General Research Issues in Multimedia Database Systems

HANAN SAMET

Computer Science Department and Center for Automation Research and Institute for Advanced Computer Studies, University of Maryland, College Park, Maryland ⟨*hjs@umiacs.umd.edu*⟩

Many of the database research issues involved in dealing with multimedia data are similar to those that arise in handling other nontraditional data such as spatial, image, temporal, text, document, and scientific. We focus here on multimedia database issues, which we hope can serve as a starting point for a wider discussion. Our examples are often taken from the spatial domain as this is where we have the greatest expertise, although these issues are far more general.

Why do we want a database? The natural and simple answer is to be able to store and *retrieve* data efficiently. Notice the emphasis on retrieval. We should not lose sight of this purpose. For example, it means that storing images in long fields in a relational database is usually not the answer. Long fields are usually a stopgap solution as they are just a repository for data and do not aid in its retrieval. In particular, as the data volume gets large, this solution breaks down because the tuples get too large.

We need to be able to integrate nontraditional data with traditional (e.g., alphanumeric) data. Alphanumeric data can frequently be treated just like locational data in that the records that make up the alphanumeric data are like points in a higher-dimensional space where each attribute is analogous to a spatial dimension. The difference is that spatial data have more than just a locational component. In particular, spatial data are distinguished from nonspatial data by having spatial extent. A number of attempts at integration take advantage of this analogy. However, it can also act as a straitjacket in the case of the relational model. Some examples of successful integration include spatial with nonspatial data [Aref and Samet 1990], temporal with nontemporal data including spatial data [Hjaltason and Samet 1995], document with nondocument data [Sacks-Davis et al. 1995], image with locational and nonlocational data [Samet and Soffer 1995], and so on.

Efficient retrieval is facilitated by building an index [Samet 1990a, 1990b]. This means that we need to find a way to sort the data. Surprisingly, this is not always done for such applications (e.g., it is absent in the Photobook image database system [Pentland et al. 1994]). The index should be compatible with the data that are being stored, and we also need to choose an appropriate zero or reference point for it. The index should be implicit rather than explicit, as it is impossible to foresee all possible queries in advance. For example, in the spatial domain, assuming a relational model, it is impractical to have an attribute for each spatial relationship (e.g., north, northeast, left, etc.). Instead, the index should enable us to derive these relationships on the fly. As a more concrete example, an explicit index would sort two-dimensional locational data on the basis of distance from a given point x; yet this would not be very useful if we wanted to have the locations sorted with respect to a different point y. In particular, we would have to resort them. In contrast, an implicit index means that we do not have to re-sort the data for queries other than updates [Hjaltason and Samet 1995].

How do we decide which attributes to sort? We can make use of primary and secondary indices. Alternative indices are based on structure (e.g., SGML and HGML [Berners-Lee 1993]). How do we deal with the attributes for which we did not build an index? At times, an index is built for the application at hand (e.g., objects in an image database) but nothing is done for the interrelationship between the various objects (e.g., QBIC [Niblack et al. 1993]) so that we cannot retrieve objects in a spatial relation to other objects.

Do we sort on the basis of the Cartesian product of a subset of the attributes? Conventional data are usually indexed by building a separate index for each of a number of attributes. Indices for individual attributes are acceptable as long as the queries do not make use of any combination of attribute values. However, not all combinations of attributes are meaningful if the dimensional units of the combined attributes differ. For example, given attribute age (in years) and weight (in pounds), the Cartesian product of the age and weight attributes is not very useful because we are unlikely to want to determine the nearest record to John Jones in terms of age and weight, in part, because we do not have a commonly accepted notion of the year-pound unit. Nevertheless, other combinations of attributes are useful, such as Boolean combinations that result in queries like range or partial range.

How are the data modeled? There are many issues here. We use the relational model to illustrate some of them. A central issue is how to incorporate the index: should it be a separate relation or an attribute in the relation? The latter can be achieved by use of foreign keys or common symbolic attributes. To incorporate temporal data, we have the dichotomy between having one tuple per instance of time or interval of time versus a solution that distributes the time intervals over the attribute values in a tuple. There is also the tradeoff between a large relation in terms of many tuples (e.g., one tuple per pixel in a region or one tuple per piece of a spatial object) versus one tuple per object. An example of the latter is a bounding box, which requires twice as many attribute values as there are dimensions to the data (i.e., the locations of two diagonally opposite corners). An equivalent approach is to use a representative point in parameter space for each object. When the shape of the spatial object is nonstandard and hence does not lend itself to a simple parameterization, the location of one point in the spatial object can serve as a representative of the object. This point can then be used as an index to access a spatial data structure (e.g., an array or a region quadtree, etc.), which can then be traversed using a process such as connected component labeling to obtain the rest of the object.

We need to identify the possible queries and find their analogues in conventional databases. For example, a map in a spatial database can be viewed as a relation in a relational database. Thus in this case, we need to define the map analogues of relational projection, selection, and join. A difference immediately apparent in the spatial case is the presence of spatial attributes as well as of spatial output that may require the construction of a spatial index as part of the operations that involve spatial attributes.

How do we interact with the database? SQL, a standard interaction method, may not be enough. In some domains (e.g., spatial and image) a graphical query language may be more appropriate. Note that the input and output of nontraditional data in raw form is commonplace. The real issue is that these data must usually be processed further in order to extract more meaningful information (e.g., interpreted). This means that indices should be constructed for the output. As a further example, images may have to be treated by edge-detection algorithms and voice data may have to be processed by filters. It is desirable that such processing be done within the scope of the database system because it requires the manipulation of huge amounts of data. This demonstrates the need for such systems to be extensible in the sense that it should be possible to integrate into such systems specialist methods involving input/output data processing, and querying techniques. As part of the interaction, we may also want to browse through the data. Thus existing browsers must be extended to permit browsing on the basis of the nonspatial attributes and their indices. For example, in the case of spatial data, we may wish to examine cities with population greater than one million inhabitants in the order of their distance from Chicago, St. Louis, and so on [Hjaltason and Samet 1995].

What strategy should be used in answering queries that mix traditional data with multimedia data? We need query-optimization rules that are domain-sensitive. This requires the identification of selectivity factors and ways to measure them without responding to the entire query. The development of relevant data-sampling techniques is critical here. In addition, it also depends on whether an index exists on the multimedia data. If not, then we should select on the traditional data first. For example, suppose that we wish to find all cities within 100 miles of the Mississippi River with a population in excess of one million. We want to perform the spatial selection first if the region is small. On the other hand, the selection on the traditional data should be performed first if there are very few cities with a large population, assuming that we have an index on the population attribute.

As can be seen, the incorporation of multimedia data in database systems is an exciting and fruitful area of research with many open problems.

ACKNOWLEDGMENTS

The comments of Aya Saffer are greatly appreciated. This work was supported in part by the National Science Foundation under Grant IRI-92-16970.

REFERENCES

AREF, W. G. AND SAMET, H. 1990. An approach to information management in geographical applications. In *Proceedings of the Fourth International Symposium on Spatial Data Handling* (Zurich, July), 589–598.

BERNERS-LEE 1993. HyperText transfer protocol (HTTP). Geneva, Switzerland. Available via ftp://info.cern.ch/pub/www/doc/http-spec.ps.

GADIA, S. 1993. Parametric databases: Seamless integration of spatial, temporal, belief and ordinary data. *ACM SIGMOD Rec. 1* (March), 15–20.

HJALTASON, G. AND SAMET, H. 1995. Ranking in spatial databases. In *Advances in Spatial Databases—4th Symposium SSD '95* M. J. Egenhofer and J. R. Herring, Eds., Lecture Notes in Computer Science 951. Springer-Verlag, Berlin, 83–95.

NIBLACK, W., BARBER, R., EQUITZ, W., FLICKNER, M., GLASMAN, E., PETKOVIC, D., AND YANKER, P. 1993. The QBIC project: Querying images by content using color, texture, and shape. In *Proceedings of the SPIE Conference on Storage and Retrieval of Image and Video Databases, Volume 1908* (San Jose, CA, Feb.), 173–187.

PENTLAND, A., PICARD, R. W., AND SCARLOFF, S. 1994. Photobook: Content-based manipulation of image databases. In *Proceedings of the SPIE Conference on Storage and Retrieval of Image and Video Databases II, Volume 2185* (San Jose, CA, Feb.), 34–47.

SACKS-DAVIS, R., KENT, A., RAMAMOHANARAO, K., THOM, J., AND ZOBEL, J. 1995. Atlas: A nested relational database system for text applications. *IEEE Trans. Knowl. Data Eng. 7*, 3 (June), 454–470.

SAMET, H. 1990a. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley, Reading, MA.

SAMET, H. 1990b. *The Design and Analysis of Spatial Data Structures* Addison-Wesley, Reading, MA.

SAMET, H. AND SOFFER, A. 1995 A map acquisition, storage, indexing, and retrieval system. In *Proceedings of the Third International Conference on Document Analysis and Recognition* (Montreal, Canada, Aug.), 992–996.