

## Sorting in Space and Words

Hanan Samet

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

**Abstract**—Techniques are reviewed for representing multi-dimensional spatial data geometrically and textually based on sorting it. These ideas are also used for metric data where only a distance function indicating the degree of similarity between all object pairs in the dataset are available.

**Keywords**—Spatial data, metric data, spatiotextual data, sorting

### I. OVERVIEW

Spatial data has traditionally been specified geometrically and explicitly. The ability to access it quickly depends on its representation and is important in game programming, computer graphics, visualization, solid modeling, computer vision and geographic information systems (GIS), not to mention mapping applications on smartphones which are largely the direct result of web-based location services such as Microsoft Bing Maps, Google Maps and Earth, as well as Apple Maps. In fact, in 2016 O’Beirne observed that Google Maps had 1 billion unique users every month which means that 1 out of 8 people in the world uses it each month<sup>1</sup>.

The efficient representation and manipulation of such data means that it must be stored in a database and be capable of quick retrieval which involves search. Invariably, the efficiency of searching is dependent on the extent to which the underlying data is sorted. The conventional definition of the term *sort* is that it is a verb meaning:

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

The sorting is encapsulated by the data structures that are used to represent the spatial data thereby making it more accessible. In fact, the term *access structure* or *index* is often used as an alternative to the term *data structure* in order to emphasize the importance of 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 given an individual such as Bill, we can use the ordering to find the person closest in weight to Bill. Similarly, we can use the same ordering to also find the person closest in weight to John. Unfortunately, in two dimensions and higher, such a solution does not

always work. In particular, suppose we sort all cities in the US by their distance from Chicago. This is fine for finding the closest city to Chicago, say with population greater than 200,000. However, we cannot use the same ordering to find the closest city to New York with population greater than 200,000, without resorting the cities.

The problem is that for two dimensions and higher, the notion of an ordering does not exist unless a dominance relation holds (e.g., [41])—that is, a point  $a = \{a_i | 1 \leq i \leq d\}$  is said to dominate a point  $b = \{b_i | 1 \leq i \leq d\}$  if  $a_i \leq b_i, 1 \leq i \leq d$ . Thus the only way to ensure the existence of an ordering is to linearize the data as can be done with a space-filling curve (e.g., [54]). 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 sample query, the reference point for the query changes (e.g., from Chicago to New York). Such an ordering is a natural byproduct when we sort objects by spatial occupancy.

Spatial data has traditionally been overwhelmingly specified geometrically (e.g., in two dimensions, as latitude-longitude pairs of numbers). Unfortunately, users of many new applications find the geometric specification to be cumbersome as they don’t think of a location in this way, don’t know it in this way, and are not accustomed to communicate it to others in this way. Instead, they are used to specify a location textually. The advantage of a textual specification is that it is easy to communicate especially on smartphone devices where a keyboard is always present. The text also 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 “Washington” can be interpreted both as a point or as an area, and the user need not be concerned with this question. The drawback of the textual specification of location data is that it is ambiguous. In particular, there are many possible locations named “Washington” and they must be resolved (toponym resolution) [32], [34], [35], [43], [55]. 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 (toponym recognition) [31]. This can be the case when processing documents such as newspapers [33], [57], [60], [64], tweets [13], [14], [24], [72], [80], [81], blogs, etc. Being able to handle textual specifications enables the

<sup>1</sup><https://www.justinobeirne.com/cartography-comparison>

development of map query interfaces to a wide range of spatially-referenced data thereby enabling the development of new applications such as disease tracking [28], [29], brand remediation [1], crime tracking [78], and the Wikipedia [42].

Another growing trend is to specify locations implicitly via, for example, an embedded GPS which provides the user's physical location. Most common is a combination of an implicit and explicit specification to yield an approximate specification. In particular, a map, coupled with the ability to pan and to vary the zoom level at which the world is viewed, provides an inherent granularity to the location specification process which facilitates this approximate specification.

In fact, the act of pointing at a location (i.e., by the appropriate positioning of a pointing device with the aid of panning) and making the interpretation of the precision of this positioning specification dependent on the zoom level is equivalent to using spatial synonyms, which are the hallmarks of approximate specifications. For example, a user posing a query seeking a concert in Manhattan would be satisfied by a concert in Harlem by proximity, New York City by containment, and Brooklyn by being a sibling borough of New York. Thus users no longer need to know the exact name or position of the sought location. Thus the touch interface is an implicit access structure to the data accomplished with direct manipulation. Of course, an index must be built whose access is achieved by software that translates the screen coordinates using nearest neighbor techniques as in a "pick" operation in computer graphics.

This tutorial provides a brief overview of hierarchical spatial data structures and related algorithms that make use of them. We describe hierarchical representations of points, lines, collections of small rectangles, regions, surfaces, and volumes. For region data, we point out the dimension-reduction property of the region quadtree and octree, as well as how to navigate between nodes in the same tree, thereby leading to the popularity of these representations. We also demonstrate how to use these representations for both raster and vector data. In the case of nonregion data, we show how these data structures can be used to find nearest neighbors which is critical when using machine learning techniques. In particular, we show how to do it in an incremental fashion so that the number of objects need not be known in advance (e.g., [16], [17], [18]). These methods are not restricted to a setting where proximity is measured in terms of "as the crow flies," but instead can also be used to support proximity in a graph such as a road network (e.g., [66], [67], [68], [69], [70], [71]). They can also be used with different metrics such as a Hausdorff distance [39].

These spatial data structures have been used in VASCO [9] which consists of a set of spatial index JAVA<sup>TM</sup> applets at <http://www.cs.umd.edu/~hjs/quadtree/> that enable web users to experiment with a number of hierarchical representations (e.g., [51], [52], [54]) for different spatial data types, and see animations of how they support a number of search queries (e.g., nearest neighbor and range queries).

They have also been used in the SAND Spatial Browser [10], [12], [56] and the QUILT system [59], [74] as well as in peer-to-peer settings [77]).

## II. METHODS BASED ON SPATIAL OCCUPANCY

There are two methods of representing spatial data. The first uses an object hierarchy that initially aggregates objects into groups based on their spatial proximity and then uses proximity to further aggregate the groups thereby forming a hierarchy. Queries are facilitated by also associating a minimum bounding box with each object and group of objects (e.g., the R-tree [15] and R\*-tree [8]) as this makes it easy to test if a point can possibly lie within the area spanned by the object or group of objects. A negative answer means that no further processing is required for the object or group. A positive answer means that further tests must be performed. Thus the minimum bounding box serves to avoid wasting work.

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. This means that if a search fails to find an object in one path starting at the root, then it does not mean that the object will not 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 is to simply redefine the decomposition and aggregation associated with the object hierarchy method so that the minimum bounding rectangles are decomposed into disjoint rectangles, thereby also implicitly partitioning the underlying objects that they bound. In this case, the partition of the underlying space is heavily dependent on the data and is said to be at arbitrary positions (e.g., the k-d-B-tree [44] and R<sup>+</sup>-tree [73]).

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., [27]), also the standard indexing method for maps. 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 resolved by using a variable resolution representation such as one of the quadtree variants (e.g., [54]) 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*. In this case we can say that the objects are sorted into cells which act like bins (i.e., buckets). The PR quadtree [40], [52] and its bucket variants are examples of such a structure for points, while the PM quadtree family [20], [36], [62] (see also the related PMR quadtree [19], [38]) 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 [7]). An alternative, as exemplified by the PK-tree [53], [79], 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], [26] and their three-dimensional octree analogs [22], [37], have also been used widely to represent and operate on region data in two and three dimensions, respectively. In particular, algorithms have been devised for converting between them and many representations including rasters [48], medial axis transforms [47], [49], terrain models [75], as well as operations such as connected component labeling [63], distance [45], and image dilation [2], [3]. Many of these operations are implemented by traversing the actual quadtrees/octrees and performing the appropriate operation on each node and its neighbors [30], [46], [50], [61]. Quadtrees and their variants are distinguished from pyramids (e.g., [76]) which are multiresolution data structures useful in spatial data mining [4].

The drawback of the disjoint method is that when 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 queries such as those that seek the length of all objects in a particular spatial region must remove duplicate objects before reporting the total length. Nevertheless, methods have been developed that avoid these duplicates by using the geometry of the type of the data that is being represented (e.g., [5], [6], [11]). 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). This means that the underlying representation is good for operations between two different data sets (e.g., a spatial join [21], [25]) as their representations are in registration (i.e., we can correlate occupied and unoccupied space in the two data sets, which is not easy when the positions of the partitions are not constrained as happens in object hierarchy methods even though the resulting decomposition of the underlying space is disjoint).

### III. REPRESENTATIONS FOR SIMILARITY SEARCHING

Methods analogous to those that we described have also been used in cases where the only information that we have available is a distance function that indicates the degree of similarity (or dis-similarity) between all pairs of the  $N$  objects. Usually the distance function  $d$  is required to obey the triangle inequality, be non-negative, and be symmetric, in which case it is known as a *metric* and also referred to as a *distance metric*. Given a distance function, we usually partition and index the objects with respect to their distance from a few selected objects. There are two basic partitioning schemes: *ball partitioning* and *generalized hyperplane partitioning* [54]. In ball partitioning, the data set is partitioned based on distances from one distinguished

object, into the subset that is inside and the subset that is outside a ball around the object. In generalized hyperplane partitioning, two distinguished objects  $p_1$  and  $p_2$  are chosen and the data set is partitioned into two sets based on which of the two distinguished objects is the closest. Observe that both schemes partition the underlying data set into spatial-like zones. The difference is that the boundaries of the zones are more well-defined for ball partitioning methods as they can be expressed explicitly using a small number of objects and a known distance value. In contrast, for generalized hyperplane partitioning methods, the boundaries of the zones are usually expressed implicitly in terms of the distinguished objects, instead of explicitly, which may require quite a bit of computation to determine. In fact, very often, the boundaries cannot be expressed explicitly as, for example, in the case of an arbitrary metric space (in contrast to a Euclidean space) where we do not have a direct representation of the ‘generalized hyperplane’ that separates the two partitions.

### ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Grant IIS-13-20791.

### REFERENCES

- [1] A. Abdelrazek, E. Hand, and H. Samet. Brands in NewsStand: Spatio-temporal browsing of business news. In *GIS'15*, Seattle, WA, Nov. 2015. Article 97.
- [2] A. Amir, A. Efrat, P. Indyk, and H. Samet. Efficient regular data structures and algorithms for location and proximity problems. In *FOCS'99*, pages 160–170, New York, Oct. 1999. Also in *Algorithmica*, 30(2):164–187, 2001.
- [3] C.-H. Ang, H. Samet, and C. A. Shaffer. A new region expansion for quadtrees. *IEEE TPAMI*, 12(7):682–686, July 1990.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] F. Brabec and H. Samet. Visualizing and animating R-trees and spatial operations in spatial databases on the worldwide web. In *Visual Database Systems (VDB4)*, pp. 123–140, L'Aquila, Italy, May 1998.
- [10] F. Brabec and H. Samet. Client-based spatial browsing on the world wide web. *IEEE Internet Computing*, 11(1):52–59, Jan/Feb 2007.
- [11] 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.
- [12] C. Esperança and H. Samet. Experience with SAND/Tcl: a scripting tool for spatial databases. *JVLC*, 13(2):229–255, Apr. 2002.
- [13] C. Fu, J. Sankaranarayanan, and H. Samet. Weibostand: Capturing Chinese breaking news using Weibo. In *SIGSPATIAL LBSN'14*, pages 41–48, Dallas, TX, Nov. 2014.
- [14] N. Gramsky and H. Samet. Seeder finder - identifying additional needles in the Twitter haystack. In *SIGSPATIAL LBSN'13*, pages 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. *Comp. Sci. 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] A. Jackoway, H. Samet, and J. Sankaranarayanan. Identification of live news events using Twitter. In *SIGSPATIAL LBSN'11*, pages 25–32, Chicago, Nov. 2011.
- [25] E. Jacox and H. Samet. Iterative spatial join. *TODS*, 28(3):268–294, Sept. 2003.
- [26] A. Klinger. Patterns and search statistics. In J. S. Rustagi, editor, *Optimizing Methods in Statistics*, pp. 303–337. Academic Press, New York, 1971.
- [27] D. E. Knuth. *The Art of Computer Programming: Sorting and Searching*, volume 3. Addison-Wesley, Reading, MA, second edition, 1998.
- [28] R. Lan, M. D. Adelfio, and H. Samet. Spatio-temporal disease tracking using news articles. In *SIGSPATIAL HealthGIS'14*, pages 31–38, Dallas, TX, Nov. 2014.
- [29] R. Lan, M. D. Lieberman, and H. Samet. The picture of health: map-based, collaborative spatio-temporal disease tracking. In *SIGSPATIAL HealthGIS'12*, Redondo Beach, CA, Nov. 2012.
- [30] 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.
- [31] M. D. Lieberman and H. Samet. Multifaceted toponym recognition for streaming news. In *SIGIR'11*, pp. 843–852, Beijing, July 2011.
- [32] 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.
- [33] 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.
- [34] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Geotagging: Using proximity, sibling, and prominence clues to understand comma groups. In *GIR'10*, Zurich, Switzerland, Feb. 2010. Article 6.
- [35] 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.
- [36] 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.
- [37] D. Meagher. Geometric modeling using octree encoding. *CGIP*, 19(2):129–147, June 1982.
- [38] R. C. Nelson and H. Samet. A population analysis for hierarchical data structures. In *SIGMOD*, pp. 270–277, San Francisco, May 1987.
- [39] S. Nutanong, E. H. Jacox, and H. Samet. An incremental Hausdorff distance calculation algorithm. *PVLDB*, 4(8):506–517, Aug. 2011.
- [40] J. A. Orenstein. Multidimensional tries used for associative searching. *INFOPL*, 14(4):150–157, June 1982.
- [41] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.
- [42] G. Quercini and H. Samet. Uncovering the spatial relatedness in Wikipedia. In *GIS'14*, pages 153–162, Dallas, TX, Nov. 2014.
- [43] 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.
- [44] 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.
- [45] H. Samet. Distance transform for images represented by quadtrees. *IEEE TPAMI*, 4(3):298–303, May 1982.
- [46] H. Samet. Neighbor finding techniques for images represented by quadtrees. *CGIP*, 18(1):37–57, Jan. 1982.
- [47] H. Samet. A quadtree medial axis transform. *CACM*, 26(9):680–693, Sept. 1983. Also see CORRIGENDUM, *CACM*, 27(2):151, Feb. 1984.
- [48] H. Samet. Algorithms for the conversion of quadtrees to rasters. *CVGIP*, 26(1):1–16, Apr. 1984.
- [49] H. Samet. Reconstruction of quadtrees from quadtree medial axis transforms. *CVGIP*, 29(3):311–328, Mar. 1985.
- [50] H. Samet. A top-down quadtree traversal algorithm. *IEEE TPAMI*, 7(1):94–98, Jan. 1985.
- [51] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley, Reading, MA, 1990.
- [52] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA, 1990.
- [53] H. Samet. Decoupling partitioning and grouping: overcoming shortcomings of spatial indexing with bucketing. *TODS*, 29(4):789–830, Dec. 2004.
- [54] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan-Kaufmann, San Francisco, 2006.
- [55] H. Samet. Using minimaps to enable toponym resolution with an effective 100% rate of recall. In *GIR'14*, pages 9:1–9:8, Dallas, TX, Nov. 2014.
- [56] 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.
- [57] 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*, pp. 525–528, Chicago, Nov. 2011.
- [58] H. Samet, B. C. Fruin, and S. Nutanong. DUKing it out at the smartphone mobile app mapping API corral: Apple, Google, and the competition. In *SIGSPATIAL MobiGIS'12*, Redondo Beach, CA, Nov. 2012.
- [59] H. Samet, A. Rosenfeld, C. A. Shaffer, and R. E. Webber. A geographic information system using quadtrees. *Pat. Recog.*, 17(6):647–656, Nov/Dec 1984.
- [60] 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. *CACM*, 57(10):64–77, Oct. 2014.
- [61] 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.
- [62] 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.
- [63] H. Samet and M. Tamminen. An improved approach to connected component labeling of images. In *CVPR*, pp. 312–318, Miami Beach, FL, June 1986.
- [64] 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.
- [65] H. Samet and R. E. Webber. On encoding boundaries with quadtrees. *IEEE TPAMI*, 6(3):365–369, May 1984.
- [66] J. Sankaranarayanan, H. Alborzi, and H. Samet. Efficient query processing on spatial networks. In *GIS'05*, pp. 200–209, Bremen, Germany, Nov. 2005.
- [67] J. Sankaranarayanan, H. Alborzi, and H. Samet. Distance join queries on spatial networks. In *GIS'06*, pp. 211–218, Arlington, VA, Nov. 2006.
- [68] J. Sankaranarayanan and H. Samet. Distance oracles for spatial networks. In *ICDE*, pp. 652–663, Shanghai, Apr. 2009.
- [69] J. Sankaranarayanan and H. Samet. Query processing using distance oracles for spatial networks. *IEEE TKDE*, 22(8):1158–1175, Aug. 2010.
- [70] J. Sankaranarayanan and H. Samet. Roads belong in databases. In *IEEE Data Engineering Bull.*, 33(2):4–11, June 2010.
- [71] J. Sankaranarayanan, H. Samet, and H. Alborzi. Path oracles for spatial networks. *PVLDB*, 2(1):1210–1221, Aug. 2009.
- [72] 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.
- [73] 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.
- [74] 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.
- [75] R. Sivan and H. Samet. Algorithms for constructing quadtree surface maps. In *SDH'92*, volume 1, pp. 361–370, Charleston, SC, Aug. 1992.
- [76] S. L. Tanimoto and T. Pavlidis. A hierarchical data structure for picture processing. *CGIP*, 4(2):104–119, June 1975.
- [77] E. Tanin, A. Harwood, and H. Samet. A distributed quadtree index for peer-to-peer settings. In *ICDE*, pp. 254–255, Tokyo, Apr. 2005.
- [78] F. Wajid and H. Samet. CrimeStand: Spatial tracking of criminal activity. In *GIS'16*, Burlingame, CA, Nov. 2016. Article 81.
- [79] 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.
- [80] H. Wei, J. Sankaranarayanan, and H. Samet. Finding and tracking local Twitter users for news detection. In *GIS'17*, Redondo Beach, CA, Nov. 2017. Article 126.
- [81] H. Wei, J. Sankaranarayanan, and H. Samet. Measuring spatial influence of Twitter users by interactions. In *SIGSPATIAL LENS'17*, Redondo Beach, CA, Nov. 2017. Article 2.