflect our main goals: reducing the cost for conventional data collection substantially while making it faster, approximating impossible queries by assuming a correlation between web structure and business and making precise analyses of geospatial Internet properties. We therefore believe that the spatial analysis of Internet resources will produce cheaper results as well as results sui generis.

4.1 Localized Crawling for Geospatial Data Marts

Web crawls are expensive to create and maintain. Commercial crawls such as maintained by search engines exceed a billion documents and need large server farms for storage and indexing. One of the important problems of search engines is to keep the relevant data up-to-date. For most applications, the interest of users is restricted to a small region, i.e., a tiny fraction of an entire web crawl would have been sufficient. It immediately follows that such a fraction could be updated frequently, at very little cost. Therefore, we propose the storage of partial crawls in data marts [KH00], directly targeted at special location-aware queries.

The essential question is how to gather all interesting sites, without visiting too many irrelevant pages. We propose to solve this challenge by employing a *location-aware crawler*. When trying to find out about the dealership structure in Marburg, one would for example be interested in all BMW sites from within x km of Marburg. Given some *anchor sites* known to lie in the region, the crawler starts to gather local sites. It may only follow links that do not range outside that area by more than y km. Thus, it examines the inner area and the surrounding sprawl. In addition, the crawler should be able to limit its search to a given topic, provided by some keywords.

The main problem of the location-aware crawler is the necessity to compute a preliminary location as it reaches each page. It needs a location to check, if this page is still within y km and its links should be followed. Hence, only the most inexpensive mapping techniques such as parsing `whois` entries can be applied.

Another challenge is found in the detection of anchor pages. These should be able to ”span” the desired set of local pages. In other words, all relevant pages should reachable by following a few links only. Anchor pages could be known directly to knowledge worker initializing the localized crawl. For some scenarios, this assumption could prove reasonable. In others, one would like to generate them automatically by pre-computing a set of anchor pages for all areas that might become interesting in the near future. This would typically be done by something like a web crawl, just on a much coarser granularity. Even though the range of possible techniques is large, they all have to deal with a tradeoff between the expenses for the pre-computation of good anchor pages and the number of sites the target crawl has to visit. If the anchor pages are of inferior quality, more pages have to be visited to ensure that all relevant pages are identified.

5 Related Work

Our work is closely related to many aspects of data warehousing [KH00]. For example, the three-stage architecture that we proposed in Section 2 is similar to the loading process of a data warehouse. The first stage corresponds to gathering data from different databases to be included in a data warehouse. The fusion of mappings shows similarities to data cleansing and integration as known from data warehouses.

The technique of the final stage is related to the work of [DGS00] where the notion of the *geographic scope* is introduced for a web page. The scope is computed by first assigning a location to every domain that is based on using zip codes from some unspecified section of the `whois` entries. Next, the authors propose computing the scope from a fixed set of hierarchically ordered political entities, such as `country, state, city`. For being in the scope of a page, there needs to be a significant amount of uniformly distributed links from the corresponding area to the page. The authors noticed that the results consist of web pages with a national scope and others limited to a smaller geographic scope. This suggests the idea of *local importance* as introduced in our paper. One of the problems of geographic scopes is that they are based on fixed zones. For different applications scenarios, like geography or business, useful geographic entities might look completely different. Moreover, the granularity of the hierarchy is generally rather coarse.

Up to now, the geographic enhancement of search engines is rather limited. For its national sites like *www.google.fr*, Google offers to narrow a search to domains from a specified country or to web pages in a specified language. There have been prototypes of geographic search engines by Northern Light [NL03], Overture [Ov03] and Google [Go03]. The former allowed specifying the distance in which the user would like to search. The same holds for the prototype of Google, which indicates the results in a small map. The search engine by [Dav99] requires geospatial tags. Since it relies on manual registration, its view of the Internet is extremely narrow. The authors of [DGS00] implemented a search engine, based on geographic scopes of web pages [Gra03] as described above. It narrows its focus to articles of 300 online magazines, for which the geographic scopes are pre-computed.

In contrast to all systems described above, the search engine described in Section 3 is unique in the sense that it supports dynamic balancing between page rank and distance. Additionally, it increases the quality of search results by paying special attention to local importance.

To the best of the authors' knowledge, geospatial analyses based on massive data gathered from the Web has not been examined, so far. The work of [RGM03] is closely related to ours, but does not address the problem within a geospatial context.

6 Conclusion and Future Work

In this paper, we sketched the field of geographic information retrieval. We outlined applications, techniques and future possibilities.

First, we demonstrated how to map Internet resources to geographic locations, and how to integrate the different available techniques in a three-stage architecture. In addition to spatial *locations*, we introduced the concept of *locality*, which describes the degree to which a page is connected to its geographic neighborhood.

Next, we showed how this information could be used to build a *geospatial search engine*, which allows users to search for information within proximity of a certain location. We pointed out that for this application two linear orders (page rank and spatial proximity) need to be integrated. It was emphasized that only a dynamically adaptable balance between these two factors allows flexible navigation. This search engine will certainly prove to be a powerful application and could serve as a killer application for location-based services.

Eventually, we showed how a web crawl, augmented with geospatial information, could be integrated into a data warehouse. This allows inferring information, otherwise implicitly stored in the web's structure. We proposed a spatial-aware crawler that allows the restriction to sites from a specific region. These local web crawls are stored separately in *data marts*. In order to improve local crawling, we introduced the concept of anchor pages.

None of the solutions discussed in this paper claim to be final. Instead, we tried to outline the large field of aspects and problems that arise when the WWW is used for exploiting geospatial information. Much of the future work is going to arise from implementations of the systems discussed in Sections 2, 3 and 4. All approaches discussed will have to be evaluated with respect to their applicability in production environments. Most techniques will have to be evaluated manually, since their quality is measured by the degree to which they reflect our everyday experience.

As far as the mapping of pages to locations is concerned, we expect interesting questions in the area of integrating several mappings. Another focal point of research is going to be the interpretation of link structures in order to extract geographical information. For the search engine, scalability will be the key issue. It will have to be investigated, to which degree geospatial properties can be indexed efficiently. In addition, these index structures will be judged by the degree of dynamic balancing between page rank and proximity.

We see a good deal of the future research in the area of efficient *geospatial analysis of web crawls*. The work of [RGM03] is certainly an excellent starting point, but needs to be extended to cover geospatial properties. Since we integrate them into a warehouse, applying more advanced data-mining techniques is another important issue. Also, one will want to integrate other data sets like those gained from door-to-door surveys and government statistics. *Query processing* will prove to be a key issue. Dealing with billions of documents, efficient query execution is crucial, especially since geographic data dramatically increases CPU and storage requirements. Because Internet is based on a best-effort paradigm and data mining is approximate by itself, approximate query processing will be a natural direction of research. In order to trade quality for execution time however, a clear notion of *quality* will have to be established first.

In summary, most aspects of traditional IR will have to be restudied in respect to geographic properties of Internet resources. In return, web search will reach a new level of personalization.

References

[BCG+99] O. Buyukkokten, J. Cho, H. Garcia-Molina, L. Gravano and N Shivakumar, "Exploiting Geographical Location Information of Web Pages," *WebDB (Informal Proceedings)*, 1999.

[Dav99] A. Daviel, "geotags.com," *http://geotags.com*, April 1999; accessed February 2003.

[DCMI00] Dublin Core Metadata Initiative, "Dublin Core Qualifiers," Recommendation, *http://dublincore.org/documents/dcmes-qualifiers/*, July 2000.

[DGS00] J. Ding, L. Gravano and N. Shivakumar, "Computing Geographical Scopes of Web Resources," *26th International Conference on Very Large Databases*, pp. 445–456, September 2000.

[Go03] Google, Inc., "Search by Location," *http://labs.google.com/location*; accessed September 2003.

[Gra03] L. Gravano, "GeoSearch: A Geographically-Aware Search Engine," *www.cs.columbia.edu/~gravano/GeoSearch/*; accessed February 2003.

[KH00] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2000.

[LGMR01] P.A. Longley, M.F. Goodchild, D.J. Maguire, D.W. Rhind, *Geographic Information Systems and Science*, Wiley, 2001.

[MCu01] K. S. McCurley, "Geospatial Mapping and Navigation of the Web," *Tenth International World Wide Web Conference*, May 2001.

[NL03] divine inc., "Northern Light GeoSearch," *http://www.northernlight.com/geosearch.html*; accessed February 2003.

[Ov03] Overture Services, Inc., "Local Search Demo," *http://localdemo.overture.com*; accessed September 2003.

[PTF+03] D. Papadias, Y. Tao, G. Fu and B. Seeger, "An Optimal and Progressive Algorithm for Skyline Queries," *ACM SIGMOD 2003,* June 2003.

[RGM03] S. Raghavan, H. Garcia-Molina, "Complex Queries over Web Repositories," *VLDB 2003,* September 2003.