USE OF THE SAND SPATIAL BROWSER FOR DIGITAL GOVERNMENT APPLICATIONS

By Hanan Samet, Houman Alborzi, František Brabec, Claudio Esperança, Gísli R. Hjaltason, Frank Morgan, and Egemen Tanin

Numerous federal agencies produce official statistics made accessible to ordinary citizens for searching and data retrieval. This is frequently done via the Internet through a Web browser interface. If this data is presented in textual format, it can often be searched and retrieved by such attributes as topic, responsible agency, keywords, or press release. However, if the data is of spatial nature, for example, in the form of a map, then using text-based queries is often too cumbersome for the intended audience. We describe the use of the SAND Spatial Browser to provide more power to users of these databases by enabling them to define and explore the specific spatial region of interest graphically. The SAND Spatial Browser allows users to form either purely spatial or mixed spatial/nonspatial queries intuitively, which can present information to users that might have been missed if only a textual interface was available.

The SAND Spatial Browser is part of the SAND System whose server side contains a spatial database system that facilitates organization (that is, indexing) of spatial and nonspatial data [1] to support efficient query processing. This database system handles any two or higher dimensional data with extent (for example, country boundaries, river paths), as well as point data (for example, city locations). It facilitates the response to queries involving this data such as finding the closest hazardous waste site to the border of a particular state.

Users access and manipulate spatial and nonspatial data using the SAND Spatial Browser in a manner similar to that used in spreadsheets where the map plays the same as a relation in a relational database management system. In particular, operations can be specified as compositions of maps with the output of one or more operations serving as input to other operations that can be saved for use as input to future operations. In addition, in many applications there is no need for the operation to run to completion in order to obtain the desired results. Thus the SAND Spatial Browser permits users to proceed in a pipelined fashion where the first results of an operation are fed as inputs to subsequent operations.

As an example of one of the composition queries that can be handled by the SAND Spatial Browser consider "finding the closest county (in terms of distances in the plane) to Cook County (Chicago) with a bladder cancer mortality rate for white

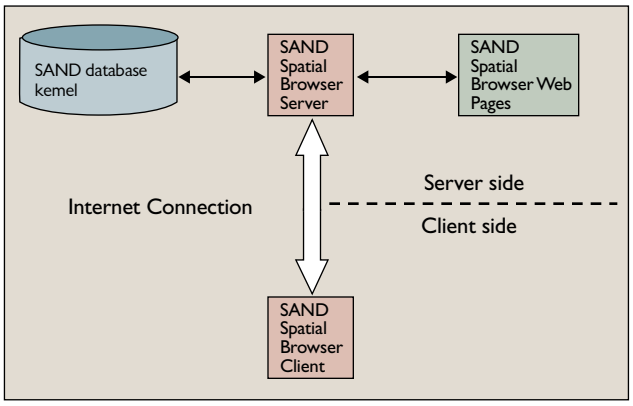

Figure 1. SAND system's client/server architecture.

males greater than 7.5 per 100,000 people in the period 1970–1994 and a population greater than 1 million." Responding to this query requires use of ranking [3] which is the process of retrieving data in the order of distance from other instances of the data or aggregates of user-defined data. The algorithms employed in the SAND Spatial Browser rank the elements incrementally, which means there is no need to compute more neighbors than are necessary. This is especially useful in our query since the nearest county may not satisfy the mortality rate and population conditions, necessitating finding the next nearest, and so on.

The SAND System is positioned somewhere between a conventional database management system (DBMS) and a Geographic Information System (GIS). It is similar in spirit, but does not have the full functionality of a GIS in the sense that its spatial analysis capabilities are limited. In addition, the design choices made in its implementation (for example, the algorithms it uses) were guided by the realization the output would be on a computer screen of relatively limited resolution, and thus it is not designed to produce maps of high accuracy. On the other hand, a small client footprint, minimal prerequisites, and ease of use all allow the SAND Spatial Browser to serve as a window into government maintained spatial databases for the general public. Thus it can be characterized as a lightweight GIS.

System Architecture

The SAND System employs a client/server architecture (Figure 1) [4] that differs from a number of Web-based mapping services (for example, MapQuest, www.mapquest.com and MapsOnUs, www.maponus.com), which perform all the calculations on the server side, and just transfer bitmaps representing results of user queries and commands to the client side. In particular, the SAND System distributes the work between the server and the client more evenly by running the actual database engine in a central location maintained by spatial database experts, while the end users acquire a Java-based client component that provides them with a gateway into the server-side SAND spatial database engine. The server and client components communicate over the public Internet (see www.cs.umd.edu/~brabec/sandjava/index.html).

The SAND Spatial Browser client is more than a simple bitmap image viewer. Instead, it operates on vector data that allows the client to execute many operations locally, such as zoom in/out or locational queries. In essence, a simpler version of the server spatial database engine is run on the client. This database keeps a copy of the relevant subset of the whole database whose full version is maintained on the server. This architecture is similar to caching, but is more sophisticated than other examples of caching, such as Web caching. In our case, when the client engine is given raw data, it evaluates queries and provides the visualization module with objects to be displayed. If it does not have enough data, it initiates communication with the server to either retrieve additional data or, if the query is more complex, to submit the query to the server and then to retrieve its results. In the case of

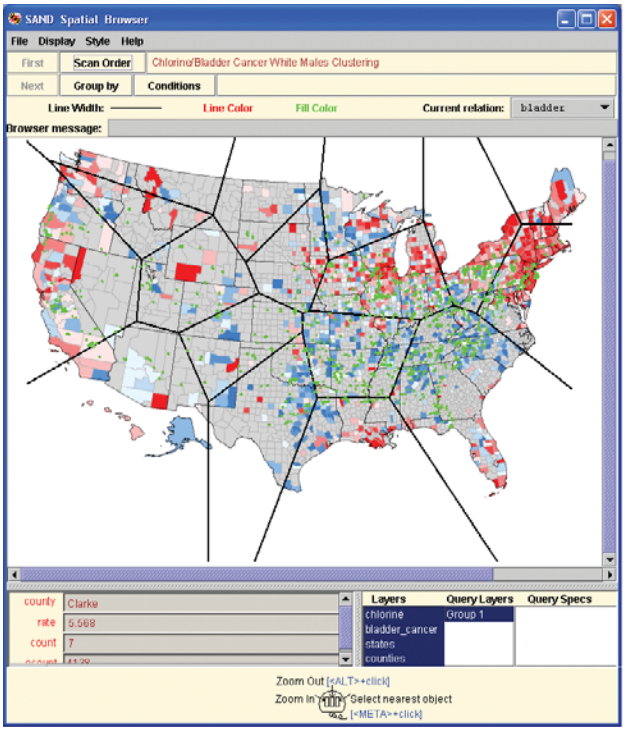

Figure 2. A sample screenshot of a possible user interaction with the SAND Spatial Browser. The relation being displayed corresponds to classes of mortality rates per 100,000 for bladder cancer for white males between 1970–1994. It is also overlaid with the result of a partition of the underlying space with respect to the 17 counties with the highest mortality rates so that each county in each partition is closer to the county with the high rate in the same partition than to any other county with a high rate. The green dots indicate locations of high chlorine emissions.

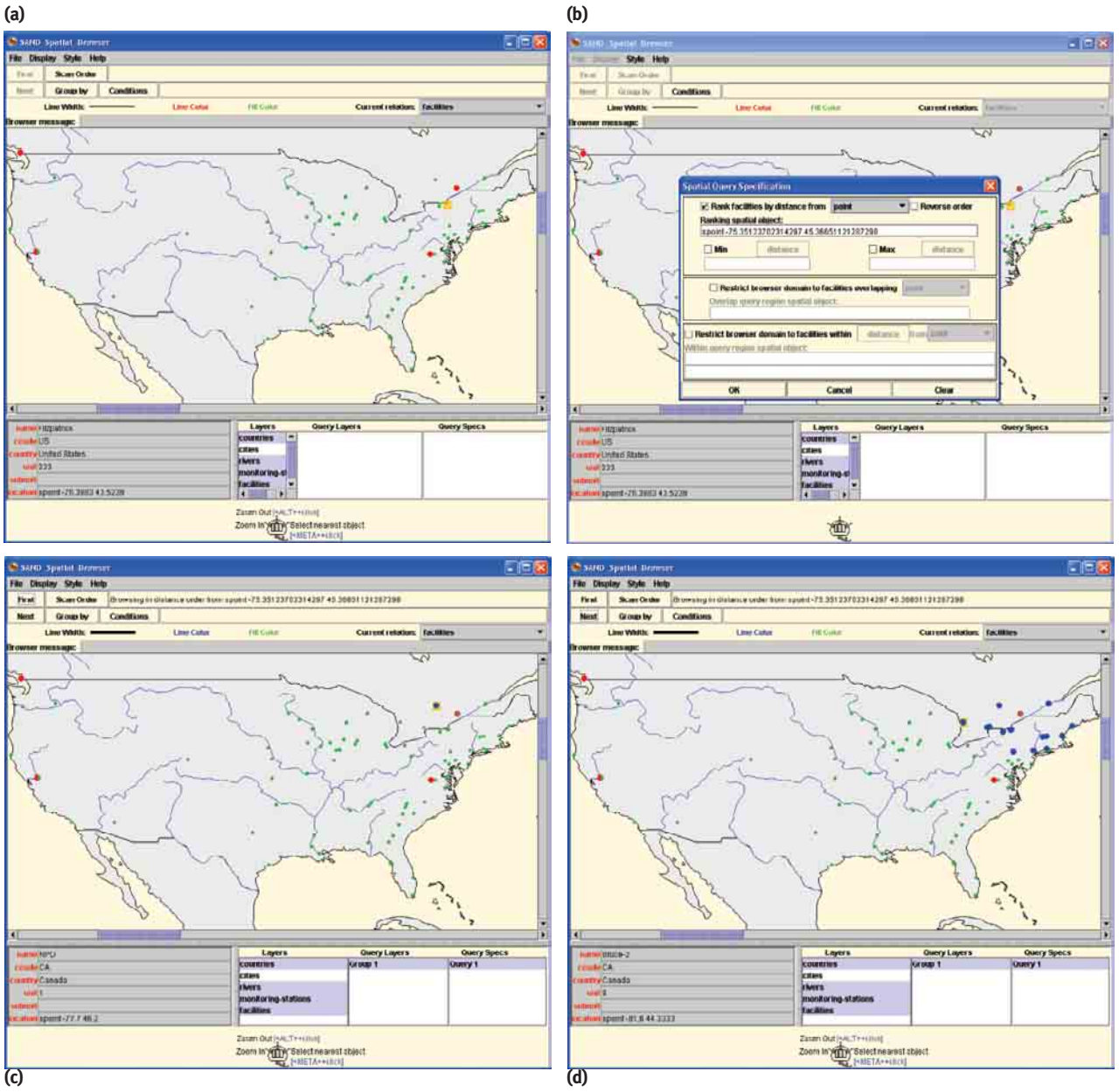

Figure 3. The nuclear facilities around a certain monitoring station along the northeastern U.S. and the Canadian border are computed. The green dots indicate nuclear facilities, the red dots indicate monitoring stations, and the blue dots indicate hits to our query. (a) displays the two relations, monitoring stations and nuclear facilities; (b) the location of a certain station is chosen for a ranking query by distance; (c) the closest facility is displayed; (d) the query continues incrementally with other hits.

Web caching, static Web pages or images are temporarily kept on the client in case the very same pages or images need to be used again in the near future. Unlike our case, in Web caching the client does not attempt to create new content from the data it stores.

Since the locally run database is only updated when additional or newer data is needed, our architecture allows the system to minimize the network traffic between the client and the server when executing the most common user-side operations such as zooming and panning. In fact, as long as the user explores one region at a time (that is, he or she is not panning all over the database), no additional data needs to be retrieved after the initial population of the client-side database. This makes the system much more responsive than the Web mapping services discussed earlier. Due to the complexity of evaluating arbitrary queries (such as, more than window queries needed for database visualization), we do not perform user-specified queries on the client. All user queries are still evaluated on the server side and the results are downloaded onto the client for displaying. However, assuming the queries are selective enough (that is, there are far fewer elements returned from the query than there are elements in the database), the response delay is usually within reasonable limits.

The SAND Spatial Browser is designed to give almost immediate access to the government data with

minimum effort to an occasional user. The trade-off is that the system is not as powerful as it could be if the fully featured server component along with all the data was also running on the client computer. For certain more serious users, it may be worth extending the effort to install and run such a full system locally. In order to accommodate such users, we have developed Approach for Peer-to-Peer Offloading the Internet (APPOINT) [6] as a means of extending the basic SAND Spatial Browser model. It introduces a centralized peer-to-peer approach to provide users with the ability to localize large volumes of data more efficiently by better utilizing the distributed network resources among active clients of the client/server architecture.

Existing Collaboration with Government Agencies and Sample Queries

FedStats is a Web site that enables ordinary citizens to access and search official statistics of numerous federal agencies without knowing in advance which agency produced them (www.fedstats.gov; also see Dippo in this section.) We have been involved in collaboration with FedStats in order to provide more power to users of FedStats by utilizing the SAND Spatial Browser to access and search official statistical data of numerous federal agencies [5].

Figure 2 is a screenshot of a user interaction with the SAND Spatial Browser. It shows the relation corresponding to mortality rates per 100,000 for bladder cancer for white males for the time period 1970–1994 obtained from the National Atlas of Cancer Mortality. We have also overlaid it with the result of a clustering-like operation available in the SAND Spatial Browser. In particular, we have shown a partition of the underlying space with respect to the 17 counties with the highest mortality rates so that each county in each partition is closer to the county with the high rate in the same partition than to any other county with a high rate. The green dots indicate locations of high chlorine emissions obtained from EPA data on the FedStats Web site. The goal is to determine if there is some spatial correlation between counties with a high incidence of bladder cancer and large chlorine emissions. As can be seen, locations with large chlorine emissions are not clustered around these counties. Thus these two events do not seem to be spatially correlated.

The scenario depicted in Figure 2 is analogous to a discrete Voronoi diagram, and is a form of spatial clustering. This clustering operation is available in the SAND Spatial Browser and is achieved by executing an incremental *distance semi-join* [2] operation where the input relation corresponding to the high chlorine emissions map is joined with the high incidence of bladder cancer map, and the join condition is based on proximity with the closest tuple pairs from the two sets being retained. Once the closest emissions-cancer pair (*a;b*) has been found, the next closest pair is found from the set of emissions tuples that excludes tuple *a* from participating. This process is continued until the closest high incidence of bladder cancer county has been found for each of the high chlorine emissions locations.

Figure 3 illustrates an example of a ranking query. In this query, nuclear facilities around a certain monitoring station along the northeastern U.S. and the Canadian border (Figure 3a) are computed in the order of their distance to this station. We first define our query by selecting the location of the station. At this point, the ranking operation starts by displaying our first hit (Figures 3b and 3c). By clicking the "Next" button we continue this operation as long as we want (Figure 3d). ■

This research was supported in part by the NSF under grants EIA-99-00268, EIA-99-01636, EAR-99-05844, IIS-00-86162, and EIA-00-91474.

References

1. Esperança, C. and Samet, H. Experience with SAND/Tcl: A scripting tool for spatial databases. *J. Visual Languages and Computing 13*, 2 (Apr. 2002), 229–255.
2. Hjaltason, G.R., and Samet, H. Incremental distance join algorithms for spatial databases. In *Proceedings of the ACMSIGMOD Conference.* Hass, L. and Tiwary, A., Eds. (Seattle, WA, June 1998), 237–248.
3. Hjaltason, G.R. and Samet, H. Distance browsing in spatial databases. *ACM Trans. Database Systems 24,* 2 (June 1999) (June 1999), 265–318. Also, *Advances in Spaatial Databases—4h International Symposium.* M.J. Engenhofer and J.R. Herring, eds., and Springer-Verlag Lecture Notes in Computer Sciences 951. (Aug. 1995), Portland, ME, and Computer Science TR-3919, University of Maryland, College Park, MD.
4. Samet, H. and Brabec, F. Remote thin-client access to spatial database systems. In *Proceedings of the 2nd National Conference on Digital Government Research.* (Los Angeles, CA, May 2002), 75–82.
5. Samet, H., Brabec, F., and Hjaltason, G.R. Interfacing the SAND spatial browser with FedStats data. In *Proceedings of the 1st National Conference on Digital Government Research.* (Los Angeles, CA, May 2001), 41–47.
6. Tanin, E. and Samet, H. APPOINT: An approach for peer-to-peer offloading the Internet. In *Proceedings of the 2nd National Conference on Digital Government Research* (Los Angeles, CA, May 2002), 99–105.

Hanan Samet (hjs@cs.umd.edu) is a professor of computer science and a member of the Center for Automation Research and the Institute for Advanced Computer Studies at the University of Maryland, College Park.
Houman Alborzi (houman@cs.umd.edu) is a graduate research assistant in the computer science department at the University of Maryland, College Park.
František Brabec (brabec@umiacs.umd.edu) is a graduate research assistant in the computer science department at the University of Maryland, College Park.
Claudio Esperança (esperanc@lcg.ufrj.br) is an associate professor at the Graduate Program in Systems and Computer Science at the Federal University of Rio de Janeiro, Brazil.
Gísli R. Hjaltason is an assistant professor of computer science at the University of Waterloo, Waterloo, Canada.
Frank Morgan (frank@umiacs.umd.edu) is a graduate research assistant in the computer science department at the University of Maryland, College Park.
Egemen Tanin (egemen@cs.umd.edu) is an assistant research scientist in the computer science department at the University of Maryland, College Park.