AUTOMATIC GEO-REFERENCING OF WEB-PAGES

Thomas Lindgaard
IT-snedkeren
Bjerg-Thomsensvej 22, DK-8410 Rønde, Denmark
thomas@it-snedkeren.dk

Kaj Grønbæk
Department of Computer Sciences, University of Aarhus
Åbogade 34, DK-8200 Århus N, Denmark
kgronbak@daimi.au.dk

ABSTRACT

This paper presents an approach to automatic geo-referencing of web pages. The approach is based on automatic recognition of address information in Web-pages. The paper presents the GRef prototype which consists of a geo-indexing crawler and a map-oriented search interface. GRef thus deals with the task of placing web-pages in geographic space based on the document contents – more specifically postal addresses found within the pages. Design issues for the development of geo-based indexing of Web pages are discussed.

KEYWORDS

Geo-spatial hypermedia, GIS, Web crawler, Web-indexing, geo-tags.

1. INTRODUCTION

The vast amount of data available today makes it impossible for anyone to keep up-to-date with even limited fields of interest. Indexes and Search Engines address the problems of navigating the Information Space and extracting the documents we need. Indexing is primarily done on the basis of full text indexing or indexing of names of authors, titles, and subjects, and many indexes (directory services) are still being handled manually.

If the document collection to be indexed consists of web-pages, however, characteristics such as the names of authors are no longer as readily available, and indexing by hand is completely infeasible. Additionally, locating documents based on author or subject is not always the best way. Consider for example someone wanting to find a restaurant in a certain area. Here a GPS or map based search would be more natural than a pure keyword search. Moreover, many professions like geologists and biologists study geographical areas. For instance, a geologist looking for documents pertaining to a specific area would be better served by a Search Engine capable of limiting a search to that particular area rather than having to sift through the on-topic but not geographically accurate results of a “global” Search Engine. A method for geo-referencing scientifically interesting documents using entities such as place names, animal habitats, and land use data is described in [ref gipsy]. In this article we describe the GRef prototype – a method for automatic geo-referencing of ordinary web-pages using phone numbers and other characteristics found in the pages.

The following sections describe related work, proposed algorithms, and the implemented GRef prototype.

2. RELATED HYPERMEDIA RESEARCH

Hypertext first saw the light of day in the article *As we may think* by Vannevar Bush (1945). Here he described the Memex (Memory Extender) – a machine capable of storing vast amounts of information, which would enable the user to quickly and easily bring up and browse stored documents. Additionally, users could make trails through a number of documents by creating links from one document to another. Since then many different types of hypermedia systems including the Web (Berners-Lee 1992) and Open Hypermedia (Grønbæk 1999, Meyrowitz 1989) have emerged. Recent hypermedia developments combine these into geo-spatial hypermedia (Grønbæk 2002) and general context-aware hypermedia (Bouvin 2003).

The field of geo-spatial hypermedia lies in the concept of linking information to coordinates. In geo-spatial hypermedia, the spatiality is tied to an object in the real world, e.g. a building or a country or the world per se. Thus in geo-spatial hypermedia it is the geography described in the document which is interesting rather than its relation to other documents in the collection (e.g. documents on the same topic).

Geo-spatial hypermedia is closely related with elements in the real world, and access to the system while away from the office often becomes an issue. This dictates a number of hardware requirements such as GPS-receivers, laptop computers, and access to the hypermedia system over a wireless link - the hardware aspect is particular to the field of geo-spatial hypermedia.

A more general kind of hypermedia systems are the context-aware ones which go beyond locations and combine the sensing of a number of context parameters in order to look up hypermedia structures and content relevant to the inferred context. The GRef system presented below would be very useful for the location-based delivery of context-aware hypermedia.

3. PREVIOUS WORK IN GEO-REFERENCING

Most information systems in operation are based on manually created indexes using tokens such as author name, title, and document type. Explicit geographic references are typically not present in such systems, but sometimes side effects of the chosen indexing scheme make limited geographical searching possible.

Although it was not the intent of the Library of Congress to develop a standard subject protocol for document indexing, it has in some degree become one due to widespread international adoption of the procedures developed. There are three geographic Subject Headings (LCSH):

1. A topical subject heading followed by a geographic subdivision (e.g. ART – PARIS).
2. A place name followed by a topical subdivision (e.g. U.S. - HISTORY).
3. Phrases beginning with a geographic adjective (e.g. NORWEGIAN LITERATURE).

These headings make it possible to search for documents with a geographic relevance, but the level of geographic detail is low, and there are several problems associated with the headings.

1. **Inconsistency**: There exists no rule telling the indexer which heading to choose.
2. **Scatter:** The use of subdivision can fragment document references. Someone searching for information on "automation of library processes" will need to search under the headings LIBRARIES - AUTOMATION, LIBRARIES - <country> - AUTOMATION, etc.
3. **Updating:** Updating a large index is time-consuming work – subject headings are not kept up-to-date, and place names may change over time.
4. **Subdivision scoping**: Geographic terms are hierarchical and, for instance, there is no rule telling the indexer whether islands should appear as subdivisions to countries or as top level headings.

Another way of indexing documents is to extract words from (parts of) the documents; this is easy to do automatically (rather than manually) thus drastically reducing the cost of building an index. Documents are retrieved by probabilistically matching search words with the index. In this setting, though, geographic references are even more accidental than in systems using LCSH.

The first system to attempt automatic geo-referencing of documents was GIPSY (Woodruff 1994). New technology made it possible to create and handle much larger indices and to use more complex indexing algorithms. The strategy used was to extract place names and other terms from the documents and use these to estimate the area to which the document referred. For each name or term found a polygon was added to a map covering the corresponding area, and, thus, after parsing a document the map would contain a “skyline” of (overlapping) polygons. The high points of this skyline then corresponded to the areas referenced in the document.

McCurley (2001) researched different possibilities for geo-referencing web-pages using both place names, addresses, and more web-specific entities. He identified two contexts from which to extract geographic information, namely content-based and entity-based – these methods are also used in the GRef prototype and are described in more detail in section 5.2.

4. COORDINATES ARE BETTER THAN NAMES

Studies have shown that using textual representation of geographic references causes several different problems - even in the best kept indexes. Below is a list of advantages found in coordinate-based Geographic Information Retrieval systems (GIR) compared to the name-based systems (Larson 1996):

1. Unaffected by name changes or political boundaries.
2. It is easy to connect coordinate based indexes with a spatial browser interface and Geographic Information Systems (GIS) data.
3. With a system based on coordinates it is possible to calculate distances.

While it is true that a document, which has already been indexed, is not affected by changing borders and names in a given area, it is not entirely true, that for instance political boundaries do not play a part in systems based on coordinates. If the boundaries change in the time between the writing of a document, and the time it is indexed, the resulting coordinates will no longer be correct (if, of course, the system indexer has been updated with the new boundaries). This will perhaps not be a big problem for current documents, but it is a valid concern when indexing older documents or documents describing historical events - they must be indexed using names, geography, and political boundaries corresponding to the point in time, they describe!

5. ADDING GEOGRAPHY TO THE WEB

In 1992 Tim Berners-Lee et al described the vision for the World Wide Web as follows:

> *Pick up your pen, mouse, or favorite pointing device and press it on a reference in this document – perhaps to the author's name, or organization, or some related work. Suppose you are directly presented with the background material – other papers,* the author's coordinates *[my emphasis], the organization's address and its entire telephone directory.*
>
> (Berners-Lee 1992)

Most of this vision has come true – the Web is now the largest document collection ever created, and each document is accessible simply through a click of the mouse. But coordinates are still virtually non-existent.

5.1 Geo-referencing done by the author

There are several ways in which the author can specify the geographic location to which a document refers – one of these is to use geo-tags. Geo-tags (http://geotags.com/geo/geotags2.html) are placed in the HEAD-section of an HTML-document, and using these the author can supply both exact coordinates and place names.

But even though several standards exist, such author-supplied tagging is hardly ever used. In the experiments conducted less than one in ten thousand web-pages contained such tags. Therefore, geographic information has to be inferred using the means available to us.

5.2 Inferring geo-references

There are several different ways of guessing the geographic location of web-pages – they fall into two categories: entity-based and content-based.

5.2.1 Entity-based

The entity-based methods involve aspects of the computing system itself – IP addresses, DNS, and Whois information. Since web-servers are usually located close to a connection point in the Internet backbone, the coordinates of the server can be approximated by using the coordinates of the connection point. Similarly, it has been shown that 25 percent of all second-level domains belong to a geographic top-level domain (e.g. `.dk`) – sometimes further geographic subdivisions enable us to narrow down the area even more (e.g. `sf.ca.us` which points to the city of San Francisco). When registering a domain name the addresses of the administrator and registrant are recorded, and with a little luck this information can also be used in our efforts to infer the coordinates. For a more in-depth description of the methods outlined here see (McCurley 2001).

As a rule, however, the content-based methods are more precise, and the entity-based methods are best used as a heuristic against which the results of the content-based methods can be compared.

5.2.2 Content-based

The contents of the individual web-pages is an indispensable source of geographic information. Reading the geo-tags as described above also belongs in this category. One of the more imprecise content-based methods is to look at the language used in the page – if the page is written in Chinese then it probably belongs in China. However, languages like English are spoken in every corner of the world, so just based on the language there is no way to tell exactly where a web-page in English belongs. For smaller languages like Danish, however, we can get fairly precise results.

Phone numbers written using the standard for international numbers contain both a country code and an area/city code – with a bit of luck a database mapping phone numbers to exact coordinates can be obtained thereby boosting the precision. However, not all numbers are written using this standard – often geographic codes are evident from the context of the web-page and are thus left out of the number. Furthermore, the use of whitespace, parenthesis, dashes, etc. varies a lot, and thus it can be hard to simply identify the phone numbers in the first place.

In order to use postal addresses, a database mapping addresses to coordinates is needed, but such a database is not available for all countries. McCurley (2001) used the Tiger/Zip+4 database distributed by the Postal Service combined with another database in order to get the coordinates. In his prototype, only the centroid of each postal code was used giving him an average resolution of 40 square miles. However, in some cases the resolution was considerably lower – the postal code for northern Alaska covers 27,000 square miles. The GRef prototype geo-references web-pages in the `.dk` domain using an address database with a resolution of one meter, see next section for details.

Finally, we include addresses (URLs) of the pages themselves.

6. AN ALGORITHM FOR AUTOMATIC GEO-REFERENCING

The algorithm for (basic) geo-referencing of a document is quite simple: run through a document and find something that we can attach a location to. But it requires a lot of CPU time, since the task of "finding something" is very time consuming.

If for instance we have a list of 1,000 different tourist attractions and their geographic location and 100,000 documents, then this means matching 100,000 documents (each stripped of all HTML-tags) with a 1,000 different strings containing the names, i.e. 1,000,000 string searches. Additionally, it cannot be done using fewer string searches since one document can contain references to any number of tourist attractions, and thus we cannot stop looking for names in a document once we have found the first - all names must be looked for in all documents. This is not to say that simple string matching is the right thing to do, though – but it does give us an idea of the amount of work waiting.

```
D = {document body}
E = <list of entities that can provide a geographic position>
foreach d in D
   foreach e in E
      if e found in d
         retrieve coordinates from database
         tag d with coordinates of e
```

The lines above describe a general geo-referencing algorithm, but they do not reveal where most of the work lies, namely in the line "`if e found in d`". In the algorithm `E` is meant to cover all the different ways of "finding something" that can lead us to a set of coordinates. For example, `E` could be the list `<points of interest, addresses>`, meaning that we want to try to locate e.g. a number of tourist attractions and/or something like a postal address and use these entities to fix the document on the map.

Addresses consist of several different parts which together form the address: each zip code corresponds to a certain place name, and within the area defined by the zip code there are a number of street names, and, finally, for each street there are some numbers.

If a document contains all the parts necessary to create a legal address, and if the parts are found within a limited span of the text, then we can be fairly certain that what we have found actually *is* an address. Thus, the structure of addresses in itself provides us with a certain degree of address validation.

6.1 Data-model for a geo-index

The GRef prototype stores information about the web-pages visited during the crawl is stored in a page-centric database, and the pages themselves are stored in a page repository. The database contains pointers to the cached copies of the pages. The database schema is shown below:

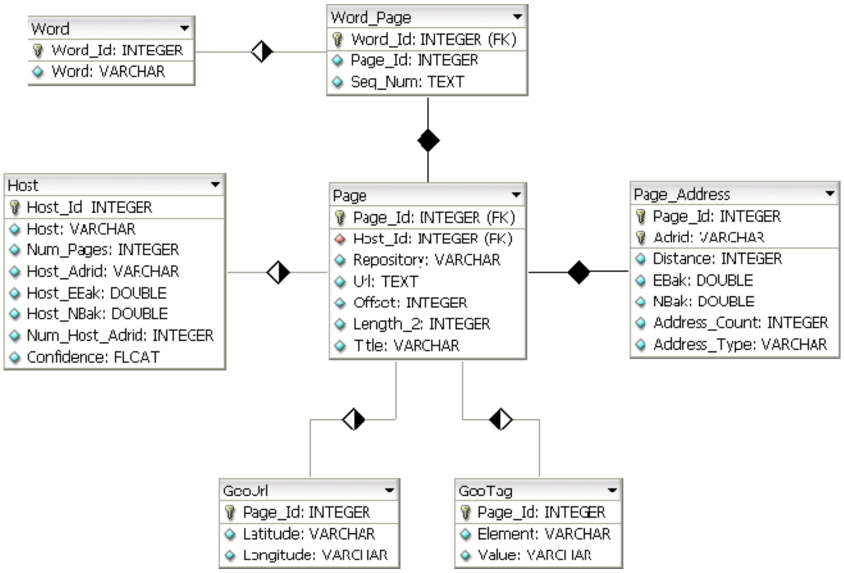


Figure 1. Database schema

The `word` and `word_page` tables implement a mapping from words to pages and thus make it possible to find pages matching a number of search words. For each address found in a page, a record is stored in the `page_address` table, and finally, geo-tags and geo-urls provided by the author are stored in the corresponding tables.

The page repository is just a collection of text files of up to 50 Mbytes in which the web-pages are stored one after another. Given a file name, an offset, and a length a page can be extracted from the repository.

6.2 Implementation

The structure of an address is hierarchic - zip code and city provide us with a well-defined geographic region, the street narrows down that region, and the number points out the exact spot. Looking for the parts of an address in this order has the simple but important property of cutting down the size of the search space as quickly as possible. The algorithm is shown below and a description of how it works follows.

```
DEFINE threshold = 100

words      = <list of words in document>
zip_codes  = <list of valid zip codes>
zips_found = intersect(words, zip_codes)

foreach zip in zips_found
    if "zip city" found in document
        pos1 = <position in document>
        foreach street in zip
            if "street <number>" found in document
                pos2 = <position in document>
                if pos1 > pos2
                    # street and number found before zip city
                    if pos2 - pos1 < threshold
                        # all parts found within 100 characters
                        tagDocumentWithFoundAddress()
```

First all possible zip codes in the document are found. For each zip code it is checked whether or not the corresponding place name follows immediately after the zip code (i.e. the zip code and place name may only be separated by whitespace). If this is the case, then a similar search for street names and numbers is performed. Finally, it is checked that the distance from street and number to zip code and city is within some threshold (currently 100 characters), and that the street name comes first. If all these checks are completed successfully, the document is referenced with the coordinates of the found address. Documents containing several addresses are referenced once per address.

6.3 Regular expressions and walking the thin line

Names of cities and streets are just strings, but simply searching for a street name like *H. C. Andersens vej* in the documents is sure to return poor results. Different use of punctuation, abbreviations, and whitespace as well as spelling mistakes necessitates the use of slightly more sophisticated methods. Thus, regular expressions are used in the prototype.

Using the street name above as an example, the list below shows a number of variations that are bound to show up in practice:

1. H. C. Andersens vej
2. H.C. Andersensvej
3. H. C. Andersensvej
4. Hans Christian Andersens vej

The challenge now is to create a regular expression capable of matching all of these variations (or as many as possible), while at the same time avoiding false positives – it is a thin line to walk! Through trial and error and a lot of checking done by hand the algorithm for creating the regular expressions ended up looking like this:

```
street_name = <list of words in street name>
last_word   = pop(street_name)

regexp = ''
foreach word in street_name
    regexp += sub_string(word, 0, 2) + \w*[ \.-]*
regexp += last_word
```

The algorithm does the following: single-word street names are matched exactly, but whenever the names consist of more words the regular expression is created using the first two characters of the first words and all of the last word. Between these (sub-)strings any number of characters followed by any number of whitespace, periods, and dashes are allowed. Thus, again using the street name from the example above, the algorithm converts the string *H. C. Andersens vej* into the regular expression below (case is disregarded):

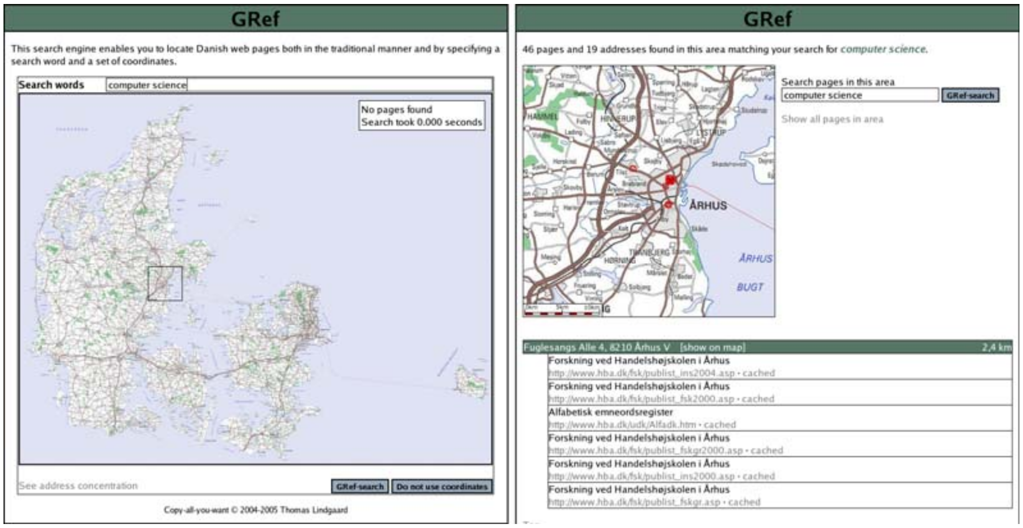
```
H\w*[ \.-]*C\w*[ \.-]*An\w*[ \.-]*vej
```

This expression will match all of the five variations of the street name shown in the list above, but, more importantly, false positives very seldom occur. Naturally some addresses will slip through the filter and sometimes a string which is not an address will be mistaken for one. Using just the first character of the first words causes a dramatic increase in the number of false positives, however, and using three characters causes a not-quite-as-dramatic decrease in the number of addresses identified. The reason for including all of the last word is that, if abbreviations are used then (for Danish street names) it is always the first words which are not spelled out and not the last – other rules apply in other countries (e.g. *S. Grandview Hwy.* in Vancouver, Canada).

Regular expressions are only used for street names – names of cities are matched exactly. However, more than half of the city names in our database contain either whitespace, periods, or dashes, thus suggesting a similar use of regular expressions; but in practice the problem is not nearly as serious as is the case for street names.

6.4 Search interface

The interface created for GRef consists only of a text box for search words and a map. At the bottom are two buttons: one for searching for the search words (if any) within the specified area and one for searching “globally” (i.e. without using the coordinates). Screenshots of the interface are shown in Figure 2.



This search engine enables you to locate Danish web pages both in the traditional manner and by specifying a search word and a set of coordinates.

Search words: computer science

No pages found
Search took 0.000 seconds

Copy-all-you-want © 2004-2005 Thomas Lindgaard

46 pages and 19 addresses found in this area matching your search for *computer science*.

Search pages in this area: computer science

Fuglesangs Alle 4, 8210 Århus V [show on map] 2,4 km

Forskning ved Handelshøjskolen i Århus
http://www.hba.dk/fsk/publist_ins2004.asp • cached

Forskning ved Handelshøjskolen i Århus
http://www.hba.dk/fsk/publist_fsk2000.asp • cached

Alfabetisk emneordsregister
http://www.hba.dk/udk/Alfadk.htm • cached

Forskning ved Handelshøjskolen i Århus
http://www.hba.dk/fsk/publist_fskgr2000.asp • cached

Forskning ved Handelshøjskolen i Århus
http://www.hba.dk/fsk/publist_ins2000.asp • cached

Forskning ved Handelshøjskolen i Århus
http://www.hba.dk/fsk/publist_fskgr.asp • cached

Figure 2. GRef interface

The results of a search are shown both as crosshairs on a more detailed map and in a list ordered by addresses – these addresses are ordered by distance. Unlike traditional Search Engines the relevance of a page is not determined by an algorithm like the Google PageRank – if the page contains an address in the area and matches the search words then it is relevant to the query. However, rather than just showing all pages in an area, matching pages could be sent through a PageRank-like filter in order to produce better results. A new search within the same area can be executed using the form next to the detailed map.

It is possible to simply browse the pages found in a given area by simply executing a search without coordinates; doing this returns all pages in the area. However, given a large number of pages this approach would not work, because the map would be covered with crosshairs. A more sophisticated browse-interface could allow the user to specify the kind of pages to show based on a directory structure – for instance, the user could be interested in `Restaurants -> Fast food`.

6.5 Results

How well does it work? An absolute answer to this question would require manually running through all geo-referenced documents and checking the found addresses to see if some were missed. However, spot checking the results gained from running the algorithms on a 140,000 page web-crawl show that very few addresses slip through unnoticed. In fact the only addresses known to have slipped past share one of the following characteristics:

1. The address is not written (entirely) in Danish.
2. The street name and/or city name is abbreviated (or misspelled).
3. The address database does not contain that exact house number for the found street. A geo-reference pointing to the street could be added, but this is not done in the prototype.

Using the algorithms described here, the prototype was able to geo-reference 10 percent of the pages visited.

7. INFERRING DOMAIN-ADDRESSES

Having 10 percent of the pages geo-referenced is a good start, but we can do even better! Since the document body used in the prototype consists of web-pages, we can use the structure of these to infer addresses for other pages. In this context, "structure" means the addresses (the URLs) of the pages: two pages on a given host are related in some way, whereas pages from different domains are not (in general).

In a world where pages from a given domain only point to one particular address, it would be very easy to infer that other pages not containing addresses in that same domain should point in the same direction. In the real world, however, we have no such guarantee. Interestingly, on more than two-thirds of the visited domains (for which a page containing an address was found) the pages only point to a single address – pages containing addresses were found on 17 percent of the visited domains, but this fraction may increase given a more “complete” crawl (i.e. a crawl where more pages on each domain are visited).

Here “domain” refers to the second level of a domain name – if the domain name in question is www.google.com then the second-level domain is *google*. Domain addresses could be extended to the next levels, but in the prototype everything below the second level is ignored. These addresses are stored in the host table in the database and thus apply to all pages on that host/in that domain, see Figure 1.

By locating the domains for which the pages only contain a single address and geo-referencing the remaining pages with the coordinates of this address, the percentage of geo-referenced pages could be boosted from 10 to 20 percent in 140,000 page web-crawl. Furthermore, the pages on multi-address domains could be referenced in a similar way (albeit at a lower level of confidence).

8. AUTOMATIC VERSUS MANUAL GEO-REFERENCING

Automatic geo-referencing may not always be the preferred solution. If the system being built is meant for tourists visiting a certain area, then it may be better to let the individual businesses in the area add their own information to the system, for example through a web-service in which the businesses can request a visit from the geo-referencing crawler or in which explicit coordinates for the business can be entered.

When using automatic geo-referencing to build an index of pages for a system like the tourist system, it is hard to avoid “noise” in the form of pages not actually related to the task at hand. Automatic systems are more suitable for a general-purpose Search Engine than systems aiming at a more narrow area of application.

9. COMPARING WITH GOOGLE

Google recently launched two new services: Google Local and Google Maps. In essence, they are just two versions of the same system – one with a text-based interface and one using maps. The two systems are the result of integrating the database maintained by the Yellow Pages with the page index created by Google thereby providing the users with a system capable of locating the web-sites of the businesses found in the Yellow Pages. In the resulting system the user no longer needs to browse through a directory structure as in the paper edition of the Yellow Pages, e.g. instead of looking up `Restaurants -> Fast food` the user just searches for fast food in a given area.

In contrast, GRef is aimed at general geo-referencing of web-pages – not just pages belonging to businesses. Furthermore, through the use of domain addresses, GRef is capable of geo-referencing a larger fraction of the web-pages and is thus capable of leading users to pages matching a query but not containing addresses while still restricting the search geographically – the task for the Google services is simply to take to user to the main page of site belonging to a business or, alternatively, to a hub page leading to a number of different businesses (e.g. a list of hotels).

10. CONCLUSION

The paper has presented a novel approach to automatic indexing of Web-pages based on addresses in the content of the pages. A GRef prototype for geo-indexing of Danish pages based on addresses in contents has been developed. The approach has to be adapted to different language areas since the structure of representation of addresses differs from language to language. Currently, no attempt is made to identify false positives, but some validation could be done by combining content based-indexing with entity-based and even geo-tag-based indexing.

REFERENCES

Berners-Lee, T. et al, 1992, World Wide Web: The Information Universe. *Electronic Networking: Research, Applications and Policy*, 1 (2), 74-82.

Bouvin, N.O. et al, 2003. HyCon: A Framework for Context-aware Mobile Hypermedia. *Hypermedia.* 9 (1), 59-88.

Bush, V. 1945. As we may think. *Interactions.* 3 (2), 35-46.

Grønbæk, K. and Trigg, R.H., 1999. *From Web to workplace: designing open hypermedia systems*. Boston, MA: MIT Press.

Grønbæk, K. et al. 2002. Towards geo-spatial hypermedia: concepts and prototype implementation. In *Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia (College Park, Maryland, USA, June 11 - 15, 2002).* New York, NY: ACM Press, 117-126.

Larson, R.R, 1996. Geographic Information Retrieval and Spatial Browsing. In: Smith, L. and Gluck, M., eds, *GIS and Libraries: Patrons, Maps and Spatial Information*, University of Illinois, 81-124.

Lindgaard, T., 2005. Geo-spatial Indexing and Searching. Thesis (M.Sc.). University of Aarhus. Available online at http://it-snedkeren.dk/GRef/Geo-spatial_Indexing_and_Searching.pdf.

McCurley, K.S., 2001. Geospatial mapping and navigation of the web. In *Proceedings of the 10th international Conference on World Wide Web (Hong Kong, Hong Kong, May 01 - 05, 2001).* New York, NY: ACM Press, 221-229.

Meyrowitz, N. 1989. The missing link: why we're all doing hypertext wrong. In Barrett, E., ed, *The Society of Text: Hypertext, Hypermedia, and the Social Construction of information*. Cambridge, MA: Mit Press, 107-114.

Woodruff, A.G. and Plaunt, C., 1994. GIPSY: Georeferenced Information Processing SYstem. *Journal of the American Society for Information Science. (Oct. 1994).* 45 (9), 645-655.