In *GIS'11: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Agrawal, I. Cruz, C. S. Jensen, E. Ofek, and E. Tanin, eds., pages 489–492, Chicago, November 2011.

Searching Web Documents as Location Sets

Marco D. Adelfio Sarana Nutanong Hanan Samet

Center for Automation Research, Institute for Advanced Computer Studies,
Department of Computer Science, University of Maryland
College Park, MD 20742, USA
{marco, nutanong, hjs}@cs.umd.edu

ABSTRACT

A geographic search system named GeoXLS is presented, which enables users to submit a set of locations as a query object Q and to find documents containing locations similar to those in Q. Search results come from a collection of geotagged web documents, specifically a vast collection of spreadsheets obtained from the Web. The results are ranked according to their similarity to Q, using one of several user-selected similarity measures related to the Hausdorff distance. GeoXLS allows users to answer queries such as "I know the locations of n entities of type X. What sets of data contain points similar to my query points?" For example, given a set Q of known impact craters, find documents that contain locations similar to those in Q and beyond. In essence, this allows someone to "complete the set" by identifying sets containing similar locations. GeoXLS provides capabilities analogous to a standard keyword search engine, but with keywords specified geographically. In contrast to a search engine that handles only text queries, our geographic search system is capable of returning search result documents that are not exact matches to the query. For example, searching with query points in "Washington, DC", "Denver, Colorado", and "Chicago, Illinois" could return documents related to colleges with actual locations in "College Park, Maryland", "Boulder, Colorado", and "Evanston, Illinois", which are similar spatially, but not textually. GeoXLS can be useful in a wide variety of knowledge domains where the data can be represented as a collection of point sets.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*

General Terms

Algorithms, Design

Keywords

Similarity search, Spatial databases, Query processing.

ACM SIGSPATIAL GIS '11, November 1-4, 2011. Chicago, IL, USA

1. INTRODUCTION

Online search engines have put knowledge at the fingertips of anyone with an Internet connection and the ability to express their search query textually. However, not all queries can be easily expressed textually, and new search systems are attempting to address this by allowing different, non-textual query formats. For example, systems are being developed to allow users to use an image or an audio clip as a query [3, 9].

For locational data, systems such as online map services allow users to interact with locational data in an intuitive manner. For example, a user can zoom into an area of interest and issue a query to display primary schools in the area. The system then displays results ranked according to the distance from the center of the zoomed area. As can be seen, this type of system works well with cases where each data entry can be represented as a single location (e.g., a school or a gas station).

However, for cases where each data entry may comprise multiple locations, a single query point or a simple range query may be insufficient to fully specify a search query. For example, a search for disease outbreaks that spread geographically in a similar manner to a new outbreak requires entering a set Q of locations from the new outbreak and comparing Q to a database of previous outbreaks. There are many other circumstances where doing a full *"point set to point set"* query fits the problem definition most closely. To accommodate queries such as these, we have developed a novel geographic search system called GeoXLS.

In this paper, we present GeoXLS, which enables users to submit a set of locations as a query object Q and to find documents containing locations similar to those in Q. The results are ranked according to their similarity to Q, using one of several user-selected similarity measures related to the Hausdorff distance.

As an example application, we focus on a collection of geotagged [5] spreadsheets retrieved from the Web. Our spreadsheet collection contains lists of locations like universities, airports, and national parks. We use this spreadsheet collection to demonstrate how GeoXLS can be used to "complete the set" by identifying sets containing similar locations.

To demonstrate the versatility of GeoXLS, we also show that it can be applied to two other types of geotagged web documents in addition to the spreadsheet data. First, we apply GeoXLS to a collection of disease outbreak data. Users can specify a set Q of locations and find disease outbreaks that contain locations similar to those in Q. Second, we use a collection of geotagged news articles where each article may contain multiple locations. Using this dataset, users may is-

In *GIS'11: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Agrawal, I. Cruz, C. S. Jensen, E. Ofek, and E. Tanin, eds., pages 489–492, Chicago, November 2011.

sue a query like "Find events that are related to these n geographic locations." For example, to understand interactions between the United States and China related to military activities in Libya, a user can place points in Washington, Beijing, and Tripoli as a search query, which returns multiple relevant articles.

The rest of this paper is organized as follows. In Section 2, we describe the measures used to determine the similarity between two point sets. Section 3 provides an outline of the components of GeoXLS, while in Section 4, we give a narrative of query examples. Section 5 contains conclusions and directions for future research.

2. SIMILARITY MEASURES

The ability to determine the degree of dissimilarity (or the distance) between two point sets is crucial to GeoXLS. In this section, we explain how we compute the distance between two locations given in the spherical coordinate system (latitude and longitude). We then explain how the distance between two location sets can be computed. Related work and theoretical discussion on similarity search can be found in our research paper [1].

2.1 Distance between two locations

We use the equirectangular projection [8] to approximate the earth surface as a 360×180 square-unit rectangle. The distance between any two points (represented as lat/long pairs) is measured as the Euclidean distance on this projection where each degree is considered one unit length, i.e.,

$$\textsc{Dist}(\boldsymbol{p_1}, \boldsymbol{p_2}) = ||\boldsymbol{p_1} - \boldsymbol{p_2}||.$$

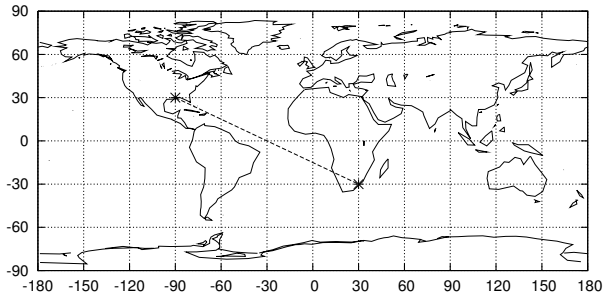


Figure 1: Equirectangular projection of the Earth's surface onto a 360×180 square-unit rectangle.

As shown in Figure 2.1, the distance from $(-90, 30)$ to $(30, -30)$ is given as

$$\sqrt{(-90-30)^2 + (30-(-30))^2} = 134.16 \text{ units.}$$

Although the equirectangular projection introduces distortions, it allows us to approximate the Earth's surface as a Cartesian space and to use existing R-Tree libraries for our implementation. The main contribution of our work is formalization of methods to perform similarity search over a collection of point sets. As future work, we plan to adopt a more accurate measure such as the great circle or the ellipsoidal distance [8] to our search methods.

2.2 Distance between two sets of locations

We now describe the distance measures that we use for comparing point sets. GeoXLS allows users to select between the following two distance measures.

- **Hausdorff distance:** the maximum discrepancy of one point set with respect to another.
- **Modified Hausdorff distance:** the average discrepancy of one point set with respect to another.

Since both measures are directional, the distance of a point set S with respect to a query point set Q can be measured in three different directions: (i) from Q to S, (ii) from S to Q, and (iii) bidirectional (symmetric). The definitions of the Hausdorff distance for the three directions are given in Table 1, where the distance function $\textsc{MinDist}(\boldsymbol{a}, B)$ is defined as $\textsc{min}\{\textsc{Dist}(\boldsymbol{a}, \boldsymbol{b}) : \boldsymbol{b} \in B\}$.

Table 1: Hausdorff Distance

Direction	Definition
$Q \Rightarrow S$	$\textsc{HausDist}(Q, S)$ $= \textsc{max}\{\textsc{MinDist}(\boldsymbol{q}, S) : \boldsymbol{q} \in Q\}$
$S \Rightarrow Q$	$\textsc{HausDist}(S, Q)$ $= \textsc{max}\{\textsc{MinDist}(\boldsymbol{s}, Q) : \boldsymbol{s} \in S\}$
$Q \Leftrightarrow S$	$\textsc{SymHausDist}(Q, S)$ $= \textsc{max}\{\textsc{HausDist}(Q, S), \textsc{HausDist}(S, Q)\}$

Figure 2 shows a query point set Q and a collection $\mathcal{D}$ of $\{R, S, T\}$ with the following Hausdorff distances.

- $Q \Rightarrow R$: $\textsc{HausDist}(Q, R) = 8$ units.
- $R \Rightarrow Q$: $\textsc{HausDist}(R, Q) = 3$ units.
- $Q \Rightarrow S$: $\textsc{HausDist}(Q, S) = 4$ units.
- $S \Rightarrow Q$: $\textsc{HausDist}(S, Q) = 7$ units.
- $Q \Rightarrow T$: $\textsc{HausDist}(Q, T) = 5$ units.
- $T \Rightarrow Q$: $\textsc{HausDist}(T, Q) = 5$ units.

Hence, the order of similarity search results for each direction is given as follows.

- $\langle Q \Rightarrow S : 4, Q \Rightarrow T : 5, Q \Rightarrow R : 8 \rangle$.
- $\langle R \Rightarrow Q : 3, T \Rightarrow Q : 5, S \Rightarrow Q : 7 \rangle$.
- $\langle Q \Leftrightarrow T : 5, Q \Leftrightarrow S : 7, Q \Leftrightarrow R : 8 \rangle$.

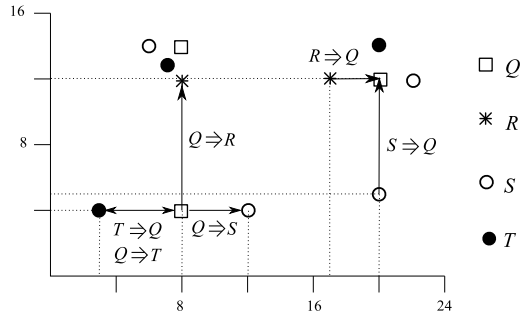


Figure 2: Four point sets Q, R, S and T in Euclidean space and the Hausdorff distances to/from R, S and T with respect to Q.

Since $\textsc{HausDist}(Q, S)$ is a measure of the maximum discrepancy of Q with respect to S, the measure can be sensitive to outliers. Specifically, if there is only one object $\boldsymbol{q}$ in Q that is far away from S, then the distance from that object $\boldsymbol{q}$ to S will be used as the resultant distance. That is, the measure disregards the majority of points in Q which are much closer to S. To mitigate this problem, a variant of the Hausdorff distance called the *modified Hausdorff distance (MHD)* [6] can be used to spread out the effect of outliers over the entire point set Q. Similar to the Hausdorff distance, the modified Hausdorff distance of a point set S with respect to a query point set Q can be measured as $\text{MHD}(Q, S)$, $\text{MHD}(S, Q)$, and $\textsc{SymMHD}(Q, S)$. The definitions of these distance functions are given in Table 2.

Table 2: Modified Hausdorff Distance

Direction	Definition
$Q \Rightarrow S$	$\text{MHD}(Q,S) = \sum\{\text{MinDist}(\boldsymbol{q},S) : \boldsymbol{q} \in Q\}/\lvert Q\rvert$
$S \Rightarrow Q$	$\text{MHD}(S,Q) = \sum\{\text{MinDist}(\boldsymbol{s},Q) : \boldsymbol{s} \in S\}/\lvert S\rvert$
$Q \Leftrightarrow S$	$\text{SymMHD}(Q,S) = \max\{\text{MHD}(Q,S), \text{MHD}(S,Q)\}$

Efficient methods to compute the Hausdorff and modified Hausdorff distances are presented in existing literature [6, 7].

3. SYSTEM COMPONENTS

In this section, we describe the components of GeoXLS. As shown in Figure 3, GeoXLS consists of the following components: (i) document geotaggers, (ii) RDBMS, (iii) Hausdorff spatial index, and (iv) web application.

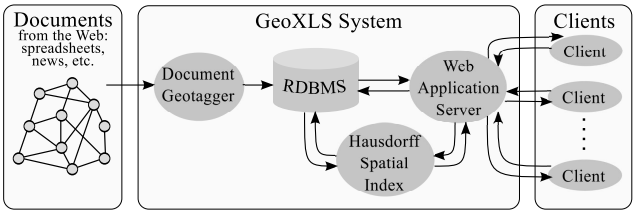


Figure 3: GeoXLS system architecture.

Document Geotaggers. Before users can search for web documents, those documents must be processed and stored. This is accomplished by the *document geotagger* module, which performs different actions depending on the data source (e.g., spreadsheets, news articles, disease outbreaks, etc.). Each data source has a dedicated submodule, which takes an individual document as input and returns a list of locations with associated lat/long values. For example, the spreadsheet import submodule performs a multi-step procedure to locate Microsoft Excel (XLS) files on the Web, identify blocks of data within each document (e.g., by excluding non-data rows such as titles, column headers, and notes), and associate a geographic location with each data row if possible. This procedure is known as *geotagging* a spreadsheet [5]. We have developed similar submodules for geotagging news articles and disease outbreak documents. This concept can also be adapted to other document types [4].

RDBMS. The results of the document geotagging process are stored in a relational database management system (*RDBMS*) which maintains information about each geotagged document and its associated points. For example, in the case of spreadsheets this includes the original source URI and title of each document, as well as the row number, place name, and lat/long for each associated point. This enables efficient ad hoc access to each web document's attributes without requiring that the original documents be read during every GeoXLS query.

Hausdorff Spatial Index. In addition to the RDBMS, we developed a Python and C++ module to perform Hausdorff searches, which effectively serves as a Hausdorff spatial index over our database of point sets. That is, the Hausdorff spatial index component is used to answer questions of the following general form for a specified query point set q (note that we express the query using SQL syntax, but the Hausdorff index is not actually integrated into the RDBMS):

```
1:      SELECT pt_set
2:        FROM collection
3:       WHERE NumPoints(pt_set) < maxPoints
4:    ORDER BY HausDist(q, pt_set)
5:       LIMIT k;
```

The search system accepts the following parameters:

- q. Query points can either be specified as a set of lat/long values or by using another point set that already exists in the index.
- $k \geq 1$, specifies the number of search results to return.
- $maxPoints \geq 1$, the maximum point set size to consider for the search. Some documents have tens of thousands of points with wide geographic coverage. Without this parameter, these documents can cause search results to be full of large, mostly irrelevant point sets.

The distance function (Line 4) and the order of its arguments are determined by two additional parameters: *Hausdorff Type* and *Direction*. The following table shows how different combinations result in different settings.

Direction\Type	HausDist	MHD
FromQuery	`HausDist(q, pt_set)`	`MHD(q, pt_set)`
ToQuery	`HausDist(pt_set, q)`	`MHD(pt_set, q)`
Symmetric	`SymHausDist(q, pt_set)`	`SymMHD(q, pt_set)`

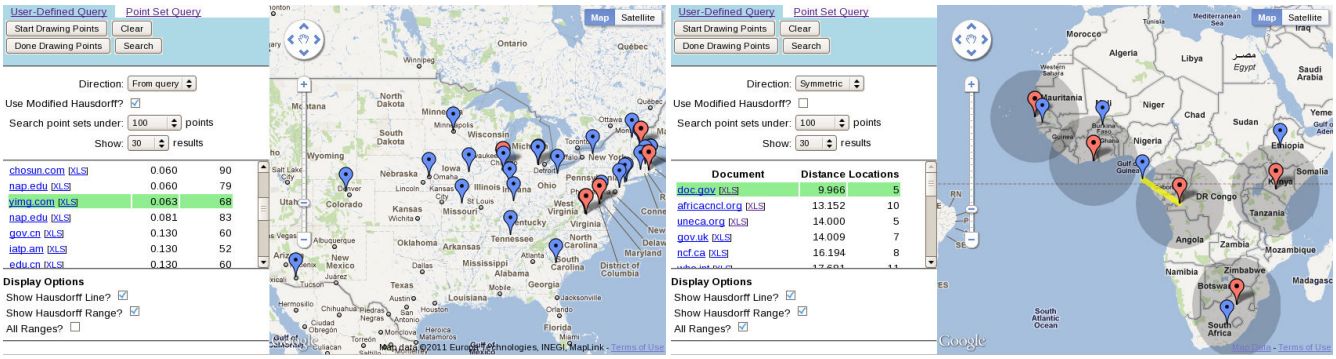
The Hausdorff spatial index is reconstructed periodically to incorporate additions to each document collection, and is kept in memory to support efficient query response times.

Web Application. The final component, which connects the RDBMS and Hausdorff spatial index to the end-user, is the GeoXLS web application. The application is composed of a back-end server written in Python, and a client-side web application written in JavaScript.

The main function of the web application server is to generate the HTML pages that the client interacts with and handle asynchronous requests for data from the client. The current implementation of the server handles two types of asynchronous HTTP requests: *search query requests*, and *document information requests*. Search query requests are submitted with the array of parameters listed earlier in this section, and then passed to the Hausdorff index to execute the query. The server takes the resulting list of document identifiers and adds relevant information (such as the URI of the document or other metadata). The combined results are used to generate an HTML listing of search results. Document information requests are submitted when a user selects a document in the search result listing. To handle these requests, the server queries the RDBMS for a list of points associated with that document, and returns that list along with any point-specific attributes.

The client-side web application serves as the user interface for GeoXLS. It provides HTML input elements for user-specified search parameters, communicates with the web application server to perform search queries, and provides a Google Maps-based interface for entering query points and browsing the point sets of search result documents. A search query session typically consists of the following user actions:

- *Specify query points.* This is done by (i) clicking "Start Drawing Points"; (ii) clicking appropriate locations in the map pane to add those to an array of query points; and then (iii) clicking "Done Drawing Points".
- *Select query parameters.* Users can specify the direction for the Hausdorff search, whether to use MHD or



(a) (b)

Figure 4: GeoXLS search results where the query points represent (a) the locations of six universities in the U.S. and (b) the locations of five cities in Africa. The query points are shown as red markers, the selected search result is highlighted in green, and the point set representing the selected search result is shown using blue markers. The Hausdorff distance is illustrated as a line and circles around the query points in (b).

HausDist, the maximum point set size to consider in the search, and the number of results to display.

- *Execute search.* After users click the "Search" button, the search parameters are sent to the server. Results are displayed in tabular format.
- *Select display options.* When browsing search results, additional map overlays can assist users in interpreting the results. In particular, the Hausdorff distance can be displayed as: (i) a line between the points that determine the Hausdorff distance, or (ii) a circle around one or all of the markers in the appropriate point set.
- *Browse search results.* When users select documents from the search results, the application makes document information requests to the server, which returns the associated locations and their attributes. These are displayed on the map as blue markers, and additional overlays are rendered depending on the currently selected display options.

4. QUERY EXAMPLES

A screenshot of GeoXLS is shown in Figure 4(a). The user has specified the locations of six universities in the U.S. as query points and executed a search using the modified Hausdorff distance. Based on the values of the selected search parameters, the search result listing contains 30 spreadsheets with 100 or fewer points, ordered by each document's modified Hausdorff distance from the query point set. When executing a FromQuery search, GeoXLS returns point sets *containing but not restricted to* locations similar to the query points. Since documents mentioning these six universities are likely to include other universities, the search results contain documents with locations of various universities including those near the query set. Note that this containment relationship is reversed for ToQuery, and the containment relationship is bidirectional for Symmetric.

Figure 4(b) shows another example of search results. The chosen query locations are large cities in Africa, possibly where the user's organization has international headquarters. The search results consist of many spreadsheets that focus on Africa, possibly published by other groups that the user's organization can partner with. The line and circle overlays illustrate the geometric properties of this measure. The yellow line connects the two points that determine the symmetric Hausdorff distance in this example, and the grey circles around the query points use the Hausdorff distance as their radii. Due to the definition of the Hausdorff distance, this means that each circle contains at least one point in the result set (blue marker).

5. CONCLUSIONS AND FUTURE WORK

GeoXLS is a novel system for searching collections of point sets. The search results are not based on proximity of a single query point or a single result point to others, but rather the results are based on the overall similarity of the query and result point sets. GeoXLS supports multiple similarity measures related to the Hausdorff distance, and has been used to search several large collections of geotagged web documents.

As future work, we plan to extend the scope of the document geotagging modules to accommodate other web document domains, such as Wikipedia. We also plan to include semantic filtering which allows users to specify sources and types of documents included in search results. This work could also be used to support feature-based queries as in spatial data mining (e.g., [2]).

Acknowledgements. This work was supported in part by the NSF under Grants IIS-10-18475, IIS-09-48548, IIS-08-12377, CCF-08-30618, and IIS-07-13501.

6. REFERENCES

[1] M. D. Adelfio, S. Nutanong, and H. Samet. Similarity search on a large collection of point sets. In *GIS*, 2011.
[2] W. G. Aref and H. Samet. Efficient processing of window queries in the pyramid data structure. In *PODS*, pages 265–272, 1990.
[3] M. Flickner, H. S. Sawhney, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32, 1995.
[4] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. STEWARD: architecture of a spatio-textual search engine. In *GIS*, pages 186–193, 2007.
[5] M. D. Lieberman, H. Samet, J. Sankaranaryanan, and J. Sperling. Spatio-textual spreadsheets: Geotagging via spatial coherence. In *GIS*, pages 524–527, 2009.
[6] R. Lipikorn, A. Shimizu, and H. Kobatake. A modified exoskeleton and a Hausdorff distance matching algorithm for shape-based object recognition. In *CISST*, pages 507–511, 2003.
[7] S. Nutanong, E. H. Jacox, and H. Samet. An incremental Hausdorff distance calculation algorithm. *PVLDB*, 4(8):506–517, 2011.
[8] J. P. Snyder. *Flattening the Earth: Two Thousand Years of Map Projections*. University Of Chicago Press, 1997.
[9] A. Wang. An industrial strength audio search algorithm. In *ISMIR*, 2003.