In *Proc. of the IEEE International Symposium on Multimedia (ISM2015)*,
Miami, FL, December 2015.

Location Specification and Representation in Multimedia Databases

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

***Abstract*— Techniques for the specification and representation of the locational component of multimedia data are reviewed. The focus is on how the locational component is specified and also on how it is represented. For the specification component we also discuss textual specifications. For the representation component, the emphasis is on a sorting approach which yields an index to the locational component where the data includes both points as well as objects with a spatial extent.**

I. Introduction

The increasing availability of computing power has led to a wider accessibility of multimedia data. Multimedia data comes in many forms and includes documents, images, videos, and audio as well as spatial data of multiple dimensionality such as points, vectors, regions, and volumes. In addition, the multimedia data may have a spatial component corresponding to its location which is usually, but not always, a point in the associated vector space in which the multimedia data is embedded. The multimedia data may also have a temporal component where the key parameters are transaction time and valid time. When the data has both a spatial and temporal component, then it is termed spatiotemporal data. Multidimensional spatial data is differentiated from conventional multidimensional data by the fact that the spatial data has extent, but this need not always be the case.

In this paper we focus on the specification and representation of the locational component of multimedia data. Section II discusses the various ways of specifying the locations of multimedia data while Section III reviews a number of representations of the location of multimedia data as well as multidimensional multimedia spatial data with extent. Concluding remarks are drawn in Section IV.

II. Specification of Locations of Multimedia Data

The rise of the use of the world wide web and the ease with which multimedia objects can be accessed, regardless of their nature and/or physical location has had a significant impact on our life and how we interact with our environment, Multimedia data is now often online and can be queried by location. Viewing the results can be enhanced by adding the ability to view them dynamically through actions such as browsing (e.g., [13], [72], [73], [94] in the case of spatial data that is represented on a map) or manipulating what is termed a spatial spreadsheet [24]. In particular, the web has made it easier to find and retrieve multimedia data by location (i.e., index it) regardless of whether the location is specified explicitly or, increasingly more importantly, implicitly by virtue of the physical location of the user.

The explicit specification of location has traditionally been geometric (e.g., as latitude-longitude pairs of numbers). Unfortunately, this is not easy as most people don't usually know of a location in this way. Even more important is the fact that they don't know where to obtain this information and are not used to communicate or receive it from others in this way. Instead, they are used to specify a location textually which includes verbally. A textual specification has numerous benefits. First of all, it is very useful for multimedia data in the form of documents whhch have undergone limited preprocessing. Second, it is very easy to communicate it in this way especially when using smartphones where a textual capability for such communication is always present. Of course, there is always the verbal communication option which is closely related to the textual option especially when making use of speech generation and recognition as is done, for example, by Siri on Apple devices.

Another important advantage is that a textual specification acts like a polymorphic type in the sense that one size fits all. In particular, depending on the application which makes use of this information, a term such as "Los Angeles" can be interpreted both as a point or as an area, and the user need not be concerned with this question.

The principal drawback of the textual specification of location data is that it is potentially ambiguous. First of all, we must be sure that the textual entity actually corresponds to a location (termed a *toponym*) rather than the name of a person, an organization, or an object, to name a few of many possible alternative interpretations. The process of determining whether a textual entity is a toponym is termed *toponym recognition* (e.g., [34]). Moreover, having identified the textual entity as a toponym such as "San Jose", there are many possible locations named "San Jose" and they must be resolved. This process is known as em toponym resolution (e.g., [35], [70]). Moreover, in some cases we are not even sure that the term "Washington" denotes a location as it could be a reference to the name of a person. Both of these issues can arise when processing documents such as newspaper articles [36], [71], [75], [82], [98], tweets [14], [25], [91], blogs, etc. The process of understanding and converting location text to its geometric specification is known as *geotagging*(e.g., [1], [3], [28], [33], [37], [38], [41], [48], [49]) and is beyond the scope of this paper.

The implicit specification of location is achieved in a number of ways. The simplest is to use the IP address of device used to connect to the Internet. However, the most common is through the use of an embedded GPS capability which provides the user's physical location. Nevertheless, many users disable this feature for privacy reasons. This is especially noticeable when looking at tweets and observing how few tweeters actually transmit their location along with their tweets.

Touch interfaces on smartphone devices are increasingly being used to combine an implicit and an explicit specification to yield an approximate specification. In particular, observe that a map, coupled with the ability to pan and to vary the zoom level at which the underlying data is viewed, facilitates this approximate specification. For example, as we zoom in on Los Angeles on a map, we are targeting more specific parts of it. This has a direct effect on queries for objects at particular locations. For example, if we are seeking an art exhibit in Hollywood, then depending on the zoom level we would be satisfied by the return of an art exhibit in Burbank (by being in proximity) or by Los Angeles (by containment). Thus we see the touch interface serving as an implicit access structure to the data accomplished with direct manipulation. Of course, an index is still required along with software that translates the screen coordinates (via use of some nearest neighbor techniques) to the ones used by the index.

III. Representations of Locations of Multimedia Data

Applications involving multimedia data increasingly make use of its location and is increasingly being stored in a database. The existence of the database means that the data stored therein must be retrieved and this involves search. Search is facilitated by sorting the underlying data. The conventional definition of the verb *sort* is:

1) To put in a certain place or rank according to kind, class, or nature
2) To arrange according to characteristics.

The location data that is sorted is known as *spatial data* and range from including just the location of the data to also including its extent (i.e., the space that it spans). The sorting is captured by the data structure that is used to represent the spatial data. This data structure is usually referred to as an *access structure* or *index* to emphasize the connection to sorting.

Notwithstanding the above definition, sorting usually implies the existence of an ordering. Orderings are fine for one-dimensional data. For example, in the case of individuals we can sort them by their weight and find people closest in weight to different individuals without having to resort the data. This is not the case in two dimensions and higher where changing the reference point to which we are finding the nearest means having to resort the data. One way to achieve an ordering is to linearize the data as can be done, for example, using a space-filling curve (e.g., [51], [68]). The problem with such an approach is that the ordering is explicit. Instead, what is needed is an implicit ordering so that we do not need to resort the data when, for example in our two-dimensional example, the reference point changes (e.g., from St. Louis to Los Angeles). Such an ordering is a natural byproduct when we sort objects by spatial occupancy, and is the subject of the remainder of this section.

Methods that are based on sorting the spatial objects by spatial occupancy essentially decompose the underlying space from which the data is drawn into regions called *buckets* where objects in close proximity are ideally in the same bucket or at least in buckets that are close to each other in the sense of the order in which they would be accessed in the case of a false hit.

There are two principal methods of representing spatial multimedia data. The first is to use an object hierarchy that initially aggregates objects into groups based on their spatial proximity and then use proximity to further aggregate the groups thereby forming a hierarchy. Queries are facilitated by also associating a minimum bounding box which could also be a sphere with each object and group of objects to enable a quick determination if a point can possibly lie within the area spanned by the object or group of objects. A negative response means that no further processing is required for the object or group, while a positive response requires more tests (e.g., R-tree [15] and R*-tree [9]).

The drawback of the object hierarchy approach is that the resulting hierarchy of bounding boxes leads to a non-disjoint decomposition of the underlying space. Therefore, if a search fails in one path starting at the root, then it is not necessarily the case that no other object will be found in another path starting at the root.

The second method is based on a recursive decomposition of the underlying space into disjoint blocks so that a subset of the objects are associated with each block. There are several ways to proceed. The first way is to redefine the decomposition and aggregation associated with the object hierarchy method so that the minimum bounding boxes (rectangles in this case) are decomposed into disjoint rectangles, thereby partitioning the underlying objects that they bound (e.g., k-d-B-tree [50] and R$^+$-tree [92]).

The second way is to partition the underlying space at fixed positions so that all resulting cells are of uniform size, which is the case when using the uniform grid (e.g., [31]), The drawback of the uniform grid is the possibility of a large number of empty or sparsely-filled cells when the objects are not uniformly distributed. This is overcome by making use of a variable resolution representation such as one of the quadtree variants (e.g., [68]) where the subset of the objects that are associated with the blocks are defined by placing an upper bound on the number of objects that can be associated with each block (termed a *stopping condition* for the recursive decomposition process) and also often referred to as a *bucket capacity*. The PR quadtree [47], [66] and its bucket variants are examples of such a structure for points, while the PM quadtree family [20], [40], [77], [84] (see also the related PMR quadtree [19], [43], [44]) is an example of a variable resolution representation for collections of straight line segment objects such as those found in polygonal subdivisions as well as higher dimensions (e.g., faces of three-dimensional objects as in the PM octree [8]). An alternative known as a PK-tree [67], [100], makes use of a lower bound on the number of objects that can be associated with each block (termed an *instantiation* or *aggregation* threshold).

Quadtrees [23], [30] and their three-dimensional octree analogs [22], [42]. have also been used widely for representing and operating on image data in two and three dimensions, respectively (e.g., [63]) and find use in GIS and medical applications. In particular, algorithms have been devised for converting between them and numerous representations such as binary arrays [52], boundary codes [12], [53], [83], rasters [54], [60], [93], medial axis transforms [59], [61], terrain models [95], boundary models [96], constructive solid geometry (CSG) [78], as well as operations such as connected component labeling [56], [80], [81], perimeters [55],

[79], distance [57], image dilation [4], [2], computing Euler numbers [11], and ray tracing [64]. Many of these operations are implemented by traversing the actual quadtrees/octrees and performing the appropriate operation on each node and its neighbors [32], [58], [62], [64], [76]. Quadtrees and their variants are to be distinguished from pyramids (e.g., [97]) which are multiresolution data structures useful in spatial data mining [5], [99].

The drawback of the disjoint method is that when the objects have extent (e.g., line segments, rectangles, and any other non-point objects), then an object may be associated with more than one block. This means that objects will be reported a multiple number of times. Nevertheless, methods have been developed that avoid these multiple reports by making use of the geometry of the type of the data that is being represented (e.g., [6], [7], [10]). Note that the result of constraining the positions of the partitions means that there is a limit on the possible sizes of the resulting cells (e.g., a power of 2 in the case of a quadtree variant). However, this means that the underlying representation is good for operations between two different data sets (e.g., a spatial join [21], [26], [27]) as their representations are in registration. This means that it is easy to correlate occupied and unoccupied space in the two data sets. This is hard to do in methods based on an object hierarchy where the positions of the partitions are not constrained as is the case for the quadtree variants. For a recent empirical comparison of these representations with respect to multidimensional point data, see [29].

IV. Concluding Remarks

We have reviewed a number of ways of specifying the locations of multimedia data as well as the representations of its location and its extent. It is important to note that the choice of these representations is often influenced by the type of operations that are applied to the data. The most common operation is one of retrieval and this is what has the highest influence on the choice that is made. Another important operation is similarity retrieval based on the use of a similarity measure which ideally is a distance metric. This means means that it must obey the triangle inequality (i.e., transitivity), be non-negative, and be symmetric. In this case, the objects are represented in a space on the basis of their distance from one or more reference objects. In the latter case, the objects are associated with their nearest reference object. These representations are often similar to the methods that are used to index data in a vector space with the difference that the partitions of the underlying metric space are implicit rather than explicit as is the case in the vector space. For more on these representations, see [18], [68].

As we pointed out, the ability to sort spatial and metric data is particularly useful for proximity queries usually where proximity is measured in terms of as "the crow flies" (e.g., [16], [17], [69]). However, these representations can also be used to support proximity in a graph such as a road network (e.g., [46], [74], [85], [86], [87], [88], [89], [90]). They can also be used with different metrics such as a Hausdorff distance [45].

Acknowledgments: This work was supported in part by the National Science Foundation under Grants IIS-12-19023 and IIS-13-20791.

References

[1] M. D. Adelfio and H. Samet. Structured toponym resolution using combined hierarchical place categories. In *GIR'13*, pp. 49–56, Orlando, FL, Nov. 2013.

[2] A. Amir, A. Efrat, P. Indyk, and H. Samet. Efficient algorithms and regular data structures for dilation, location and proximity problems. *Algorithmica*, 30(2):164–187, 2001.

[3] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-Where: Geotagging web content. In M. Sanderson, K. Järvelin, J. Allan, and P. Bruza, editors, *SIGIR'04*, pp. 273–280, Sheffield, United Kingdom, July 2004.

[4] C.-H. Ang, H. Samet, and C. A. Shaffer. A new region expansion for quadtrees. *IEEE TPAMI*, 12(7):682–686, July 1990.

[5] W. G. Aref and H. Samet. Efficient processing of window queries in the pyramid data structure. In *PODS'90*, pp. 265–272, Nashville, TN, Apr. 1990.

[6] W. G. Aref and H. Samet. Uniquely reporting spatial objects: yet another operation for comparing spatial data structures. In *SDH'92*, pp. 178–189, Charleston, SC, Aug. 1992.

[7] W. G. Aref and H. Samet. Hashing by proximity to process duplicates in spatial databases. In *CIKM'94*, pp. 347–354, Gaithersburg, MD, Dec. 1994.

[8] D. Ayala, P. Brunet, R. Juan, and I. Navazo. Object representation by means of nonminimal division quadtrees and octrees. *TODS*, 4(1):41–59, Jan. 1985.

[9] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an efficient and robust access method for points and rectangles. In *SIGMOD*, pp. 322–331, Atlantic City, NJ, June 1990.

[10] J.-P. Dittrich and B. Seeger. Data redundancy and duplicate detection in spatial join processing. In *ICDE*, pp. 535–546, San Diego, CA, Feb. 2000.

[11] C. R. Dyer. Computing the Euler number of an image from its quadtree. *CGIP*, 13(3):270–276, July 1980.

[12] C. R. Dyer, A. Rosenfeld, and H. Samet. Region representation: boundary codes from quadtrees. *CACM*, 23(3):171–179, Mar. 1980.

[13] C. Esperança and H. Samet. Experience with SAND/Tcl: a scripting tool for spatial databases. *JVLC*, 13(2):229–255, Apr. 2002.

[14] N. Gramsky and H. Samet. Seeder finder - identifying additional needles in the Twitter haystack. In *LBSN'13*, pp. 44–53, Orlando, FL, Nov. 2013.

[15] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *SIGMOD*, pp. 47–57, Boston, June 1984.

[16] A. Henrich. A distance-scan algorithm for spatial access structures. In *GIS'94*, pp. 136–143, Gaithersburg, MD, Dec. 1994.

[17] G. R. Hjaltason and H. Samet. Distance browsing in spatial databases. *TODS*, 24(2):265–318, June 1999.

[18] G. R. Hjaltason and H. Samet. Incremental similarity search in multimedia databases. Computer Science Technical Report TR–4199, University of Maryland, College Park, MD, Nov. 2000.

[19] G. R. Hjaltason and H. Samet. Speeding up construction of PMR quadtree-based spatial indexes. *VLDBJ*, 11(2):109–137, Oct. 2002.

[20] E. G. Hoel and H. Samet. Efficient processing of spatial queries in line segment databases. In *SSD'91*, pp. 237–256, Zurich, Aug. 1991.

[21] E. G. Hoel and H. Samet. Benchmarking spatial join operations with spatial output. In *VLDB*, pp. 606–618, Zurich, Sept. 1995.

[22] G. M. Hunter. *Efficient computation and data structures for graphics*. PhD thesis, Department of Electrical Engineering and Computer Science, Princeton University, Princeton, NJ, 1978.

[23] G. M. Hunter and K. Steiglitz. Operations on images using quad trees. *IEEE TPAMI*, 1(2):145–153, Apr. 1979.

[24] G. S. Iwerks and H. Samet. The spatial spreadsheet. In *VISUAL99*, pp. 317–324, Amsterdam, The Netherlands, June 1999.

[25] A. Jackoway, H. Samet, and J. Sankaranarayanan. Identification of live news events using Twitter. In *LBSN'11*, pages 25–32, Chicago, Nov. 2011.

[26] E. Jacox and H. Samet. Iterative spatial join. *TODS*, 28(3):268–294, Sept. 2003.

[27] E. Jacox and H. Samet. Spatial join techniques. *TODS*, 32(1):7, Mar. 2007.

[28] C. B. Jones, R. S. Purves, P. D. Clough, and H. Joho. Modelling vague places with knowledge from the Web. *IJGIS*, 22(10):1045–1065, 2008.

[29] Y. J. Kim and J. M. Patel. Rethinking choices for multi-dimensional point indexing: making the case for the often ignored quadtree. In *CIDR 2007*, pp. 281–291, Asilomar, CA, Jan. 2007.

[30] A. Klinger. Patterns and search statistics. In J. S. Rustagi, editor, *Optimizing Methods in Statistics*, pp. 303–337. Academic Press, New York, 1971.

[31] D. E. Knuth. *The Art of Computer Programming: Sorting and Searching*, volume 3. Addison-Wesley, Reading, MA, second edition, 1998.

[32] M. Lee, L. De Floriani, and H. Samet. Constant-time neighbor finding in hierarchical tetrahedral meshes. In *SMI'01*, pp. 286–295, Genova, Italy, May 2001.

[33] J. L. Leidner and M. D. Lieberman. Detecting geographical references in the form of place names and associated spatial natural language. *SIGSPATIAL Special*, 3(2):5–11, 2011.

[34] M. D. Lieberman and H. Samet. Multifaceted toponym recognition for streaming news. In *SIGIR'11*, pp. 843–852, Beijing, July 2011.

[35] M. D. Lieberman and H. Samet. Adaptive context features for toponym resolution in streaming news. In *SIGIR'12*, pp. 731–740, Portland, OR, Aug. 2012.

[36] M. D. Lieberman and H. Samet. Supporting rapid processing and interactive map-based exploration of streaming news. In *GIS'12*, pp. 179-188, Redondo Beach, CA, Nov. 2012.

[37] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Geotagging: Using proximity, sibling, and prominence clues to understand comma groups. In *GIR'10*, page 6, Zurich, Switzerland, Feb. 2010.

[38] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *ICDE*, pp. 201–212, Long Beach, CA, Mar. 2010.

[39] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. STEWARD: architecture of a spatio-textual search engine. In *GIS'07*, pp. 186–193, Seattle, WA, Nov. 2007.

[40] M. Lindenbaum, H. Samet, and G. R. Hjaltason. A probabilistic analysis of trie-based sorting of large collections of line segments in spatial databases. *SIAM J. Comp.*, 35(1):22–58, Sep. 2005.

[41] B. Martins, H. Manguinhas, and J. Borbinha. Extracting and exploring the geo-temporal semantics of textual resources. In *ICSC'08*, pages 1–9, Santa Clara, CA, Aug. 2008.

[42] D. Meagher. Geometric modeling using octree encoding. *CGIP*, 19(2):129–147, June 1982.

[43] R. C. Nelson and H. Samet. A consistent hierarchical representation for vector data. In *SIGGRAPH*, pp. 197–206, Dallas, TX, Aug. 1986.

[44] R. C. Nelson and H. Samet. A population analysis for hierarchical data structures. In *SIGMOD*, pp. 270–277, San Francisco, May 1987.

[45] S. Nutanong, E. H. Jacox, and H. Samet. An incremental Hausdorff distance calculation algorithm. *PVLDB*, 4(8):506–517, Aug. 2011.

[46] S. Nutanong and H. Samet. Memory-efficient algorithms for spatial network queries. In *ICDE*, pages 649–660, Brisbane, Australia, Apr. 2013.

[47] J. A. Orenstein. Multidimensional tries used for associative searching. *INFOPL*, 14(4):150–157, June 1982.

[48] R. S. Purves, P. Clough, C. B. Jones, A. Arampatzis, B. Bucher, D. Finch, G. Fu, H. Joho, S. A. K, S. Vaid, and B. Yang. The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the internet. *IJGIS*, 21(7):717–745, 2007.

[49] G. Quercini, H. Samet, J. Sankaranarayanan, and M. D. Lieberman. Determining the spatial reader scopes of news sources using local lexicons. In *GIS'10*, pp. 43–52, San Jose, CA, Nov. 2010.

[50] J. T. Robinson. The K-D-B-tree: a search structure for large multidimensional dynamic indexes. In *SIGMOD*, pp. 10–18, Ann Arbor, MI, Apr. 1981.

[51] H. Sagan. *Space-Filling Curves*. Springer-Verlag, New York, 1994.

[52] H. Samet. Region representation: quadtrees from binary arrays. *CGIP*, 13(1):88–93, May 1980.

[53] H. Samet. Region representation: quadtrees from boundary codes. *CACM*, 23(3):163–170, Mar. 1980.

[54] H. Samet. An algorithm for converting rasters to quadtrees. *IEEE TPAMI*, 3(1):93–95, Jan. 1981.

[55] H. Samet. Computing perimeters of images represented by quadtrees. *IEEE TPAMI*, 3(6):683–687, Nov. 1981.

[56] H. Samet. Connected component labeling using quadtrees. *JACM*, 28(3):487–501, July 1981.

[57] H. Samet. Distance transform for images represented by quadtrees. *IEEE TPAMI*, 4(3):298–303, May 1982.

[58] H. Samet. Neighbor finding techniques for images represented by quadtrees. *CGIP*, 18(1):37–57, Jan. 1982.

[59] H. Samet. A quadtree medial axis transform. *CACM*, 26(9):680–693, Sept. 1983. Also see CORRIGENDUM, *CACM*, 27(2):151, Feb. 1984.

[60] H. Samet. Algorithms for the conversion of quadtrees to rasters. *CVGIP*, 26(1):1–16, Apr. 1984.

[61] H. Samet. Reconstruction of quadtrees from quadtree medial axis transforms. *CVGIP*, 29(3):311–328, Mar. 1985.

[62] H. Samet. A top-down quadtree traversal algorithm. *IEEE TPAMI*, 7(1):94–98, Jan. 1985.

[63] H. Samet. An overview of quadtrees, octrees, and related hierarchical data structures. In R. A. Earnshaw, editor, *Theoretical Foundations of Computer Graphics and CAD*, pp. 51–68. Springer-Verlag, Berlin, West Germany, 1988.

[64] H. Samet. Implementing ray tracing with octrees and neighbor finding. *Computers & Graphics*, 13(4):445–460, 1989.

[65] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley, Reading, MA, 1990.

[66] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA, 1990.

[67] H. Samet. Decoupling partitioning and grouping: overcoming shortcomings of spatial indexing with bucketing. *TODS*, 29(4):789–830, Dec. 2004.

[68] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan-Kaufmann, San Francisco, 2006.

[69] H. Samet. K-nearest neighbor finding using MaxNearestDist. *IEEE TPAMI*, 30(2):243–252, Feb. 2008.

[70] H. Samet. Using minimaps to enable toponym resolution with an effective 100% rate of recall. In *GIR'14*, Dallas, TX, Nov. 2014.

[71] H. Samet, M. D. Adelfio, B. C. Fruin, M. D. Lieberman, and B. E. Teitler. Porting a web-based mapping application to a smartphone app. In *GIS'11*, pages 525–528, Chicago, Nov. 2011.

[72] H. Samet, H. Alborzi, F. Brabec, C. Esperança, G. R. Hjaltason, F. Morgan, and E. Tanin. Use of the SAND spatial browser for digital government applications. *CACM*, 46(1):63–66, Jan. 2003.

[73] H. Samet, A. Rosenfeld, C. A. Shaffer, and R. E. Webber. A geographic information system using quadtrees. *Pattern Recognition*, 17(6):647–656, Nov/Dec 1984.

[74] H. Samet, J. Sankaranarayanan, and H. Alborzi. Scalable network distance browsing in spatial databases. In *SIGMOD*, pp. 43–54, Vancouver, Canada, June 2008.

[75] H. Samet, J. Sankaranarayanan, M. D. Lieberman, M. D. Adelfio, B. C. Fruin, J. M. Lotkowski, D. Panozzo, J. Sperling, and B. E. Teitler. Reading news with maps by exploiting spatial synonyms. *Communications of the ACM*, 57(10):64–77, Oct. 2014.

[76] H. Samet and C. A. Shaffer. A model for the analysis of neighbor finding in pointer-based quadtrees. *IEEE TPAMI*, 7(6):717–720, Nov. 1985.

[77] H. Samet, C. A. Shaffer, and R. E. Webber. Digitizing the plane with cells of non-uniform size. *INFOPL*, 24(6):369–375, Apr. 1987.

[78] H. Samet and M. Tamminen. Bintrees, CSG trees, and time. *SIGGRAPH*, pp. 121–130, San Francisco, July 1985.

[79] H. Samet and M. Tamminen. Computing geometric properties of images represented by linear quadtrees. *IEEE TPAMI*, 7(2):229–240, Mar. 1985.

[80] H. Samet and M. Tamminen. An improved approach to connected component labeling of images. In *CVPR*, pp. 312–318, Miami Beach, FL, June 1986.

[81] H. Samet and M. Tamminen. Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE TPAMI*, 10(4):579–586, July 1988.

[82] H. Samet, B. E. Teitler, M. D. Adelfio, and M. D. Lieberman. Adapting a map query interface for a gesturing touch screen interface. In *WWW'11 (Companion Volume)*, pp. 257–260, Hyderabad, India, Mar. 2011.

[83] H. Samet and R. E. Webber. On encoding boundaries with quadtrees. *IEEE TPAMI*, 6(3):365–369, May 1984.

[84] H. Samet and R. E. Webber. Storing a collection of polygons using quadtrees. *TOGS*, 4(3):182–222, July 1985.

[85] J. Sankaranarayanan, H. Alborzi, and H. Samet. Efficient query processing on spatial networks. In *GIS'05*, pp. 200–209, Bremen, Germany, Nov. 2005.

[86] J. Sankaranarayanan, H. Alborzi, and H. Samet. Distance join queries on spatial networks. In *GIS'06*, pp. 211–218, Arlington, VA, Nov. 2006.

[87] J. Sankaranarayanan and H. Samet. Distance oracles for spatial networks. In *ICDE*, pp. 652–663, Shanghai, Apr. 2009.

[88] J. Sankaranarayanan and H. Samet. Query processing using distance oracles for spatial networks. *IEEE TKDE*, 22(8):1158–1175, Aug. 2010.

[89] J. Sankaranarayanan and H. Samet. Roads belong in databases. *IEEE Data Engineering Bulletin*, 33(2):4–11, June 2010.

[90] J. Sankaranarayanan, H. Samet, and H. Alborzi. Path oracles for spatial networks. *PVLDB*, 2(1):1210–1221, Aug. 2009.

[91] J. Sankaranarayanan, H. Samet, B. Teitler, M. D. Lieberman, and J. Sperling. TwitterStand: News in tweets. In *GIS'09*, pp. 42–51, Seattle, WA, Nov. 2009.

[92] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R^+-tree: a dynamic index for multi-dimensional objects. In *VLDB*, pp. 71–79, Brighton, United Kingdom, Sept. 1987.

[93] C. A. Shaffer and H. Samet. Optimal quadtree construction algorithms. *CVGIP*, 37(3):402–419, Mar. 1987.

[94] C. A. Shaffer, H. Samet, and R. C. Nelson. QUILT: a geographic information system based on quadtrees. *IJGIS*, 4(2):103–131, Apr.–June 1990.

[95] R. Sivan and H. Samet. Algorithms for constructing quadtree surface maps. In *SDH'92*, volume 1, pp. 361–370, Charleston, SC, Aug. 1992.

[96] M. Tamminen and H. Samet. Efficient octree conversion by connectivity labeling. *SIGGRAPH* pp. 43–51, Minneapolis, MN, July 1984.

[97] S. L. Tanimoto and T. Pavlidis. A hierarchical data structure for picture processing. *CGIP*, 4(2):104–119, June 1975.

[98] B. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling. NewsStand: A new view on news. In *GIS'08*, pp. 144–153, Irvine, CA, Nov. 2008.

[99] W. Wang, J. Yang, and R. R. Muntz. STING: a statistical information grid approach to spatial data mining. In *VLDB*, pp. 186–195, Athens, Greece, Aug. 1997.

[100] W. Wang, J. Yang, and R. Muntz. PK-tree: a spatial index structure for high dimensional point data. In *FODO'90*, pp. 27–36, Kobe, Japan, Nov. 1998.