

Towards a Spatial Search Engine Using Geotags

Jan Torben Heuer, Sören Dupke

Institute for Geoinformatics, University of Münster, Germany
{jan.heuer, soeren.dupke}@uni-muenster.de

Abstract. We introduce the idea of a spatial search engine based on geotags. Geotags are keywords linked to a concrete position. User generated geotags are available from Web 2.0 portals like Flickr¹ or Google Maps². We collected a sample dataset of about 300.000 geotags. In the following we explain a prototype implementation of a search engine and describe how to compute the spatial relevance of a tag. The last section gives an outlook about our research goals in this area and discusses the challenges and possible benefits of integrating semantic information.

1 INTRODUCTION

Geotagging is becoming increasingly popular. More and more consumer devices like digital cameras and mobile phones include GPSsensors (Toronai 2007). This allows to easily combine attributive information with spatial location. The idea of a geotag is that the information transported by a tag is enriched by a spatial location. People share their geotagged contents on platforms like Flickr or the Google Earth Community³. Beside these communities, the number of private web pages that incorporate Yahoo- and Google Maps is also growing. Daily, thousands of new tags are annotated with a position and linked to information in form of text, images, videos or other browseable content. This makes geotags a rich information source.

We understand a *geotag* as a pair of a keyword and a position. The position can be encoded in any standard geographic reference system, as long as it is convertible to WGS84.

Spatial relevance is the probability that the keyword of a geotag belongs to the assigned real-world position. A good example might be the tag *Eifel Tower*. Almost all geotags with the keyword *Eifel Tower* are limited to a small area. These geotags have a probability close to 100% to point to the *Eifel Tower* in Paris, while few geotags for shops in China selling *Eifel Tower* miniatures have a very low probability.

¹ <http://www.flickr.com/>

² <http://maps.google.com/>

³ <http://bbs.keyhole.com/>

Another example, the keyword *house* has geotags distributed almost equally all over the world. It is improbable to find a certain house, thus the geotags for house have a low spatial relevance.

The use of cluster-algorithms allows us to determine if geotags form a cluster of high density and therefore are likely to belong to the same feature. This approach allows us to define the probability to a cluster of geotags rather than for each tag separately.

We collected a sample dataset of geotags from the web and implemented a search engine to extract, filter and rank the relevant information from these geotags. This allows the use of this huge amount of data in other applications that process spatial information.

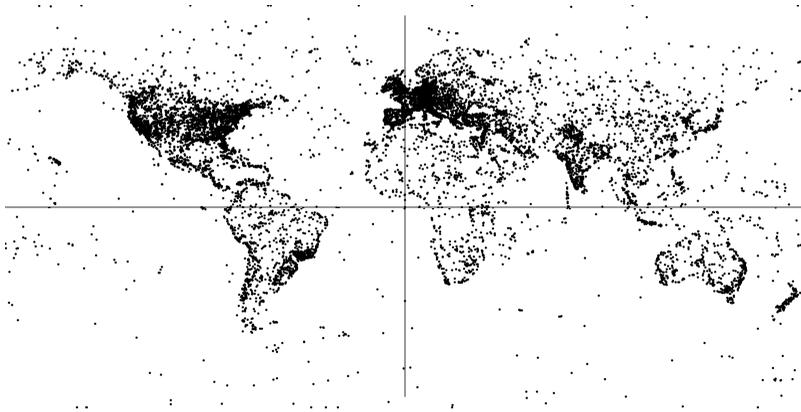


Figure 1: Positions of all geotags in our database

2 RELATED WORK

Traditional search-engines use a text and link-based algorithm (Brin 1998) to rank the data. Our geotags are not linked to each other. In order to rank the data by its spatial relevance, a spatial processing function has to be added. Markowetz (2007) describes how other search engines for spatial information compare the content and structure of a website with a database of postal codes, country or city names to assign a spatial position. In contrast to these search engines our dataset already contains precise positions. These positions are used in spatial algorithms to make a statement about the relatedness between a tag and a position.

The idea of finding spatially relevant tags is also described by Rattenbury et al. (2007). However this example focuses on inferring spatial and

temporal semantics from tags. Our search engine uses a clusterfinding algorithm based on a Delaunay triangulation (Leonidas 1992), which is unaffected by the shape of the clusters (Eldershaw 1997). A similar scenario is described for storage of huge spatial datasets (Estivill 1999).

3 SPATIAL PROCESSING

The search engine shall return a ranked list of geotags for a given keyword. The first step is divided into finding and optimizing the clusters. In the next step we calculate the spatial relevance from the properties of the cluster.

3.1 Finding Clusters

Our algorithm finds the positions that are relevant for a certain geotag. To distinguish these from the tags that are distributed all over the world, the algorithm has to find clusters with a high density. The Delaunay triangulation described in (Leonidas1992) creates a graph $G = (V, E)$ with geotags as nodes V and edges $E = v_1 \times v_2$. To find the clusters, we introduce a threshold t and remove those Edges where the geographic distance $d(n_1, n_2) > t$.

Figure 2 shows a graph for the geotag *Wales*. The edges longer than the threshold distance are grey and the resulting clusters are black. You can identify Wales in Great Britain and New South Wales in Australia: the resulting clusters with a density.



Figure 2: Sample diagram for the tag Wales

3.2 Calculating the Spatial Relevance

It is not sufficient to find clusters because a large number of tags forms big clusters, which are not referring to real world objects. The tag *house*, shown in figure 3 is a good example. A deeper analysis of the resulting clusters is necessary:

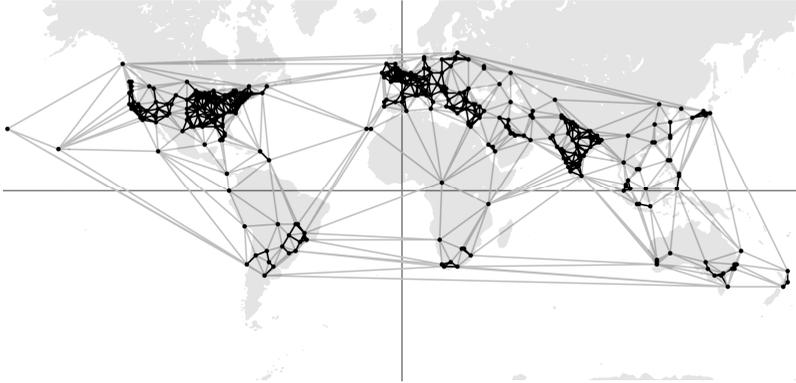


Figure 3: A sample Delaunay diagram for the tag *house*.

We compute clusters for a predefined set of thresholds (e.g.: $t = 1, 2, \dots, 50$). In order to calculate the spatial relevance of the geotags, we use the following properties:

- *total points in the cluster*
- *area of cluster*
- *total clusters*
- *total geotags*

Of course, a lot more properties are possible but for a first prototype we limited ourselves to the ones above.

The challenge now is to find the relation and weight between the properties and the correct value for the clusters' threshold. At this point, we don't use a fixed formula, but we introduce the neural network: We create clusters for different thresholds and use the neural network to figure out which threshold creates the optimal clustering for a tag.

The neural network weights the properties and calculates the probability. Our search engine can now present a ranked list of geotags, based on their spatial relevance.

4 CONCLUSIONS

Most clusters we found represent well-known cities around the world, but these are already available through Yahoo or Google. Our goal is to retrieve spatial information from geotags that represent activities, events or other points of interest. This can be tags like “hiking” which show favored hiking-areas around the world.

Markowetz (2007) used postal codes and address databases to assign positions to webresources (Section 2). The geotags that are filtered and ranked by our search engine can be used as an additional datasource. With more and more geotags we also expect to reduce the threshold of the cluster. This allows a finer distinction between clusters and the engine will be able to identify smaller features.

5 FUTURE WORK

We plan to extend our search engine with spatial operators like intersection and union for our clusters in order to be able to create more complex queries. The example: “hiking AND Chile” will only return search results within the hiking and the Chile cluster. A central aspect of our ongoing work will be the training of our neural network. At the moment a manual training based on black and white lists of geotags is sufficient, but with an increasing number of tags, we propose to adopt existing spatial-semantic information, for example the placelist from the Alexandria Digital Library Gazetteer (ADLG 1999).

REFERENCES

- Alexandria digital library gazetteer (1999) santa barbara ca: Map and imagery lab, davidson library, university of california, santa barbara. copyright uc regents. <http://www.alexandria.ucsb.edu/gazetteer>.
- Brin, S. and L. Page (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.
- Eldershaw, C. and M. Hegland (1997) Cluster analysis using triangulation. In B. J. Noye, M. D. Teubner, and A. W. Gill, editors, *Computational Techniques and Applications*, pages 201–208. CTAC97, World Scientific, 1997.

- Estivill-Castro, V. and Ickjai Lee (1999) AMOEBA: Hierarchical clustering based on spatial proximity using Delaunay triangulation. Technical Report 99-05, Callaghan 2308, Australia.
- Leonidas J. Guibas, Donald E. Knuth, and Micha Sharir (1992) Randomized incremental construction of delaunay and voronoi diagrams. *Algorithmica*, 7(4):381–413.
- Markowitz, A., Yen-Yu Chen, Torsten Suel, Xiaohui Long, and Bernhard Seeger (2005) Design and implementation of a geographic search engine. In *WebDB*, pages 19–24.
- Rattenbury, T. , Nathan Good, and Mor Naaman (2007). Towards extracting flickr tag semantics. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1287–1288, New York, NY, USA. ACM Press.
- Torniai, C., S.Battle, and S. Cayzer.(2007) *Sharing, Discovering and Browsing Geotagged Pictures on the Web*, chapter *Sharing, Discovering and Browsing Geotagged Pictures on the Web*. Springer Book.